

Boundary integral equations for BVPs, and their high-order Nyström quadratures: a tutorial

Alex Barnett

June 22, 2014

Abstract

We build up the essentials required to understand and code high-order and spectrally-accurate quadratures for Nyström methods to solve boundary integral equations (BIE). We show how these are used to solve common elliptic boundary value problems in \mathbb{R}^2 , using dense linear algebra for tutorial purposes. Simple, readable, modular Matlab codes, and theory and coding exercises, are included. The singular schemes and ideas for \mathbb{R}^3 are only sketched.

Contents

1	Quadrature for 1D functions	2
1.1	Periodic trapezoid rule (PTR)	2
1.2	Composite quadrature using “panels”	3
1.3	Parametrization of curves in \mathbb{R}^2	4
2	Boundary integral equations (BIE)	5
2.1	Evaluation of the solution	5
3	Nyström method for solving the BIE, smooth kernel case	6
4	Nyström quadratures for singular kernels	8
4.1	Local correction to the PTR weights (Kapur–Rokhlin)	9
4.2	The auxiliary nodes idea (Kolm–Rokhlin, Alpert)	10
4.2.1	Surfaces in \mathbb{R}^3	11
4.3	The explicit split of log singularity idea	11
4.3.1	Global correction of the PTR weights (Kress)	11
4.3.2	Explicit-split local panel corrections (Helsing)	12
4.4	PDE-based singular quadratures: QBX for \mathbb{R}^2 and \mathbb{R}^3	12

This document, and the accompanying code examples and exercises directory `code`, can be downloaded from <http://math.dartmouth.edu/~fastdirect>. The codes are in Matlab, for convenience; most production codes would be in a compiled language.¹

¹For compatibility with fast direct solvers our codes provide individual matrix elements, even though this is much slower when the full matrix is needed as it is for our codes.

1 Quadrature for 1D functions

A N -point quadrature rule approximates an integral on $[a, b]$ via a weighted sum of N function evaluations,

$$\int_a^b f(s)ds \approx \sum_{j=1}^N w_j f(s_j)$$

where the “rule” consists of the weights w_j and the nodes $s_j \in [a, b]$. The rule is designed to be accurate for all f in some class.

The only way to create very high-order methods is to understand the case of smooth (C^∞ or analytic) functions. When elliptic PDEs have locally smooth coefficients and data, solutions are locally smooth. This is very common in applications, eg, wave scattering from a piecewise smooth surface.

1.1 Periodic trapezoid rule (PTR)

For 2π -periodic functions on $[0, 2\pi)$, the rule $w_j = 2\pi/N$, $s_j = 2\pi j/N$, $j = 1, \dots, N$, is excellent; it has super-algebraic convergence if f is smooth, ie error is $O(N^{-k})$ for all $k > 0$. (Contrast the non-periodic trapezoid rule.)

Theorem 1.1 (Davis '57, [17, Thm. 12.6]). *Let $f(s)$ be 2π -periodic, and bounded and analytic in $|\operatorname{Im} s| \leq \alpha$, then the PTR error is $O(e^{-\alpha N})$.*

Our integrands will involve fundamental solutions arising from the PDE, and the integrals are often with respect to a source point traversing a closed curve (parametrized by $[0, 2\pi)$). The Davis theorem thus explains why when the target point comes close to the curve, we pay a price by needing a larger N (in inverse proportion) to preserve accuracy. The next example illustrates this.

Example 1.1. The Laplace equation has a fundamental solution

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

where $x, y \in \mathbb{R}^2$. The “single layer potential” with smooth density σ on the unit circle at target $x = (0, 0.9) \in \mathbb{R}^2$ is then

$$u(x) := \int_0^{2\pi} \Phi(x, y(s))\sigma(s)ds$$

where the source point is $y(s) = (\cos s, \sin s)$. We wish to approximate this numerically via the PTR—how can we predict the convergence?

We associate \mathbb{R}^2 with the complex plane, ie $x, y \in \mathbb{C}$, and realize $u(x) = \operatorname{Re} \frac{1}{2\pi} \int_0^{2\pi} \log \frac{1}{x - e^{is}} \sigma(s)ds$. Let's assume σ is sufficiently smooth. The integrand has a singularity when $x = e^{is}$, ie the complex parameter value $s = -i \log x$, so is analytic only in a strip $|\operatorname{Im} s| \leq \alpha$ for $\alpha < |\log 0.9|$. The convergence rate then approaches $|\log 0.9| \approx 0.105$. Thus to reach 10^{-14} we expect $N \approx \log 10^{-14} / \log 0.9 \approx 306$. This large N grows as the target x approaches the unit circle, illustrating the *close evaluation* problem (see work of Helsing, Barnett, Quaife).

Exercise 1 (“5h-rule”). By assuming the constant in the big- O in Davis' theorem is of order 1, show that 14-digit accuracy results if there is no singularity (eg, target point) closer than $5h$, where $h = 2\pi/N$ is the node spacing.

Exercise 2. Measure (by plotting) the convergence type and rate of the PTR on $[0, 2\pi)$ for the real-valued integrand $f(s) = a/(a^2 + \sin^2 \frac{s}{2})$ where $a > 0$ is some real constant. [Hints: use the converged value as the “exact” answer. Use the Matlab code template `ptrconv_template.m`]. As $a \rightarrow 0^+$ what does f do and what happens to the rate? Why? Use the region of analyticity of f to add the predicted rate to your plot.

Exercise 3. Test the claim on N made in the Example 1.1.

Exercise 4. Another way in which the integrand can inhibit convergence is by being oscillatory, as happens with high frequency waves. Let $k \in \mathbb{Z}$. Prove that the PTR is exact for the (entire) function e^{iks} once $N > |k|$. To how many “points per wavelength” does this critical N correspond? How does it compare to the Nyquist frequency?

1.2 Composite quadrature using “panels”

The PTR is mostly convenient only for integration in 1D, and when there is a “uniform amount of complexity”, ie there aren’t big regions where the integrand is smooth but others where it’s rapidly changing. This latter case arises in applications and motivates using composite (panel-wise) quadratures, since panels of vastly different sizes can be combined, and adaptivity (by splitting panels) is easy.

Consider integration on the single “panel” $[-1, 1]$. When s_j are zeros of Legendre polynomials and the weights w_j optimal, this is Gauss–Legendre quadrature. This is also exponentially accurate for analytic integrands, but “distance” in the complex plane is measured differently.

Theorem 1.2 ([18, Thm. 19.3]). *Let f be bounded and analytic in the ellipse² with foci ± 1 and semiaxes $\cosh \alpha$ and $\sinh \alpha$ for some $\alpha > 0$. Then p -node Gauss–Legendre quadrature has error $O(e^{-2\alpha p})$.*

The ellipse arises as the image of lines parallel to the real axis under the cosine map. The proof involves showing exponential decay in the Chebyshev coefficients. The point is that the convergence rate suffers if the integrand has a nearby singularity.

We can now handle $[0, 2\pi)$ or any other interval by breaking into subintervals, each of which uses a fixed p -node Gauss rule. If there is not too much variation in geometry or integrand smoothness then all panels may be the same size. Here’s $n = 5$ panels with $p = 16$ (vertical lines show panel junctions):



For smooth integrands, convergence with respect to n , hence with respect to $N = np$, is now algebraic of the high order $2p$ if $f \in C^{2p}$ (combining standard results on the Gauss rule [8, (2.7.12)] with a theorem on composite rules [8, Sec. 2.4]). In practice for large $p \geq 10$ this is hard to tell apart from spectral.

Each of these panels has its own Bernstein ellipses: if the integrand has no singularities closer than, say, one panel length from the real line, then the Chebyshev coefficients of the integrand on each panel will decay very fast, and the $p = 16$ rule will have accuracy close to machine precision. Considering kernel integrands with a target point x as in Example 1.1, this explains a criterion for x being “far” vs “near” a panel. If x is “near”, either a new rule is needed, or the panel should be further subdivided.

Exercise 5. Use the $p = 16$ point Gauss–Legendre rule (use `gauss.m`) on n panels of size decreasing by powers of two ($[1/2, 1]$, $[1/4, 1/2]$, etc) to accurately integrate $f(x) = x^{-1/2}$ on $[0, 1]$, and check the convergence rate with $N = np$. This is the idea for BIEs resulting from domains with corners; this subdivision scheme is called “dyadic”. BONUS: design a more optimal subdivision for general power-law singularities at 0.

Exercise 6. Compare the convergence of PTR and composite Gauss rules on band-limited functions such as e^{iks} . [to finish]

²This “Bernstein” ellipse is usually characterized by parameter $\rho = e^\alpha$.

1.3 Parametrization of curves in \mathbb{R}^2

To integrate with respect to arc length ds on a closed curve Γ we parametrize by a 2π -periodic function $Z : \mathbb{R} \rightarrow \mathbb{R}^2$ where $Z([0, 2\pi)) = \Gamma$, and change variable to coordinate s . This generalizes Example 1.1. Let f be a function defined on Γ , then

$$I := \int_{\Gamma} f(y) ds_y = \int_0^{2\pi} f(Z(s)) |Z'(s)| ds \approx \sum_{j=1}^N f(Z(s_j)) |Z'(s_j)| w_j = \sum_{j=1}^N f(x_j) \tilde{w}_j$$

where $|Z'(s)|$ is the “speed” (length of the velocity vector), and w_j and s_j are a rule for $[0, 2\pi)$. Note that the nodes are $x_j := Z(s_j)$, and the weights acquire a speed factor $\tilde{w}_j := |Z'(s_j)| w_j$.

There are infinitely many parametrizations; some will be better than others in terms of accuracy if a non-adaptive quadrature rule is used (PTR or uniform panels).

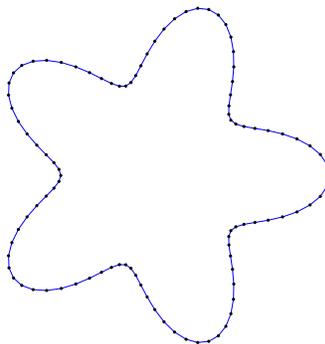
A simple test curve is the “smooth star” shape. In Matlab, we use complex numbers to represent points in \mathbb{R}^2 , and a struct `G` conveniently to contain Z , Z' and other useful curve properties. We can create a 5-pointed star with “wobbliness” $a = 0.3$ via `G = smoothstar(0.3,5)`; given the following function.

```
function G = smoothstar(a,w)
% SMOOTHSTAR - set up [0,2pi) parametrization of smooth star closed curve
R = @(t) 1 + a*cos(w*t); Rp = @(t) -w*a*sin(w*t); Rpp = @(t) -w*w*a*cos(w*t);
G.Z = @(t) R(t).*exp(1i*t); G.Zp = @(t) (Rp(t) + 1i*R(t)).*exp(1i*t);
G.Zpp = @(t) (Rpp(t) + 2i*Rp(t) - R(t)).*exp(1i*t);
```

We can set up a PTR with $N = 100$ on this curve via `G = curvquad(G,'ptr',100)`, which adds the locations `x`, normals `nx`, and curvatures at each node, etc, into the curve struct as follows.

```
function G = curvquad(G,rule,N)
% CURVQUAD - set up underlying quadrature for a closed curve struct
if strcmp(rule,'ptr')
    G.s = 2*pi*(1:N)/N; G.w = 2*pi/N*ones(N,1);
end
s = G.s; G.x = G.Z(s); G.sp = abs(G.Zp(s)); G.nx = -1i*G.Zp(s)./G.sp;
G.cur = -real(conj(G.Zpp(s)).*G.nx)./G.sp.^2; % curvature
```

Here is the resulting curve showing its nodes:



Exercise 7. Expand the above function to set up a N -code curve quadrature based upon Gauss panels of order $p = 16$.

2 Boundary integral equations (BIE)

We set up the simplest BIE for the interior Laplace problem with Dirichlet data. Kress [17, Ch. 6] is an excellent reference. Let Ω be the interior of a simple closed curve Γ . The BVP to solve is,

$$\Delta u = 0 \quad \text{in } \Omega \quad (2)$$

$$u = f \quad \text{on } \Gamma \quad (3)$$

Recalling that Φ is the fundamental solution (1) to the PDE, we make the ansatz that u is represented by a *double-layer potential*, namely

$$u(x) = \int_{\Gamma} \frac{\partial \Phi(x, y)}{\partial n_y} \sigma(y) ds_y = \int_{\Gamma} \frac{n_y \cdot (x - y)}{2\pi|x - y|^2} \sigma(y) ds_y = \int_0^{2\pi} \frac{n(s) \cdot (x - Z(s))}{2\pi|x - Z(s)|^2} \sigma(s) |Z'(s)| ds \quad (4)$$

where $n_y = n(s)$ means the outwards-pointing unit normal at boundary point $y = Z(s)$, and, abusing notation slightly, $\sigma(s) = \sigma(y(s))$. Since it is a linear combination of fundamental solutions, u satisfies (2) in Ω , and also in $\mathbb{R}^2 \setminus \overline{\Omega}$ (but these two solutions are not obviously related to each other.)

We are left only with (3) to satisfy; this gives an equation for the unknown “density” function σ . Potential theory states that the DLP (4) has limiting values on the curve which differ on the two sides. Using $u^{\pm}(x) := \lim_{h \rightarrow 0^+} u(x + hn_x)$ to denote these two limits at $x \in \Gamma$ (+ is the exterior, – the interior), we have the *jump relation*

$$u^{\pm}(x) = (D\sigma)(x) \pm \frac{1}{2}\sigma(x), \quad x \in \Gamma, \quad (5)$$

where $D : C(\Gamma) \rightarrow C(\Gamma)$ is the *integral operator* with a “dipole” kernel

$$k(x, y) = \frac{\partial \Phi(x, y)}{\partial n_y} = \begin{cases} \frac{n_y \cdot (x - y)}{2\pi|x - y|^2}, & x \neq y \\ -\frac{\kappa(x)}{4\pi}, & x = y \end{cases} \quad x, y \in \Gamma \quad (6)$$

Note that the diagonal limit requires the curvature function κ . Since u^- must equal f , (5) gives the linear integral equation

$$(D - \frac{1}{2})\sigma = f$$

This is Fredholm 2nd kind, which has wonderful consequences for both analysis (existence) and stable computation. This is called the *indirect* BIE; the direct BIE is similar but with $\partial u / \partial n$ the unknown, and the operator involves an adjoint.

Writing this using the parameters $s, t \in [0, 2\pi)$, ie target $x = Z(t)$ and source $y = Z(s)$,

$$\frac{\sigma(t)}{2} - \int_0^{2\pi} k(Z(t), Z(s)) |Z'(s)| \sigma(s) ds = f(t) \quad \text{for all } t \in [0, 2\pi) \quad (7)$$

Exercise 8. Check the diagonal limit of the continuous kernel (6).

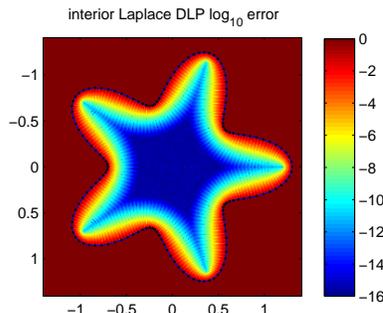
2.1 Evaluation of the solution

Assume for now that the density function σ has been solved for. The solution potential u may then be evaluated at any point x using (4), approximated by a quadrature rule on Γ .

A word about testing. For the Laplace DLP in \mathbb{R}^2 or \mathbb{R}^3 , a classic test case for correctness of this is the constant density $\sigma \equiv 1$ which generates $u(x) = -1$ for $x \in \Omega$ but $u(x) = 0$ outside; a similar test exists for Stokes. For the SLP, or for the Helmholtz equation, the only tests involve combinations of SLP and DLP, or solving an actual BVP.

Exercise 9. Write a routine `evalDLP.m` which given σ , the curve struct \mathbf{G} , and a list of target points, evaluates (4) at these target points using the quadrature rule in \mathbf{G} . [Matlab hint: vectorize the loop over targets.] Check it with the provided code `testevalDLP.m` which plots the \log_{10} of the error for above constant test case.

You should get something like this. Notice the utility of the $5h$ -rule: the result is accurate only at least $5h$ from the boundary, where h is the local node spacing. (See [3] to understand how the map Z distorts this from the $[0, 2\pi)$ case.)



Efficient and accurate quadratures for evaluation of u right up to the boundary in \mathbb{R}^2 or \mathbb{R}^3 is an active research area [2, 13, 3, 15]. For your amusement I include Laplace \mathbb{R}^2 close-evaluation codes improving upon [13]; see `testevalDLP_close.m`.

Exercise 10. Prove the above claim about the potential u generated by a unit DLP density for Laplace equation in \mathbb{R}^d .

3 Nyström method for solving the BIE, smooth kernel case

We now approximate the solution of an integral equation on the periodic interval $[0, 2\pi)$,

$$\sigma(t) + \int_0^{2\pi} K(t, s)\sigma(s)ds = f(t) \quad \text{for all } t \in [0, 2\pi)$$

noting that (7) is of this form. Kress [17, Ch. 12] is an excellent reference. We insert a fixed 2π -periodic quadrature rule s_j and w_j to get

$$\sigma(t) + \sum_{j=1}^N K(t, s_j)w_j\sigma(s_j) = f(t) \quad \text{for all } t \in [0, 2\pi) \quad (8)$$

We then let $\sigma_j = \sigma(s_j)$, and enforce equality at each node,

$$\sigma_i + \sum_{j=1}^N K(s_i, s_j)w_j\sigma_j = f(x_i), \quad i = 1, \dots, N.$$

This is a linear system for the vector $\sigma = \{\sigma_j\}_{j=1}^N$,

$$(I + A)\sigma = \mathbf{f} \quad (9)$$

with matrix elements

$$A_{ij} = K(s_i, s_j)w_j. \quad (\text{plain Nyström matrix elements}) \quad (10)$$

Remark 3.1 (FMM). Recall from (7) that K contains a speed factor, so in terms of the PDE kernel k (6) we have $A_{ij} = -2k(x_i, x_j)|Z'(s_j)|w_j$. The prefactor -2 is unimportant here; the key is that the “speed weights” $|Z'(s_j)|w_j$ depend only on the source point index j , so may be rolled into the dipole source strengths if the fast multipole method is used for the matvec $\sigma \rightarrow A\sigma$ in an iterative solver.

Having solved (9) for σ , there are two common ways to reconstruct an approximation to the function σ from the samples σ_j :

1. Nyström interpolation, which uses sections $K(\cdot, s_j)$ of the kernel itself, by reusing (8) [17, Sec. 12.2]. This is more accurate but less convenient.
2. Local high-order polynomial, or global spectral, interpolation.

The accuracy of the samples σ_j , and the Nyström interpolant, inherit the convergence properties of the underlying quadrature scheme applied to the integrands $K(s_i, \cdot)\sigma(\cdot)$. This result is due to Anselone and explained by Kress [17]. Eg, for Gauss panels, we expect order $2p$, but when local interpolation is used on panels we expect only order p . The second is the most common.

Remark 3.2. *If this underlying quadrature rule used for Nyström is sufficient for evaluation of u at the desired targets, no interpolation of σ is necessary; we simply use the existing samples σ_j .*

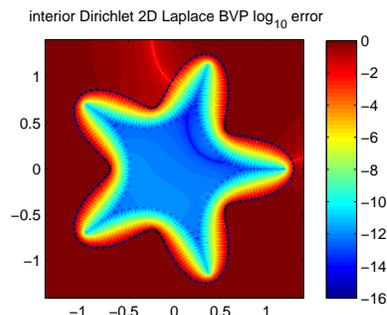
We illustrate this with the interior Dirichlet Laplace BVP in 2D. For convenience with fast direct solvers, our interface will be a function that returns just one matrix element.

```
function a = Dnystmatel(G,i,j)
% DNYSTMATEL - return one element of 2D Laplace double-layer Nystrom matrix D
if i~=j
    d = G.x(i)-G.x(j); a = (1/(2*pi)) * real(conj(G.nx(j))*d)/abs(d)^2;
else
    a = -G.cur(j)/(4*pi); % diagonal limit formula
end
a = a * G.sp(j)*G.w(j); % speed weights
```

The following short code `lapbvp.m` uses the above four routines to solve the BVP. We choose boundary data coming from the exact solution $f(x, y) = \log |(x, y) - (1, 1.5)|$.

```
G = smoothstar(.3,5); N = 150; G = curvquad(G,'ptr',N); % set up curve
f = @(z) log(abs(z-(1+1.5i))); % known Laplace soln in 2D (z is complex)
rhs = -2*f(G.x); % bdry data at nodes
A = nan(N,N); for i=1:N, for j=1:N, A(i,j) = -2*Dnystmatel(G,i,j); end, end
sigma = (eye(N) + A) \ rhs; % dense solve
g = -1.4:0.01:1.4; [xx yy] = meshgrid(g); t = xx + 1i*yy; % targets for plot
u = evalDLP(t,G,sigma); % evaluate soln
figure; imagesc(g,g,log10(abs(u-f(t)))); caxis([-16 0]); colorbar; axis xy equal
```

Here's the resulting error plot with the nodes also shown. Note that the error far from the boundary is around 10^{-11} ; this reflects errors in σ_j rather than due to evaluation too close to the boundary.



Remark 3.3. *Note that Galerkin methods are an alternative to Nyström methods. Since they require integration over twice as many dimensions as Nyström (eg, 4D for surfaces rather than 2D), yet are no more*

efficient in terms of N the number of unknowns, we avoid them. They are usually low-order, yet apparently have certain advantages in coupling to FEM, and for analysis.³

One huge advantage of BIE over direct discretization methods for solving BVPs is that they handle exterior problems with no extra effort.

Exercise 11. Figure out the (tiny!) changes to make to the above code solve the Laplace BVP *exterior* to Γ with the decay condition $u(x) = O(1)$ as $|x| \rightarrow \infty$ [17, Sec. 6.1]. Be sure to use boundary data for a known solution that obeys this condition. [Hint: you will notice $\dim \text{Null}(I+A) = 1$, but that the backwards-stable dense solve does not mind.]

Exercise 12. Wrap the above code in a loop over N to plot the convergence of the maximum solution error over a set of points far from the curve. What geometric quantities control the exponential convergence rate? (This is I believe an open question [3]).

Exercise 13. Test the accurate close-evaluation code by replacing the `evalDLP` call with the corresponding line from `testevalDLP_close.m`, and compare their accuracy at $N = 200$.

Exercise 14. [Advanced] Edit `curvquad.m` to include dyadically-refined panels towards $s = 0$, and test the BVP solution accuracy on a domain with a corner at $s = 0$.

4 Nyström quadratures for singular kernels

The above case of smooth kernel⁴ is very rare in applications, occurring only for the double-layer representations for Laplace and Stokes in \mathbb{R}^2 . Anything in \mathbb{R}^3 , and Helmholtz and/or more exotic boundary conditions in \mathbb{R}^2 , requires handling singular kernels, ie those which are non-smooth, or blow up, at $x = y$. We stick to *weakly singular* kernels, ie, those with an integrable singularity at $x = y$, that is, $|k(x, y)| \leq C/|x - y|^\gamma$ for some $\gamma < 1$ for curves in 2D ($\gamma < 2$ for surfaces in 3D).

The most common 2D case is $|k(x, y)| \leq C \log|x - y|$, which we illustrate with the exterior radiative Helmholtz BVP,

$$(\Delta + k^2)u = 0 \quad \text{in } \mathbb{R}^2 \setminus \bar{\Omega} \tag{11}$$

$$u = f \quad \text{on } \Gamma \tag{12}$$

$$\frac{\partial u}{\partial r} - iku = o(r^{-1/2}), \quad r := |x| \rightarrow \infty \tag{13}$$

which has a unique solution (see [7] for more details). If one chooses f the negative of an incident wave, then u is the scattered wave for a time-harmonic scattering problem at frequency k .

The 2D Helmholtz fundamental solution is

$$\Phi(x, y) = \frac{i}{4} H_0^{(1)}(k|x - y|) \tag{14}$$

A robust representation turns out to need an admixture of double- and single-layer potentials (the combined-field integral equation or “CFIE”), eg,

$$u(x) = \int_{\Gamma} \left(\frac{\partial \Phi(x, y)}{\partial n_y} - i\eta \Phi(x, y) \right) \sigma(y) ds_y \tag{15}$$

³Cultural note: researchers in the US tend to use Nyström (and read Kress) whereas those in Europe use Galerkin. (Everybody looks east?)

⁴In fact the 2D Laplace double-layer $K(s, t)$ was analytic with respect to s and t , for Γ an analytic curve.

abbreviated $u = (D - i\eta S)\sigma$. We choose $\eta = k$. The jump relations are the same as for Laplace, hence the BIE will be $(D - i\eta S - \frac{1}{2})\sigma = f$. The kernel of S has a singularity of type $\log|x-y|$, and the kernel of D , while continuous, is not smooth, having a singularity of type $|x-y|^2 \log|x-y|$. Thus we need new quadrature methods (even if η were zero the plain smooth formula (10) would give only 3rd-order convergence). See the last three lines of the next code for implementation of the kernel. Also see `evalCFIEhelm.m` which evaluates (15) at target points.

Summary: we care about (in 2D) kernels of the form smooth plus log-singular times smooth. We wish to find matrix elements A_{ij} such that

$$\sum_{j=1}^N A_{ij}\sigma_j \approx \int_0^{2\pi} K(s_i, s)\sigma(s)ds \quad \text{for all } i = 1, \dots, N \quad (16)$$

holds to high accuracy, so that the linear system well approximates the action of the operator given only the samples σ_j .

4.1 Local correction to the PTR weights (Kapur–Rokhlin)

Recall the “plain” PTR weights $w_j = 2\pi/N$ in (10). The simplest way to achieve high-order is to adjust these weights in a band around the diagonal in a manner that integrates accurately kernels of the form smooth plus smooth times a periodic log singularity [14] (and see our review [10, Sec. 3]). Here is the 6th-order scheme:

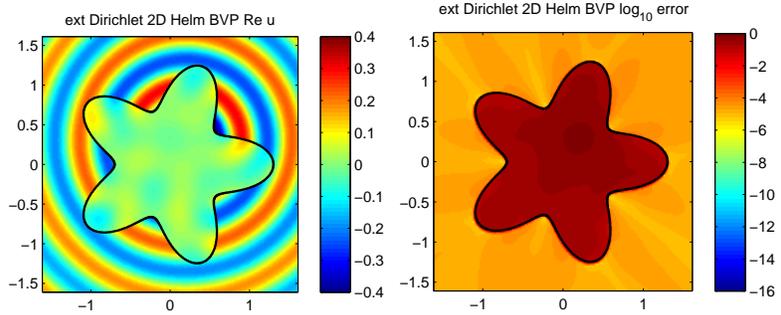
```
function a = CFIEnystKR(G,i,j,k,eta)
% CFIEENYSTKR - ij element of 2D Helmholtz CFIE Nystrom matrix, Kapur-Rokhlin corr
g6 = [4.967362978287758 -16.20501504859126 25.85153761832639 ...
      -22.22599466791883 9.930104998037539 -1.817995878141594]; % 6th order
sw = G.sp(j)*G.w(j); % speed weight
N = numel(G.x); l = mod(i-j,N); if l>N/2, l=N-l; end % index distance i to j
if l>0 & l<=6, sw = sw * (1 + g6(l)); end % apply correction to sw
if i==j, a = 0; return; end % kill diagonal
d = G.x(i)-G.x(j); kr = k*abs(d); % last 3 lines do CFIE kernel:
costheta = real(conj(G.nx(j)).*d)./abs(d); % theta angle between x-y & ny
a = (1i/4) * (k*costheta*besselh(1,kr) - 1i*eta*besselh(0,kr)) * sw;
```

Note that the multiplicative weight correction depends only on l , the index distance between i and j . The diagonal $i = j$ is set to zero, so no information about the kernel behavior there is used, and the price to pay is oscillatory weights of size up to 25 times their plain values. This loses digits and introduces spurious eigenvalues into A ; the 10th order rule is even worse and we don’t recommend it [10, Sec. 3].

The next code is the complete BVP solver `helmbvp.m`; compare it to `lapbvp.m`, noting for instance the sign change for exterior vs interior.

```
G = smoothstar(.3,5); N = 500; G = curvquad(G,'ptr',N); % set up curve
k = 10; eta = k; % wavenumber, SLP mixing amount
f = @(z) besselh(0,k*abs(z-(0.2+0.3i))); % known soln: interior source
rhs = 2*f(G.x); % bdry data at nodes
A = nan(N,N); for i=1:N, for j=1:N, A(i,j) = 2*CFIEnystKR(G,i,j,k,eta); end, end
sigma = (eye(N) + A) \ rhs; % dense solve
g = -1.6:0.02:1.6; [xx yy] = meshgrid(g); t = xx + 1i*yy; % targets for plot
u = evalCFIEhelm(t,G,sigma,k,eta); % evaluate soln
figure; imagesc(g,g,log10(abs(u-f(t)))); caxis([-16 0]); colorbar; axis xy equal
```

Here is the solution and error; we get only 5 digits, since the constant in the $O(N^{-6})$ is huge [10]. Due to Matlab’s overhead and slow Hankel implementation, the code is also around $100\times$ slower than it need be.

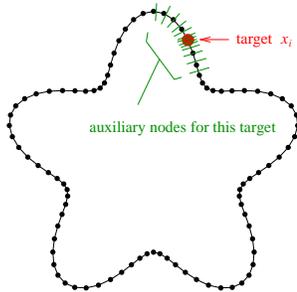


Remark 4.1. For surfaces in \mathbb{R}^3 , an analogous idea is due to Haroldsen–Meiron [11], and has been used in the vortex methods fluids community (Goodman, Beale, Siegel, Ambrose, etc). The Laplace double-layer kernel takes the form $k(x, y) = \frac{n_y \cdot (x - y)}{4\pi|x - y|^3}$, which on a general smooth surface has a $1/|x - y|$ singularity. The PTR is used in both coordinates on the surface, creating a regular grid, and the local weight corrections are found by implicitly writing down Richardson extrapolation from a grid of twice the spacing, giving $O(h^3)$. More extrapolations gives higher order. A smooth cut-off is needed if this is to be only a local correction.

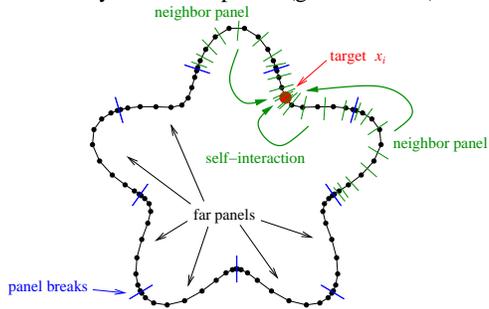
4.2 The auxiliary nodes idea (Kolm–Rokhlin, Alpert)

We return to (8) and notice that we could replace the fixed N nodes, which we call the “underlying” quadrature rule, with a new set of nodes specially adapted to the location of each desired target $t = x_i$ and the nature of the known singularity in $K(t, s)$. These auxiliary nodes are used to approximate the integral in (16). The auxiliary nodes lie only near the target point. The distant interactions (matrix elements with i far from j) continue to use the remainder of the underlying nodes, hence the plain rule. For logarithmic singularities on curves in \mathbb{R}^2 , this is common for both the PTR (using a trapezoid rule endpoint correction nodes due to Alpert [1]), and for Gauss panels (using Kolm–Rokhlin or “generalized Gaussian” auxiliary nodes):

Auxiliary node idea: PTR case (Alpert)



Auxiliary node idea: panels (gen. Gaussian)



Because of their local nature, only $O(N)$ Nyström matrix elements need to be corrected from the smooth formula (10), hence they are FMM-compatible.

The corrected entries require kernel and speed function evaluations at the auxiliary nodes, which are easy. They also need $\sigma(s)$ evaluations at the new nodes, which must be *interpolated* from the available unknowns σ_j , since otherwise new unknowns would have to be introduced. To be clear: the auxiliary nodes do *not* introduce new degrees of freedom; they merely allow one to fill Nyström matrix elements as in (16) to give a high-order solution. Interpolation is usually by a local Lagrange interpolation from the nearby nodes (for PTR) or from the set of p nodes on the source panel (for panels).

We review and test Alpert corrections and Kolm–Rokhlin panel corrections in [10]. My own code for Alpert corrections for 2D Helmholtz kernels can be found in `MPSpack` [4].

Exercise 15. Download the Alpert nodes and weights code `MPSpack/@quadr/QuadLogExtraPtNodes.m` and check that they integrate smooth plus log times smooth correctly.

4.2.1 Surfaces in \mathbb{R}^3

The same idea is used for surfaces in 3D. Underlying quadratures may either be product global PTR (for spheres and tori and their smooth deformations), or rectangular or triangular panels (which are better for adaptivity and connecting to engineering/CAD surfaces).

For an underlying PTR in both variables, or for very large panels for which it is unwieldy to cover with auxiliary nodes, the radial integration method of Bruno–Kunyansky [6] is successful. This uses the idea of a high-order smooth cut-off function (partition of unity) to split the integral into a near part, done via a new auxiliary set of nodes in polar coordinates (using the interpolant of the density), and a far part which uses the underlying PTR quadrature and the smooth formula (10).

With an underlying panel quadrature (triangles), Bremer has pioneered auxiliary node methods. Distant panel interaction matrix elements follow the smooth formula (10), whereas special auxiliary nodes on cut triangles are used for the self-interaction of a panel, and adaptive triangular subdivision is used for the interaction with neighbors. Many auxiliary nodes per panel are needed, but at least 12 digits is possible. See Bremer–Gimbutas [5].

4.3 The explicit split of log singularity idea

We can make better use of analytic information, and split the kernel $K(t, s)$ into the form

$$K(s, t) = K_1(s, t) \log\left(4 \sin^2 \frac{s-t}{2}\right) + K_2(s, t)$$

where K_1 and K_2 are both smooth (in fact analytic if Γ is) in both variables. The point of the strange log term is to factor out a 2π -periodic logarithmic singularity with a simple known Fourier series.

4.3.1 Global correction of the PTR weights (Kress)

For the Helmholtz kernels, with an underlying PTR, the analytic split idea has been favored by Kress, who gives the analytic form of K_1 and the needed diagonal limits $K_1(t, t)$ and $K_2(t, t)$ for the S and D operators [16]. We review this in [10, Sec. 6]. The second kernel K_2 is smooth, and the underlying weights are used. For the first kernel a *product quadrature* is used, based upon the Fourier series for the periodized log, giving a circulant matrix $R_{i-j} = R_j(s_i)$ where

$$R_j(t) = -\frac{4\pi}{N} \left[\sum_{n=1}^{N/2-1} \frac{1}{n} \cos n(s_j - t) + \frac{1}{N} \cos \frac{N}{2}(s_j - t) \right], \quad j = 1, \dots, N. \quad (17)$$

A simple code for the R_{ij} matrix is thus

```
n = N/2; m = 1:n-1; Rj = -2*pi*ifft([0 1./m 1./n 1./m(end:-1:1)]);
R = toeplitz([Rj(1) Rj(end:-1:2)], Rj);
```

Kress quadrature beats all other known schemes in terms of error convergence with N [10], when adaptive (panels) are not needed. These Kress quadratures are implemented in `MPSpack`.

Exercise 16. Use the above, and the formulae for K_1 and K_2 from [16], to code up the Kress scheme for the Helmholtz CFIE.

Remark 4.2. *Kress' scheme is not compatible with the conventional FMM, due to the dense circulant matrix; however, we expect it to still have low-rank off-diagonal blocks, hence be compatible with fast direct solvers.*

4.3.2 Explicit-split local panel corrections (Helsing)

Helsing recently had the idea to apply Kress’ explicit Helmholtz split to panel quadratures, hence construct self- and neighbor-interactions for panels for 2D Helmholtz *without* auxiliary nodes, building upon his Laplace and elasticity quadratures. He is then able to use panel schemes for log singularities based upon several creative ideas [13]: the path-independence of Cauchy integrals in the complex plane, representation of the density as a Taylor series in the complex plane z variable *not* with respect to the curve parameter s , recurrence relations for integrals of log times monomials, and backwards-stable solution of ill-conditioned Vandermonde matrices. This is intrinsically 2D, although he has adapted it for axisymmetric Helmholtz, and has cool ideas for 3D, eg [arXiv:1301.7276v1](https://arxiv.org/abs/1301.7276v1).

Helsing has a tutorial here which includes Helmholtz panel quadratures (but you’ll have to ignore the stuff about RCIP, a corner-compression scheme): <http://ctr.maths.lu.se/na/staff/helsing/Tutor/index.html>

4.4 PDE-based singular quadratures: QBX for \mathbb{R}^2 and \mathbb{R}^3

Quadrature by eXpansion (QBX) was recently invented [15], building on the close-evaluation scheme of Barnett [3] for the Laplace and Helmholtz equations.

The idea is that if the data f is smooth, then the solution u is smooth up to $\partial\Omega$ and analytically continues some distance beyond it. Therefore a P -term *local expansion* (analogous to a Taylor series), eg for Helmholtz,

$$u(x) \approx \sum_{m=-P}^P c_m J_m(k|x-x_0|) e^{i \text{angle}(x,x_0)}$$

(eg see Martinsson slides on FMM) about a “center” x_0 near $\partial\Omega$ converges up to and on $\partial\Omega$ as $P \rightarrow \infty$. The coefficients of this local expansion can be found as a boundary integral (via the addition formula), which for accuracy needs a refined (auxiliary) set of nodes on which the interpolated σ is needed. The refinement factor is $\beta = 2$ to 4. QBX has convergence proofs [3, 9].

QBX may be used as a local correction (self and neighbor panels), or as a *global* evaluation scheme eg extracting local expansions from the downwards FMM pass (ask the Greengard group). The advantage of the global version is that lower P can be used because the singularities at the edges of the neighboring panels present in any local scheme are absent. I don’t know if global QBX is compatible with fast direct solvers. QBX is essentially dimension-independent, ie, is just as easy in 3D as in 2D, and handles a variety of singular kernels. A write-up of the 3D performance is in progress.

References

- [1] B. K. Alpert, *Hybrid Gauss-trapezoidal quadrature rules*, SIAM J. Sci. Comput. **20** (1999), 1551–1584.
- [2] K. Atkinson and Y.-M. Jeon, *Algorithm 788: Automatic boundary integral equation programs for the planar Laplace equation*, ACM Trans. Math. Software **24** (1998), 395–417.
- [3] A. H. Barnett, *Evaluation of layer potentials close to the boundary for Laplace and Helmholtz problems on analytic planar domains*, SIAM J. Sci. Comput. **36** (2014), no. 2, A427–A451.
- [4] A. H. Barnett and T. Betcke, *MPSpack: A MATLAB toolbox to solve Helmholtz PDE, wave scattering, and eigenvalue problems*, 2008–2012, <http://code.google.com/p/mpspack>.
- [5] J. Bremer and Z. Gimbutas, *A Nyström method for weakly singular integral operators on surfaces*, J. Comput. Phys. **231** (2012), 4885–4903.

- [6] O. P. Bruno and L. A. Kunyansky, *A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications*, J. Comput. Phys. **169** (2001), 80–110.
- [7] David Colton and Rainer Kress, *Inverse acoustic and electromagnetic scattering theory*, second ed., Applied Mathematical Sciences, vol. 93, Springer-Verlag, Berlin, 1998.
- [8] P. J. Davis and P. Rabinowitz, *Methods of numerical integration*, Academic Press, San Diego, 1984.
- [9] C. L. Epstein, L. Greengard, and A. Klöckner, *On the convergence of local expansions of layer potentials*, 2012, [arXiv:1212.3868](https://arxiv.org/abs/1212.3868).
- [10] S. Hao, A. H. Barnett, P. G. Martinsson, and P. Young, *High-order accurate Nyström discretization of integral equations with weakly singular kernels on smooth curves in the plane*, Adv. Comput. Math. **40** (2014), no. 1, 245–272.
- [11] D. J. Haroldsen and D. I. Meiron, *Numerical calculation of three-dimensional interfacial potential flows using the point vortex method*, SIAM J. Sci. Comput. **20** (1998), no. 2, 648–683.
- [12] J. Helsing and R. Ojala, *Corner singularities for elliptic problems: Integral equations, graded meshes, quadrature, and compressed inverse preconditioning*, J. Comput. Phys. **227** (2008), 8820–8840.
- [13] ———, *On the evaluation of layer potentials close to their sources*, J. Comput. Phys. **227** (2008), 2899–2921.
- [14] S. Kapur and V. Rokhlin, *High-order corrected trapezoidal quadrature rules for singular functions*, SIAM J. Numer. Anal. **34** (1997), 1331–1356.
- [15] A. Klöckner, A. H. Barnett, L. Greengard, and M. O’Neil, *Quadrature by expansion: a new method for the evaluation of layer potentials*, J. Comput. Phys. **252** (2013), no. 1, 332–349.
- [16] R. Kress, *Boundary integral equations in time-harmonic acoustic scattering*, Mathl. Comput. Modelling **15** (1991), 229–243.
- [17] Rainer Kress, *Linear integral equations*, second ed., Appl. Math. Sci., vol. 82, Springer, 1999.
- [18] L. N. Trefethen, *Approximation theory and approximation practice*, SIAM, 2012, <http://www.maths.ox.ac.uk/chebfun/ATAP>.

Solutions to exercises

1. Easy: $e^{-5 \cdot 2\pi} \approx 10^{-14}$.
2. Exponential, rate is strip half-width $\alpha = 2 \operatorname{Im} \sin^{-1} ia$, gets smaller (worse) as $a \rightarrow 0^+$, and f becomes arbitrarily peaked. See `ptrconv.m`
- 3.
4. Exact integral is zero for $k \neq 0$. Proof for PTR uses $\sum_{j=1}^N e^{2\pi ijk/N} = 0$ for k a multiple of N , zero otherwise.
5. See [12, Sec. 5].
6. Gauss quadrature needs N at least a factor $\pi/2$ higher than the PTR.
7. See `curvquad.m`
8. Calculus.
9. See `evalDLP.m`
10. Choose $u \equiv 1$ and $v = \Phi(x, \cdot)$ in Green's 2nd identity [17, Example 6.16].
11. Main change is the sign of the identity, ie $I + 2D = 2f$.
- 12.
13. You should find close to machine precision for the close-evaluation code; see commented line in `lapbvp.m`
14. haven't tried this recently, but for non-reentrant corners only a few refinements are needed.
15. See `MPSpack/test/testalpertquadr.m`
16. See codes `MPSpack/@layerpot/D.m` etc