Electronic Transactions on Numerical Analysis.
Volume 8, pp. 138-153, 1999.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

DISCRETE WAVELET TRANSFORMS ACCELERATED SPARSE PRECONDITIONERS FOR DENSE BOUNDARY ELEMENT SYSTEMS*

KE CHEN †

Abstract. We consider a construction of efficient preconditioners, using discrete and fast wavelet transforms, for dense and unsymmetric linear systems that arise from boundary elements. The wavelet compression property combined with operator splitting result in much improved preconditioners, in terms of both eigenspectra clustering and inverse approximations, taking the form of band matrices with wrap-around boundaries. With our new non-standard wavelet transform, the transformed matrix is permuted to band forms. It is shown that, to have band matrices, one has to use a smaller number of wavelet levels. Numerical experiments using the iterative methods of conjugate gradients based on the normal equations (CGN) and generalised minimal residuals (GMRES) are reported.

Key words. fast wavelet transforms, dense linear systems, sparse preconditioners, conjugate gradient, boundary elements.

AMS subject classifications. 65F10, 65N38, 45E05.

1. Introduction. For the numerical solution of boundary value problems modelled by partial differential equations, the usual choice of solution methods are between the finite element method and the finite difference method. These methods are of domain type because the entire domain must be discretized. An efficient alternative, the boundary type, is the boundary element method (BEM) where only the boundary of the domain is needed for discretization. Therefore the dimension of the problem is reduced by one. See [5, 6, 9] and the references therein.

However, in contrast to domain type methods where sparse stiffness matrices are generated, the boundary element method produces a full coefficient matrix. Hence iterative solvers are usually used; see [5, 12] and the references therein. As far as the generation of the coefficient matrix is concerned, the process is highly parallelizable. Recently we found that parallel direct methods based on a flexible elimination technique appear to be quite efficient; see [24, 13].

This paper considers the construction of efficient sparse preconditioners for the BEM using fast wavelet transforms. We employ the preconditioned conjugate gradient type iterative solvers. The preconditioners we propose are generally applicable and can be easily combined with existing BEM software. Related work can be found in [4, 7, 8].

It should be remarked that there is other related and recent work treating the full boundary element matrix. This work falls into two categories: (i) The panel clustering technique of [21] and the multipole method of [20]. In this case, the objective is to calculate matrix-vector multiplications in less than $O(n^2)$ operations, where n is the dimension of the coefficient matrix. The idea is to view matrix-vector multiplications as an approximation to an integral and seek alternative and faster ways to compute this integral by exploiting different expansion properties of the integrand function in 'near' and 'far' fields. This method assumes that a kernel function is smooth away from the collocation point and is thus not suitable for oscillatory kernels. (ii) The wavelet method as [7] and [17]. In this case the objective is to obtain a nearly sparse coefficient matrix instead of a full matrix. In fact one only computes and stores a sparse matrix. This can be achieved by replacing the usual piecewise polynomial basis by a wavelet basis. The main difficulty with this method lies in constructing useful and applicable

* Received May 29, 1998. Accepted for publication November 20, 1999. Recommended by L. Reichel.
†Department of Mathematical Sciences, The University of Liverpool, Liverpool L69 3BX, UK. k.chen@liv.ac.uk and http://www.liv.ac.uk/maths/applied.

wavelets. Practically, to avoid the calculation (and storage) of small entries, the method is somewhat problem dependent (e.g. on forms of kernels as with (i)).

As far as preconditioning is concerned, both categories of methods cannot employ more general (efficient) preconditioners. In the first category, it is possible to use operator-splitting-based preconditioners because all near field entries (band form) are available. In the second category, apart from a diagonal scaling, it is not obvious what other preconditioners can be implemented. Also because the compressed sparse matrix possesses an unusual 'finger'-like pattern, solving the associated linear system by direct sparse solvers can result in many fill-ins and is therefore not practical. We believe our present work (on new non-standard wavelets) offers a way to construct possible band preconditioners for iterative methods.

We concentrate on design of new preconditioners for the traditional BEM. The robustness and strength of our algorithms comes from the fact that they can effectively deal not only with singular BIE's but also singular operator equations with perturbations of an oscillatory nature where compressions-and-expansions-based methods cannot deal effectively. The work presented here, both on permuting wavelet matrices into near band matrices and on accelerating operator splitting type preconditioners, appear to be new. Some numerical experiments illustrating the results will be shown.

To proceed, denote an $n \times n$ dense linear system by

$$Ax = b. \tag{1.1}$$

Here A is usually unsymmetric from boundary elements using the collocation approach or the Nystrom quadrature method. Our work may be extended to Galerkin methods, but such methods are practically more expensive to implement and less often used. Refer to [5, 9, 11, 12] and the references therein.

2. Sparse preconditioners. As is known, if A arises from a discretization of integral equations with compact operators, the eigenspectrum of A affects the performance of the iterative methods. For the conjugate gradients based on the normal equations, the eigenspectrum of both A and $A^\top A$ are clustered thus leading to fast convergence; this may not be true for matrices form other applications such as from discretizing partial differential equations directly.

For integral equations with non-compact operators, we proposed in [12] operator-splitting-based matrix preconditioners, which have corresponding operator counterparts such that the preconditioned matrix as well as the preconditioned operator and their normals have clustered eigenvalues (as in the case of compact operators). Therefore iterative solvers such as conjugate gradients applied to the normal equation (CGN) and to the generalized minimal residuals method (GMRES) will be efficient for the preconditioned systems.

More precisely, for equation (1.1), we have considered the following splitting of A, and the associated preconditioned system

$$A = D + C, \qquad \text{and} \qquad (I + D^{-1}C)x = D^{-1}b, \tag{2.1}$$

where $M^{-1} = D^{-1}$ is the preconditioner. Here D is a simple band matrix (with wrap-round boundaries), For the case of band $\mu = 3$, the matrix is simply

$$D = \begin{pmatrix} A_{1,1} & A_{1,2} & & & A_{1,n} \\ A_{2,1} & A_{2,2} & A_{2,3} & & \\ & A_{3,2} & \ddots & \ddots & \\ & & \ddots & \ddots & A_{n-1,n} \\ A_{n,1} & & & A_{n,n-1} & A_{n,n} \end{pmatrix}.$$

This kind of choice for M^{-1} is for regularization of a usually singular boundary integral operator. Clearly with the above choice of $\mu = 3$, M^{-1} will not approximate A^{-1} well. It has been found that, to further improve the above preconditioner, increasing the band-width μ alone is not sufficient as the improvements are only marginal. In fact, improved results were observed in [12] by reducing μ to 2.

Here we shall discuss a way of combining operator splitting with the discrete wavelet transforms to derive a new and improved preconditioner. In terms of operator splitting, our new preconditioner M^{-1} will be effectively the inverse of a nearly full matrix $\widetilde{D}$. Therefore M^{-1} will approximate A^{-1} better, iterative methods will work better, and more importantly discrete wavelet transforms ensure a fast implementation.

3. The standard discrete wavelet transforms. We first consider the use of standard discrete wavelet transforms (DWT's), based on Daubechies' compactly supported orthogonal wavelets, for matrix A in (1.1). Refer to [7, 22, 25, 26, 29].

Recall that wavelets form a basis of $\mathcal{L}_2$. That is, given a function f, it can be written as an infinite linear combination of the wavelets and the wavelet coefficients uniquely determine the function although most coefficients may be small and nearly zero.

For a given vector s, from vector space $\mathcal{R}^n$, one may construct an infinite periodic sequence of period n and use it as coefficients of a scaling function $f_L(x)$ in some fixed subspace V_L of $\mathcal{L}_2$ where L is an integer. As these subspaces of $\mathcal{L}_2$ form a multiresolution analysis, for example orthogonal partition of subspaces

$$V_L = V_r \bigoplus W_r \bigoplus W_{r+1} \bigoplus \cdots \bigoplus W_{L-1}, \tag{3.1}$$

then in the equivalent space (partition) $f_L(x)$ has a new representation whose coefficients may be viewed as a new vector w mapped from s by wavelets. This is the starting point of a discrete wavelet transform. See [8, 29, 30].

The above transform $W : \ s \mapsto w$ (i.e. $w = Ws$) is implemented by the well known pyramidal algorithm. To discuss it briefly and specify the matrix structure of W, let m be the order of compactly supported wavelets with m lowpass filter coefficients $c_0, c_1, \cdots, c_{m-1}$ and $m/2$ vanishing moments. Assume $n = 2^L$ and r is an integer such that $2^r < m$ and $2^{r+1} \geq m$. Note $r = 0$ for $m = 2$ (Haar wavelets) and $r = 1$ for $m = 4$ (Daubechies order 4 wavelets); see [8, 26, 29]. Denote by $s = s^{(L)}$ a column vector of A at the wavelet level L. Then the standard pyramidal algorithm transforms the vector $s^{(L)}$ to

$$w = \left[(s^{(r)})^\top \ (f^{(r)})^\top \ (f^{(r+1)})^\top \ \cdots \ (f^{(L-1)})^\top \right]^\top$$

in a level by level manner, that is,

$$\begin{array}{ccccccccccccc}
s^{(L)} & \rightarrow & s^{(L-1)} & \rightarrow & s^{(L-2)} & \rightarrow & \cdots & \rightarrow & s^{(\nu)} & \rightarrow & \cdots & \rightarrow & s^{(r)} \\
& \searrow & & \searrow & & \searrow & & \searrow & & \searrow & & \searrow & \\
& & f^{(L-1)} & & f^{(L-2)} & & \cdots & & f^{(\nu)} & & \cdots & & f^{(r)}
\end{array}$$

where the vectors $s^{(\nu)}$ and $f^{(\nu)}$ are of length 2^ν. Notice that the sum of these lengths is $n = 2^L$ since

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

Suppose $\Phi_m(x)$ denotes the wavelet function. Then we expect the new vector w to be nearly sparse because the usual function moment conditions

$$\int_{-\infty}^{\infty} \Phi_m(x) x^p dx = 0 \qquad \text{for} \qquad p = 0, 1, \cdots, m/2 - 1$$

are equivalent to vector moment conditions

$$\sum_{k=0}^{m-1} (-1)^k k^p c_k = 0 \qquad \text{for} \qquad p = 0, 1, \cdots, m/2 - 1.$$

Here the larger m is, the better is the compression in w is, but the compact support is larger as well.

At a typical level ν, $s^{(\nu)}$ and $f^{(\nu)}$ are collections of scaling and wavelet coefficients respectively. In matrix form, w expressed as

$$w = P_{r+1} W_{r+1} \cdots P_{L-1} W_{L-1} P_L W_L s^{(L)} \equiv W s^{(L)}, \tag{3.2}$$

where

$$P_\nu = \begin{pmatrix} \overline{P}_\nu & \\ & J_\nu \end{pmatrix}_{n \times n}$$

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

$$W_\nu = \begin{pmatrix} \overline{W}_\nu & \\ & J_\nu \end{pmatrix}_{n \times n}$$

with $\overline{W}_\nu$ an orthogonal (sparse) matrix of size $2^\nu = 2^L - k_\nu = n - k_\nu$ and J_ν is an identity matrix of size k_ν. Here $k_L = 0$ and $k_\mu = k_{\mu+1} + 2^\mu$ for $\mu = L - 1, \cdots, r + 1$.

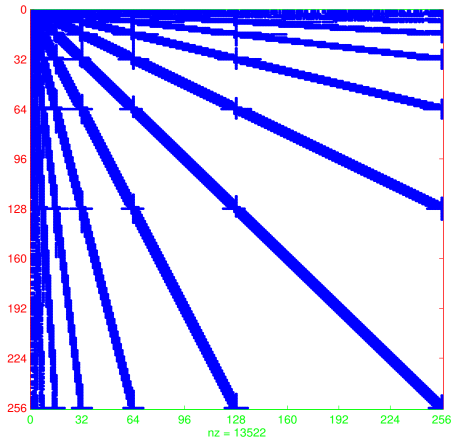

FIG. 3.1. $\widetilde{A}$ *after thresholding with TOL*=10^{-4}, $LEV = 8$ *and* $n = 256$

The one level transformation matrix $\overline{W}_\nu$ is a compact quasi-diagonal block matrix. For example, with the Daubechies' order $m = 4$ wavelets with $m/2 = 2$ vanishing moments, it

ETNA
Kent State University
etna@mcs.kent.edu

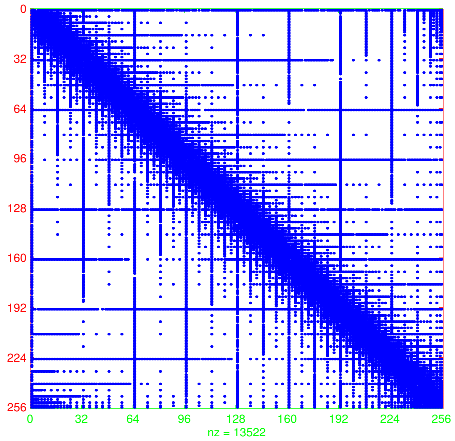

FIG. 3.2. $\widehat{W}A\widehat{W}^\top = P(WAW^\top)P^\top$ *with TOL*=10^{-4}*, LEV* $= 8$ *and* $n = 256$

is

$$\overline{W}_\nu = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & & & & & & \\ d_0 & d_1 & d_2 & d_3 & & & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & & & \\ & & d_0 & d_1 & d_2 & d_3 & & & & \\ & & & & \ddots & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \ddots & \ddots & & \\ & & & & & & c_0 & c_1 & c_2 & c_3 \\ & & & & & & d_0 & d_1 & d_2 & d_3 \\ c_2 & c_3 & & & & & & & c_0 & c_1 \\ d_2 & d_3 & & & & & & & d_0 & d_1 \end{pmatrix},$$

where the filtering coefficients $\{c_i, d_i\}$ are known to be $c_0 = (1+\sqrt{3})/(4\sqrt{2})$, $c_1 = (3+\sqrt{3})/(4\sqrt{2})$, $c_2 = (3-\sqrt{3})/(4\sqrt{2})$, $c_3 = (1-\sqrt{3})/(4\sqrt{2})$ and $d_k = (-1)^k c_{m-1-k}$ as usual.

Table 3.1 shows a typical example of applying a 3-level pyramidal algorithm to a vector $s = a = s^{(4)}$ of size $n = 2^L = 2^4$, where 3 forms (1 for components, 2 for vectors, 3 for matrices) of transform details are shown. Clearly local features in a are scattered in Wa, that is, the standard DWT is not centred. A full DWT step for matrix A, transforming column and row vectors, respectively, will be $\widetilde{A} = WAW^\top$. It is known that $\widetilde{A}$ has a 'finger'-like sparsity pattern above some threshold. This is demonstrated in Fig.3.1 where one example from [7, ex.1], where A defined by

$$A_{ij} = \begin{cases} \frac{1}{i-j}, & i \neq j, \\ 0, & i = j, \end{cases} \tag{3.3}$$

with $n = 256$ and with a threshold tolerance of 10^{-4}.

Let $\overline{A}$ be such a 'finger'-like sparse matrix, truncated from $\widetilde{A}$ after some thresholding. For such matrices, matrix-vector products can be formed very efficiently. That is to say, if it is desirable to solve $\overline{A}u = z$ by some iterative methods without preconditioning, then as far as certain applications are concerned we have obtained an efficient implementation. Here $u = Wx$ and $z = Wb$.

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 3.1
Standard DWT for a vector a of size 16 ($m = 4$, $L = 4$, $r = 1$)

	$Form$	Lev 4	Lev 3		Lev 2	
1		a_1	$\times$	$\times$	$\times$	$\times$
2		a_2	$\times$	0	$\times$	0
3		0	0	0	0	$\times$
4		0	0	0	0	$\times$
5		0	0	0	$\times$	$\times$
6		0	0	0	$\times$	0
7		0	0	$\times$	$\times$	$\times$
8	1	0	0	0	$\times$	$\times$
9		0	0	$\times$	$\times$	$\times$
10		0	0	0	0	0
11		0	0	0	0	0
12		0	0	0	0	0
13		0	0	0	0	0
14		0	0	0	0	0
15		0	$\times$	0	0	0
16		0	$\times$	$\times$	$\times$	$\times$
	2	$s^{(4)}$	$\overline{W}_4 s^{(4)}$	$s^{(3)}$ $f^{(3)}$	$\overline{W}_3 s^{(3)}$ $f^{(3)}$	$s^{(2)}$ $f^{(2)}$ $f^{(3)}$
	3	$s^{(4)}$	$W_4 s^{(4)}$	$P_4 W_4 s^{(4)}$	$W_3 P_4 W_4 s^{(4)}$	$P_3 W_3 P_4 W_4 s^{(4)}$

However, approximately, $\overline{A}$ is spectrally equivalent to the original matrix A. Therefore, as the original problem requires preconditioning, we need to precondition $\overline{A}u = z$. Indeed, [28] reports that with GMRES, the number of iteration steps for $\widetilde{A}u = z$ and $\overline{A}u = z$ are the same.

Moreover, it is not an easy task to use $\overline{A}$ as a preconditioner because solving the linear system $\overline{A}u = z$ results in many fill-ins. We also note that such a 'finger'-like matrix also gives problems in designing approximate sparse inverses; see [10]. Below we consider a new implementation of DWT's avoiding the generation of such 'finger'-like matrices.

4. A new non-standard DWT. To begin with, we re-consider how a DWT should ideally deal with a local feature like that in the vector a shown in Table 3.1 before introducing our proposed changes to the basic pyramidal algorithm. We first define a new one-level DWT

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 4.1
New DWT for a vector a of size 16 *(*$m = 4$*,* $L = 4$*,* $r = 1$*)*

	$Form$	Lev 4	Lev 3		Lev 2	
1		a_1	$\times$	$\times$	$\times$	$\times$
2		a_2	$\times$	$\times$	$\times$	$\times$
3		0	0	0	$\times$	$\times$
4		0	0	0	0	0
5		0	0	0	0	0
6		0	0	0	0	0
7		0	0	0	0	0
8	1	0	0	0	0	0
9		0	0	0	$\times$	$\times$
10		0	0	0	0	0
11		0	0	0	$\times$	$\times$
12		0	0	0	0	0
13		0	0	0	$\times$	$\times$
14		0	0	0	0	0
15		0	$\times$	$\times$	$\times$	$\times$
16		0	$\times$	$\times$	$\times$	$\times$
	3	$s^{(4)}$	$\widehat{W}_4 s^{(4)}$	$\widehat{W}_4 s^{(4)}$	$\widehat{W}_3 \widehat{W}_4 s^{(4)}$	$\widehat{W}_3 \widehat{W}_4 s^{(4)}$

matrix (similar to $\overline{W}_\nu$ in (3.2))

$$
\widehat{W}_\nu = \begin{pmatrix}
c_0 & \varnothing & c_1 & \varnothing & c_2 & \varnothing & \cdots & c_{m-1} & & & \\
\varnothing & \mathcal{I} & \varnothing & \varnothing & \varnothing & \varnothing & \cdots & \varnothing & & & \\
d_0 & \varnothing & d_1 & \varnothing & d_2 & \varnothing & \cdots & d_{m-1} & & & \\
\varnothing & \varnothing & \varnothing & \mathcal{I} & \varnothing & \varnothing & \cdots & \varnothing & & & \\
 & & & & c_0 & \varnothing & \ddots & \ddots & & & \\
 & & & & \varnothing & \mathcal{I} & \ddots & \ddots & & & \\
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \ddots & \ddots & \ddots & \ddots & \\
c_2 & \varnothing & \cdots & c_{m-1} & & & & c_0 & \varnothing & c_1 & \varnothing \\
\varnothing & \varnothing & \cdots & \varnothing & & & & \varnothing & \mathcal{I} & \varnothing & \varnothing \\
d_2 & \varnothing & \cdots & d_{m-1} & & & & d_0 & \varnothing & d_1 & \varnothing \\
\varnothing & \varnothing & \cdots & \varnothing & & & & \varnothing & \varnothing & \varnothing & \mathcal{I}
\end{pmatrix}_{n \times n} . \tag{4.1}
$$

Here $\mathcal{I}$ is an identity matrix of size $2^{(L-\nu)} - 1$ and $\varnothing$'s are block zero matrices. For $\nu = L$, both $\mathcal{I}$ and $\varnothing$ are of size 0 i.e. $\widehat{W}_L = \overline{W}_L = W_L$.

Further a new DWT (from (3.1)) for a vector $s^{(L)} \in \mathcal{R}^n$ can be defined by

$$\widehat{w} = \widehat{W} s^{(L)}$$

with

$$\widehat{W} = \widehat{W}_{r+1} \widehat{W}_{r+2} \cdots \widehat{W}_L \tag{4.2}$$

based on $(L + 1 - r)$ levels. For $L = 4$ and $m = 4$ (Daubechies' wavelets for $n = 2^L = 16$), Table 4.1 shows details of a 3-level transform using $\widehat{W}_4$ and $\widehat{W}_3$. One may verify that, unlike

ETNA
Kent State University
etna@mcs.kent.edu

Table 3.1, the very last column ($\widehat{w}$) in Table 4.1 would possess a locally centered sparse structure depending only on the number of wavelet levels and not on size n.

For a matrix $A_{n\times n}$, the new DWT would give

$$\widehat{A} = \widehat{W} A \widehat{W}^\top.$$

Now to relate $\widehat{A}$ to $\widetilde{A}$ from a standard DWT, or rather $\widehat{W}$ to W, we define a permutation matrix

$$P = P_L^\top P_{L-1}^\top \cdots P_{r+2}^\top P_{r+1}^\top, \tag{4.3}$$

where matrices P_k's come from (3.2). Firstly, by induction, we can prove the following

$$\widehat{W}_k = \left(\prod_{\ell=1}^{L-k} P_{k+\ell}\right)^\top W_k \left(\prod_{\ell=1}^{L-k} P_{k+\ell}\right) \qquad \text{for } k = r+1, r+2, \cdots, L,$$

that is,

$$\begin{aligned}
\widehat{W}_L &= W_L,\\
\widehat{W}_{L-1} &= P_L^\top W_{L-1} P_L,\\
&\vdots\\
\widehat{W}_{r+1} &= P_L^\top P_{L-1}^\top \cdots P_{r+2}^\top W_{r+1} P_{r+2} \cdots P_{L-1} P_L.
\end{aligned}$$

Secondly, we can verify that

$$\begin{aligned}
PW &= \left(P_L^\top P_{L-1}^\top \cdots P_{r+1}^\top\right)\left(P_{r+1} W_{r+1} \cdots P_L W_L\right)\\
&= \widehat{W}_{r+1}\left(P_L^\top P_{L-1}^\top \cdots P_{r+2}^\top\right)\left(P_{r+2} W_{r+2} \cdots P_L W_L\right)\\
&\vdots\\
&= \widehat{W}_{r+1}\widehat{W}_{r+2}\cdots \widehat{W}_{L-2} P_L^\top W_{L-1}\left(P_L W_L\right)\\
&= \widehat{W}_{r+1}\widehat{W}_{r+2}\cdots \widehat{W}_L\\
&= \widehat{W}.
\end{aligned}$$

Consequently from $\widehat{W} = PW$, $\widehat{A} = P\widetilde{A}P^\top$.

The practical implication of these relations is that the new DWT can be implemented in a level by level manner, either directly using $\widehat{W}_\nu$'s (via $\widehat{W}$) or indirectly using P_ν's (via P) after a standard DWT, and we obtain the same result.

To illustrate the new DWT, we used it to compress the matrix A shown in Fig.3.1 and the transformed matrix, which has an essentially band-like pattern, is shown in Fig.3.2. Clearly such a pattern may be used more advantageously than a finger-like one (for example, in the applications of [10]).

As far as preconditioning is concerned, to solve (1.1), we propose the following:

Algorithm 4.1.

1. *Apply a DWT to* $Ax = b$ *to obtain* $\widehat{A}u = z$*;*
2. *Select a suitable band form* M *of* $\widehat{A}$*;*
3. *Use* M^{-1} *as a preconditioner to solve* $\widehat{A}u = z$ *iteratively.*

Here the band size of M determines the cost of a preconditioning step. If this size is too small, the preconditioner may not be effective. If the size is so large (say nearly $n-1$) that the preconditioner approximates $\widehat{A}^{-1}$ very accurately, then one may expect that 1 or 2 iterations are sufficient for convergence but each iteration is too expensive. Therefore, we shall next examine the possibility of constructing an effective preconditioner based on a relatively small band.

ETNA
Kent State University
etna@mcs.kent.edu

5. Band matrices under the new DWT. To discuss combining the new DWT with the operator splitting ideas in next section, we first consider the process of transforming a band matrix. Let LEV denote the actual number of wavelet levels used $(1 \le LEV \le (L+1-r))$. Mainly we try to address this question: under what conditions does the new DWT transform a band matrix A into another band matrix $\widehat{A}$ (instead of a general sparse matrix)? Here, by a band matrix, we mean a usual band matrix with wrap-round boundaries. The correct condition turns out to be that LEV should be chosen to be less than $(L+1-r)$. This will be stated more precisely in Theorem 5.3.

To motivate the problem, we show in Figs. 3 and 4 respectively the DWT of a diagonal matrix using $LEV = 5$. Here with $m = 4$ (so $r = 1$) and $L = 8$, a band structure in $\widehat{A}$ is achieved by not using the maximum level $LEV = L - r + 1 = 8$. For a given band matrix A, to establish the exact band width for the transformed matrix $\widehat{A}$ under the new DWT, we need to view the one-step transformation matrix $\widehat{W}_\nu$ as a band matrix. For ease of presentation, we first introduce some definitions and a lemma before establishing the main theorem.

Definition 5.1 (Band(α, β, k)**).** *A block band matrix $A_{n\times n}$, with wrap-round boundaries and blocks of size $k \times k$, is called a* Band(α, β, k) *if its lower block band width is α and upper block band width β (both including but not counting the main block diagonal).*

Note that when $k = 1$, we write Band$(\alpha, \beta, 1)$ = Band(α, β).

Definition 5.2 (Band(α, β, k, τ)**).**
A Band(α, β, k) *matrix $A_{n\times n}$ is called* Band(α, β, k, τ) *if each $k \times k$ block has a total band width of $2*k-1-2*\tau$, that is, if there are τ bands of zeros at both ends of the anti-diagonal.*

LEMMA 5.1. *For non-negative integers $\alpha, \beta, \gamma, \delta, k, \tau$, the following results hold:*
1. Band(α, β)Band(γ, δ) = Band$(\alpha + \gamma, \beta + \delta)$*;*
2. Band(α, β, k)Band(γ, δ, k) = Band$(\alpha + \gamma, \beta + \delta, k)$*;*
3. Band(α, β, k) = Band$((\alpha + 1)k - 1, (\beta + 1)k - 1)$*;*
4. Band(α, β, k, τ) = Band$((\alpha + 1)k - 1 - \tau, (\beta + 1)k - 1 - \tau)$.

LEMMA 5.2. *With Daubechies' order m wavelets, the one-step transformation matrix $\widehat{W}_\nu$, for $n = 2^L$, is* Band$(0, (m/2 - 1), 2^{(L-\nu+1)}, 2^{(L-\nu)} - 1)$ *and is therefore* Band$(2^{(L-\nu+1)} - 2^{(L-\nu)}, 2^{(L-\nu)}m - 2^{(L-\nu)})$.

Proof. Note that the band information of $\widehat{W}_\nu$ does not actually involve L or n, and the apparent involvement of L is due to index ν. It suffices to consider $\nu = L$. Then $\widehat{W}_L$ is consisted of 2×2 blocks with $m/2$ blocks on each row. That is, it is a Band$(0, m/2-1, 2, 0)$ matrix. Then use Lemma 5.1 to complete the proof. □

Remark 5.1. *Clearly* Band(α, β, k) = Band$(\alpha, \beta, k, 0)$*. However, for block matrices, it is often necessary to keep them as blocks until a final step in order to obtain improved results. For example, with Lemma 5.1.2- 5.1.3,* Band$(1, 1, 2)$Band$(1, 1, 2)$ = Band$(2, 2, 2)$ = Band$(5, 5)$*, but with Lemma 5.1.3,*

$$\text{Band}(1,1,2)\text{Band}(1,1,2) = \text{Band}(3,3)\text{Band}(3,3) = \text{Band}(6,6).$$

Similarly as blocks, Band$(3, 2, 4)$Band$(1, 1, 4)$ = Band$(4, 3, 4)$ = Band$(19, 15)$ *but as bands,*

$$\text{Band}(3,2,4)\text{Band}(1,1,4) = \text{Band}(15,11)\text{Band}(7,7) = \text{Band}(22,18)$$

— an over-estimate! This suggests that if the band matrix A is Band$(\alpha, \beta, 2)$*, then Theorem 5.3 below can be further improved.*

THEOREM 5.3. *Assume that $A_{n\times n}$ is a* Band(α, β) *matrix. Then the new DWT of ℓ levels, based on Daubechies' order m wavelets, transforms A into $\widehat{A}$ which is at most a* Band(λ_1, λ_2) *matrix with*

$$\lambda_1 - \alpha = \lambda_2 - \beta = m(2^{(\ell-1)} - 1).$$

Proof. For the new DWT with ℓ levels, the transform is $\widehat{A} = \widehat{W} A \widehat{W}^\top$

$$\widehat{W} = \widehat{W}_{L-\ell+2} \widehat{W}_{L-\ell+3} \cdots \widehat{W}_L.$$

From Lemma 5.2, the total lower and upper band widths of $\widehat{W}$ will be respectively

$$low = \sum_{\nu=L-(\ell-2)}^{L} \left(2^{L-\nu+1} - 2^{L-\nu}\right) \qquad \text{and} \qquad up = \sum_{\nu=L-(\ell-2)}^{L} (m-1)2^{L-\nu}.$$

Therefore the over estimate for the lower band width of $\widehat{A}$ will be

$$\lambda_1 = \alpha + low + up = \alpha + \sum_{\nu=L-(\ell-2)}^{L} 2^{L-\nu} = \alpha + m(2^{(\ell-1)} - 1).$$

Similarly we get the result for λ_2 and the proof is complete. ☐

Note that as indicated before, parameters λ_1, λ_2 do not depend on the problem size n. When $\alpha = \beta$ for A, $\lambda_1 = \lambda_2$ for $\widehat{A}$. For a diagonal matrix A with distinct diagonal entries, for instance, a *Band*$(0,0)$ matrix, with $m = 4$, $LEV = \ell = 5$, and $n = 256$, the standard DWT gives a ‘finger’-like pattern in $\widetilde{A}$ as shown in Fig.5.1 while the new DWT gives a *Band(*$46,46)$ matrix in $\widehat{A}$ as shown in Fig.5.2. Here Theorem 5.3 gives over estimates $\lambda_1 = \lambda_2 = 4(2^{(5-1)} - 1) = 60$, a *Band(*$60,60)$ matrix.

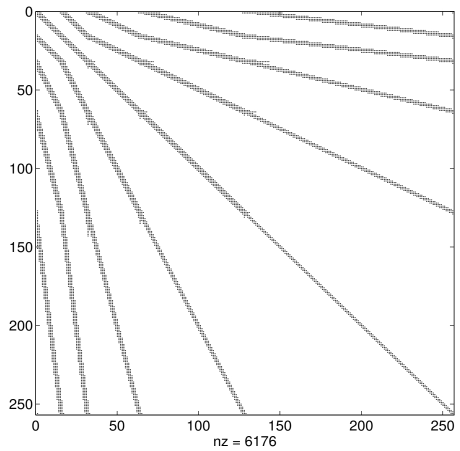

FIG. 5.1. $WAW^\top$ *for a diagonal matrix A* ($LEV = 5$ *and n = 256)*

6. Applications to preconditioning a linear system. Consider the transformed linear system $\widehat{A} y = \widehat{b}$ after a new DWT applied to $Ax = b$, where $\widehat{A} = \widehat{W} A \widehat{W}^\top$, $x = \widehat{W}^\top y$ and $\widehat{b} = \widehat{W} b$. We hope to select an efficient preconditioner M^{-1} to matrix $\widehat{A}$ based on operator (matrix) splitting, that is, $M = \widehat{D}$ and $\widehat{A} = \widehat{D} + \widehat{C}$.

The main issue to bear in mind is the following: any partition $\widehat{D}$ of $\widehat{A}$ corresponds to a partition D of matrix A (via an inverse DWT process). The latter partition, directly related to operators, must include all singularities; this will ensure a good eigenvalue distribution for the preconditioned matrix and its normal ([12]). Such a selection idea seems difficult to realize. Therefore we consider the reverse process.

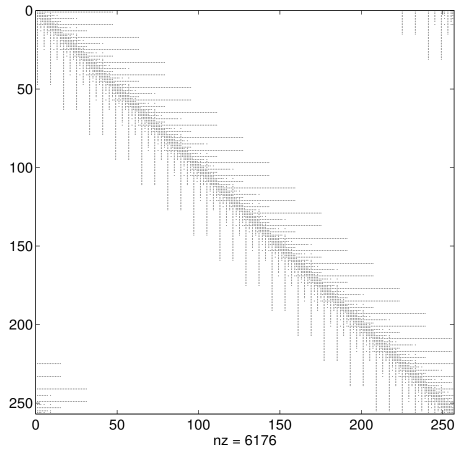

FIG. 5.2. $P(WAW^\top)P^\top$ *for a diagonal matrix A (LEV* $= 5$ *and* n *= 256)*

The strategy that we take is to start with a preliminary partition $A = D + C$ and consider the linear relationships of A, D, C under a DWT, where we may assume as in §2.1 D is a *Band(*α, α) for some integer α (say $\alpha = 1$ for a tridiagonal matrix). First, apply the new DWT with a smaller number $\ell = LEV \leq (L + 1 - r) - 2 = L - r - 1$) of wavelet levels, to give

$$\widehat{A}y = (\widehat{D} + \widehat{C})y = \widehat{b}, \tag{6.1}$$

where $\widehat{D} = \widehat{W}D\widehat{W}^\top$, $\widehat{C} = \widehat{W}C\widehat{W}^\top$ and $\widehat{b} = \widehat{W}b$. Now $\widehat{D}$ is also a band matrix (with wrap-around boundaries) and more specifically it is at most *Band(*λ, λ) matrix with λ as predicted according to Theorem 5.3. Let B denote the *Band(*λ, λ) part of matrix $\widehat{A}$, and we can identify the composition of B in terms of $\widehat{D}, \widehat{C}$. Specifically $B = \widehat{D} + C_d$, where $\widehat{D}$ is enclosed in B and C_d is the band part of matrix $\widehat{C}$ that falls into the sparsity pattern of B. Secondly, partition matrix $\widehat{C} = C_d + C_f$ via B. That is, C_f contains the remaining elements of $\widehat{C}$. Finally, with $B = \widehat{D} + C_d$, we effectively partition the coefficient matrix of (6.1) by $(\widehat{D} + \widehat{C}) = (D_f + C_f) = B + C_f$, with $D_f = B = \widehat{D} + C_d$. Thus $M^{-1} = D_f^{-1}$ will be used as a preconditioner. Using inverse transforms, one can see that using the sparse matrix $M = D_f$ is spectrally equivalent to using a full matrix to precondition matrix A. So the use of DWT is a way to achieve this purpose efficiently.

It should be remarked that, in implementations, the wavelet transform is applied to A, not to D and C separately. The above discussion is mainly to identify the band structure B (see Fig.5.2) and to explain the exact inclusion of $\widehat{W}D\widehat{W}^\top$ (or D) in matrix M. Thus a new algorithm can be stated as follows:

Algorithm 6.1.

1. *Decide on an operator splitting* $A = D + C$ *with* D *a band matrix;*
2. *Apply the new DWT to* $Ax = b$ *to obtain* $\widehat{A}y = \widehat{b}$*;*
3. *Determine a band-width* μ *from Theorem 5.3 (to find B that encloses* $\widehat{D}$*);*
4. *Select the preconditioner as the inverse of a band-width* μ *matrix of* $\widehat{A}$ *and use it to solve* $\widehat{A}y = \widehat{b}$ *iteratively.*

Here the band size $\mu = \lambda_1 + \lambda_2 + 1$ (total band width) is known in advance, once m (wavelet order) and ℓ (wavelet levels) have been selected, and generally small, with respect to problem size n. For example if D is *Band(*$1, 1$) (tridiagonal), with $m = 6$ and $\ell = 3$, the total band width for B will be $\mu = 2 * 6 * (2^{3-1} - 1) + 1 = 37$.

Note that Algorithm 2 includes Algorithm 1 as a special case if λ is chosen as a fixed integer in the second step.

Remark 6.1. *Recall that wavelets compress well for smooth (or for smooth parts of) functions and operators. Here matrix C, after cutting off the non-smooth parts D of A, is smooth except on the bands near the cuts (artificially created). So most nonzeros in $\widehat{C}$ will be centered around these cuts and C_d will be significant.*

7. Numerical results. To test the preconditioner, as proposed in Algorithm 2, we have solved the following two examples.

Example 1 - Singular BIE for Helmholtz equation. The exterior Helmholtz equation (see [1])

$$(\nabla^2 + \mathtt{k}^2)\phi(p) = 0$$

is of importance in acoustic scattering problems. The interior boundary Γ is the ellipse $(x/0.5)^2 + (y/2)^2 = 1$. For Neumann's boundary conditions, a unique BIE formulation due to Burton and Miller is the following:

$$(-\frac{1}{2}\mathcal{I} + \mathcal{M}_{\mathtt{k}} + i\eta\mathcal{N}_{\mathtt{k}})\phi = [\mathcal{L}_{\mathtt{k}} + i\eta(\frac{1}{2}\mathcal{I} + \mathcal{M}_{\mathtt{k}}^T)]\frac{\partial \phi}{\partial n}.$$

Here $\mathcal{L}_{\mathtt{k}}$ and $\mathcal{M}_{\mathtt{k}}$ are the usual single and double layer potential operators respectively,

$$(\mathcal{L}_{\mathtt{k}}\phi)(p) = \int_\Gamma G_{\mathtt{k}}(p,q)\phi(q)dS, \qquad (\mathcal{M}_{\mathtt{k}}\phi)(p) = \int_\Gamma \frac{\partial G_{\mathtt{k}}}{\partial n_q}\phi(q)dS.$$

$Theoperator\mathcal{M}_{\mathtt{k}}^T$ is the adjoint of $\mathcal{M}_{\mathtt{k}}$ and $\mathcal{N}_{\mathtt{k}}$ is the hypersingular operator. Recall that the 2D Green function is $G_{\mathtt{k}}(p,q) = \frac{i}{4}H_0^{(1)}(\mathtt{k}|p-q|)$ where $H_0^{(1)}$ is the Hankel function. Refer to [1, 2, 3]. Here we have tested the case of wavenumber $\mathtt{k} = 10$ and $\eta = 1$ using the collocation method.

Example 2 - Cauchy SIE. Singular integral equations (SIE) of Cauchy type are important in fracture mechanics. Consider a Cauchy SIE

$$\left\{ \begin{array}{ll} \frac{1}{\pi}\int_{-1}^1 \frac{w(t)\phi(t)}{t-x}dt + \int_{-1}^1 \frac{(t^2-x^2)^2}{t^2+x^2}w(t)\phi(t)dt = f(x), & x \in (-1,1), \\ \frac{1}{\pi}\int_{-1}^1 w(t)\phi(t)dt = 0, & \end{array} \right. \tag{7.1}$$

with the exact solution $\phi(x) = x|x|$, where $w(t) = (1-t^2)^{-1/2}$; see [14, 15, 18, 23] for full details. Here

$$f(x) = \frac{2}{\pi}\left((1 + x^2\omega\log\left|\frac{(1-x)\omega+1}{(x-1)\omega+1}\right|\right) \qquad \text{with} \qquad \omega = \frac{1}{\sqrt{1-x^2}}.$$

We choose this example because the integral equation resembles a singular BIE in the sense that the operator has a singular principal part and a smooth part.

Tables 7.1-7.5 show the number of iteration steps required to reduce the residual error to be of the same magnitude as the discretization error; (see [27] for an earlier use of this kind of stopping criteria). This error refers to the root mean square (RMS) error of the computed solution against the exact solution at all nodal points. The symbol '*' indicates a very slow (or no) convergence. Recall that m denotes the order of Daubechies' wavelets, ℓ denotes the number of wavelet levels used, M^{-1} is the preconditioner, and λ refers to a *Band*(λ,λ) matrix

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 7.1
Example 1: Convergence steps of CGN (Daubechies' $m = 4$)

Size	LEV	Diagonal $D = D_1$			Tridiagonal $D = D_2$			Case
N	ℓ	Band λ	Steps	$M = D$	Band λ	Steps	$M = D$	$M = I$
64	2	4	14	17	5	14	15	19
	3	12	13		13	13		
128	2	4	17	32	5	16	18	41
	3	12	16		13	16		
	4	28	14		29	14		
256	2	4	22	72	5	22	24	91
	3	12	19		13	19		
	4	28	17		29	17		
	5	60	19		61	19		
512	2	4	27	162	5	26	42	199
	3	12	20		13	20		
	4	28	20		29	20		
	5	60	19		61	19		
1024	2	4	29	332	5	29	88	420
	3	12	25		13	25		
	4	28	21		29	21		
	5	60	21		61	21		

for selecting M. The value of λ is calculated according to Theorem 5.3. As a DWT takes O(N) operations and a band solver takes about O($N\lambda^2$) operations, the one-off setting up of the preconditioner should not take more than one (full) matrix vector multiplication and this implies $\lambda \approx \sqrt{N}$. For Table 7.2, the "$\ell = LEV = 5$" case is expensive (included here only for a comparison) and $\ell = LEV = 3$ or 4 are more suitable. Therefore each preconditioning step takes about O(N) operations if DWT level ℓ is fixed (say 3) and so CPU times for each case are proportional to the number of iterations.

The results clearly demonstrate that for dense linear systems arising from singular BIE's, preconditioning is necessary and all preconditioners give faster convergence than the unpreconditioned case of $M = I$. The performance of the CGN and GMRES are similar. Further, simple operator splitting preconditioners are effective (with a speed up of a factor up to 5) but their performance are much improved after our DWT acceleration (with a further speed up of factor up to 4). In particular, as our theory predicted, the choice of band size λ is sufficient as it is based on an over estimate. This can be observed more clearly in Tables 7.2 and 7.4 with $m = 8$.

8. Conclusions. Simple sparse preconditioners (diagonal, bi-diagonal, tri-diagonal) based on operator splitting can be useful for solving dense boundary element systems iteratively. The discrete wavelet transforms can be utilized independently to design alternative sparse preconditioners based on approximate inverse approximation.

We have proposed to combine both ideas to generate new and more efficient preconditioners. To implement this idea, one has to solve the transformed preconditioning equation with a 'finger'-like matrix. Here matrix permutations are proposed to convert 'finger'-like matrices into band ones so that the preconditioning step can be efficiently carried out. To this end, we have presented a new DWT that can be implemented directly.

Thus the combination of a DWT with an operator splitting gives rise to a more effective preconditioner, without much increase of the computational work. Numerical results have

TABLE 7.2
Example 1: Convergence steps of CGN (Daubechies' $m = 8$)

Size	LEV	Diagonal $D = D_1$			Tridiagonal $D = D_2$			Case
N	ℓ	Band λ	Steps	$M = D$	Band λ	Steps	$M = D$	$M = I$
64	2	8	13	17	9	13	15	19
	3	24	10		25	10		
128	2	8	15	32	9	15	18	41
	3	24	15		25	15		
	4	56	9		57	9		
256	2	8	18	72	9	18	24	91
	3	24	17		25	17		
	4	56	15		57	15		
	5	120	10		121	10		
512	2	8	23	162	9	23	42	199
	3	24	19		25	19		
	4	56	18		57	18		
	5	120	21		121	21		
1024	2	8	29	332	9	29	88	420
	3	24	22		25	22		
	4	56	20		57	20		
	5	120	19		121	19		

also confirmed this conclusion.

For boundary integral problems with fixed singularities [19], induced by geometric singularities, the method proposed here would still work since such fixed singularities occur along the diagonal. For dense matrix problems from other applications (for example, image processing), singularities in the matrix may not be along the diagonal and it may not always be possible to permute to a block form as above. This problem is beyond the scope of the present paper but more research work is needed.

9. Acknowledgement. The author wishes to thank the anonymous referee for making helpful comments.

REFERENCES

[1] S. AMINI, P. J. HARRIS, AND D. T. WILTON, *Coupled Boundary and Finite Element Methods for the Solution of the Dynamic Fluid-Structure Interaction Problem*, SPRINGER-VERLAG, 1992.

[2] S. AMINI S. AND N. MAINES, *Iterative solutions of boundary integral-equations*, IN BOUNDARY ELEMENT METHODS 14, C.A. BREBBIA, ED., CM PUBLICATIONS, 1994.

[3] ———, *Qualitative properties of boundary integral operators and their discretizations*, MATHEMATICS TECHNICAL REPORT MCS-95-12, UNIVERSITY OF SALFORD, UK, 1995.

[4] ———, PRECONDITIONED KRYLOV SUBSPACE METHODS FOR BOUNDARY ELEMENT SOLUTION OF THE HELMHOLTZ EQUATION. TECHNICAL REPORT MCS-95-05, DEPT OF MATHS AND COMP. SCI., UNIVERSITY OF SALFORD, UK, 1995.

[5] K. ATKINSON, *The Numerical Solution of Integral Equations of the Second Kind*, CAMBRIDGE UNIV. PRESS, CAMBRIDGE , 1997.

[6] O. AXELSSON AND V. A. BARKER, *Finite element solution of boundary values problems: theory and computation*. ACADEMIC PRESS, ORLANDO, FLA., 1984.

[7] G. BEYLKIN, R. COIFMAN, AND V. ROKHLIN, *Fast wavelet transforms and numerical algorithms I*, COMM. PURE APPL. MATH., 44 (1991), PP. 141-183.

[8] D. BOND AND S. VAVASIS, *Fast wavelet transforms for matrices arising from boundary element methods*, COMPUTER SCIENCE RESEARCH REPORT TR-174, CORNELL UNIVERSITY, 1994.

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 7.3
Example 1: Convergence steps of GMRES(5) (Daubechies' $m = 4$)

Size N	LEV ℓ	Diagonal $D = D_1$ Band λ	Steps	$M = D$	Tridiagonal $D = D_2$ Band λ	Steps	$M = D$	Case $M = I$
64	2	4	6	20	5	7	7	14
	3	12	6		13	6		
128	2	4	10	31	5	10	15	27
	3	12	8		13	8		
	4	28	6		29	6		
256	2	4	10	37	5	10	35	58
	3	12	10		13	7		
	4	28	10		29	8		
	5	60	8		61	8		
512	2	4	27	45	5	27	46	145
	3	12	12		13	12		
	4	28	8		29	12		
	5	60	12		61	12		
1024	2	4	47	53	5	47	51	306
	3	12	19		13	19		
	4	28	13		29	13		
	5	60	13		61	13		

[9] C.A. BREBBIA, J.C.F. TELLES, AND L.C. WROBEL, *Boundary Element Techniques: Theory and Applications in Engineering*, SPRINGER-VERLAG, BERLIN, 1984.

[10] *T. F. Chan, W. P. Tang, and W. L. Wan*, *Wavelet Sparse Approximate Inverse Preconditioning*, BIT, 37 (1997), PP. 644-660.

[11] K. CHEN, *Preconditioning boundary element equations*, IN BOUNDARY ELEMENTS, IMPLEMENTATION AND ANALYSIS OF ADVANCED ALGORITHMS : PROCEEDINGS OF THE TWELFTH GAMM-SEMINAR, W. Hackbusch and G. Wittum, eds.,Braunschweig/Wiesbaden, Vieweg, 1996, pp. 57-67.

[12] K. CHEN, *On a class of preconditioning methods for dense linear systems from boundary elements*, SIAM J. Sci. Comput., 20 (1998), pp. 684-698.

[13] K. CHEN. AND C. H. LAI, PARALLEL SOLUTIONS OF BOUNDARY ELEMENT EQUATIONS BY A FLEXIBLE ELIMINATION PROCESS, Contemp. Math., 218 (1998), pp. 311-317.

[14] J. A. CUMINATO, *On the uniform convergence of a collocation method for a class of singular integral equations*, BIT, 27 (1987), pp. 190-202.

[15] F. ERDOGAN AND G. D. GUPTA, *On the numerical solution of singular integral equations*, Quart. Appl. Math., 30 (1972), pp. 525-534.

[16] W., DAHMEN, *Wavelet and multiscale methods for operator equations*, Acta Numerica, 6, Cambridge Univ. Press, 1997.

[17] W. DAHMEN, B. KLEEMANN, S. PROSSDORF, AND R. SCHNEIDER, *Multiscale methods for the solution of the Helmholtz and Laplace equations*, in Boundary Element Topics, Wolfgang L. Wendland. ed., Springer-Verlag, 1996.

[18] G. H. Golub and C. Van Loan, *Matrix Computation*, 2nd ed, John Hopkins Univ. Press, Baltimore, 1989.

[19] I. G. GRAHAM AND G. A. CHANDLER, *Higher order methods for linear functionals of solutions of second kind integral equations*, SIAM J. Numer. Anal, 25 (1988), pp. 1118-1137.

[20] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comp. Phys., 73 (1987), pp. 325-348.

[21] 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.

[22] A. HARTEN, *Multiresolution representation and numerical algorithms: a brief review*, CAM report 94-12, Dept of Mathematics, UCLA, USA, 1994.

[23] N. I. IOAKIMIDIS AND P. S. THEOCARIS, *A comparison between the direct and the classical numerical methods for the solution of Cauchy type singular integral equations*, SIAM J. Numer. Anal., 17 (1980), pp. 115-118.

[24] C. H. LAI, *An extension of Purcell's vector method with applications to panel element equations Comput.*

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 7.4
Example 1: Convergence steps of GMRES(5) (Daubechies' $m = 8$)

Size	LEV	Diagonal $D = D_1$			Tridiagonal $D = D_2$			Case
N	ℓ	Band λ	Steps	$M = D$	Band λ	Steps	$M = D$	$M = I$
64	2	8	7	20	9	7	7	14
	3	24	5		25	5		
128	2	8	8	31	9	8	15	27
	3	24	7		25	7		
	4	56	6		57	6		
256	2	8	11	37	9	11	35	58
	3	24	11		25	11		
	4	56	9		57	9		
	5	120	5		121	5		
512	2	8	13	45	9	13	46	145
	3	24	12		25	12		
	4	56	12		57	12		
	5	120	11		121	11		
1024	2	8	34	53	9	34	51	306
	3	24	13		25	13		
	4	56	13		57	13		
	5	120	14		121	14		

TABLE 7.5
Example 2: Convergence steps of GMRES(9) with $\ell = 3$ and $m = 4$

Size	Case	Tridiagonal $D = D_2$		Bi-diagonal $D = D_3$	
N	$M = I$	Steps	$M = D$	Steps	$M = D$
64	19	2	30	3	10
128	80	3	40	3	13
256	*	4	50	5	23
512	*	3	46	7	24
1024	*	4	38	10	39

Math. Appl., 33 (1997), pp. 101-114.
[25] D. MILLER, *Wavelet transforms and linear algebra*, MSc thesis, Inst. Adv. Sci. Comput., University of Liverpool, UK, 1995.
[26] WILLIAM H. PRESS, SAUL A. TEUKOLSKY, WILLIAM T. VETTERLING, AND BRIAN P. FLANNERY,*Numerical Recipes in C. The Art of Scientific Computing. 2nd ed.*, Cambridge Univ. Press, Cambridge, 1992.
[27] L. REICHEL, *Parallel iterative methods for the solution of Fredholm integral equations of the second kind*, in Hypercube Multiprocessors 1987, M. T. Heath, ed. SIAM publications, pp. 520-529, 1987.
[28] R. SCHNEIDER, P. L. LEVIN, AND M. SPASOJEVIC,*Multiscale compression of BEM equations for electrostatic systems* IEEE Trans. Dielectrics Elec. Insulation, 3 (1996), pp. 482-493.
[29] G. STRANG AND T. NGUYEN, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA, 1996.
[30] W. SWELDENS, *The construction and application of wavelets in numerical analysis*, PhD. thesis, Catholic University of Leuven, Belgium, 1994.