Precondition for Boundary Integral Equations*
(Preliminary Version)

Stephen A. Vavasis

TR 90-1098A
February 1990

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*This work was supported in part by a gift from Xerox Webster Research Center and in part by the U.S. Army Research Office through the Mathematical Sciences Institute of Cornell University.

Preconditioning for Boundary Integral Equations*
(Preliminary Version)

Stephen A. Vavasis
Department of Computer Science
Cornell University

(Revision of March 27, 1990)

Abstract

We propose new classes of preconditioners for the linear systems arising from a boundary integral equation method. The problem under consideration is Laplace's equation in three dimensions. The system arising in this context is dense and unsymmetric. Our preconditioners, which are based on solving small linear systems at each node, reduce the number of iterations in some cases by a factor of 20. Two iterative methods are considered: conjugate gradient on the normal equations and GMRES of Saad and Schultz. For a simple model problem, we demonstrate the exact relationship between the preconditioners and the resulting condition number of the system. Our analysis suggests that the condition number of the preconditioned system is decreased by a factor asymptotically greater than any constant.

*This work supported in part by a gift from Xerox Webster Research Center and in part by the U.S. Army Research Office through the Mathematical Sciences Institute of Cornell University

1 Boundary Integral Equations

The *boundary integral equation method* (BIEM) is a technique for solving certain kinds of boundary value problems including Laplace's equation. The specific problem under consideration is: given a compact connected region $\Omega \in \mathbb{R}^3$ find a solution to $\nabla^2\phi = 0$ given Neumann or Dirichlet boundary data. The Neumann and Dirichlet boundary data can be expressed in terms of one another via integral equations. By discretizing these integral equations, one arrives at a dense linear system relating the Neumann and Dirichlet data on the boundary. If Dirichlet data is given, it can be solved for the Neumann data or vice versa. Once both sets of boundary data are known, further integrations yield interior values of ϕ.

Our application for Laplace's equation is a free-surface irrotational incompressible fluid flow in three dimensions. The data determined by BIEM in this case is the normal derivative of the velocity potential on the free surface.

Accordingly, each time step of the fluid calculation requires a new solution of Laplace's equation, which in turn requires the solution of the dense unsymmetric system of linear equations arising from BIEM.

For the remainder of this section we explain the details of the integral equations and linear systems. The boundary integral equation method is credited to Hess and Smith [1967], Jaswon [1963], and Symm [1963]. For more information about the method, we refer the reader to the series edited by Brebbia [1986].

Suppose ϕ satisfies Laplace's equation. Let z be a point on $\partial\Omega$. Let $g_z(x)$ be the function

$$g_z(x) = \frac{1}{2\pi\|x - z\|}.$$

For now and for the rest of the paper, $\|\cdot\|$ will denote the 2-norm. Lower case letters are used for scalars and for 2- or 3-dimensional vectors. Boldface is used for vectors of higher dimension. Capital letters are used for matrices. With this definition of g, the following identity always holds

$$c\phi(z) = \int_{\partial\Omega} \frac{\partial\phi}{\partial n}(x) g_z(x) dS - \int_{\partial\Omega} \frac{\partial g_z}{\partial n}(x)\phi(x) dS. \tag{1}$$

Here c measures the angle at the point z and would be 1 if the boundary is smooth. There is an analogous equation for 2-dimensional Laplace problems with a different choice for the function g (called the *Green's function*). See Section 5. Note that the integrals in this equation are improper because the function g_z and its derivative are infinite at $x = z$. The integrals still have a well-defined value, but these singularities make numerical methods more complicated.

Suppose we divide the surface of Ω into triangular elements joined by nodes. Let the nodes be $\{z_1, \ldots, z_n\}$. Then an equation like (1) holds for each z_i. We can approximate the solution to the Laplace equation by assuming that the ϕ and $\partial\phi/\partial n$ are piecewise linear interior to each element and are therefore determined by their values at nodal points. This means that there are $2n$ variables $\phi(z_1), \ldots, \phi(z_n)$ and $\partial\phi/\partial n(z_1) \ldots, \partial\phi/\partial n(z_n)$. The integrations in (1) are approximated with Gaussian quadrature; the integral on each triangular element can be expressed as a linear combination of the variables at its vertices. This gives a dense linear system of n equations in the $2n$ variables; the system looks like this:

$$\begin{aligned} c_1\phi(z_1) &= \textstyle\sum_{j=1}^{n} d_{1j}\partial\phi/\partial n(z_j) - \sum_{j=1}^{n} e_{1j}\phi(z_j), \\ &\vdots \\ c_n\phi(z_n) &= \textstyle\sum_{j=1}^{n} d_{nj}\partial\phi/\partial n(z_j) - \sum_{j=1}^{n} e_{nj}\phi(z_j). \end{aligned} \tag{2}$$

In this system of equations the coefficients d_{ij} and e_{ij} arise from evaluating the Green's function. Accordingly, all of the coefficients (c's, d's and e's) are determined entirely by the geometry of $\partial\Omega$. See Atkinson [1985] for a description of methods to discretize this system.

In a well-posed Laplace problem, for each point on the boundary either $\partial\phi/\partial n$ or ϕ is specified at the point. Therefore, in the above system of equations, n out of the $2n$ variables will be specified and n will be unknown. Since (2) has n equations, we use these equations to solve for the remaining unknowns. This system will in general be dense and unsymmetric.

It follows that the system of equations that needs to be solved is determined entirely by $\partial\Omega$ and by the specification of which kind of boundary data is provided for each node.

Once the values of ϕ and $\partial\phi/\partial n$ are known, there is an equation similar to (1) to compute any interior value of ϕ.

In some applications of BIEM interior values are not needed. An application we are addressing is three-dimensional fluid flow governed by Laplace's equation with a moving boundary. In order to update the fluid position it is enough to know $\partial\phi/\partial n$ everywhere on the boundary.

The remainder of the paper is organized as follows. In Section 2 we discuss iterative methods for solving (2). In Section 3 we introduce our classes of preconditioners for this problem. In Section 4 we report on our computational experience with the algorithm. In Section 5 we give an analysis of one of the preconditioners for a simple model problem. We show that the condition number is reduced by a factor of $\sqrt{\ln n}$ for the model. Finally, in Section 6 we discuss computational complexity issues connected with the BIEM solution.

2 Iterative methods for linear systems

An *iterative* method for solving a system of linear equations $A\mathbf{x} = \mathbf{b}$ is a method that generates a sequence of vectors that converge to the true solution. Applied to a dense $n \times n$ system, one iteration of an iterative method ordinarily requires $O(n^2)$ steps. Solving the system with elimination requires $O(n^3)$ steps. Accordingly, the hope is that a good iterative method will require only a few iterations and be faster than elimination.

The two iterative methods under consideration in this paper are GMRES and conjugate gradient applied to the normal equations. The GMRES algorithm is due to Saad and Schultz [1986] and operates directly on the original system. Conjugate gradient, due to Hestenes and Stiefel [1952], only works for symmetric positive definite systems. Accordingly, CG is applied to $A^TA\mathbf{x} = A^T\mathbf{b}$ instead of $A\mathbf{x} = \mathbf{b}$. Note that the product A^TA is never actually formed—this would ruin the bound on the running time. Instead, whenever a product $A^TA\mathbf{y}$ is needed by conjugate gradient, we carry out the product as two matrix vector multiplications $A^T(A\mathbf{y})$. This remark about matrix products also applies to case described below when preconditioners are inserted into the problem.

It is known (see Golub and Van Loan [1989]) that the convergence rate of conjugate gradient can be bounded in terms of the condition number of the matrix of coefficients. There is no corresponding result known for GMRES, but our computational experience suggests that condition numbers play a role in that algorithm as well.

Suppose that one had a nonsingular matrix P such that PA had a lower condition number than A alone. Then the system of equations $PA\mathbf{x} = P\mathbf{b}$ has the same solution as $A\mathbf{x} = \mathbf{b}$. Moreover, an iterative method applied to

$PA\mathbf{x} = P\mathbf{b}$ might converge faster because of the lower condition number. Such a matrix P is called a *left preconditioner*. Ideally one wants to have PA close to the identity matrix.

Each iteration of conjugate gradient requires one matrix-vector multiply operation. Since we are working on the normal equations, we need two matrix-vector multiplies. Finally, if we use preconditioning then the system becomes $A^T P^T PA$ so we need four matrix-vector multiplies.

Each "outer" iteration of GMRES requires m matrix-vector multiplies, where m is a parameter specified by the user. For preconditioned GMRES $2m$ multiplies are required.

3 Preconditioners for BIEM

In this section we introduce three classes of preconditioners tailored to the BIEM problem described in the introduction. All three of our preconditioners are based on the same idea, which is as follows. The rows of preconditioner P are generated independently and can be done in parallel. Let the ith column of P^T be denoted as $\mathbf{p}_i$. As mentioned in the last section, we would ideally like to have $A^T\mathbf{p}_i = \mathbf{e}_i$, where $\mathbf{e}_i$ is the ith column of the identity matrix.

For our preconditioners we take the following approach. From the matrix A and other data we determine some small list L of indices drawn from $\{1, \dots, n\}$ such that the variables and constraints in L have the most impact on variable i. Then we solve the small system of equations $\bar{A}\bar{\mathbf{p}}_i = \bar{\mathbf{e}}_i$ where the bars over the variables indicate that we are deleting all the rows and columns except for the ones in L. Once this solution is known, we expand it back to the entries of P. This is done for all rows of P; the rows can be generated in parallel.

The brief description in the last paragraph completely explains how to compute our classes of preconditioners. All that remains is the explanation of how L is chosen and how to expand the solution of the small system back to the row of P. In all cases the number of flops to compute the preconditioner is equal to $O(k^3 n)$ where k is the average size of the small system solved for each node.

Preconditioner MN

Preconditioner MN stands for "mesh neighbor." Recall that each variable of the linear system arising in a BIEM problem corresponds to a node on the surface of the region. We say that two nodes are *neighbors* if there is an edge of the triangulation joining them. The choice of L for node i is as follows: we let L be i together with all the indices of neighbors of node i. After the small system involving these rows and columns is solved, then we scatter the entries of the solution vector $\bar{\mathbf{p}}_i$ back to their original coordinate positions in $\mathbf{p}_i$, and we fill in the remaining positions of $\mathbf{p}_i$ with zeros.

The motivation for Preconditioner MN is as follows. The matrix coefficient relating node i and node j comes up roughly from a formula like $1/\|x_i - x_j\|$. Therefore, the further apart two nodes are, the less impact we would expect a change at one node to have on the other. Since neighboring nodes are the most interrelated, we put them in our preconditioner.

Note that the preconditioner generated in this manner will be sparse, and in particular, its sparsity pattern will mirror the connectivity of the mesh. In general, sparsity is considered a very desirable feature of a preconditioner. See, for example, Manteuffel [1979] or Kolotilina and Yeremin [1986].

For our application sparsity is an added benefit but not a major goal. The reason is that A is already dense, so we have to carry out a dense $n \times n$ matrix-vector multiply operation each iteration. Therefore, the additional multiplication involving P is not too expensive relatively speaking even if P is dense.

Preconditioner ME

Preconditioner ME stands for "matrix entries." The way to choose L for a variable i for this preconditioner is as follows. If a_{ij}, a_{ji} satisfy $|a_{ij}a_{ji}| \geq t|a_{ii}a_{jj}|$ then include j in L. Here, t denotes some user-specified tolerance. Note that this preconditioner does not depend on any BIEM structure and hence could be applied to an arbitrary matrix.

After the small system is solved, the solution vector is scattered back into P as in Preconditioner MN.

This preconditioner is similar to the local matrix idea of Bensen and Frederickson [1982].

Preconditioner HC

Preconditioner HC stands for "hierarchical clustering." Hierarchical clustering, an idea due to Rokhlin [1985], classifies nodes based on how far away they lie from x_i.

The idea of hierarchical clustering is as follows. We say that nodes x_j for $j \in C$ lie in a *cluster* of diameter d if all the nodes are distance at most d from each other. Suppose that x_i is distance at least td from cluster C, where $t > 1$ is a user-specified parameter. Then the coupling in matrix A between the nodes of cluster C and node i will not depend very much on which node of C is selected. Therefore, we could summarize all the nodes of C with a single node.

Preconditioner HC divides the domain into a hierarchy of clusters. The largest cluster contains all the nodes. The second level clusters each contain about half the nodes of the top-level cluster and so on. The cluster splitting is done by passing planes through clusters to yield smaller clusters. The hierarchy of clusters is prepared before the computation of P.

Then the list L for a node i is determined as follows. A cluster C is said to be acceptable for i if the distance from x_i to all nodes in C is at least td, where d is the diameter of C. We check whether the top level cluster is acceptable for i (in general this cluster would never be acceptable). If so, then A will have the representative of that cluster. Otherwise, we recursively try out both clusters below the top level in the hierarchy. At each stage, if a cluster is acceptable then we include its representative. If it is unacceptable we branch to the next level. This procedure must terminate, because the bottom clusters in the hierarchy are individual nodes whose diameter is 0.

The list L is generated in this manner, and then the small system for i is solved. In Preconditioners MN and ME the elements of the small system solution were scattered throughout $\mathbf{p}_i$ and the remaining elements were filled with zeros. This preconditioner will be dense; the entry at position m of $\mathbf{p}_i$ is taken to be equal to the entry in $\bar{\mathbf{p}}_i$ corresponding to the representative of m's cluster.

This preconditioner is the most expensive and complicated of the three to compute.

4 Computational experience

We tried all three preconditioners out on three different test problems. For each problem we tried GMRES with $m = 4$, GMRES with $m = 8$, and conjugate gradient on the normal equations. We also noted the condition numbers of the system with and without preconditioners. These experiments were carried out using MATLAB. The results are summarized in Tables 1 to 3. The numbers in the three middle columns are number of iterations (in the case of GMRES, the number of "outer" iterations). The last column is the 2-condition number. In addition, a picture of the domain used to generate the matrix appears above each table. In all of the domains we used mixed boundary conditions.

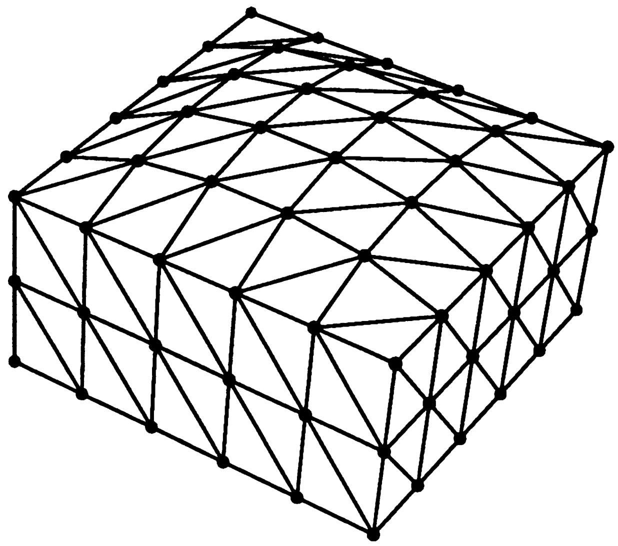
The most striking feature of these tables is the drastic reduction in the number of iterations when comparing preconditioned systems to the case with no preconditioner. In some cases the reduction is over a factor of 20.

The three preconditioners themselves seemed to have similar properties. We notice that Preconditioner ME generally did slightly worse than the other two. Preconditioner HC did the best in the domain reported in Table 3. One possible reason is that in the "pinched" section of the domain, Preconditioner MN does not allow for the fact that non-neighboring nodes may be closely spaced.

5 Analysis of a simple model problem

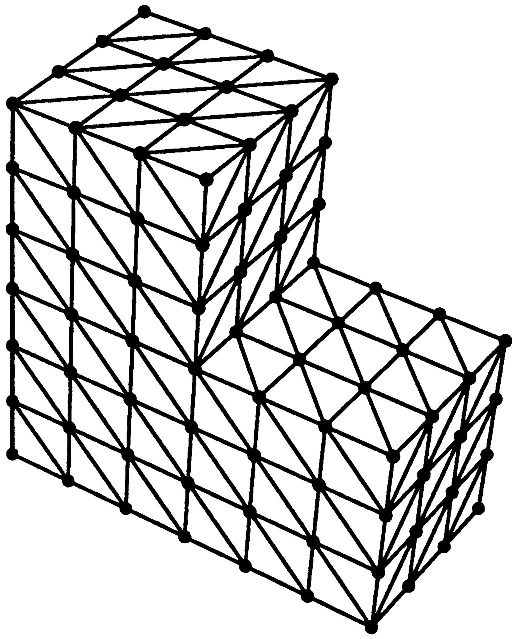
Let $\Omega \subset \mathbb{R}^2$ be the disk of radius ρ. We will try to solve Laplace's equation with Dirichlet boundary data on this region. We provide an asymptotic analysis of the condition number of the dense matrix and of Preconditioner MN. We will argue that there is an asymptotic reduction of $\sqrt{\ln n}$ in the condition number. The argument is not rigorous because we do not establish bounds to connect eigenvalues of the continuous problem with eigenvalues of the discretized problem. Such bounds are possible (see Atkinson [1976]), but we have not yet carried them out.

We observe that any real-valued function u satisfying Laplace's equation on the disk has a conjugate v such that $f(z) = u(z) + iv(z)$ is a complex analytic function. This function can therefore be expanded as a power series. Accordingly, we examine the effect of our preconditioner on the real



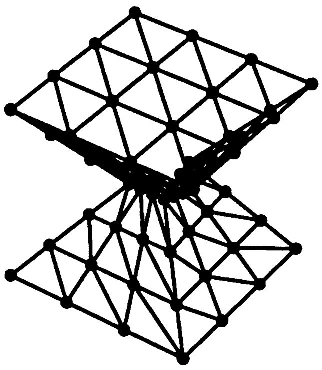
Precond.	GMRES $m=4$	GMRES $m=8$	C.G.	κ_2
none	249	86	184	45.7
MN	7	3	32	3.2
ME	8	4	35	3.8
HC	7	3	39	4.6

Table 1: Domain D_1 with $n = 92$.



Precond.	GMRES $m = 4$	GMRES $m = 8$	C.G.	κ_2
none	323	86	243	45.7
MN	7	3	45	5.3
ME	9	4	55	6.7
HC	6	3	49	5.3

Table 2: Domain D_2 with $n = 132$.



Precond.	GMRES $m=4$	GMRES $m=8$	C.G.	κ_2
none	305	86	243	75.1
MN	7	3	44	5.9
ME	9	4	41	9.6
HC	6	3	36	3.9

Table 3: Domain D_3 with $n = 92$.

part and imaginary parts of the complex analytic function $f(z) = z^k$. Let these functions be g_k, g'_k.

A computation shows that g_k maps the point at (r, θ) to $r^k \cos(k\theta)$. This point is mapped to $r^k \sin(k\theta)$ by g'_k. The boundary of Ω can be parameterized by

$$\gamma(\theta) = \rho(\cos\theta, \sin\theta).$$

We can easily compute the partial derivatives.

$$\begin{aligned}
\frac{\partial g_k}{\partial n}(\gamma(\theta)) &= \frac{\partial g_k}{\partial r}(\gamma(\theta)) \\
&= k\rho^{k-1}\cos(k\theta) \\
&= (k/\rho) g_k(\gamma(\theta))
\end{aligned}$$

A similar result holds for g'_k. Let L be the set of continuous real-valued functions on $\partial\Omega$. Let $\mathbf{A} : L \to L$ be the following integral operator. The function $\mathbf{A}(f)$ evaluated at point $\gamma(t)$ is defined to be:

$$\mathbf{A}(f)(\gamma(t)) = -\frac{\rho}{4\pi}\int_0^{2\pi} \ln(\|\gamma(t) - \gamma(s)\|^2) \cdot f(\gamma(s))\, ds$$

The kernel of this integral operator is the Green's function for a 2-dimensional Laplace equation.

Let $\mathbf{B} : L \to L$ be the following integral operator.

$$\mathbf{B}(f)(\gamma(t)) = \frac{\rho}{2\pi}\int_0^{2\pi} \frac{\langle \gamma(t) - \gamma(s), (\cos s, \sin s)\rangle}{\|\gamma(t) - \gamma(s)\|^2} \cdot f(\gamma(s))\, ds$$

The kernel here is the normal derivative of Green's function. The notation $\langle \cdot, \cdot \rangle$ denote the inner product on $\mathbb{R}^2$.

Then we have the identity

$$(\mathbf{I} + \mathbf{B})(f) = \mathbf{A}\left(\frac{\partial f}{\partial n}\right)$$

whenever f is a Laplace solution—this identity is the basis for the boundary integral equation method. Here, $\mathbf{I}$ denotes the identity operator. In particular, this holds for each function g_k. We know already the normal derivative of g_k; plugging this in gives

$$(\mathbf{I} + \mathbf{B})(g_k) = (k/\rho)\mathbf{A}(g_k)$$

and similarly for g_k'. We want to evaluate these integrals; the integral $\mathbf{A}(g_k)$ appears difficult to solve symbolically, so we work on $\mathbf{B}(g_k)$. If we substitute $\gamma(t) = \rho(\cos t, \sin t)$ then we get the formula

$$\begin{aligned}
\mathbf{B}(g_k)(\gamma(t)) &= \frac{\rho}{2\pi}\int_0^{2\pi} \frac{\rho(\cos t - \cos s)\cos s + \rho(\sin t - \sin s)\sin s}{\rho^2(\cos t - \cos s)^2 + \rho^2(\sin t - \sin s)^2} \cdot g_k(\gamma(s))\,ds \\
&= \frac{1}{2\pi}\int_0^{2\pi} \frac{\cos(s-t)-1}{2-2\cos(s-t)} \cdot \rho^k \cos(ks)\,ds \\
&= \frac{\rho^k}{2\pi}\int_0^{2\pi} (-1/2)\cos(ks)\,ds \\
&= 0
\end{aligned}$$

as long as $k \geq 1$. This is because $\cos(ks)$ goes through an integral number of periods provided $k \geq 1$. The result holds for g_k' as well.

Thus, $\mathbf{B}(g_k) = 0$ for $k \geq 1$, hence the preceding identity becomes

$$g_k = (k/\rho)\mathbf{A}(g_k).$$

This shows that for $k \geq 1$, g_k and g_k' are eigenfunctions of the operator $\mathbf{A}$ with eigenvalue equal to ρ/k.

When we discretize the problem with n evenly spaced nodes $v_1, \ldots, v_n$ around $\partial\Omega$, we would expect the discretization of operator $\mathbf{A}$, say matrix A, to have a similar property. In particular, we would expect that the vector whose entries are periodic with frequency k would have ρ/k as its eigenvalue This can only hold for k up to $n/2$, since beyond that point the periodic vector aliases to a lower frequency.

In fact, A is a circulant symmetric matrix, so its eigenvectors are indeed vectors of cosines and sines. In particular, if we define

$$\mathbf{w}_k = \begin{pmatrix} \cos 0 \\ \cos 2\pi k/n \\ \vdots \\ \cos 2\pi(n-1)k/n \end{pmatrix}$$

then $\mathbf{w}_k$ is an eigenvector. Similarly, $\mathbf{z}_k$ defined in terms of the sine function is an eigenvector. The analysis of the continuous case tells us that the eigenvalue of $\mathbf{w}_k$ or $\mathbf{z}_k$ will be roughly like ρ/k if $k \leq n/2$ and roughly like

n	$\kappa_2(A)$
10	8.89
20	17.69
40	35.31
80	70.54

Table 4: Condition numbers of A for a disk with $\rho = 2$

$\rho/(n+1-k)$ if $k \geq n/2$. Finally, we must figure out the eigenvector $\mathbf{w}_0$, i.e., the constant vector. The function $\mathbf{A}(g_0)$ comes out to be

$$\begin{aligned}\mathbf{A}(g_0)(\rho\cos t, \rho\sin t) &= -\frac{\rho}{4\pi}\int_0^{2\pi}\ln(\rho^2(2-2\cos(s-t)))\,ds \\ &= -\frac{\rho}{4\pi}\int_0^{2\pi}\ln(\rho^2)\,ds - \frac{\rho}{4\pi}\int_0^{2\pi}\ln(2-2\cos s)\,ds \\ &= -\rho\ln(\rho^2)/2\end{aligned}$$

The second integral in the second line is zero, as was verified with numerical integration. Thus, g_0, a constant function, is transformed to another constant function by $\mathbf{A}$.

Therefore, A is expected to have an eigenvalue for $\mathbf{w}_0$ that is independent of n.

We conclude that the condition number of A should grow proportionally with n since the eigenvector with the largest eigenvalue will be either $\mathbf{w}_0$ or $\mathbf{w}_1$ and will have fixed magnitude, and the smallest eigenvalue is at $\mathbf{w}_{n/2}$, when the magnitude will be proportional to $1/n$. In fact, this condition number result was established more rigorously and generally by Richter [1978], but we need to present analysis to estimate each eigenvalue.

These assertions are borne out by our computational experience; Table 4 indicates the condition number of A in the case $\rho = 2$ for varying values of n.

Next, we turn to preconditioner MN. Let us call the preconditioning matrix P. Note that A and P are symmetric in this case, so we can talk about row and columns equivalently. In the problem on the disk, each node has two nearest neighbors, therefore each column of P will have three nonzero entries. A way to describe the ith column of P is as follows. The

ith column of P is the solution to

$$A \begin{pmatrix} 0 \\ \vdots \\ 0 \\ p_{i-1,i} \\ p_{ii} \\ p_{i,i+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \times \\ \vdots \\ \times \\ 0 \\ 1 \\ 0 \\ \times \\ \vdots \\ \times \end{pmatrix}$$

where $\times$ denotes "don't care." The matrix A comes from an integral operator. The above equation is approximately the same as choosing a piecewise linear function on $\partial\Omega$ nonzero at nodes v_{i-1}, v_i, v_{i+1} and zero at all other nodes such that the result of applying the integral operator to this function is 0 at v_{i-1}, 1 at v_i, and 0 at v_{i+1}.

By symmetry $p_{i-1,i} = p_{i+1,i}$, so there are really two quantities that need to be determined; let us denote $p_{i,i}$ as x and $p_{i-1,i} = p_{i+1,i}$ as y. By symmetry these quantities are independent of i.

The continuous integral equations that need to be solved involve integrating terms like $\ln(\cos(t))$ which seem difficult to analyze. Accordingly, we simplify the problem by assuming that the five nodes $v_{i-2}, v_{i-1}, v_i, v_{i+1}, v_{i+2}$ are actually collinear points in $\mathbb{R}$ with spacing $2\pi\rho/n$. We are looking for a function f piecewise linear such that

$$\begin{aligned} f(v_{i-2}) &= 0, \\ f(v_{i-1}) &= y, \\ f(v_i) &= x, \\ f(v_{i+1}) &= y, \\ f(v_{i+2}) &= 0. \end{aligned}$$

There are three equations that need to be solved, but the first and third are equivalent by symmetry. These equations are:

$$-\frac{\rho}{4\pi} \int_{v_{i-2}}^{v_{i+2}} \ln|s - v_{i-1}|^2 \cdot f(s)\, ds = 0,$$

$$
\begin{aligned}
-\frac{\rho}{4\pi}\int_{v_{i-2}}^{v_{i+2}} \ln|s-v_i|^2 \cdot f(s)\,ds &= 1,\\
-\frac{\rho}{4\pi}\int_{v_{i-2}}^{v_{i+2}} \ln|s-v_{i+1}|^2 \cdot f(s)\,ds &= 0.
\end{aligned}
$$

Therefore, there are two equations in two unknowns, x and y. The above integrals can be analytically integrated by breaking each integration into four intervals. We omit the details but present the results, which are,

$$
\begin{aligned}
(\ln\epsilon + 3.89)x + (2\ln\epsilon + 7.17)y &= 0\\
(\ln\epsilon + 2.5)x + (2\ln\epsilon + 7.77)y &= c/\epsilon
\end{aligned}
$$

where $c = -2\pi\rho$ and $\epsilon = 2\pi\rho/n$. We focus on the first equation, which tells us that

$$
\begin{aligned}
x/y &= -\frac{2\ln\epsilon + 7.17}{\ln\epsilon + 3.89}\\
&= -2 + \frac{0.61}{\ln\epsilon + 3.89}
\end{aligned}
$$

As ϵ tends to zero we see that x/y tends to $-2 + c'/\ln\epsilon$ for some constant $c' > 0$. Ignoring the scaling factor we can take $y = 1$; then $x \approx -2 - c'/\ln n$, so preconditioning matrix P has the form $P = P_0 - (c'/\ln n)I$. Here, P_0 is the circulant matrix with -2's on the diagonal, 1's on the positions immediately adjacent to the diagonal, and I is the identity matrix. This is because, by definition, P has the entry x on the diagonal an y on the off-diagonal.

Since this matrix is also circulant and symmetric, the vectors $\mathbf{w}_k$ and $\mathbf{z}_k$ defined above are eigenvectors for P. Let us compute the eigenvalue of $\mathbf{w}_k$. Using some trigonometric identities we discover that $P_0\mathbf{w}_k = (2\cos(2\pi k/n) - 2)\mathbf{w}_k$. Therefore, the eigenevalue of P for $\mathbf{w}_k$ is

$$
2\cos(2\pi k/n) - 2 - c'/\ln n.
$$

The same holds for $\mathbf{z}_k$.

Figure 1 shows a plot of the magnitude of the eigenvalue versus k produced by MATLAB. The eigenvectors used to make this plot were the complex-valued eigenvectors $\mathbf{w}_k + i\mathbf{z}_k$. There are three sets of eigenvalues in the plot. The solid line is the eigenvalue in A; the dashed line is the eigenvalue in P; and the dotted line is the eigenvalue in PA. Notice how the eigenvalues of P are of large magnitude roughly in the places where the eigenvalues of A are small. This is the desired feature of the preconditioner. Note that these plots include some scaling factors not described above. Continuing our analysis, the eigenvectors of PA, the preconditioned system of equations, are also $\mathbf{w}_k$, and $\mathbf{z}_k$, and the kth eigenvalue for $1 \leq k \leq n/2$ is proportional to the negative of

$$e_k = \frac{2 - 2\cos(2\pi k/n) + c'/\ln n}{k}$$

Notice that e_k is always positive. We now intend to put upper and lower bounds on e_k.

In order to argue about this function of k and n, we note that there is a constant $c'' > 0$ such that

$$c''t^2 \leq 2 - 2\cos t \leq t^2$$

for all t between 0 and π. This follows from the Taylor series expansion.

Using the upper bound, we observe that

$$e_k \leq \frac{(2\pi)^2 k}{n^2} + \frac{c'}{k \ln n}.$$

Notice the right hand side is a function of k of the form $ak + b/k$. Such a function is always convex, hence the maximum is at one of the endpoints. The two endpoints in this case are $k = 1$ and $k = n/2$. Plugging both in, we find that $e_k \leq c^{(3)}/\ln n$ where $c^{(3)}$ is a constant.

For the lower bound, we use the fact that

$$e_k \geq \frac{(2\pi)^2 k}{c'' n^2} + \frac{c'}{k \ln n}.$$

Again, the right hand side is a convex function of k and the minimum is the point where both terms are equal, i.e., the point when

$$k^2 = \frac{c' c'' n^2}{(2\pi)^2 \ln n}.$$

At this point, k is proportional to $n/\sqrt{\ln n}$ and the lower bound on e_k is of the form $c^{(4)}/(n\sqrt{\ln n})$.

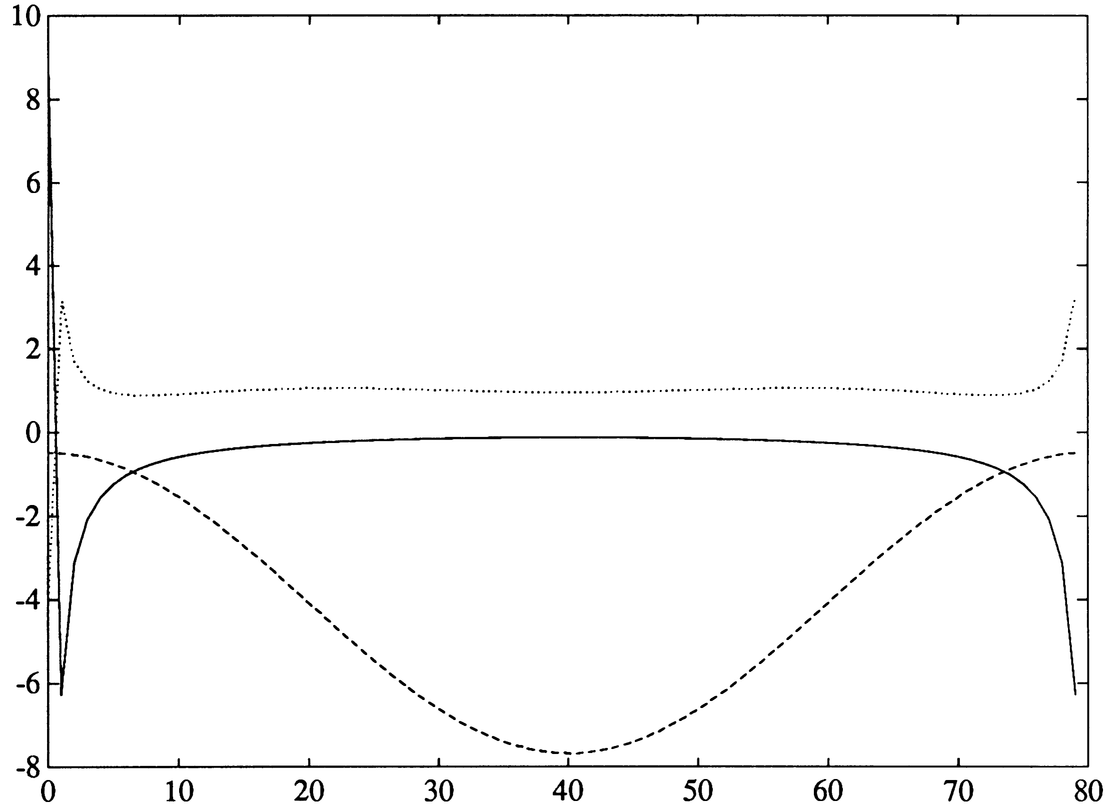


Figure 1: The eigenvalues of A, P and PA plotted against frequency

There is one eigenvector not accounted for by this analysis, namely, $\mathbf{w}_0$. Recall that for fixed ρ the eigenvalue of $\mathbf{w}_0$ for A was a constant. We compute that the eigenvalue of P is $c'/\ln n$. Therefore, this fits in with the upper and lower bounds.

The condition number of PA is bounded by the maximum eigenvalue over the minimum. We therefore conclude that

$$\kappa_2(PA) \leq \frac{c^{(3)}/\ln n}{c^{(4)}/n\sqrt{\ln n}}$$

which implies that $\kappa_2(PA) \leq O(n/\sqrt{\ln n})$. This concludes our claim that for a simple model problem, the condition number of A is reduced by a factor growing faster than any constant.

6 Complexity issues

Suppose we could establish a constant upper bound on the number of iterations required by a preconditioned iterative method for BIEM. This would imply a running time of $O(n^2)$ to solve the dense system of equations, clearly an improvement over the running time $O(n^3)$ of Gaussian elimination.

In fact, Rokhlin claims a running time of $O(n)$ to solve this system of equations. His argument has two parts: first, and most important, he shows an ingenious way to multiply the coefficient matrix A by an arbitrary vector in $O(n)$ steps (which is the main work in an iterative method). Second, he asserts that the number of iterations required by an iterative method is bounded by a constant independent of n.

This assertion, however, applies to an integral equation formulation based on potential layers and different from (1). In particular, we do not know of a method to recover $\partial\phi/\partial n$ on the boundary from Rokhlin's method given Dirichlet data. This normal derivative often has physical meaning as explained in the introduction.

Accordingly, it remains an interesting open problem to establish asymptotic complexity results on how long it takes to solve the BIEM system of linear equations arising from (1).

Acknowledgement

The author thanks Prof. Kendall Atkinson for providing some of the background material for this paper.

References

K. E. Atkinson [1976], *A Survey of Numerical Methods for the Solution of Fredholm Integral Equations of the Second Kind*, SIAM Publications, Philadelphia.

K. E. Atkinson [1985], Piecewise polynomial collocation for integral equations on surfaces in three diemnsions, *J. of Integral Equations*, 9:25-48.

M. W. Bensen and P. O. Frederickson [1982], Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems, *Utilitas Mathematica*, 22:127-140.

C. A. Brebbia [1986] (ed.), *Topics in Boundary Element Research*, Springer-Verlag, Berlin.

G. H. Golub and C. F. Van Loan [1989], *Matrix Computations*, Johns Hopkins University Press, Baltimore.

J. L. Hess and A. M. O. Smith [1967], Calculation of potential flow about arbitrary bodies, *Progress in Aeronautical Sciences, vol. 8* (D. Kückemann, ed.). Pergamon Press, London.

M. A. Jaswon [1963], Integral equation methods in potential theory, I, *Proc. Royal Soc. Ser. A* 275:23-32.

M. R. Hestenes and E. Stiefel [1952], Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Stand.* 49:409-36.

L. Yu. Kolotilina and A. Yu. Yeremin [1986], On a family of two-level preconditionings of the incomplete block factorization type, *Sov. J. Numer. Anal. Math. Model.* 4:293-320.

T. A. Manteuffel [1979], Shifted incomplete Cholesky factorization, *Sparse Matrix Proceedings, 1978* ed. I. S. Duff and G. W. Stewart, SIAM Publications, Philadelphia.

G. R. Richter [1978], Numerical solution of integral equations of the first kind with unsmooth kernels, *SIAM J. Numer. Anal.* 15:511-522.

V. Rokhlin [1985], Rapid solution of integral equations of classical potential theory, *J. Compu. Phys.* 60:187-207.

Y. Saad and M. H. Schultz [1986], GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comp.* 7:856-869.

G. T. Symm [1963], Integral equation methods potential theory, II, *Proc. Royal. Soc. Ser. A* 275:33-46.