An Operator Splitting Preconditioner for Matrices Arising from a Wavelet Boundary Element Method for the Helmholtz Equation

Stuart C. Hawkins
Department of Mathematical Sciences,
University of Liverpool, Liverpool L69 7ZL, UK.

Ke Chen
Department of Mathematical Sciences,
University of Liverpool, Liverpool L69 7ZL, UK.

Paul J. Harris
School of Computing, Mathematical and Information Sciences,
University of Brighton, Brighton BN2 4GJ, UK.

An operator splitting type preconditioner is presented for fast solution of linear systems obtained by Galerkin discretisation of the Burton and Miller formulation for the Helmholtz equation. Our approach differs from usual boundary element treatments of the three dimensional scattering problem because we use a basis of biorthogonal wavelets. Such wavelets result in a sparse linear system and that facilitates preconditioning and makes matrix vector products cheap to form. In this Part I of our work, we implement a biorthogonal wavelet transform on a closed surface in three dimensions. Numerical results demonstrate the gains in efficiency that are already achievable with this convenient but non-optimal implementation.

1. Introduction

We are concerned with the fast solution of the exterior Helmholtz equation, which models the steady state acoustic field in a fluid surrounding an object[1,7]. A boundary element method using a Galerkin scheme with a biorthogonal wavelet basis is used. Our primary focus is on the construction of a new preconditioner for the resulting sparse linear system. Previous work[7] used an operator splitting preconditioner for the dense linear system obtained from a boundary element method.

The Helmholtz equation is reformulated as a boundary integral equation (BIE) using the Burton and Miller formulation[1]. This formulation possesses a unique solution for all frequencies. Although most boundary integral operators have good spectral properties, the Burton and Miller formulation involves the hypersingular operator, which produces ill conditioning in the discretised operator that grows roughly linearly with the number of unknowns. We shall consider problems with a relatively low wavenumber.

Solution of the BIE using a traditional single scale boundary element method

(BEM) produces a linear system with a dense matrix having n^2 nonzero entries, where n is the number of unknowns. Beylkin, Coifman and Rokhlin[2] showed that the use of a wavelet basis produces a linear system with a matrix that has many small entries. Perturbing the system by thresholding the matrix yields a sparse matrix but retains a specified accuracy in the solution. Developments by Dahmen, Prössdorf and Schneider[12], Petersdorff and Schwab[26], Lage and Schwab[21], Harbrecht[16], and Harbrecht and Schneider[17] yield sophisticated wavelet discretisation and compression methods with the key properties: (a) the matrix has $O(n \log^a n)$ nonzero entries, for some small integer a; (b) the asymptotic order of convergence of the method is not changed by the compression.

Optimal implementation of the wavelet discretisation requires designing suitable quadrature rules, which are applicable over entire manifolds. This is in contrast to simple quadrature rules that are applied over small subdomains in a traditional BEM. Here, in Part I of our work, we consider the use of a discrete biorthogonal-wavelet transform (DWT) for dense linear systems coming from a BEM. Our motivation in this work is to study the discretised operator, whose sparsity can be used to facilitate preconditioning and fast iterative solution of the linear system. In the sequel we present an optimal implementation using wavelets without computing a dense BEM matrix, although the two methods are equivalent. We remark that it is not a trivial task to generalise the familiar one dimensional DWT over a periodic domain to the two dimensional DWT over an unstructured manifold. We use a technique described by Harbrecht[16] and Dahmen and Schneider[13] that represents the manifold as a union of patches on which wavelets can be defined and glued together. We describe an easy construction for the matrices representing the wavelet transform so that the wavelet transform is easy to implement.

Biorthogonal high order tensor product wavelets, defined on complicated geometries, were proposed by Harbrecht[16,17] for solving Laplace's equation. We use these wavelets for solving the Helmholtz equation. Our formulation includes the hypersingular operator for the Helmholtz problem, which we discretise using the high order Galerkin scheme of Harris and Chen[18]. The scheme was developed for a single scale basis on a triangular mesh. We adapt the scheme for tensor product wavelets defined on a curved surface with an underlying computational mesh that is rectangular. Without using the new Galerkin formulation one would be forced to use first order wavelets, whose compression would not be sufficient to realise the efficiency that we observe here. We believe that the discretisation of this Helmholtz equation using high order wavelets is novel. Application to complicated geometries is straightforward.

We now remark on preconditioning issues. Firstly, the complex matrix obtained from Galerkin discretisation is not Hermitian because the underlying integral operators are not Hermitian. This is in contrast to the real matrix obtained from the Laplacian operator, which is symmetric. It follows that the Conjugate Gradient Method[20,22] cannot be used for solution of the linear system so we use GMRES[24,25].

Algorithm 1.

(1) **Boundary Element Method.** Obtain linear system $A_{\mathbf{V}^{J+1}}x = b$.

(2) **Wavelet Transform.**
 (a) Apply DWT to A and b to get $A_{\mathbf{\Psi}^{J+1}}$ and $b_{\mathbf{\Psi}^{J+1}}$.
 (b) Threshold $A_{\mathbf{\Psi}^{J+1}}$ to get $\mathring{A}_{\mathbf{\Psi}^{J+1}}$.

(3) **Preconditioner.**
 (a) Threshold $\mathring{A}_{\mathbf{\Psi}^{J+1}}$ to get M.
 (b) Reorder $\mathring{A}_{\mathbf{\Psi}^{J+1}}$,$M$ and $b_{\mathbf{\Psi}^{J+1}}$ using spiral ordering.
 (c) Restrict sparsity pattern of M to bordered-band.
 (d) Compute LU factorisation of M

(4) **Solve Linear System** Solve $\mathring{A}_{\mathbf{\Psi}^{J+1}}\widetilde{x} = b_{\mathbf{\Psi}^{J+1}}$ using GMRES preconditioned with M.

Secondly, norm equivalence results for the wavelet basis mean that for elliptic problems there exists a diagonal preconditioner for the linear system that yields a condition number that is independent of the number of unknowns. However, even for elliptic problems, the bounded condition number can still be large in practice, and fast convergence of the iterative solver is not ensured by a low condition number when the iteration matrix is non-Hermitian. Attention has been given to further preconditioning. Cohen and Masson[10] show that it is sometimes important to incorporate into the preconditioner the off diagonal finger pattern present in the matrix. This is even more important when treating the hypersingular operator. We present a preconditioning strategy related to the operator splitting method[4,5,7]. There an essentially banded approximation to the matrix is constructed by splitting the boundary integral operator into two operators, involving "local" interactions and "far" interactions respectively. Because it is essentially banded this matrix is easy to invert by LU factorisation. This approach is complicated in the wavelet or multiscale setting because "local" has different meanings at and between different scales. The key to successfully using this technique in the wavelet setting is to carefully order the wavelet basis so that the easy factorisation property is retained.

Our method for solving the Helmholtz equation is summarised in Algorithm 1. The boundary element method is described in Section 2. The wavelet transform and the thresholding strategy are described in Section 3. A straightforward construction of the wavelet transform matrices is described in Section 4. The preconditioner is described in Section 5. The effectiveness of the preconditioner is demonstrated by numerical results presented in Section 6. In the sequel, Steps (1) and (2) will be replaced by the optimal implementation that computes the matrix $\mathring{A}_{\mathbf{\Psi}^{J+1}}$ directly from the wavelet basis.

2. The BIE Formulation

We briefly describe how to apply the boundary element method for the Helmholtz equation. For completeness we include some details on the boundary integral reformulation of the exterior Helmholtz equation problem.

Let $D \subseteq \mathbb{R}^3$ be a bounded region with surface Γ and exterior $D_+ = \mathbb{R}^3 \setminus (D \cup \Gamma)$. Consider the Helmholtz equation

$$\nabla^2 u(p) + k^2 u(p) = 0, \quad p \in D_+ \cup \Gamma, \tag{2.1}$$

where $k > 0$ is the wavenumber, with a Neumann boundary condition and with the Sommerfeld radiation condition $\lim_{r\to\infty} r\left(\frac{\partial u}{\partial r} - iku\right) = 0$. The single and double layer potential operators, respectively, are

$$(L_k u)(p) = \int_\Gamma G_k(p,q)u(q)ds_q, \quad (M_k u)(p) = \int_\Gamma \frac{\partial G_k}{\partial n_q}(p,q)u(q)ds_q,$$

where $G_k(p,q)$ is the free space Green's function for the Helmholtz equation and n_q is the unit outward normal to Γ at q. An application of Green's theorem yields the BIE formulation

$$-\frac{1}{2}u(p) + (M_k u)(p) = (L_k \frac{\partial}{\partial n_p} u)(p), \quad p \in \Gamma. \tag{2.2}$$

Since (2.2) does not possess a unique solution for all k (and those values of k for which a unique solution does not exist are generally not known) we differentiate (2.2) along n_p to give

$$-\frac{1}{2}\frac{\partial}{\partial n_p}u(p) + (N_k u)(p) = (M_k^T \frac{\partial}{\partial n_p} u)(p), \quad p \in \Gamma, \tag{2.3}$$

where

$$(N_k u)(p) = \frac{\partial}{\partial n_p}(M_k u)(p), \quad (M_k^T u)(p) = \frac{\partial}{\partial n_p}(L_k u)(p).$$

Taking a linear combination of (2.2) and (2.3) gives the Burton and Miller formulation

$$(-\frac{1}{2}I + M_k + \alpha N_k)u(p) = (\alpha \frac{1}{2}I + \alpha M_k^T + L_k)\frac{\partial}{\partial n_p}u(p), \quad p \in \Gamma, \tag{2.4}$$

where α is a coupling parameter. It can be shown that (2.4) has a unique solution for all $k > 0$ provided the imaginary part of α is nonzero[3]. We choose $\alpha = i/k$ to yield an almost optimal condition number[1] for the operator on the left hand side of (2.4).

For brevity write $A = -(1/2)I + M_k + \alpha N_k$ and denote by f the function on the right hand side of (2.4), which is derived from the Neumann boundary condition.

A is an operator of order 1 and $A : H^{1/2}(\Gamma) \to H^{-1/2}(\Gamma)$. We seek a solution of the weak form of (2.4), that is, we seek $u \in H^{1/2}(\Gamma)$ so that

$$\langle Au, v\rangle = \langle f, v\rangle, \quad \forall v \in H^{1/2}(\Gamma), \tag{2.5}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product $\langle u, v\rangle = \int_\Gamma u(q)\overline{v(q)}ds$.

The Galerkin solution is obtained by replacing $H^{1/2}(\Gamma)$ in (2.5) by some $V_h \subseteq H^{1/2}(\Gamma)$ to seek $u \in V_h$ so that

$$\langle Au, v\rangle = \langle f, v\rangle, \quad \forall v \in V_h.$$

Writing $A_{\mathbf{V}} = \langle A\mathbf{V}, \mathbf{V}\rangle$, $f_{\mathbf{V}} = \langle f, \mathbf{V}\rangle$ where $\mathbf{V} = [v_1, v_2, \ldots, v_n]$ is a basis for V_h, yields the linear system

$$A_{\mathbf{V}} u_{\mathbf{V}} = f_{\mathbf{V}}.$$

Each entry $(A_{\mathbf{V}})_{i,j}$ requires evaluation of the double integral

$$\int_\Gamma v_j(p)\ (N_k v_i)(p)\ ds_p = \int_\Gamma v_j(p) \int_\Gamma \frac{\partial^2 G_k}{\partial n_p \partial n_q}(p,q)\ v_i(q)\ ds_q ds_p. \tag{2.6}$$

When the supports of v_i and v_j intersect the inner integral on the right hand side of (2.6) cannot be integrated directly because it is hypersingular. However, when the basis $\mathbf{V}$ consists of piecewise constant functions this integral can be reformulated into an expression involving only weakly singular integrals[1,23]. This reformulation of the inner integral only can be used in both Galerkin or collocation methods. We require a reformulation of the full double integral developed by Harris and Chen[18] for high order Galerkin schemes because piecewise constant wavelets do not give sufficient compression for our purposes. The new expression for the right hand side of (2.6) then involves only weakly singular integrals that can be integrated without the use of special quadrature rules.

3. Biorthogonal Wavelets and Matrix Compression

In this section we describe how to apply the discrete wavelet transform to obtain the representation of the boundary element matrix in the wavelet basis. We then describe how to sparsify the matrix without compromising the accuracy of the solution. We include a brief review of the properties of biorthogonal wavelets but refer to Section 4 for the construction of the wavelets on a specific domain.

We start with a multiresolution analysis of nested subspaces

$$V_{j_0} \subseteq V_{j_0+1} \subseteq \ldots \subseteq L^2(\Gamma), \qquad \overline{\bigcup_{j=j_0}^{\infty} V_j} = L^2(\Gamma),$$

where V_{j_0} is the subspace at the coarsest scale. Each space V_j has a basis of piecewise linear *scaling functions* $\Phi_j = \{\phi_{j,k} \mid k \in S_j\}$ where S_j is an index set. Almost all of the scaling functions have the same form and thus satisfy the refinement equation

$$\phi_{j,k} = \sum_i \alpha_{i-k} \phi_{j+1,i}$$

where the coefficients α_i are finite in number and are fixed. This is precisely the kind of refinement equation satisfied by, for example, the Daubechies 4 scaling function[14]. When Γ is not infinite or periodic, some of the scaling functions that are near to the edge of Γ must be modified. This introduces a new refinement equation for each of the modified scaling functions. It is then convenient to combine the refinement equations in the more general form

$$\mathbf{\Phi}_j = \mathbf{\Phi}_{j+1} M_{j,0} \tag{3.7}$$

for some refinement matrix $M_{j,0} \in \mathbb{R}^{|S_{j+1}|\times|S_j|}$, with $\mathbf{\Phi}_j$ denoting the row vector with entries $\phi_{j,k}$.

Each V_{j+1} can be decomposed $V_{j+1} = V_j \oplus W_j$. Each space W_j has a basis of *wavelets* $\Psi_j = \{\psi_{j,k} \mid k \in T_j\}$ where $T_j = S_{j+1} \setminus S_j$. Since $W_j \subseteq V_{j+1}$ we can write

$$\mathbf{\Psi}_j = \mathbf{\Phi}_{j+1} M_{j,1} \tag{3.8}$$

for some refinement matrix $M_{j,1} \in \mathbb{R}^{|S_{j+1}|\times|T_j|}$, with $\mathbf{\Psi}_j$ denoting the row vector with entries $\psi_{j,k}$.

There also exist duals $\widetilde{\Phi}_j = \{\widetilde{\phi}_{j,k} \mid k \in S_j\}$ and $\widetilde{\Psi}_j = \{\widetilde{\psi}_{j,k} \mid k \in T_j\}$ that have analogous properties to the primals Φ_j and Ψ_j, and which satisfy

$$\begin{aligned} \langle \mathbf{\Phi}_j, \widetilde{\mathbf{\Phi}}_j \rangle &= I, \qquad & \langle \mathbf{\Psi}_j, \widetilde{\mathbf{\Phi}}_j \rangle &= 0, \\ \langle \mathbf{\Phi}_j, \widetilde{\mathbf{\Psi}}_j \rangle &= 0, \qquad & \langle \mathbf{\Psi}_j, \widetilde{\mathbf{\Psi}}_j \rangle &= I. \end{aligned}$$

In particular we have

$$\widetilde{\mathbf{\Phi}}_j = \widetilde{\mathbf{\Phi}}_{j+1} \widetilde{M}_{j,0}, \qquad \widetilde{\mathbf{\Psi}}_j = \widetilde{\mathbf{\Phi}}_{j+1} \widetilde{M}_{j,1}.$$

A basis of biorthogonal wavelets Ψ_j can be easily obtained[13,16] from a non-biorthogonal basis $\breve{\Psi}_j = \{\breve{\psi}_{j,k} \mid k \in T_j\}$ of W_j. Suppose that $\breve{\Psi}_j$ satisfies the refinement relation

$$\breve{\mathbf{\Psi}}_j = \mathbf{\Phi}_{j+1} \breve{M}_{j,1}. \tag{3.9}$$

Biorthogonal wavelets are then obtained using the relation

$$M_{j,1} = \breve{M}_{j,1} - M_{j,0} L_j \tag{3.10}$$

$$\mathbf{\Psi}_j = \breve{\mathbf{\Psi}}_j - \mathbf{\Phi}_j L_j \tag{3.11}$$

for $L_j = \widetilde{M}_{j,0}^T \breve{M}_{j,1}$. This ensures that $\langle \mathbf{\Psi}_j, \widetilde{\mathbf{\Phi}}_j \rangle = 0$ so that the Ψ_j are biorthogonal wavelets. The relation (3.9) is used in Section 4, where $\breve{M}_{j,1}$ is introduced to simplify the construction of wavelets on a surface comprising several square patches.

The refinement matrices $M_{j,0}$ and $M_{j,1}$ contain the coefficients for the wavelet transform, and depend on the geometry of the surface Γ. Full details of how to compute these matrices, for a given surface, are presented in Section 4.

The representation of A in the wavelet basis is obtained by applying a discrete wavelet transform to the boundary element matrix. First obtain a representation of A in the finest scaling function basis $\mathbf{\Phi}_{J+1}$ using

$$A_{\mathbf{\Phi}_{J+1}} = C_{J+1} A_{\mathbf{V}_{J+1}} C_{J+1}^T$$

where C_{J+1} is a sparse matrix satisfying

$$\mathbf{\Phi}_{J+1} = \mathbf{V}_{J+1} C_{J+1}^T,$$

and $A_{\mathbf{V}_{J+1}}$ denotes the boundary element matrix. Fix $J > j_0$. Then the representation in the wavelet basis $\mathbf{\Psi}^{J+1} = [\mathbf{\Phi}_{j_0}, \mathbf{\Psi}_{j_0}, \mathbf{\Psi}_{j_0+1}, \ldots, \mathbf{\Psi}_J]$ is obtained with

$$A_{\mathbf{\Psi}^{J+1}} = X_{j_0}^T X_{j_0+1}^T \ldots X_J^T A_{\mathbf{\Phi}_{J+1}} X_J \ldots X_{j_0+1} X_{j_0} \tag{3.12}$$

where X_j represents a single level of the wavelet transform at level $j+1$. In matrix form

$$X_j = \begin{pmatrix} M_{j,0} & M_{j,1} & 0 \\ 0 & 0 & I_{|S_{J+1}|-|S_{j+1}|} \end{pmatrix},$$

where I_n denotes the $n \times n$ identity matrix.

The vanishing moments of the wavelet basis mean that many of the entries of $A_{\mathbf{\Psi}^{J+1}}$ are sufficiently small that they can be discarded. Analysis of the operator and the wavelets[16,17] can predict *a priori* the location of these entries. Denote the convex hull of the support of $\psi_{j,k}$ by $\Theta_{j,k}$. We apply

$$\left(\widetilde{A}_{\mathbf{\Psi}^{J+1}}\right)_{(j,k)(j',k')} = \begin{cases} 0, & \text{for } \operatorname{dist}(\Theta_{j,k}, \Theta_{j',k'}) > B_{jj'}, \\ (A_{\mathbf{\Psi}^{J+1}})_{(j,k)(j',k')}, & \text{otherwise,} \end{cases} \tag{3.13}$$

which is a simplification of Equation 39 in[17]. The parameter $B_{jj'}$ depends on the order $2q$ of A, the vanishing moments $\widetilde{d}$ of the wavelets, and the regularity d of the wavelets:

$$B_{jj'} = a \max\left\{2^{-\min\{j,j'\}}, 2^{\frac{2J(\delta-q)-(j+j')(\delta+\widetilde{d})}{2(\widetilde{d}+q)}}\right\},$$

where $a > 1$, and δ satisfies $d < \delta < \widetilde{d} + 2q$.

In practice we replace the condition $\operatorname{dist}(\Theta_{j,k}, \Theta_{j',k'}) > B_{jj'}$ in (3.13) by the more easily evaluated condition

$$\|\mathbf{n}_{j,k} - \mathbf{n}_{j',k'}\|_2 - r_{j,k} - r_{j',k'} > B_{jj'}, \tag{3.14}$$

where $r_{j,k}$ approximates the radius $\sup_{\mathbf{x}\in\Theta_{j,k}} \|\mathbf{x} - \mathbf{n}_{j,k}\|_2$ of the support $\Theta_{j,k}$ of $\psi_{j,k}$. We set

$$r_{j,k} = \max_l \left\{ \|\mathbf{n}_{j,k} - \mathbf{n}_{j+1,l}\|_2 \ : \ [M_{j,0}, \breve{M}_{j,1}]_{(j+1,l)(j,k)} \neq 0 \right\}.$$

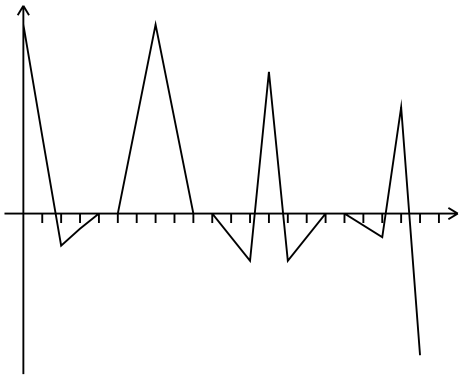
Further compression can be achieved by *a posteriori* thresholding $\widetilde{A}_{\mathbf{\Psi}^{J+1}}$. We apply

$$\left(\mathring{A}_{\mathbf{\Psi}^{J+1}}\right)_{(j,k)(j',k')} = \begin{cases} 0, & \text{for } \left|\left(\widetilde{A}_{\mathbf{\Psi}^{J+1}}\right)_{(j,k)(j',k')}\right| \leq \epsilon_{jj'}, \\ \left(\widetilde{A}_{\mathbf{\Psi}^{J+1}}\right)_{(j,k)(j',k')}, & \text{otherwise,} \end{cases} \tag{3.15}$$

where

$$\epsilon_{jj'} = \widetilde{c} \min\left\{2^{-|j-j'|}, 2^{-2(J-(j+j')/2)(\delta-q)/(\widetilde{d}+q)}\right\} 2^{2Jq} 2^{-2d'(J-(j+j')/2)},$$

Fig. 1. Edge scaling function and scaling function, wavelet and edge wavelet.



for some $d' > d$ and some parameter $\widetilde{c}$. This is Equation 43 in[17].

The ability to predict, using (3.13), the pattern of $\widetilde{A}_{\mathbf{\Psi}^{J+1}}$ allows efficient computation of this matrix by only computing directly from the wavelet basis the entries that are nonzero. An efficient scheme to do this is presented by Harbrecht[16,17], but will only be implemented in the sequel to this work. Here we focus on preconditioning $\mathring{A}_{\mathbf{\Psi}^{J+1}}$ for relatively small J. In Section 5 we see that the sparse structure makes construction of a preconditioner for $\mathring{A}_{\mathbf{\Psi}^{J+1}}$ easier than the construction of a preconditioner for $A_{\mathbf{V}_{J+1}}$. Since application of the wavelet transform is an $O(N_{J+1})$ procedure we suggest that preconditioning for $A_{\mathbf{V}_{J+1}}$ be implemented with our new preconditioner after applying a wavelet transform.

4. Wavelet Filters and Complexity

In this section we show how to apply the discrete wavelet transform used in the previous section. We assume that the surface Γ can be written as the union $\Gamma_1 \cup \Gamma_2 \cup \ldots \cup \Gamma_N$ of patches, each the image of the unit square under its corresponding diffeomorphism.

The refinement matrices $M_{j,0}$ and $M_{j,1}$ involved in the transform in the two dimensional case are too large to be explicitly stated here. Instead we explicitly state the refinement matrices for the one dimensional case and show how to obtain the refinement matrices for the two dimensional case from them. The one dimensional wavelets are derived[11,13,16] from biorthogonal wavelets on $\mathbb{R}$ developed by Cohen, Daubechies and Feauveau[9]. The wavelet are illustrated in Figure 1.

A glueing process takes refinement matrices or wavelets for a single patch and yields refinement matrices or wavelets for several patches that are joined together. However, the transform can be applied without consideration of these details, for which we refer to Harbrecht[16].

We start with the one dimensional case. Define $\mathbb{N}_k = \{0, 1, 2, \ldots, k\}$ and consider the mesh $S_j^{[0,1]} = \{x_k \mid k \in \mathbb{N}_{2^j}\}$ with nodes $x_k = k2^{-j}$. Then $s_j^{[0,1]}$ is an evenly spaced mesh over $[0,1]$. The refinement matrices $M_{j,0}^{[0,1]}$ and $M_{j,1}^{[0,1]}$ in (3.7) and (3.8) are given by

$$M_{j,0}^{[0,1]} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & & & & & & \\ \frac{7}{12} & \frac{1}{2} & & & & & \\ -\frac{1}{6} & 1 & & & & & \\ -\frac{1}{12} & \frac{1}{2} & \frac{1}{2} & & & & \\ & & 1 & \cdot & & & \\ & & \frac{1}{2} & \cdot & \frac{1}{2} & & \\ & & & \cdot & 1 & & \\ & & & & \frac{1}{2} & \frac{1}{2} & -\frac{1}{12} \\ & & & & & 1 & -\frac{1}{6} \\ & & & & & \frac{1}{2} & \frac{7}{12} \\ & & & & & & 1 \end{pmatrix}, \quad M_{j,1}^{[0,1]} = \frac{1}{\sqrt{2}} \begin{pmatrix} -\frac{3}{4} & & & & & & \\ \frac{7}{16} & -\frac{1}{8} & & & & & \\ -\frac{1}{8} & -\frac{1}{4} & & & & & \\ -\frac{1}{16} & \frac{3}{4} & -\frac{1}{8} & & & & \\ & -\frac{1}{4} & -\frac{1}{4} & \cdot & & & \\ & -\frac{1}{8} & \frac{3}{4} & \cdot & -\frac{1}{8} & & \\ & & -\frac{1}{4} & \cdot & -\frac{1}{4} & & \\ & & -\frac{1}{8} & \cdot & \frac{3}{4} & -\frac{1}{8} & \\ & & & \cdot & -\frac{1}{4} & -\frac{1}{4} & \\ & & & & -\frac{1}{8} & \frac{3}{4} & -\frac{1}{16} \\ & & & & & -\frac{1}{4} & -\frac{1}{8} \\ & & & & & -\frac{1}{8} & \frac{7}{16} \\ & & & & & & -\frac{3}{4} \end{pmatrix}.$$

To facilitate the glueing process for wavelets on a surface comprising several patches, we introduce $\breve{M}_{j,1}^{[0,1]}$ constructed using

$$\breve{M}_{j,1}^{[0,1]} = M_{j,1}^{[0,1]} + M_{j,0}^{[0,1]} L_j^{[0,1]}, \tag{4.16}$$

where $L_j^{[0,1]}$ is chosen so that the first and last rows of $\breve{M}_{j,1}^{[0,1]}$ are zero. This corresponds to modifying the edge wavelets. Thus

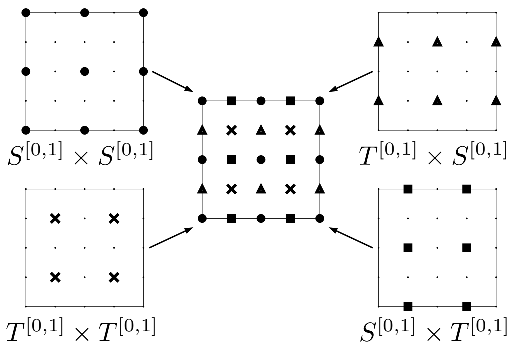
$$\breve{M}_{j,1}^{[0,1]} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & & & & & & \\ \frac{7}{8} & -\frac{1}{8} & & & & & \\ -\frac{1}{4} & -\frac{1}{4} & & & & & \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} & & & & \\ & -\frac{1}{4} & -\frac{1}{4} & \cdot & & & \\ & -\frac{1}{8} & \frac{3}{4} & \cdot & -\frac{1}{8} & & \\ & & -\frac{1}{4} & \cdot & -\frac{1}{4} & & \\ & & -\frac{1}{8} & \cdot & \frac{3}{4} & -\frac{1}{8} & \\ & & & \cdot & -\frac{1}{4} & -\frac{1}{4} & \\ & & & & -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ & & & & & -\frac{1}{4} & -\frac{1}{4} \\ & & & & & -\frac{1}{8} & \frac{7}{8} \\ & & & & & & 0 \end{pmatrix}, \qquad L_j^{[0,1]} = \begin{pmatrix} \frac{3}{4} & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & \frac{3}{4} \end{pmatrix}.$$

We can now consider the two dimensional case. Let $\square = [0,1] \times [0,1]$ and consider the mesh $S_j^{\square} = S_j^{[0,1]} \times S_j^{[0,1]}$ on $\square$. The mesh is illustrated in Figure 2.

Analogues of $M_{j,0}^{[0,1]}$ and $M_{j,1}^{[0,1]}$ on this mesh are given by[16]

$$\begin{aligned} M_{j,0}^{\square} &= M_{j,0}^{[0,1]} \otimes M_{j,0}^{[0,1]}, \\ \breve{M}_{j,1}^{\square} &= [\breve{M}_{j,1}^{[0,1]} \otimes \breve{M}_{j,1}^{[0,1]}, \breve{M}_{j,1}^{[0,1]} \otimes M_{j,0}^{[0,1]}, M_{j,0}^{[0,1]} \otimes \breve{M}_{j,1}^{[0,1]}], \end{aligned} \tag{4.17}$$

Fig. 2. Block ordering of a mesh on $\square$.



where $A \otimes B$ denotes the matrix Kronecker product

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ a_{21}B & a_{22}B & \dots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \dots & a_{nm}B \end{bmatrix}$$

for $n \times m$ matrix A and matrix B.

Similarly, we have

$$L_j^{\square} = [L_j^{[0,1]} \otimes L_j^{[0,1]}, L_j^{[0,1]} \otimes I_{|S_j^{[0,1]}|}, I_{|S_j^{[0,1]}|} \otimes L_j^{[0,1]}], \tag{4.18}$$

where, as before, I_n denotes the $n \times n$ identity matrix. Now $M_{j,1}^{\square}$ is obtained using (3.10):

$$M_{j,1}^{\square} = \breve{M}_{j,1}^{\square} - M_{j,0}^{\square} L_j^{\square}.$$

This is the reverse process of (4.16) and can be viewed as a reversal of the modification of the edge wavelets, which has propagated from $\breve{M}_{j,1}^{[0,1]}$ to $\breve{M}_{j,1}^{\square}$ via (4.17). The matrices $M_{j,0}^{\square}$ and $M_{j,1}^{\square}$ are required to implement the discrete wavelet transform (3.12) on $\square$.

Any ordering of nodes in a subset V of $S_j^{[0,1]} \times S_j^{[0,1]}$ can be represented by a vector whose entries index the nodes in the correct order. We may order V row at a time from the bottom to the top. This is often called the *lexicographical* or *natural* ordering. To facilitate a clear description of the discrete wavelet transform on $\square$ we introduce the following lexicographical node orderings:

$$\begin{array}{cl} \bullet & s_j \text{ of } S_j^{[0,1]} \times S_j^{[0,1]} \\ \times & t_{j,1} \text{ of } T_j^{[0,1]} \times T_j^{[0,1]} \\ \blacktriangle & t_{j,2} \text{ of } T_j^{[0,1]} \times S_j^{[0,1]} \\ \blacksquare & t_{j,3} \text{ of } S_j^{[0,1]} \times T_j^{[0,1]}. \end{array}$$

Let t_j denote the concatenation $[t_{j,1}, t_{j,2}, t_{j,3}]$. Then applying a single level of discrete wavelet transform to a vector z yields $\widetilde{z}$ with

$$\begin{bmatrix} \widetilde{z}(s_j) \\ \widetilde{z}(t_j) \end{bmatrix} = [M^{\square}_{j,0}, M^{\square}_{j,1}]^T z(s_{j+1})$$

where we use the Matlab notation for indexing multiple entries of a vector.

The multilevel discrete wavelet transform of z on $\square$ can now be easily described with the pyramid algorithm.

$$\begin{tikzcd}
 & \widetilde{z}(t_J) & \widetilde{z}(t_{J-1}) & & \widetilde{z}(t_{j_0}) \\
z(s_{J+1}) \arrow[r] \arrow[ur] & \widetilde{z}(s_J) \arrow[r] \arrow[ur] & \widetilde{z}(s_{J-1}) \arrow[r] \arrow[ur] & \cdots \arrow[r] \arrow[ur] & \widetilde{z}(s_{j_0})
\end{tikzcd}$$

The discrete wavelet transform on Γ is applied on a patch by patch basis. Let Γ consist of N patches $\Gamma_1, \Gamma_2, \ldots, \Gamma_N$, with each patch Γ_i the image of $\square$ under some diffeomorphism γ_i. We require that adjoining patches match at their common boundaries, so that when $\Gamma_i \cap \Gamma_j \neq \emptyset$ there exists an affine mapping $\theta_{ij} : \square \to \square$ such that, whenever $\xi \in \Gamma_i \cap \Gamma_j$ then $\xi = \gamma_i(s)$ and $\xi = \gamma_j(\theta_{ij}(s))$ for some $s \in \square$.

The mapping γ_i induces a mesh on Γ_i via $S_j^{\Gamma_i} = \gamma_i(S_j^{\square})$ and $S_j^{\Gamma} = \cup_{i=1}^{N} S_j^{\Gamma_i}$. Also $T_j^{\Gamma_i} = \gamma_i(T_j^{\square})$ and $T_j^{\Gamma} = \cup_{i=1}^{N} T_j^{\Gamma_i}$. Clearly some nodes will be contained within more than one patch. Let $r(\xi_k)$ denote the number of patches that contain node ξ_k.

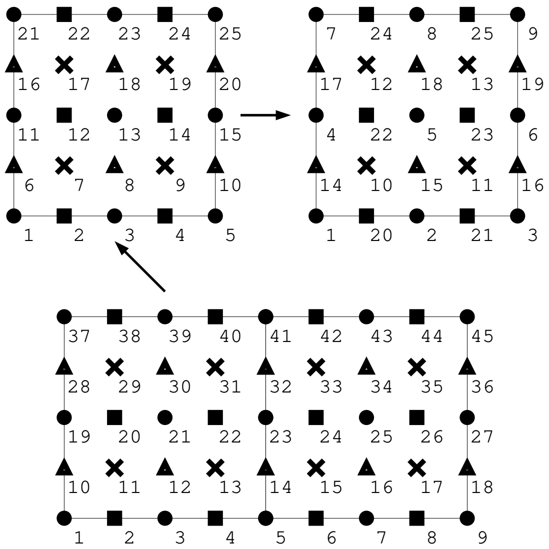
Let s_j^i, t_j^i be orderings of $S_j^{\Gamma_i}$ induced by s_j, t_j. For example, see the mesh illustrated in Figure 3, which has two patches. The wavelet transform $\widetilde{z}$ of a vector z on Γ is given by Algorithm 2[16].

5. Preconditioner

In this section we describe an effective sparse preconditioner for $\mathring{A}_{\boldsymbol{\Psi}^{J+1}}$ that incorporates off diagonal finger pattern entries. The preconditioner is a sparse approximation to $\mathring{A}_{\boldsymbol{\Psi}^{J+1}}$ that is cheap to invert by LU factorisation. This property is achieved by limiting the number of nonzeros in the preconditioner, and manipulating the sparsity pattern for the preconditioner using a reordering.

Preconditioners of this type are well established for matrices arising from boundary element methods for one dimensional periodic boundaries. The main idea there is to apply a modified discrete wavelet transform to the matrix. Chen[6] applies a permuted discrete wavelet transform DWTPer, which effectively orders the wavelet basis, from left to right, by the location of the supports of the wavelets. This creates a loose banded structure from the larger entries in the matrix. Ford[15] applies a further permutation after applying DWTPer so that rows and columns containing large entries corresponding to scaling functions are moved to the bottom and right of the matrix. This creates a *bordered band* structure from the larger entries in the matrix.

Fig. 3. Orderings on a surface comprising 2 patches. Global ordering (bottom). Ordering s_2^1 (top left). Ordering $[s_1^1, t_1^1]$ (top right).



Algorithm 2.

$\widetilde{z} = 0$

for $j = J, J-1, \ldots, j_0$

  for $k = 1, \ldots, N$

    $\widetilde{z}(s_j^k) = \widetilde{z}(s_j^k) + (M_{j,0}^{\square})^T \mathrm{diag}(r(s_{j+1}^k))^{-1} z(s_{j+1}^k)$

    $\widetilde{z}(t_j^k) = \widetilde{z}(t_j^k) + (\check{M}_{j,1}^{\square})^T \mathrm{diag}(r(s_{j+1}^k))^{-1} z(s_{j+1}^k)$

  end

  for $k = 1, \ldots, N$

    $\widetilde{z}(t_j^k) = \widetilde{z}(t_j^k) - (L_j^{\square})^T \mathrm{diag}(r(s_j^k))^{-1} z(s_j^k)$

  end

  $z = \widetilde{z}$

end

Note that, in contrast to the work presented here, the DWT in DWTPer and DWTPerMod is applied without consideration of the basis functions used to obtain the boundary element matrix. It follows that the resulting matrix cannot be identified as resulting from a Galerkin scheme with a wavelet basis.

DWTPer or DWTPerMod cannot be used for boundary integral equations on

Fig. 4. At level 4 there are $N_4 = 1538$ unknowns and the preconditioning level is 2. (a) The unordered matrix $A_{\mathbf{\Psi}^4}$ has a structure unsuited to LU factorisation. (b) The reordered matrix $A_{\mathbf{\Psi}^4}$ has a broadly banded structure, with (almost) dense blocks at the bottom and right. This is more suited to LU factorisation. (c) The matrix after Stage 2 of the preconditioning. This matrix approximates A but is much sparser. (d) The new preconditioner is a bordered band matrix $M \approx A_{\mathbf{\Psi}^4}$ obtained by chopping the matrix in (c) to a bordered band structure.

../Figures/spyw.eps

(a)

../Figures/spya.eps

(b)

../Figures/spyspai.eps

(c)

../Figures/spym.eps

(d)

surfaces because the simple one dimensional wavelet transforms on which they are based do not exist, and because there is no simple left to right ordering for nodes on a general surface. We develop an ordering of the wavelet basis that, like DWTPer, produces a banded pattern in the matrix. Further permutation ensures that coarse level basis functions (scaling functions and coarse wavelets) are ordered last to give, like DWTPerMod, a bordered band matrix.

Figure 4 (a) illustrates the structure of $\widetilde{A}_{\mathbf{\Psi}^{J+1}}$. Matrices with this form are difficult to invert directly because a lot of fill in occurs when an LU factorisation $M = LU$ is computed. Let M be obtained by thresholding $\widetilde{A}_{\mathbf{\Psi}^{J+1}}$ using (3.15) with a larger choice of $\widetilde{c}$ than that used to obtain $\mathring{A}_{\mathbf{\Psi}^{J+1}}$. Thus M approximates $\mathring{A}_{\mathbf{\Psi}^{J+1}}$, and contains entries from the finger pattern, but is sparser.

Our approach has three stages. First we attempt to place all nonzeros of M close to the diagonal by applying a *spiral ordering*[18]. Now M is almost banded. However, some rows and columns of M, corresponding to scaling function or coarse level wavelets, will be (almost) dense. The spiral ordering leaves these spread throughout the matrix, where they can cause fill in in the LU factorisation. The second stage is to reorder the wavelets again so that the dense rows are moved to the bottom of M, and the dense columns are moved to the right of M[15].

Figures 4 (b) and (c) illustrate the structure of $\widetilde{A}_{\mathbf{\Psi}^{J+1}}$ and the sparser $M \approx \widetilde{A}_{\mathbf{\Psi}^{J+1}}$ after the reordering has been applied. M has block structure

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$$

where M_{12}, M_{21}, and M_{22} are (almost) dense, and M_{11} is almost banded.

The third stage is to discard any entries of M_{11} outside a banded pattern to further limit the cost of computing the LU factorisation of M. Figure 4 (d) illustrates the final structure of M.

One may view this bordered band pattern of M as being dictated by factorisation considerations—the reordering is a way to obtain a good approximation to $\mathring{A}_{\mathbf{\Psi}^{J+1}}$ from this pattern.

Consider now the details of the reorderings applied. Since each wavelet in the basis corresponds to a node in the mesh, reordering the wavelet basis is equivalent to reordering the nodes. Suppose the nodes are already numbered in some way so that a reordering of the nodes can be represented by a vector, whose jth entry is the existing index of the new jth node.

Let $\mathcal{J}$ be the *spiral ordering* obtained from Algorithm 3. Under a reordering with $\mathcal{J}$ the matrix M is almost banded. To see this recall from (3.13) that, loosely speaking, $M_{(j,k)(j',k')}$ is nonzero whenever nodes k and k' are close together. In the spiral ordering such nodes are placed close together.

However, the meaning of *close together* depends on j and j'. When j or j' are small the entry $M_{(j,k)(j',k')}$ may be nonzero even when nodes k and k' are far apart. This accounts for the dense or almost dense rows and columns in M, without which M would be banded under the spiral ordering.

The ordering obtained from Algorithm 4 addresses this problem by moving the (almost) dense rows to the bottom of M, and moving the (almost) dense columns to the right of M. In Algorithm 4 we specify the rows/columns that we consider to be (almost) dense by the level J_p of their corresponding wavelets.

6. Numerical Experiments

In this section we present results illustrating the performance of the new preconditioner for two Helmholtz problems solved using GMRES. We consider the effects of problem size, restarting GMRES, sparsity of the preconditioner, and block size in the preconditioner. In all cases GMRES is applied until the relative residual norm is reduced by a factor of 10^{-8}, or until a limit of 100 iterations is reached. Where a preconditioner M is used we implement right preconditioning. Except where otherwise stated the experiments were performed on a PC with a 1GHz processor.

Algorithm 3.
Let $\mathbf{n}_{\mathcal{J}_1}$ denote the node that has the largest z coordinate.
for $j = 1, \ldots, |S_{J+1}| - 1$
    $\mathcal{J}_{j+1} = \operatorname{argmin}_{k \in S_J \setminus \{\mathcal{J}_1, \ldots, \mathcal{J}_j\}} \|\mathbf{n}_k - \mathbf{n}_{\mathcal{J}_j}\|_2 + \|\mathbf{n}_k - \mathbf{n}_{\mathcal{J}_1}\|_2$
end
Note: a tie is decided by minimising the angle in the xy-plane between the projection of $\mathbf{n}_k$ and the x-axis.

Algorithm 4.
$k = 1$
for $j = 1, \ldots, |S_{J+1}|$
    if level$(\mathbf{n}_{\mathcal{J}_j}) > J_p$
        $\mathcal{I}_k = \mathcal{J}_j$
        $k = k + 1$
    end
end
for $j = 1, \ldots, |S_{J+1}|$
    if level$(\mathbf{n}_{\mathcal{J}_j}) \leq J_p$
        $\mathcal{I}_k = \mathcal{J}_j$
        $k = k + 1$
    end
end

Previous work by the authors[19] described the preconditioner and presented results for the scattering problem exterior to the unit sphere with 1538 unknowns. Here we demonstrate the preconditioner for problems with 386, 1538 and 6146 unknowns. We present results for a range of preconditioners, obtained by varying parameters of the construction. Results are obtained for the scattering problems exterior to the unit sphere, and to a truncated-wedge, which models a wall-shaped structure.

Table 1 shows how the costs of assembling the preconditioner and solving the preconditioned system using GMRES(100) depend on the thresholding parameter $\widetilde{c}$ used to obtain M for the level $J = 4$ problem on the sphere with $N_4 = 1538$ unknowns and a preconditioning level $J_p = 2$. As $\widetilde{c}$ decreases the preconditioner becomes denser, and becomes a better approximation to $\mathring{A}_{\mathbf{\Psi}^{J+1}}$. This is reflected in the decrease in the number of GMRES iterations required to solve the problem. However, the increase in density of the preconditioner causes the LU factorisation time and back/forward substitution time to increase—in fact the benefit of fewer GMRES iterations is offset by these increases. The best strategy here is to use the sparsest preconditioner.

Table 2 shows how the costs of assembling the preconditioner and solving the preconditioned system using GMRES(100) depend on the preconditioning level for the level $J = 4$ problem on the sphere with $N_4 = 1538$ unknowns. The thresholding parameter $\widetilde{c}$ is fixed so that the behaviour with respect to J_p can be seen clearly. We choose $\widetilde{c} = 1.0 \times 10^{-4}$, which was shown in Table 1 to give the best performance. As J_p increases the preconditioner becomes denser. The consequences of this are (1) the LU factorisation time becomes high, and (2) the cost of back and forward substitution with L and U increases. Despite a corresponding decrease in the number of GMRES iterations we see that both the GMRES time and the LU factorisation time increase with J_p.

Sometimes the memory available for the solution of the linear system is limited, and it is necessary to restart the GMRES iteration[24,25]. Poor spectral properties of the iteration matrix typically cause the total number of GMRES iterations to increase when restarting is enforced. Table 3 shows how the number of GMRES(restart) iterations depends on restart for the level $J = 4$ problem on the sphere with $N_4 = 1538$ unknowns. The thresholding parameter $\widetilde{c}$ and preconditioning level J_p are fixed so that the behaviour with respect to the restart parameter can be seen clearly. We choose $\widetilde{c} = 1.0 \times 10^{-4}$ and $J_p = 2$, which were shown in Table 2 to give the best performance. As the restart value decreases the number of iterations required to solve the unpreconditioned linear system increases sharply until GMRES fails to converge. In contrast the number of iterations required to solve the preconditioned system increases only slightly.

Table 4 shows how the costs of assembling the preconditioner and solving the preconditioned system using GMRES(100) depend on the thresholding parameter $\widetilde{c}$ for the level $J = 4$ problem on the wall shaped domain with $N_4 = 1538$ unknowns, and a preconditioning level $J_p = 2$. The situation is similar to that shown in Figure 1 for the sphere.

Table 5 shows how the costs of assembling the preconditioner and solving the preconditioned system using GMRES(100) depend on the level J for the problem on the sphere, with a preconditioning level $J_p = 2$. Without preconditioning the number of GMRES iterations of unpreconditioned GMRES grows as J increases but with the new preconditioner the number of GMRES iterations remains constant.

We have shown that the most effective preconditioner is the sparsest available at the lowest preconditioning level. Our results indicate that the number of iterations required with this preconditioning strategy remain roughly constant with respect to both number of unknowns, and the restart parameter in GMRES.

GMRES iteration counts for our method are comparable to those observed in previous work for single scale Galerkin[18,7] and collocation[8] boundary element methods preconditioned with operator splitting preconditioners. Each iteration in our method is cheaper than each iteration in a BEM method because our matrices are sparse and the matrices in the BEM are dense.

7. Conclusions

By application of a discrete wavelet transform over a two dimensional surface we have obtained a sparse Galerkin discretisation of the Burton and Miller formulation for the three dimensional Helmholtz equation. We have introduced a new preconditioner, designed for problems with a moderate number of unknowns, that takes advantage of the sparsity introduced by the wavelets. The performance of the preconditioner is analysed with respect to its sparsity and to the number of unknowns. The new preconditioner is shown numerically to substantially reduce the cost of solving the sparse linear system using GMRES and is shown numerically to be stable with respect to restarting the GMRES iteration. Further work detailing an optimal implementation that obtains the sparse Galerkin discretisation matrix without using a BEM formulation will be presented later.

Acknowledgement

This work was supported by EPSRC research grant number GR/R22315/01.

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. G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure. and Appl. Math.*, 37:141–183, 1991.
3. A. J. Burton and G. F. Miller. The application of integral equation methods to the numerical solution of some exterior boundary-value problems. *R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci.*, 323:201–210, 1971.
4. K. Chen. Efficient iterative solution of linear systems from discretizing singular integral equations. *Electron. Trans. Numer. Anal.*, 2:76–91, 1994.
5. K. Chen. On a class of preconditioning methods for dense linear systems arising from boundary elements. *SIAM J. Sci. Comput.*, 20:684–698, 1998.
6. K. Chen. Discrete wavelet transforms accelerated sparse preconditioners for dense boundary element systems. *Electron. Trans. Numer. Anal.*, 8:138–153, 1999.
7. K. Chen and P. J. Harris. Efficient preconditioners for iterative solution of the boundary element equations for the three dimensional helmholtz equation. *Appl. Numer. Math.*, 36:475–489, 2001.
8. K. Chen and P. J. Harris. A discontinuous boundary element method for solving the three dimensional helmholtz problem. In *Proceedings UKBIM 2003*, pages 107–116, 2003.
9. A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45:485–560, 1992.
10. A. Cohen and R. Masson. Wavelet methods for second-order elliptic problems, preconditioning and adaptivity. *SIAM J. Sci. Comput.*, 21:1006–1026, 1999.
11. W. Dahmen, A. Kunoth, and K. Urban. Biorthogonal spline-wavelets on the interval—stability and moment conditions. *Appl. Comput. Harmon. Anal.*, 6:132–196, 1999.
12. W. Dahmen, S. Prössdorf, and R. Schneider. Wavelet approximation methods for pseudodifferential equations II: Matrix compression and fast solution. *Adv. Comput. Math.*, 1:259–335, 1993.

13. W. Dahmen and R. Schneider. Composite wavelet bases for operator equations. *Math. Comp.*, 68:1533–1567, 1999.
14. I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, XLI:909–996, 1988.
15. J. Ford. An improved discrete wavelet transform preconditioner for dense matrix problems. *SIAM J. Matrix Anal. Appl.*, 25:642–661, 2003.
16. H. Harbrecht. *Wavelet Galerkin Schemes For The Boundary Element Method In Three Dimensions*. PhD thesis, Technischen Universität Chemnitz, 2001.
17. H. Harbrecht and R. Schneider. Wavelet Galerkin schemes for 3D-BEM. Technical report, Technischen Universität Chemnitz, 2002.
18. P. J. Harris and K. Chen. On efficient preconditioners for iterative solution of a Galerkin boundary element equation for the three-dimensional exterior Helmholtz problem. *J. Comput. Appl. Math.*, 156:303–318, 2003.
19. S. C. Hawkins, K. Chen, and P. J. Harris. Easily inverted approximation type preconditioners for matrices arising from wavelet discretisations of boundary integral equations. In *Proceedings UKBIM 2003*, pages 165–174, 2003.
20. M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, December 1952.
21. C. Lage and C. Schwab. Wavelet Galerkin algorithms for boundary integral equations. *SIAM J. Sci. Comput.*, 20:2195–2222, 1999.
22. C. Lanczos. Solutions of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, 49(6):33–53, December 1952.
23. W. L. Meyer, W. A. Bell, and B. T. Zinn. Boundary integral solutions of three dimensional acoustic radiation problems. *J. Sound Vibration*, 59:245–262, 1978.
24. Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, second edition, 2003.
25. Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, July 1986.
26. T. von Petersdorff and C. Schwab. Fully discrete multiscale BEM. Technical Report 95-08, Seminar fr Angewandte Mathematik, Eidgenssische Technische Hochschule, Zrich, Switzerland, 1995.

$\widetilde{c}$	% nnz M	% nnz LU	LU/s	outer	inner	GMRES/s	total/s
no preconditioner				1	66	2.30	2.30
1.0×10^{-4}	1	2	0.07	1	29	1.15	1.22
3.2×10^{-5}	1	4	0.12	1	30	1.32	1.44
1.0×10^{-5}	2	5	0.13	1	26	1.18	1.31
3.2×10^{-6}	3	5	0.14	1	26	1.18	1.32
1.0×10^{-6}	3	5	0.15	1	26	1.21	1.35

Table 1. Results for sphere with level $J = 4$, $N_4 = 1538$ and preconditioning level $J_p = 2$. Solver is GMRES(100).

J_p	% nnz M	% nnz LU	LU/s	outer	inner	GMRES/s	total/s
2	1	2	0.07	1	29	1.15	1.22
3	1	11	3.00	1	22	1.29	4.30
4	2	16	16.13	1	17	1.19	17.32

Table 2. Results for sphere with level $J = 4$, $N_4 = 1538$ and thresholding parameter $\widetilde{c} = 1.0\times10^{-4}$. Solver is GMRES(100).

restart	LU/s	outer	inner	total	GMRES/s	total/s
100	no preconditioner	1	66	66	2.30	2.30
100	0.07	1	29	29	1.15	1.22
20	no preconditioner	5	13	93	3.12	3.12
20	0.07	2	9	29	1.18	1.24
10	no preconditioner	*10*	*10*	*100*	*3.46*	*3.46*
10	0.07	3	10	30	1.23	1.29
5	no preconditioner	*20*	*5*	*100*	*3.73*	*3.73*
5	0.07	7	4	34	1.51	1.58

Table 3. Results for sphere with level $J = 4$, $N_4 = 1538$ and preconditioning level $J_p = 2$ and $\widetilde{c} = 1\times10^{-4}$. Solver is GMRES(restart). Underlined entries indicate that GMRES did not converge within a total of 100 inner iterations.

$\widetilde{c}$	% nnz M	% nnz LU	LU/s	outer	inner	GMRES/s	total/s
no preconditioner				*1*	*100*	*3.77*	*3.77*
1.0×10^{-4}	1	2	0.06	1	39	1.54	1.60
3.2×10^{-5}	1	4	0.11	1	38	1.66	1.77
1.0×10^{-5}	2	5	0.14	1	38	1.78	1.92
3.2×10^{-6}	3	5	0.14	1	38	1.81	1.95
1.0×10^{-6}	3	6	0.16	1	38	1.84	1.99

Table 4. Results for wall with level $J = 4$, $N_4 = 1538$ and preconditioning level $J_p = 2$. Solver is GMRES(100). Underlined entries indicate that GMRES did not converge within a total of 100 inner iterations.

N_J	$\widetilde{c}$	% nnz M	% nnz LU	LU/s	outer	inner	GMRES/s	total/s
386	———	no preconditioner	———		1	41	0.33	0.33
386	1.0×10^{-3}	1	2	0.01	1	29	0.24	0.25
1538	———	no preconditioner	———		1	66	4.19	4.19
1538	1.0×10^{-4}	1	2	0.22	1	29	2.08	2.30
6146	———	no preconditioner	———		1	98	29.59	29.59
6146	3.2×10^{-5}	0	0	0.74	1	30	8.78	9.53

Table 5. Results for sphere with preconditioning level $J_p = 2$. Solver is GMRES(100). These results were obtained on a Sun Sparc (Solaris 2) machine. Underlined entries indicate that GMRES did not converge within a total of 100 inner iterations.