SIAM J. SCI. COMPUT.
Vol. 20, No. 2, pp. 684–698

ON A CLASS OF PRECONDITIONING METHODS FOR DENSE LINEAR SYSTEMS FROM BOUNDARY ELEMENTS*

KE CHEN†

Abstract. In this paper we discuss several sparse-matrix-based preconditioners suitable for preconditioning dense linear systems of boundary element equations. All preconditioners involve only $O(n)$ nonzeros. We provide a framework for constructing operator-splitting-based preconditioners and use it to analyze a class of sparse preconditioners. For singular integral equations, a more efficient preconditioner is proposed that has a band-2 structure.

Key words. dense and sparse matrices, boundary integral equations, preconditioners, iterative methods, discrete wavelet transforms

AMS subject classifications. 65F10, 65R20

PII. S1064827596304058

1. Introduction.

We consider the efficient solution of dense linear systems $Au = f$ by a preconditioned iterative method. Here A is an $n \times n$ dense and nonsymmetric matrix. Such linear systems arise from the solution of boundary element equations.

The boundary element method (BEM) has proved to be a powerful alternative to finite element methods (FEMs) and finite difference methods (FDMs) for solving a class of partial differential equations (PDEs); see [4]. A lot of recent work also attempts to generalize BEM to solve more complex PDEs; see [16].

The main advantage of a BEM is its ability to reduce the problem of solving a PDE in a domain (closed or unbounded) to that of solving a boundary integral equation over the finite part of the boundary. Thus the dimension of the resulting problem is reduced compared to FEMs. For instance, in three-dimensional (3D) applications, rather than discretizing large portions of 3D space, only a two-dimensional (2D) manifold must be discretized. Discretization of boundary integral equations usually leads to a fully populated matrix, in contrast to sparse matrices which are often associated with FEMs and FDMs.

However, it is known that with a BEM the eigenvalues of these full matrices approximate those of its underlying integral operator. In particular we may say that the condition number of such matrices is independent of the number of discretization points. Here we assume that the underlying discrete operator has a norm convergence to the integral operator.

As far as iterative solution is concerned, its success, not surprisingly, largely depends on the spectral properties of the integral operator or of the matrices of discrete linear systems. This observation naturally suggests that we should study operator preconditioning in connection with matrix preconditioning. It is known that, if the underlying operator is smooth and compact, iterative methods can be very efficient without preconditioning; see [1], [10], and [12]. This paper addresses mainly the case of a noncompact operator associated with singular integral equations, where preconditioning is essential for iterative methods.

*Received by the editors May 20, 1996; accepted for publication (in revised form) April 26, 1997; published electronically September 11, 1998.
http://www.siam.org/journals/sisc/20-2/30405.html
†Department of Mathematical Sciences, M & O Building, The University of Liverpool, Liverpool L69 3BX, UK (k.chen@liv.ac.uk, http://www.liv.ac.uk/maths/applied/).

We remark that, for a general nonsymmetric matrix, an ideal eigenvalue distribution alone is not sufficient for fast convergence of most iterative solvers; see [15]. Fortunately we find that this can be sufficient for Fredholm integral equations applied by conjugate gradient iterations based on the normal equation (CGN). The reason may be explained as follows. Let the eigenvalues of the matrix $A = I - K$ approximate those of the operator $\mathcal{A} = \mathcal{I} - \mathcal{K}$. Then for a compact operator $\mathcal{K}$, its adjoint $\mathcal{K}^*$ and the product $\mathcal{K}^*\mathcal{K}$ are also compact. Hence the eigenvalues of the normal matrix A^*A approximate those of the normal operator $(\mathcal{I}-\mathcal{K}^*)(\mathcal{I}-\mathcal{K}) = \mathcal{I}-\mathcal{K}^*-(\mathcal{I}-\mathcal{K}^*)\mathcal{K}$, which is compact and thus has an ideal eigenvalue distribution. Note that such a result on the relationship of eigenvalues of A and A^*A may not hold for general matrices from other applications, such as PDE problems solved by FEMs.

In the literature, work on preconditioning of singular boundary element equations has mostly been based on algebraic considerations. The underlying ideas fall into two main categories: a) design a preconditioner that can be inverted or solved easily, and contains or represents somehow the dominant part of the contributions due to singular integrals; b) design a preconditioner that is sparse and somehow "close" to the inverse of the coefficient matrix A. Essentially, efficiency is the chief consideration, and naturally the preconditioners suggested are often sparse; refer to [2], [5], [6], [19], and [20], among others. These include the wavelet method as discussed in [3] and [8]. However, our concern here is not only efficiency but also effectiveness of preconditioning on eigenspectra.

In this paper, we first review some of these preconditioners and demonstrate their relationships. We then present our methods of integral operator splitting which are further used to construct new, and to analyze existing, preconditioners. Our theory aims to provide a justification for a class of preconditioners and points a way toward further modifications and new designs. Some numerical results are reported.

2. Boundary elements and dense linear systems. Let $\Omega \in R^2$ denote a closed 2D domain that may be interior and bounded, or exterior and unbounded, and $\Gamma = \partial\Omega$ be its (finite-part) boundary that can be parameterized by $p = (x, y) = (x(s), y(s)), a \le s \le b$. The R^3 case can be studied similarly. Then the boundary integral equation that usually arises from reformulating a PDE in Ω can be written as

$$U(p) - \int_\Gamma \bar{k}(p,q)U(q)dS_q = f(p), \qquad p \in \Gamma, \tag{1}$$

or

$$U(s) - \int_a^b k(s,t)U(t)dt = f(s), \qquad s \in [a,b], \tag{2}$$

or simply

$$(I - \mathcal{K})u = f. \tag{3}$$

Here, if the kernel function $k(s,t)$ is continuous or weakly singular, then the operator $\mathcal{K}$ is compact; however, if $k(s,t)$ is strongly singular, $\mathcal{K}$ is no longer compact (see [14]).

To solve the above equation numerically, divide the boundary Γ (interval $[a,b]$) into m boundary elements (nonintersecting subintervals $E_i = [s_{i-1}, s_i]$). On each interval E_i, we may either approximate the unknown u by an interpolating polynomial of order η, which leads to a collocation method, or apply a quadrature method (say the Gauss–Legendre rule) of η nodes and weights w_i, which gives rise to the Nyström method. Both discretization methods approximate (3) by

$$(I - \mathcal{K}_n)u_n = f, \tag{4}$$

where we can write

$$\mathcal{K}_n u = \mathcal{K}_n u_n = \sum_{j=1}^{m} \left[\sum_{i=1}^{\eta} w_i k(s, t_{ji}) u_{ji} \right], \quad u_n(t_{ji}) = U(t_{ji}) = u_{ji}, \quad \text{and} \quad n = m\eta.$$

For the collocation method, the weights w_i come from integration of the basis functions.

We use, for simplicity, vector u to denote u_{ji} at all nodes. By a collocation step in (4), we obtain a linear system of equations

$$(I - K)u = f, \qquad \text{or} \qquad Au = f, \tag{5}$$

where the matrices K and A are dense and, in general, not symmetric. The conditioning of A depends on the smoothness of the kernel function $k(s,t)$. A strong singularity (as $t \to s$) leads to noncompactness of the operator $\mathcal{K}$ and renders (5) difficult to solve by iterative methods without preconditioning. In contrast, when $\mathcal{K}$ is compact, both the CGN and multigrid methods are efficient even without preconditioning; see [1], [10], and [12]. When $\mathcal{K}$ is noncompact, both CGN and multigrid methods have been found converging very slowly.

3. Preconditioning techniques. For (5), i.e., $Au = f$, denote a preconditioner by M. Then a (left) preconditioned system can be written as $MAu = Mf$. In what follows we shall construct the preconditioner M either directly or indirectly through finding its inverse M^{-1}. Correspondingly, a preconditioning step will be denoted by $x = My$ or $M^{-1}x = y$, respectively. In general, two requirements are considered. First, a system $Mx = y$ for any vectors x, y should be efficiently solved in less than $O(n^2)$ operations, or $O(n)$ operations if a more sophisticated method like panel clustering is also used. This normally means that M or its inverse must be sparse, and so it is natural to seek sparse preconditioners. Second, the preconditioned matrix MA should possess some better properties than A does, for example, its eigenvalues are more clustered. We hope the condition number of MA will often be smaller than that of A. In this sense, the best preconditioner should approach the inverse of A.

Indeed, both requirements are reflected (at least in part) in the preconditioning techniques discussed below. The sparsity of all preconditioners will be illustrated so that the implementation will become more apparent.

3.1. Two grid based sparse column preconditioners. The success of this preconditioner, due to Yan [20], follows from the fact that an $n \times n$ sparse column matrix B, with m nonzero long columns, and its inverse B^{-1} have identical sparsity; and a system such as $Bx = y$ for $x, y \in R^n$ can be solved in only $(n-m)m$ operations. For example, with $n = 9$,

$$B = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} = \begin{pmatrix} \times & & \times & & \times & & \times \\ & \times & \times & & \times & & \times \\ & & \boxed{\times} & & \boxed{\times} & & \boxed{\times} \\ & & \times & \times & \times & & \times \\ & & \times & & \times & & \times \\ & & \boxed{\times} & & \boxed{\times} & & \boxed{\times} \\ & & \times & & \times & \times & \times \\ & & \times & & \times & & \times \\ & & \boxed{\times} & & \boxed{\times} & & \boxed{\times} \end{pmatrix},$$

where each B_{ij} denotes a 3×3 block matrix. Formally, let $n = \eta m$ with η, m integers. Then from the matrix $K = (k_1, k_2, \ldots, k_n)$ of (5), construct column vectors by

$$k'_i = \begin{cases} k_i & \text{if } \ i = \ell\eta, \quad 1 \le \ell \le m, \\ 0 & \text{otherwise,} \end{cases}$$

and define a new matrix by $K' = (k'_1, k'_2, \ldots, k'_n)$. Then use M as a preconditioner with $M^{-1} = A' = (I + \eta K')$. It may be expected that A' is "close" to $A = A_n$ because $A' \approx A_m$, which is the corresponding discrete matrix with m nodes.

It can be shown that both the inverse of M and its LU decomposition are of the same sparsity structure. Therefore, the solution of $M^{-1}x = y$ reduces to that of a smaller system $M^+x^+ = y^+$, where M^+ is an $m \times m$ principal submatrix of M^{-1} (i.e., of those boxed elements in the above case of $n = 9$ with $m = 3$). Formally, define a restriction operator R_m from n-dimensional (complex) space $\mathbf{C}^n$ to $\mathbf{C}^m$ by $R_m u = (u_\eta, u_{\eta 2}, \ldots, u_{\eta m})^T$ for $u = (u_1, \ldots, u_n)^T \in \mathbf{C}^n$ and a prolongation operator P_m from $\mathbf{C}^m$ to $\mathbf{C}^n$ by $P_m v = (v'_1, \ldots, v'_n)^T$ for $v = (v_1, \ldots, v_m)^T \in \mathbf{C}^m$, where $v'_k = v_k$ for $k = \eta\ell$, otherwise $v'_k = 0$.

ALGORITHM 1 (Two grid for $M^{-1}x = y$).

1. *Solve for v_m from an $m \times m$ system $(I + \eta K_m)v_m = R_m y$, where K_m is the submatrix extracted from K', corresponding to M^+.*
2. *Evaluate $x = y - (I + \eta K' P_m)v_m$.*

Here $M^{-1}x = y$ is used to represent the essential preconditioning step because, in iterative methods, matrices (A and M) are only used in matrix-vector multiplications.

We remark that the idea used here for approximately evaluating matrix elements, originally motivated by two grid methods, is in a way related to the panel clustering and multipole methods; see [11] and the references therein. Here we exploit the fact that neighboring elements on each row of A have come from the discretization of an integral over a certain subinterval, while in panel clustering and multipole methods we exploit the fact that each row of A multiplied by the solution vector u represents the approximation to an integral which may be evaluated in alternative ways and leads to fast matrix-free multiplications.

3.2. Mesh-neighbor-based approximate inverses. The starting point in the mesh-neighbor preconditioner of Vavasis [19] is the hope of finding a matrix M such that $AM \approx I$ and the diagonal elements of M and their immediate neighbors have more importance, since A usually comes from singular integral equations. In particular, assume that M is a quasi-tridiagonal matrix; for $n = 9$ we have

$$M = \begin{pmatrix} \times & \times & & & & & & & \times \\ \times & \times & \times & & & & & & \\ & \times & \times & \times & & & & & \\ & & \times & \times & \times & & & & \\ & & & \times & \times & \times & & & \\ & & & & \times & \times & \times & & \\ & & & & & \times & \times & \times & \\ & & & & & & \times & \times & \times \\ \times & & & & & & & \times & \times \end{pmatrix}. \tag{6}$$

Then to find $M = (p_1, p_2, \ldots, p_n)$ in terms of $A = (a_1, a_2, \ldots, a_n)$, a direct approach is used. Let $AM = I \iff Ap_i = e_i$. Then, as p_i only has three nonzero positions,

i.e., $p_{i_1}, p_{i_2}, p_{i_3}$, an approximate solution is obtained by solving a 3×3 system

$$\begin{pmatrix} A_{i_1 i_1} & A_{i_1 i_2} & A_{i_1 i_3} \\ A_{i_2 i_1} & A_{i_2 i_2} & A_{i_2 i_3} \\ A_{i_3 i_1} & A_{i_3 i_2} & A_{i_3 i_3} \end{pmatrix} \begin{pmatrix} p_{i_1} \\ p_{i_2} \\ p_{i_3} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

where $i_1 = i-1, i_2 = i, i_3 = i+1$ for columns $i = 2, \ldots, n-1$, $i_1 = n, i_2 = 1, i_3 = 2$ for $i = 1$ and $i_1 = n-1, i_2 = n, i_3 = 1$ for $i = n$, due to the wraparound nature of M.

ALGORITHM 2 (Mesh neighbors for $x = My$).

1. *(once and only once) For each column* $i = 1, \ldots, n$,

   set i_1, i_2, i_3*, and form the* 3×3 *submatrix* W *as above. Set* $e = (0, 1, 0)^T$*. Then solve a* 3×3 *system* $Wc = e$*. Set* $p_i = c$ *(the* i*th column of* M*).*

2. *Evaluate* $x = My$.

3.3. Approximate inversion techniques. Sparse approximate inverse preconditioners have been proposed for preconditioning sparse linear systems [9], [13]. They are here adapted for applications to dense linear systems $Au = f$.

The problem is to construct an approximation, M, to the inverse of the matrix A for which $\|AM - I\|$ is small in some norm. Once such an M has been constructed; we solve the preconditioned linear system $AMw = f$, $u = Mw$.

To specify the matrix M, write $\mathcal{N} = \{1, 2, \ldots, n\}$, let $\mathcal{S}$ be a given set of (i, j)s with $(i, j) \in \mathcal{N}$, i.e., a subset of $\mathcal{N} \times \mathcal{N}$, and $\mathcal{G}_{\mathcal{S}}$ be the space of all $n \times n$ matrices that have entries in positions indexed by $\mathcal{S}$, and zeros outside $\mathcal{S}$. For each column index $j = 1, \ldots, n$, define its set of row indices $\mathcal{S}_j = \{i : (i, j) \in \mathcal{S}\}$ and let its vector space $\mathcal{G}_{\mathcal{S}_j}$ contain all vectors that have entries in positions indexed by $\mathcal{S}_j$. Then the approximate inverse M is calculated by solving the least squares (LS) problem

$$\min_{M \in \mathcal{G}_{\mathcal{S}}} \|AM - I\|_F^2 = \sum_{j=1}^{n} \min_{m_j \in \mathcal{G}_{\mathcal{S}_j}} \|Am_j - e_j\|_2^2, \tag{7}$$

where $M = [m_1, \ldots, m_n]$ and $I = [e_1, \ldots, e_n]$. In theory, such an LS problem can be posed in any norm, but the Frobenius norm leads to an easier solution. The full problem of finding M is reduced to n standard LSs problems.

For singular integral equations, it seems reasonable to select $\mathcal{S}$ from the same sparsity pattern as (6).

ALGORITHM 3 (Approximate inversion for $x = My$).

1. *(once and only once) For each column* $i = 1, \cdots, n$,

   set i_1, i_2, i_3*, and form the* $n \times 3$ *submatrix* W *as above. Set* $e = e_i$ *the* i*th unit vector. Then solve an* $n \times 3$ *LS system* $Wc = e$ *by the QR method. Set* $p_i = c$ *(the* i*th column of* M*).*

2. *Evaluate* $x = My$.

3.4. Sparse LU decompositions. As with the method of mesh-neighbor-based approximate inverses, [2] assumes that the inverse of preconditioner M should have its diagonal elements and immediate neighbors possess more importance for the same reason, namely that A usually comes from singular integral equations; see also [6]. When $n = 9$, such a matrix M^{-1} is of the same sparsity as that in (6).

Here we form M^{-1} explicitly from A with $M_{ij}^{-1} = A_{ij}$ for $|i - j| \leq 1$ and $i = 1, \ldots, n$, $M_{1n}^{-1} = A_{1n}$, and $M_{n1}^{-1} = A_{n1}$. As M cannot maintain any sparsity property of M^{-1}, it is proposed, in order to solve $M^{-1}x = y$ for $x, y \in R^n$, to decompose M^{-1}

as $M^{-1} = LU$, where L and U are sparse triangular matrices. To illustrate, for $n = 9$, we have

$$M^{-1} = LU = \begin{pmatrix} P_{1,1} & & & & & & & & \\ P_{2,1} & P_{2,2} & & & & & & & \\ & P_{3,2} & P_{3,3} & & & & & & \\ & & P_{4,3} & P_{4,4} & & & & & \\ & & & P_{5,4} & P_{5,5} & & & & \\ & & & & P_{6,5} & P_{6,6} & & & \\ & & & & & P_{7,6} & P_{7,7} & & \\ & & & & & & P_{8,7} & P_{8,8} & \\ P_{9,1} & P_{9,2} & P_{9,3} & P_{9,4} & P_{9,5} & P_{9,6} & P_{9,7} & P_{9,8} & P_{9,9} \end{pmatrix} \times \begin{pmatrix} 1 & P_{1,2} & & & & & & & P_{1,9} \\ & 1 & P_{2,3} & & & & & & P_{2,9} \\ & & 1 & P_{3,4} & & & & & P_{3,9} \\ & & & 1 & P_{4,5} & & & & P_{4,9} \\ & & & & 1 & P_{5,6} & & & P_{5,9} \\ & & & & & 1 & P_{6,7} & & P_{6,9} \\ & & & & & & 1 & P_{7,8} & P_{7,9} \\ & & & & & & & 1 & P_{8,9} \\ & & & & & & & & 1 \end{pmatrix}.$$

A general algorithm is given as follows (note that only nonzero elements are stored).

ALGORITHM 4 (Sparse LU for $M^{-1}x = y$).

1. *(once and only once) Calculation of sparse LU factorization* $M^{-1} = LU$*;*
2. *forward substitution* $Lz = y$*;*
3. *backward substitution* $Ux = z$*.*

3.5. Discrete wavelet transforms. Wavelets are a class of localized functions that give rise to a basis of the space L_2. They can be used to discretize integral equations, leading to sparse matrices of entries which are larger than a small threshold tolerance. An alternative approach is to view wavelets as functions mapping periodic sequences (specifically rows and columns of A) $R^n \mapsto R^n$ linearly, and to have $\tilde{A} = WAW^T$ nearly sparse, where $Au = f$ comes from using a conventional boundary element discretization.

With the latter approach, we may consider two solution methods:

1. Apply an iterative method to $\bar{A}y = Wf$ where $u = W^T y$, as the multiplication $z = \bar{A}v$ only takes $O(n \log(n))$;
2. Use $\bar{A}$ as a preconditioner to $\tilde{A}y = Wf$,

where $\bar{A}$ is a sparse matrix taken from $\tilde{A}$ by use of a threshold.

Let m be the order of compactly supported wavelets, with $m/2$ vanishing moments, $n = 2^L$, and r an integer such that $2^r < m$ and $2^{r+1} \geq m$. Denote by $s^{(L)}$ a column vector of A at the wavelet level L. Then

$$\begin{array}{ccccccccccc} s^{(L)} & \searrow & s^{(L-1)} & \searrow & s^{(L-2)} & \searrow & \cdots & \searrow & s^{(\nu)} & \searrow & \cdots & \searrow & s^{(r)}, \\ & & d^{(L-1)} & & d^{(L-2)} & & & & d^{(\nu)} & & & & d^{(r)}, \end{array}$$

where the vectors $s^{(\nu)}$ and $d^{(\nu)}$ are of length 2^ν. Notice that the sum of these lengths gives the total $n = 2^L$ from the simple equality

$$2^L = 2^r + 2^r + 2^{r+1} + 2^{r+2} + \cdots + 2^{L-1}.$$

Note $r = 0$ for $m = 2$ (Haar wavelets) and $r = 1$ for $m = 4$ (Daubechies order-4 wavelets). Denote the transformed vector by

$$\begin{aligned} w &= \left[(s^{(r)})^T\ (d^{(r)})^T\ (d^{(r+1)})^T\ \cdots\ (d^{(L-1)})^T\right]^T \\ &= P_{r+1}W_{r+1}\cdots P_{L-1}W_{L-1}P_LW_Ls^{(L)} = Ws^{(L)}, \end{aligned}$$

where

$$P_\nu = (\bar{P}_\nu \quad J_\nu)$$

with $\bar{P}_\nu$ a permutation matrix of size $2^\nu = 2^L - k_\nu$, that is, $\bar{P}_\nu = I(1, 3, \ldots, 2^\nu - 1, 2, 4, \ldots, 2^\nu)$,

$$W_\nu = \begin{pmatrix} \bar{W}_\nu & \\ & J_\nu \end{pmatrix}$$

with an orthogonal (sparse) matrix of size $2^\nu = 2^L - k_\nu$, and J_ν is a unit diagonal matrix of size k_ν. Note that $k_L = 0$ and $k_\mu = k_{\mu+1} + 2^\mu$ for $\mu = L-1, \ldots, r+1$. With $m = 4$ (Daubechies order-4 wavelets), we have

$$\bar{W}_\nu = \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & & & & & & \\ g_0 & g_1 & g_2 & g_3 & & & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & & & \\ & & g_0 & g_1 & g_2 & g_3 & & & & \\ & & & & \ddots & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \ddots & \ddots & & \\ & & & & & & h_0 & h_1 & h_2 & h_3 \\ & & & & & & g_0 & g_1 & g_2 & g_3 \\ h_2 & h_3 & & & & & & & h_0 & h_1 \\ g_2 & g_3 & & & & & & & g_0 & g_1 \end{pmatrix},$$

where the filtering coefficients $\{h_i, g_i\}$ are known. In the tests below, we shall use $m = 4$ and study 2-level wavelets. We remark that for a general order m, there are only $L - r + 1$ wavelet levels and not L levels (as some references seem to suggest) unless we use Daubechies' order-4 wavelets $r = 1$). For Haar wavelets ($r = 0$), there are $L + 1$ levels.

A similar discussion can be put forward for transforming rows of the matrix A. The resulting matrix $\tilde{A} = WAW^T$ can be shown to be sparse at some threshold; see [3] and [8]. As far as iterative methods are concerned, we make the following remark.

REMARK 1.

1. *The matrix $\bar{A}$, truncated from $\tilde{A}$ after using a threshold, is sparse. Assume that $\bar{A}$ has V nonzeros where $V = O(n)$ (or $O(n \log n)$). Then the matrix-vector multiplication $\bar{A}x$ may only take V operations. Therefore, if an iterative method for $\bar{A}u = f$ only involves matrix-vector multiplications and the iterative solver is convergent, the wavelet method can be ideal because of an efficiency gain due to matrix compression.*
2. *As W is orthogonal, the eigenspectrum (or the condition number) of $\tilde{A}$, and thus most likely that of $\bar{A}$, is the same as that of A. Therefore, if the original matrix A requires preconditioning, due to some singular behavior of the underlying integral operator, the matrix $\bar{A}$ also needs suitable preconditioning.*

3. *If we decide to use the iterative method of CGN* [15]*, the application of wavelet transforms to columns of A may be sufficient because the normal equations of* WA *and* WAW^T *(hence approximately that of* $\bar{A}$*) are the same.*

Previously only heuristic accounts of the above sparse preconditioners have appeared in the literature. In order to understand and analyze them, we now consider operator splittings or preconditioning of singular integral operators.

4. Operator splitting strategies. Here we introduce the idea of operator splitting that originated in [6] and use the partition previously defined, $[a,b] = \bigcup_{i=1}^{m} E_i$. Accordingly we can partition the variable U and the vector u as follows: $U = (U_1, U_2, \ldots, U_m)^T$ and $u = (u_1, u_2, \ldots, u_m)^T$.

4.1. Splitting strategy 1. The operator $\mathcal{K}$ can be partitioned into matrix form as for vectors, and further, we can observe that all singularities of $\mathcal{K}$ are contained in the following operator:

$$\bar{\mathcal{K}} = \begin{pmatrix} \mathcal{K}_{1,1} & \mathcal{K}_{1,2} & & & \mathcal{K}_{1,m} \\ \mathcal{K}_{2,1} & \mathcal{K}_{2,2} & \mathcal{K}_{2,3} & & \\ & \mathcal{K}_{3,2} & \ddots & \ddots & \\ & & \ddots & \ddots & \mathcal{K}_{m-1,m} \\ \mathcal{K}_{m,1} & & & \mathcal{K}_{m,m-1} & \mathcal{K}_{m,m} \end{pmatrix}.$$

The corresponding matrix arising from K is

$$\bar{K} = \begin{pmatrix} K_{1,1} & K_{1,2} & & & K_{1,m} \\ K_{2,1} & K_{2,2} & K_{2,3} & & \\ & K_{3,2} & \ddots & \ddots & \\ & & \ddots & \ddots & K_{m-1,m} \\ K_{m,1} & & & K_{m,m-1} & K_{m,m} \end{pmatrix}.$$

Define operators $\mathcal{D} = I - \bar{\mathcal{K}}$ and $\mathcal{C} = \mathcal{K} - \bar{\mathcal{K}}$ and corresponding matrices $D = I - \bar{K}$ and $C = K - \bar{K}$. Then it can be shown that the operator $\mathcal{D}$ is bounded and $\mathcal{C}$ is compact under certain conditions. As the operator $\mathcal{D}^{-1}\mathcal{C}$ is also compact, and since 0 is the only possible point of accumulation for the eigenvalues of a compact operator, so the eigenvalues of operator $(I - \mathcal{D}^{-1}\mathcal{C})$ cluster at the point 1. Thus, when the solution of $Au = f$ is reduced to that of $[I - D^{-1}C]u = D^{-1}f$, an iterative method such as CGN can be very efficient. This is because the eigenvalues of the normal matrix A^*A will be clustered (as discussed in the introduction). Here $M = D^{-1}$, in general a block quasi-tridiagonal matrix, is used as a preconditioner, and the solution of $M^{-1}x = y$ should similarly follow Algorithm 4.

4.2. Splitting strategy 2. Recall that for many numerical methods, collocation occurs at nodal points. Therefore, we may surround all collocation points by suitable boundary domains, define an intermediate operator, and split the above partitions further in order to have preconditioning matrices with an even simpler structure.

Define an ε-cover for all nodal points by $I_i = [s_i^-, s_i^+]$ for nodes $i = 1, \ldots, n$, where $s_i^{\pm} = s_i \pm \varepsilon$. Note that s_0 and s_n refer to the same point since the boundary Γ is closed. Therefore, a new partition for the boundary (or interval $[a,b]$) is $[a,b] = \bigcup_{i=1}^{n} \left(I_i \bigcup I_i^0\right)$, where $I_i^0 = (s_{i-1}^+, s_i^-)$. Let $\Omega_\varepsilon = \bigcup_{i=1}^{n} I_i$ and $\Omega_o = \bigcup_{i=1}^{n} I_i^0$. Then $[a,b] = \Omega_\varepsilon \bigcup \Omega_0$ and $\Omega_\varepsilon \bigcap \Omega_0 = 0$. Then we may regard Ω_ε as the collocation space.

So all singularities of $\mathcal{K}$ are contained in

$$\bar{\mathcal{K}} = \begin{pmatrix} \mathcal{K}_{1,1} & & & & \mathcal{K}_{1,m} \\ \mathcal{K}_{2,1} & \mathcal{K}_{2,2} & & & \\ & \mathcal{K}_{3,2} & \ddots & & \\ & & \ddots & \ddots & \\ & & & \mathcal{K}_{m,m-1} & \mathcal{K}_{m,m} \end{pmatrix}.$$

The corresponding matrix arising from K is

$$\bar{K} = \begin{pmatrix} K_{1,1} & & & & K_{1,m} \\ K_{2,1} & K_{2,2} & & & \\ & K_{3,2} & \ddots & & \\ & & \ddots & \ddots & \\ & & & K_{m,m-1} & K_{m,m} \end{pmatrix}.$$

As in Strategy 1, we use M as the preconditioner with $M^{-1} = D = I - \bar{K}$, which is a quasi-bidiagonal matrix. To illustrate, for $n = 4$, we can decompose M^{-1} as follows:

$$M^{-1} = LU = \begin{pmatrix} M_{1,1} & & & \\ M_{2,1} & M_{2,2} & & \\ & M_{3,2} & M_{3,3} & \\ & & M_{4,3} & P_{4,4} \end{pmatrix} \begin{pmatrix} 1 & & & P_{1,4} \\ & 1 & & P_{2,4} \\ & & 1 & P_{3,4} \\ & & & 1 \end{pmatrix}.$$

Algorithm 5 (New sparse LU for $M^{-1}x = y$).

1. *(once and only once) Calculation of sparse LU factorization* $M^{-1} = LU$*;*
2. *forward substitution* $Lz = y$*;*
3. *backward substitution* $Ux = z$*.*

5. Sparse preconditioners for integral operators. Corresponding to every preconditioning technique for matrices, there must exist some underlying preconditioning strategies for integral operators. There are two difficulties associated with this idea. First, it is not always obvious how to work out such strategies, i.e., splittings. Second, not all strategies are useful in terms of singularity separation. Here it is a useful splitting of an operator (not just a splitting) that is important.

Thus, to establish the effectiveness of a given preconditioning technique for matrices, we need to find a *useful* singularity separation for the integral operator; otherwise if such an attempt fails, the preconditioner is not likely to be the most efficient. We now use the theory of the previous section to identify the operator splittings implied in the algorithms presented.

First, we see that the preconditioner of Algorithm 1 uses the splitting $\mathcal{K} = \mathcal{D} + \mathcal{C}$, where $\mathcal{D} = I - \tilde{\mathcal{K}}$, with $\tilde{\mathcal{K}} = (\tilde{\mathcal{K}}_{ij})$ and $\tilde{\mathcal{K}}_{ij} = \eta \int_{E_j/\eta} k(s,t)u_j(t)dt \approx \int_{E_j} k(s,t)u_j(t)dt = \mathcal{K}_{ij}$. Theoretically $\mathcal{C}$ is not compact but $\mathcal{C}$ should approach 0 if $m, n \to \infty$. Practically η should not be too large (otherwise the method becomes too expensive if $m \approx n$). Second, for Algorithm 2, the method splits a diagonal operator $\mathcal{D}$, which gives rise to a useful singularity separation for a class of numerical methods; see [7]. Third, for Algorithm 3, the method splits a nonsparse (i.e., full) operator which is unlikely to be a useful singularity separation. Fourth, for the preconditioner of Algorithm 4, the underlying splitting is a tridiagonal-like operator as

in section 4.1. For piecewise constant approximations or the panel method (midpoint rule), the splitting is essentially identical to that of Algorithm 2.

Therefore we may predict that Algorithms 1 and 3 are unlikely to be very efficient for noncompact operator equations.

5.1. Algorithm 1 for first kind equations. Algorithms 2–5 all apply to singular integral equations of the first kind, but Algorithm 1 is not immediately applicable. In [20], for the case where part of the integral operator may be reducible to a circulant matrix, the author suggested applying the fast Fourier transform to this part to obtain a diagonal matrix. Then the situation resembles the case for a second kind equation.

Instead of carrying out more case studies, we here propose, in view of our discussions on singularity separation, a more general method to enable Algorithm 1 to be applicable to first kind equations. First decompose the matrix $K = K_1 + K_2$, where K_1 is the diagonal matrix of K and $K_2 = (k_1, k_2, \ldots, k_n)$. Then construct new column vectors by

$$k_i' = \begin{cases} k_i & \text{if } \ i = \ell\eta, \quad 1 \le \ell \le m, \\ 0 & \text{otherwise,} \end{cases}$$

and define a new matrix by $K' = (k_1', k_2', \ldots, k_n')$. Then we can use M as a preconditioner with $M^{-1} = A' = K_1 + (\eta - 1)K'$. As before, it may be expected that A' is "close" to $A = A_n$ because $A' \approx A_m$, which is the corresponding discrete matrix with m nodes.

5.2. Matrix splitting under fast wavelet transforms. We hope to be able to isolate the matrix corresponding to the singular partition D, after a fast wavelet transform $\tilde{A} = WA = WD + WC$. Unfortunately the truncated matrices of both WD and WC possess identical sparsity structures. So isolating singularities in this way will not be fruitful.

To overcome this problem, we propose to consider the 2-level wavelets only. Then we can follow the exact spread of nonzero elements of D in WD that can give a new splitting method based on the transformed equation $WAu = Wf$. Take the example of Daubechies' order $m = 4$ wavelets with $r = L - 1$ (i.e., 2-levels), based on the splitting of section 4.2. Then the exact structure of WD is as in Fig. 1. With $n = 16$, we can illustrate the corresponding entries of WA, denoted as WA_{WD}:

(8)
$$\left(\begin{array}{cccccccccccccccc}
\times & \times & \times & \times & & & & & & & & & & & & \times \\
 & \times & \times & \times & \times & \times & & & & & & & & & & \\
 & & & \times & \times & \times & \times & \times & & & & & & & & \\
 & & & & & \times & \times & \times & \times & \times & & & & & & \\
 & & & & & & & \times & \times & \times & \times & \times & & & & \\
 & & & & & & & & & \times & \times & \times & \times & \times & & \\
 & & & & & & & & & & & \times & \times & \times & \times & \times \\
\times & \times & & & & & & & & & & & & \times & \times & \times \\
\hline
\times & \times & \times & \times & & & & & & & & & & & & \times \\
 & \times & \times & \times & \times & \times & & & & & & & & & & \\
 & & & \times & \times & \times & \times & \times & & & & & & & & \\
 & & & & & \times & \times & \times & \times & \times & & & & & & \\
 & & & & & & & \times & \times & \times & \times & \times & & & & \\
 & & & & & & & & & \times & \times & \times & \times & \times & & \\
 & & & & & & & & & & & \times & \times & \times & \times & \times \\
\times & \times & & & & & & & & & & & & \times & \times & \times
\end{array}\right). \tag{8}$$

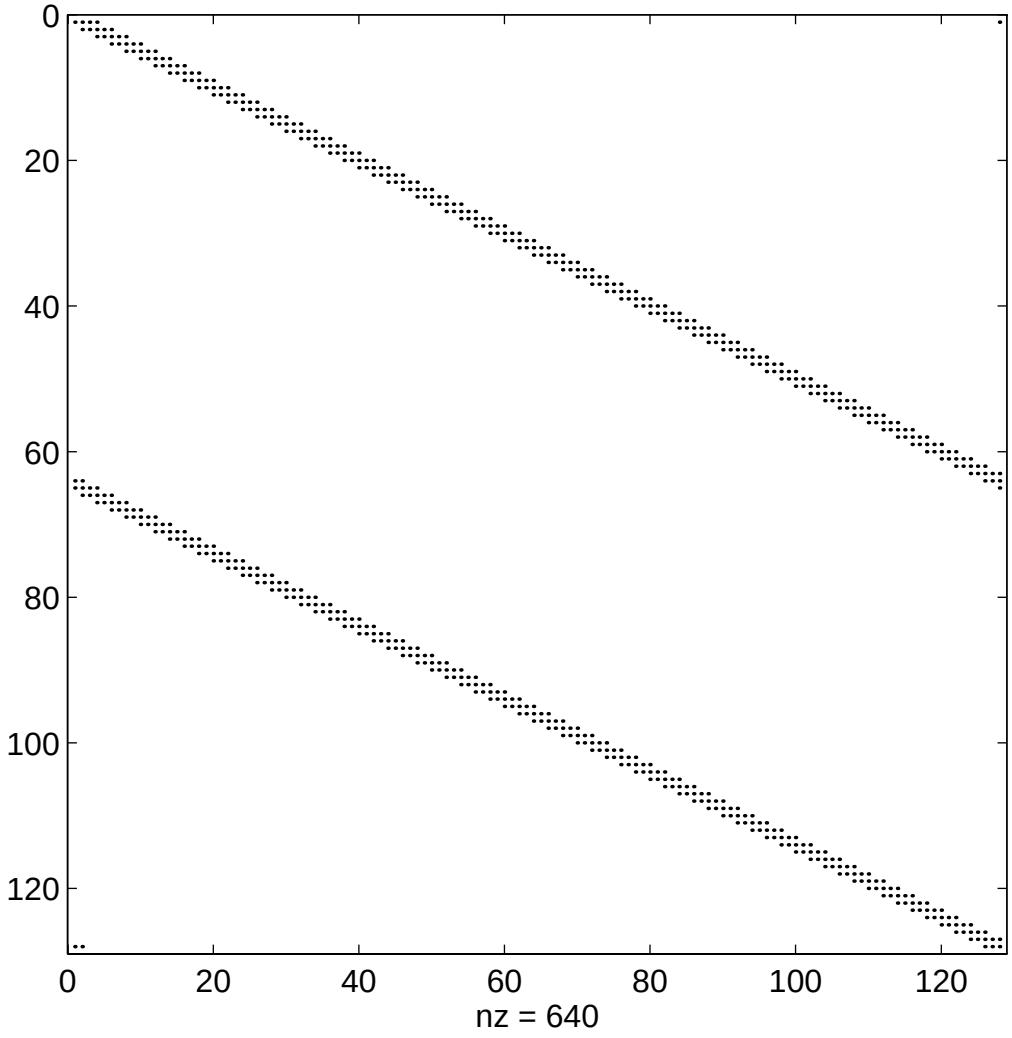

FIG. 1. *Structure of WD with $n = 128$.*

Here each "×" denotes an entry of $\tilde{A}$. Further, such a matrix WA_{WD}, that contains all entries of WD plus some contributions from WC, can be used as a preconditioner for WA. Then

$$WA_{WD}^{-1}WAu = (I + WA_{WD}^{-1}B)u = WA_{WD}^{-1}f,$$

where $B = WA - WA_{WD}$ corresponds to the discretization of nonsingular integral operators.

ALGORITHM 6 (2-level fast wavelet sparse LU for $M^{-1}x = y$).

1. *Apply the* (2*-level columnwise*) *fast wavelet transform to* $Au = f$.
2. *Set* $M^{-1} = WA_{WD}$ *as in* (8).
3. *Decompose* $M^{-1} = LU$.
4. *Solve* $Lz = y$ *and* $Ux = y$ *in each preconditioning step for* $M^{-1}x = y$.

5.3. Preconditioning for equations with geometric singularities. As a special case of the more general singular boundary integral equation, geometric sin-

TABLE 1
Operation counts for the preconditioners.

Iteration i	A1	A2	A3	A4	A5	A6
$i = 0$	$\frac{2}{3}m^3$	$21N$	$12N^2$	$8N$	$2N$	$4N$
$i \geq 1$	$(m+1)N$	$3N$	$3N$	$7N$	$6N$	$2N$

gularities may alone lead to difficulties in iterative methods without preconditioning. Our techniques can immediately be generalized to this case. The results are as in [18] and [6].

6. Numerical results. To illustrate the preconditioning techniques discussed above, we consider these specific implementations

A0	The unpreconditioned case
A1	2-grid preconditioner
A2	Mesh-neighbor method
A3	Approximate inversion method
A4	Sparse LU method (band 3)
A5	Sparse LU method (lower band 2)
A6	Sparse LU method (lower band 2) with 2-level fast wavelet transform WA

for two examples; here A1–A6 refer to Algorithms 1–6 respectively. We have mentioned that all preconditioners considered here cost $O(N)$ operations at each iteration step. To be more specific and to compare all preconditioners, we list in Table 1 the actual operation counts, i.e., specify the constant c in $O(N) = cN$. We can observe that, apart from an initial setup, no preconditioner is expensive. However, the operation count for A1 depends on m and we shall choose $m = N/8$, which makes A1 the most expensive preconditioner.

For comparison, we have chosen two iterative methods for our experiments: the CGN and GMRES(k)—the generalized minimal residual method of [17] (see also [15])—with $k = 9$. As discussed, our theory suggests that Algorithms 2, 4, and 5, all admitting useful singularity separations, should perform well with the CGN because the eigenvalues of the normal matrix cluster around 1. However, for nonsymmetric matrices, the convergence of GMRES is not determined by the eigenvalue spectrum of A; see [15]. So it is not clear whether these algorithms will be efficient with GMRES. Moreover, two steps of GMRES (as listed in the tables) are the same as one step of CGN. This is because in GMRES(9), the iterative process is restarted every nine steps, and in each step, there is one matrix-vector multiplication compared to two multiplications in CGN.

The first problem to be tested has a weak singularity (see [20]):

Problem 1: $$u(s) + \gamma \int_{-\pi}^{\pi} u(t)b(s,t)dt = f(s), \qquad s \in [-\pi, \pi],$$

which arises from the solution of the exterior Neumann problem for the Laplace equation (when $\gamma = 1$) over an elliptic boundary $p(s) = (\cos(s), \sin(s)/4)$. Here the kernel function is

$$b(s,t) = \frac{(p(t) - p(s)) \cdot n(p(t))|p'(t)|}{\pi |p(t) - p(s)|^2} = \frac{4}{\pi(17 - 15\cos(t+s))}.$$

The parameter γ is included to vary the difficulty of the problem. We specifically

TABLE 2
Convergence results of Problem 1 ($\gamma = 10$)
[Note that two GMRES(9) *steps are equivalent to* 9 *CGN steps.]*

Method	Size N	A0	A1	A2	A3	A4	A5	A6
GMRES(9)	16	2	2	1	1	13	1	2
	32	2	*	1	2	*	1	2
	64	2	*	1	2	4	2	2
	128	2	*	2	2	*	2	5
	256	2	*	2	2	3	2	*
	512	3	*	2	2	9	3	*
	1024	3	*	3	3	3	3	4
CGN	16	6	4	5	4	8	9	6
	32	10	6	6	7	10	7	10
	64	10	37	7	7	11	8	11
	128	13	52	8	9	13	8	13
	256	13	60	8	9	13	10	14
	512	14	62	10	9	13	10	15
	1024	16	62	10	10	14	11	15

choose

$$f(s) = |\sin(s)| + \frac{2\gamma}{15\pi}\left\{4\cos(s)\log\frac{17+15\cos(s)}{17-15\cos(s)} + 17\sin(s)\tan^{-1}\frac{15\sin(s)}{8}\right\}$$

so that $u(s) = |\sin(s)|$.

The second problem possesses a Cauchy singularity (see [6]):

Problem 2:
$$\begin{cases} \dfrac{1}{\pi}\displaystyle\int_{-1}^{1}\frac{w(t)\phi(t)}{t-x}dt + \int_{-1}^{1}\frac{(t^2-x^2)^2}{t^2+x^2}w(t)\phi(t)dt = f(x), \\ \dfrac{1}{\pi}\displaystyle\int_{-1}^{1} w(t)\phi(t)dt = 0, \qquad\qquad x \in (-1,1), \end{cases}$$

which has the exact solution $\phi(x) = x|x|$.

Both problems are discretized by the Nyström method, using uniform nodes for Problem 1 and Chebyshev nodes for Problem 2. The CGN is adopted; see [15]. The tolerance for residual errors (RMS norm) is set to be TOL $\in [10^{-8},\ 10^{-2}]$, so that it is of the same magnitude as the discretization error (measured using a direct method). Results of iteration steps are shown in Tables 2 and 3 for comparisons of the six preconditioners for Problems 1 and 2. From Table 2, we see that A0 for Problem 1 produces similar results to A1–A6 because of a weak singularity, which is consistent with our early assertion that preconditioning is not required for compact cases. We use "*" to denote a divergence entry. As all preconditioners (except initializing A1) only involve $O(N)$ extra operations per step, the number of iterations is thus proportional to the CPU time used.

From Table 3, we can observe that, for Problem 2, without preconditioning (A0), the CGN converges slowly, while the GMRES diverges for larger N and its performance in general is erratic. The experiments clearly suggest that A2 and A5 outperform the rest. This agrees with our theory of operator preconditioning, that is, regularization of singular operators is more significant than approximation. A more detailed study of A2 is presented in [7].

Although further and more extensive tests are needed to compare these preconditioners, our preliminary conclusion is that, for singular boundary element equations, the most robust preconditioners (Algorithms 2 and 5) should be based on singularity separations.

TABLE 3
Convergence results of Problem 2.
[Note that two GMRES(9) *steps are equivalent to* 9 *CGN steps.]*

Method	Size N	A0	A1	A2	A3	A4	A5	A6
GMRES(9)	16	5	25	21	3	6	5	3
	32	8	*	41	49	14	7	5
	64	19	*	32	97	30	10	6
	128	80	*	38	193	40	13	13
	256	*	*	35	385	50	23	75
	512	*	*	63	769	46	24	*
	1024	*	*	34	24	38	39	*
CGN	16	9	13	9	10	8	9	6
	32	19	17	11	18	11	10	9
	64	36	24	12	30	17	11	16
	128	71	32	13	39	23	13	27
	256	144	45	14	43	27	14	45
	512	291	60	14	46	30	14	42
	1024	599	69	16	47	32	15	54

7. Conclusions. In this paper we have examined a class of preconditioning methods suitable for dense linear systems arising from boundary elements. We have provided a general framework based on operator splittings for analyzing preconditioners. Such a framework is used to propose a new preconditioner (A5) that is shown to be efficient for the experiments performed. By appealing to compactness of integral operators, we have demonstrated that the normal matrix can exhibit an eigenvalue clustering property, and hence the CGN is found to be more efficient than GMRES with the proposed preconditioners.

Acknowledgments. The author wishes to thank two anonymous referees for making a lot of helpful criticisms. He is also grateful to Drs. T. B. Boffey and S. Brenner for reading the manuscript and making helpful suggestions.

REFERENCES

[1] S. AMINI AND K. CHEN, *Conjugate gradient method for second kind integral equations: Applications to the exterior acoustic problem*, Engineering Analysis with Boundary Elements, 6 (1989), pp. 72–77.
[2] S. AMINI AND N. MAINES, *Iterative solutions of boundary integral equations*, in Proc. 14th conference on Boundary Element Methods, C. A. Brebbia, ed., CM Publications, Southampton, UK, 1994, pp. 193–200.
[3] A. BOND AND S. VAVASIS, *Fast Wavelet Transforms for Matrices Arising from Boundary Element Methods*, Computer Science Research Report TR-174, Cornell University, Ithaca, NY, 1994.
[4] C. A. BREBBIA ET AL., *Boundary Element Techniques: Theory and Applications*, Springer-Verlag, Berlin, NY, 1984.
[5] F. CANNING, *Sparse approximation for solving integral equations with oscillatory kernels*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 71–87.
[6] K. CHEN, *Efficient iterative solution of linear systems from discretizing singular integral equations*, Electron Trans. Numer. Anal., 2 (1994), pp. 76–91.
[7] K. CHEN, *An analysis of sparse approximate inverse preconditioners for dense linear systems*, submitted (1998).
[8] G. BEYLKIN, R. COIFMAN, AND V. ROKHLIN, *Fast wavelet transforms and numerical algorithms* I, Comm. Appl. Pure Math., 41 (1991), pp. 141–183.
[9] J. D. F. COSGRAVE, J. C. DIAZ, AND A. GRIEWANK, *Approximate inverse preconditioners for sparse linear systems*, Internat. J. Comput. Math., 44 (1992), pp. 91–110.
[10] W. HACKBUSCH, *Multigrid Methods and Applications*, Springer-Verlag, Berlin, New York, 1985.
[11] W. HACKBUSCH AND Z. P. NOWAK, *On the fast matrix multiplication in the boundary element method by panel clustering*, Numer. Math., 54 (1989), pp. 463–491.

[12] P. W. Hemker and H. Schippers, *Multigrid methods for the solution of Fredholm integral equations of the second kind*, Math. Comput., 36 (1981), pp. 215–232.
[13] T. Huckle and M. Grote, *A new approach to parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1998), pp. 838–853.
[14] R. Kress, *Linear Integral Equations*, Springer-Verlag, Berlin, New York, 1989.
[15] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.
[16] P. W. Partridge, C. A. Brebbia, and L. C. Wrobel, *The Dual Reciprocity Boundary Element Method*, CMP, Southampton, UK., 1992.
[17] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving unsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[18] H. Schippers, *Multigrid methods for boundary integral equations*, Numer. Math., 46 (1985), pp. 351–363.
[19] S. Vavasis, *Preconditioning for boundary integral equations*, SIAM J. Matrix. Anal. Appl., 13 (1992), pp. 905–925.
[20] Y. Yan, *Sparse preconditioned iterative methods for dense linear systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1190–1200.