UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Two-Stage Preconditioners Using Wavelet Band Splitting and Sparse Approximation

Tony F. Chan
Ke Chen

July 2000

CAM Report 00-26

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

http://www.math.ucla.edu/applied/cam/index.html

Two-Stage Preconditioners Using Wavelet Band Splitting and Sparse Approximation[*]

Tony F. Chan[†]
Department of Mathematics,
University of California, Los Angeles,
405 Hilgard Ave, Los Angeles,
CA 90095-1555, USA.

and

Ke Chen[‡]
Department of Mathematical Sciences,
University of Liverpool,
Peach St, Liverpool,
L69 7ZL, UK.

AMS subject class: 65Y05, 65F35, 65Y20.

Keywords: Preconditioning, band splitting and approximation, sparse matrices, wavelets, minimization.

[*] This research work is partially supported by grants NSF ACR 97-20257, NASA Ames NAG2-1238, Sandia Lab LG-4440.

[†] Email: chan@math.ucla.edu. Web: http://www.math.ucla.edu/~chan

[‡] Email: k.chen@liverpool.ac.uk. Web: http://www.liv.ac.uk/~cmchenke

Abstract

The wavelet sparse approximate inverse preconditioners previously studied in [5, 18] are re-examined and improved for iterative solution of sparse linear systems arising from PDE's. Our new idea is to improve the approximation of a wavelet transformed matrix by banded matrices based on treating smooth and non-smooth splittings differently in a two-stage preconditioning setting. We introduce the concept of a wavelet band splitting and use it to derive a theoretical result on our two-stage preconditioners. Our preconditioner combines simple sparse scaling preconditioning with wavelet sparse approximate inversion. We propose an iterative method for finding the optimal splitting that minimises the wavelet band approximation errors for the diagonal case. Preliminary numerical experiments have been successful.

1 Introduction

The numerical solution of partial differential equations (PDE's) usually generates large systems of linear equations. Such linear systems can be too large to be solved by direct methods so iterative methods have to be developed and applied. An efficient approach is the combination of Krylov subspace methods and suitable preconditioning. Denote a linear system by

$$Ax = b, \tag{1}$$

where A is an $n \times n$ sparse and unsymmetric matrix. We consider a suitable choice of preconditioner M^{-1} for (1) so that the following preconditioned system

$$M^{-1}Ax = M^{-1}b \tag{2}$$

can be more efficiently solved by Krylov subspace methods.

To this end, there exist many types of sparse preconditioners [1], [11] [15] and [17]; see the references therein for details. Of particular interest is the so-called sparse approximate inverse (SPAI) preconditioner; see [2], [9] and [18]. One difficulty associated with SPAI is the determination of a suitable sparse pattern for the preconditioner M^{-1} that approximates the unknown matrix A^{-1}. In [5], a wavelet sparse approximate inverse preconditioner (WSPAI) was proposed using the wavelet transform to tackle this difficulty and extending the applicability of SPAI; see also [4] and [8]. Let W denote an orthogonal discrete wavelet transform (DWT) matrix and $\widetilde{A} = WAW^\top$ the representation of A in the wavelet basis; *throughout this paper a tilde will denote wavelet transformed quantities.* The idea is to look for SPAI in a wavelet basis rather than the original space for A. As WSPAI makes use of wavelet compression as well as SPAI, it is more efficient. However, it has been found that WSPAI can run into difficulties if in the wavelet space $\widetilde{A}$ cannot be accurately approximated by block or banded matrices. This is the case when the inverse of $\widetilde{A}$ has large entries away from a band part (including the main diagonal); e.g. in case of PDE's with discontinuous coefficients [5].

To improve WSPAI, following [5], we introduce for a matrix A new and fundamental concepts of *wavelet band splitting* $A = \mathbf{WB}_\mu(A) + \mathbf{WO}_\mu(A)$ and the related *wavelet band space* $\mathcal{WB}(\mu, \epsilon)$ for a bandwidth μ (where $\mathbf{WB}_\mu(A)$ becomes a band-μ matrix under a wavelet transform and $\mathcal{WB}(\mu, \epsilon)$ measures the dominance of $\mathbf{WB}_\mu(A)$ in A; see Definitions 1-2 in §2). In this new setting, whenever $A \in \mathcal{WB}(\mu, \epsilon)$, $\mathbf{WO}_\mu(A)$ is relatively small (precisely $\|\mathbf{WO}_\mu(A)\|_F^2 \le \epsilon \|\mathbf{WB}_\mu(A)\|_F^2$) and the wavelet transformed matrix $\widetilde{A}$ can be approximated accurately by a banded matrix to produce an effective preconditioner. If $A \notin \mathcal{WB}(\mu, \epsilon)$ for some tolerance ϵ, we look for a splitting of A:

$$A = X + C \qquad \text{and} \qquad A_1 = X^{-1}A = I + X^{-1}C, \tag{3}$$

such that the so-called stage 1 preconditioner, X^{-1}, smooths A so that $A_1 \in \mathcal{WB}(\mu, \epsilon)$ or $X^{-1}C \in \mathcal{WB}(\mu, \epsilon)$ (since $I \in \mathcal{WB}(\mu, 0)$ for any bandwidth $\mu \ge 0$). All we require of

X is that X is invertible (and easy to invert) and $\mathbf{WO}_\mu(A_1)$ is relatively smaller than $\mathbf{WB}_\mu(A_1)$ in some norm. Then a WSPAI type preconditioner is sought for matrix A_1. This will give rise to our two stage preconditioning strategy. In the applications that we are primarily interested in, vectors and matrices come from discretization of functions and operators, the concepts of smoothness and singularities carry over from the continuous case. In the context of wavelets, smoothness (of derivative functions) is directly connected to compression; see [3, 16]. Therefore, throughout the paper, we shall use the terms 'smoothness' and 'singularity' for vectors and matrices.

To motivate and illustrate our proposed methodology, consider firstly a smooth matrix A as characterised and tested in [3]

$$A_{ij} = \begin{cases} 1/(i-j), & \text{if } i \neq j, \\ 2, & \text{if } i = j. \end{cases} \tag{4}$$

Then $A \in \mathcal{WB}(\mu, \epsilon)$ because in the unique splitting $A = \mathbf{WB}_\mu(A) + \mathbf{WO}_\mu(A)$, even with a small bandwidth μ, $\mathbf{WO}_\mu(A)$ is small and dominated by $\mathbf{WB}_\mu(A)$ so $\widetilde{A}$, a finger-like matrix, can be approximated by a banded matrix accurately. With $n = 1024$, bandwidth $\mu = 16$, $\|\mathbf{WO}_{16}(A)\|_F^2 \leq \|\mathbf{WB}_{16}(A)\|_F^2$ i.e. $A \in \mathcal{WB}(16, 1)$. This is the case when WSPAI works well. Now consider secondly a simple diagonal modification of A from (4)

$$A^* = D + A, \tag{5}$$

with D a diagonal matrix with $D_{ii} = 10$ (odd i), -10 (even i). Then although A^* and A, after a DWT, yield an almost identical sparse pattern (finger-like), the wavelet band splitting of A^* differs from that of A. Namely $\mathbf{WO}_\mu(A^*)$ dominates $\mathbf{WB}_\mu(A^*)$ while $\mathbf{WB}_\mu(A)$ dominates $\mathbf{WO}_\mu(A)$. With $n = 1024, \mu = 16$, we found that $\|\mathbf{WO}_{16}(A^*)\|_F^2 \leq 2.34\|\mathbf{WB}_{16}(A^*)\|_F^2$ i.e. $A^* \in \mathcal{WB}(16, 2.34)$. However we hope to construct X such that $X^{-1}A$ is smoother in the sense of wavelet band splitting. For the simple choice $X = diag(A^*)$, with $n = 1024, \mu = 16$, we can observe an improvement: $\|\mathbf{WO}_{16}(X^{-1}A^*)\|_F^2 \leq 0.03\|\mathbf{WB}_{16}(X^{-1}A^*)\|_F^2$. Then $X^{-1}A^* \in \mathcal{WB}(16, 0.03)$; see Figure 1 for $\widetilde{A^*}$ and Figure 2 for $\widetilde{X^{-1}A^*}$. A further example comes from a discretization of the elliptic PDE

$$\frac{\partial}{\partial x}\left(a(x)\frac{\partial u}{\partial x}\right) + b(x)u = f(x). \tag{6}$$

If both coefficients a and b are smooth, the resulting matrix A is relatively smooth and in $\mathcal{WB}(\mu, \epsilon)$ with a small μ and $\epsilon \leq 1$. However if either a or b is discontinuous and oscillatory, matrix A is non-smooth, not in $\mathcal{WB}(\mu, \epsilon)$ for such μ, ϵ. and $\mathbf{WO}_\mu(A)$ will be more dominant than $\mathbf{WB}_\mu(A)$ in a wavelet band splitting. Then WSPAI will not provide a good preconditioner. A stage 1 preconditioning is needed before using a DWT to construct a stage 2 preconditioner.

In summary, this paper will address a class of matrix problems in the matrix space of $\mathcal{WB}(\mu, \epsilon)$ where both μ and ϵ are relatively large i.e. $\mathbf{WO}_\mu(A)$ is not small compared

Figure 1: Finger-like sparse pattern of $\widetilde{A^*}$. Note $A^* \in \mathcal{WB}(16, 2.34)$ meaning that A^* is not a smooth matrix as far as banded matrix approximations are concerned.

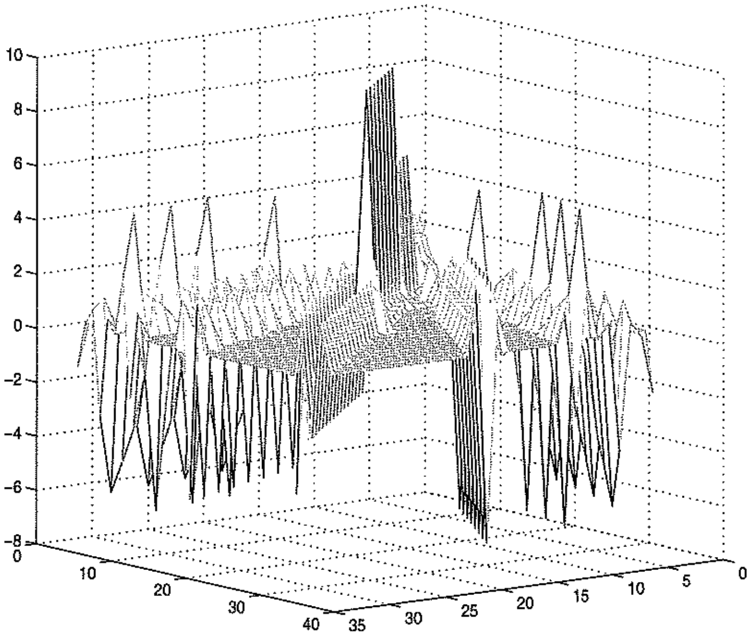
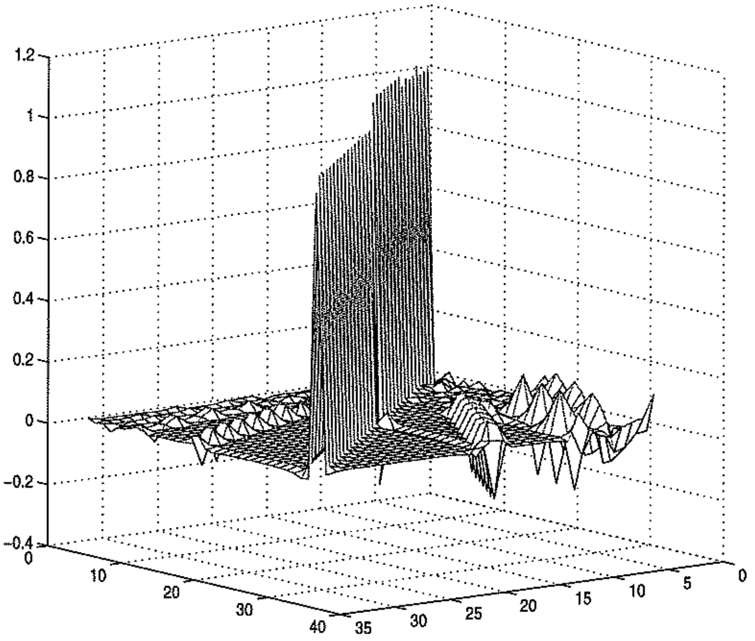


Figure 2: Finger-like sparse pattern of $\widetilde{X^{-1}A^*}$. Note that after a stage 1 preconditioning, $X^{-1}A^* \in \mathcal{WB}(16, 0.03)$ meaning that the preconditioned matrix is smoother.



to $\mathbf{WB}_\mu(A)$ for reasonably small μ. The proposed strategy of combining a preconditioning pre-processing (stage 1) and a wavelet band approximation (stage 2) gives a more robust preconditioner than WSPAI and SPAI. Our analysis will assume a diagonal non-smoothness but the method itself is more generally applicable.

The plan of the paper is as follows. In Section 2, we analyse the preconditioner WSPAI proposed in [5] using the new concept of wavelet band splittings and show that a simple way to improve the preconditioner is to allow the inclusion of off band matrix elements (the exact inclusion algorithm). But this will make the new preconditioner too expensive. In Section 3, a 2-stage preconditioner is proposed. We analyse the use of diagonal scaling as a stage 1 preconditioner and propose a minimization approach to select the best diagonal scaling preconditioner. We also consider other scaling preconditioners that are not diagonal matrices, In Section 4, we consider the complexity issue of this type of preconditioners. In Section 5, we use several numerical examples to compare the new preconditioner with WSPAI and the simple diagonal preconditioner. We present discussions of further work in Section 6.

2 A new WSPAI type algorithm

The wavelet sparse approximate inverse preconditioner (WSPAI) of [5] proposes to select the preconditioner $\widetilde{M}^{-1}$ in (2) by solving the following problem:

$$\begin{aligned}\min_M \|AM^{-1} - I\|_F &= \min_M \|WAW^\top WM^{-1}W^\top - I\|_F \\ &= \min_{\widetilde{M}} \|\widetilde{A}\widetilde{M}^{-1} - I\|_F. \qquad (7)\end{aligned}$$

The use of Frobenius norm decouples the above problem into n least squares problems for column of $\widetilde{M}^{-1}$:

$$\min_{\widetilde{m}_j} \|\widetilde{A}\widetilde{m}_j - e_j\|_2, \qquad j = 1, 2, \cdots, n. \qquad (8)$$

These problems are solved by specifying a block diagonal sparse pattern for the $\widetilde{m}_j$'s; see [5, 18].

To analyse the preconditioner, we consider a relationship between operator splitting with norm invariance and approximation by band structures. Thus this work is based on a slightly different philosophy from that of [5, 18] — we are trying to approximate better the transform matrix, not directly pursuing the smoothness of the inverse of the transform matrix.

We first illustrate how the present WSPAI method works in the new setting and demonstrate what happens with a strong singularity. Then we present an improved method and discuss its implementation. Recall the usual 2-norm for vectors $x \in \mathbf{R}^n = \mathbf{R}^{n\times 1}$ and the Forbenius norm for matrices $A = [A_1, \ldots, A_n] \in \mathbf{R}^{n\times n}$:

$$\|x\|_2 = \left(\sum_{j=1}^n x_j^2\right)^{1/2}, \qquad \|A\|_F = \left(\sum_{j=1}^n \|A_j\|_2^2\right)^{1/2}.$$

These norms are invariant under orthogonal transforms.

2.1 The vector case

For easy presentation, we first discuss an one-dimensional version of problem (7). Given $x \in \mathbf{R}^n$, we wish to find a sparse vector $y \in \mathbf{R}^n$ such that

$$\min_y \|x - y\|_2 = \min_y \|Wx - Wy\|_2 = \min_{\tilde{y}} \|\tilde{x} - \tilde{y}\|_2. \tag{9}$$

To mimic problem (7), we specify that $\tilde{y}$ will be consisted of a dense vector of size μ and a zero vector of size $(n - \mu)$. Then an approximate solution will be

$$\tilde{y}_j = \begin{cases} \tilde{x}_j, & j \leq \mu, \\ 0, & j > \mu. \end{cases}$$

We can see that this approximate solution will be fairly accurate if x represents a smooth function with weak or no singularities at x_1 because for large j, the wavelet coefficients will be small.

For example, consider $x = [8\ 7\ 6\ 5\ 4\ 3\ 2\ 1]^\top$ and use the Haar wavelet transform (with 3 levels and bandwidth $\mu = 3$). We have

$$\begin{aligned} \tilde{x} &= \begin{bmatrix} 12.7279 & -5.6569 & -2 & -2 & -0.7071 & -0.7071 & -0.7071 & -0.7071 \end{bmatrix}^\top, \\ \tilde{y} &= \begin{bmatrix} 12.7279 & -5.6569 & -2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \\ \tilde{c} = \tilde{x} - \tilde{y} &= \begin{bmatrix} 0 & 0 & 0 & -2 & -0.7071 & -0.7071 & -0.7071 & -0.7071 \end{bmatrix}^\top. \end{aligned}$$

In the wavelet basis, $\tilde{y}$ is an approximation to $\tilde{x}$ and the total energy is $\|x\|_F^2 = \|\tilde{x}\|_F^2 = \|\tilde{y}\|_F^2 + \|\tilde{c}\|_F^2 = 204$. The relative error is $\|\tilde{c}\|_F^2 / \|\tilde{x}\|_F^2 = 0.0294$, which may be acceptable. However, the approximation scheme is not so good if there is a strong singularity. For example, consider the same method for a new vector $x = [30\ 7\ 6\ 5\ 4\ 3\ 2\ 1]^\top$, that will be taken as our main example in the remainder of this section. We get

$$\begin{aligned} \tilde{x} &= \begin{bmatrix} 20.5061 & -13.4350 & -13 & -2 & -16.2635 & -0.7071 & -0.7071 & -0.7071 \end{bmatrix}^\top, \\ \tilde{y} &= \begin{bmatrix} 20.5061 & -13.4350 & -13 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \\ \tilde{c} &= \begin{bmatrix} 0 & 0 & 0 & -2 & -16.2635 & -0.7071 & -0.7071 & -0.7071 \end{bmatrix}^\top. \end{aligned}$$

Now the relative approximation error is not very small, $\|\tilde{c}\|_F^2 / \|\tilde{x}\|_F^2 = 270/1040 = 0.26$! We wish to reduce this error by modifying the scheme. To do an analysis first, we need the following definitions (that are applicable for both vectors and matrices), where we need the notation $W_b = W = W_{n \times n}$ if $m = n$ and $W_b = 1$ if $m = 1$.

Definition 1 (wavelet band splitting) *For a general matrix $B \in \mathbf{R}^{n \times m}$ $(n \geq m)$ and a given bandwidth μ, if there exists a splitting*

$$B = B_1 + B_2$$

such that under a wavelet transform B_1 becomes a banded matrix $\widetilde{B}_1$ of semi-bandwidth μ and B_2 becomes an off-banded matrix $\widetilde{B}_2$ complementing $\widetilde{B}_1$ in sparsity, i.e.

$$WB_1W_b^\top = \text{“\textit{banded matrix}”},$$

then the splitting $B = B_1 + B_2$ is called a wavelet band splitting.

Theorem 1 (uniqueness) *For orthogonal wavelet transforms, a wavelet band splitting as defined in Definition 1 is unique and F-norm invariant.*

Proof. For any given splitting $B = B_1 + B_2$, we always have $\widetilde{B} = \widetilde{B_1} + \widetilde{B_1}$ with $\widetilde{B} = WBW_b^\top$. As the banded matrix $\widetilde{B_1}$ and $\widetilde{B_2}$ complement each other, $\widetilde{B_1}$ is clearly unique. Further by orthogonality and $B_1 = W^\top \widetilde{B_1} W_b$, we have the norm invariance. ■

For clarity, following the uniqueness result, we write such a wavelet band splitting in a functional notation

$$B = \mathbf{WB}_\mu(B) + \mathbf{WO}_\mu(B), \tag{10}$$

where matrices on the right hand side may be calculated by

$$B_1 = \mathbf{WB}_\mu(B) = W^\top \left(\widetilde{B}\right)_{band} W_b = W^\top \left(WBW_b^\top\right)_{band} W_b,$$

$$B_2 = \mathbf{WO}_\mu(B) = W^\top \left(\widetilde{B}\right)_{off} = W^\top \left(WBW_b^\top\right)_{off} W_b.$$

Generally speaking, in the wavelet space, banded matrix approximations can only provide effective preconditioners if $\mathbf{WB}_\mu(B)$ in the wavelet band splitting is dominant. To characterise this dominance more precisely, we give the following definition.

Definition 2 (wavelet band space) *For a general matrix $B \in \mathbf{R}^{n\times m}$ $(n \geq m)$ and a given bandwidth μ, we say B belongs to a* wavelet band space *$\mathcal{WB}(\mu, \epsilon)$, $B \in \mathcal{WB}(\mu, \epsilon)$, if $\mathbf{WB}_\mu(B)$ dominates the wavelet band splitting of B in the sense $\|\mathbf{WO}_\mu(B)\|_F^2 \leq \epsilon\|\mathbf{WB}_\mu(B)\|_F^2$.*

Note that with the notation W_b, the above definitions apply to both a vector x $(m = 1)$ and a squares matrix A $(m = n)$. We now illustrate close relationships of wavelet band splittings, wavelet band space, smoothness and accuracy of banded approximations. Consider the above vector example with $m = 1$ and $\mu = 3$:

$$x = [30\ 7\ 6\ 5\ 4\ 3\ 2\ 1]^\top. \tag{11}$$

By inverse transforms, we find that

$$\mathbf{WB}_3(x) = W^\top \widetilde{y} = \begin{bmatrix} 18.5 & 18.5 & 5.5 & 5.5 & 2.5 & 2.5 & 2.5 & 2.5 \end{bmatrix}^\top,$$

$$\mathbf{WO}_3(x) = W^\top \widetilde{c} = \begin{bmatrix} 11.5 & -11.5 & 0.5 & -0.5 & 1.5 & 0.5 & -0.5 & -1.5 \end{bmatrix}^\top.$$

Clearly we can verify that $x = \mathbf{WB}_3(x) + \mathbf{WO}_3(x)$ and $\|\mathbf{WB}_3(x)\|_F^2 + \|\mathbf{WO}_3(x)\|_F^2 = \|x\|_F^2 = 2040$ due to norm invariance. Note that $x \in \mathcal{WB}(3, 0.59)$. We now use wavelet band splitting to explain which entries of x are more responsible for this approximation error.

Consider a non-band and linear splitting as follows $x = d + c$ (representing respectively smooth and nonsmooth parts of x) with

$$\left.\begin{aligned} d &= \begin{bmatrix} 22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \\ c &= \begin{bmatrix} 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}^\top. \end{aligned}\right\} \tag{12}$$

To see how much each part (vector) contributes to the approximation $\widetilde{y}$, one can carry out separate transforms and inverse transforms to find that $d = \mathbf{WB}_3(d) + \mathbf{WO}_3(d)$, $d = \mathbf{WB}_3(c) + \mathbf{WO}_3(c)$ represent two wavelet band splittings with

$$\begin{aligned} \mathbf{WB}_3(d) &= \begin{bmatrix} 11 & 11 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \\ \mathbf{WB}_3(c) &= \begin{bmatrix} 7.5 & 7.5 & 5.5 & 5.5 & 2.5 & 2.5 & 2.5 & 2.5 \end{bmatrix}^\top, \\ \mathbf{WO}_3(d) &= \begin{bmatrix} 11 & -11 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \\ \mathbf{WO}_3(c) &= \begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 & 1.5 & 0.5 & -0.5 & -1.5 \end{bmatrix}^\top. \end{aligned}$$

Notice the linear relations

$$\mathbf{WB}_3(d) + \mathbf{WB}_3(c) = \mathbf{WB}_3(x) \qquad \text{and} \qquad \mathbf{WO}_3(d) + \mathbf{WO}_3(c) = \mathbf{WO}_3(x)$$

hold but only $\mathbf{WB}_3(x)$ generates the approximation $\widetilde{y}$. The fact that the large vector $\mathbf{WO}_3(d)$ is not contributing to the approximation is a main source of error. Here $c \in \mathcal{WB}(3, 0.17)$ and $d \in \mathcal{WB}(3, 1) \not\subset \mathcal{WB}(3, 0.59) \not\subset \mathcal{WB}(3, 0.17)$.

Therefore, intuitively, we hope more information should be included in $\mathbf{WB}_3(x)$ in order for $\widetilde{y}$ to be a good approximation to $\widetilde{x}$; however by Theorem 1 wavelet band splitting is unique so we may have to go beyond band approximation. We suggest two related methods for improving such an approximation: exact inclusion and preconditioning — both will be generalised to the matrix case.

2.2 Exact inclusion

Here we propose an approximation based on wavelet band splitting plus exact representation of the transformed singular information. This requires a knowledge of the exact locations of transformed singular entries. An easier way to proceed with this is to do a symbolic transformation of the singular vector (such as d in (12)) and record the locations; this is especially necessary for matrix problems where an analytical trace is difficult. However for vectors, an analytical trace can be done.

Theorem 2 (trace of singularity) *Let $x_1 \neq 0$, $x = [x_1, 0, \ldots, 0]^\top \in \mathbf{R}^n$ and the standard Daubechies' order n_{dwt} wavelets be used with L levels. Then the nonzero positions in the transformed vector $z = Wx$ are located in this set of indexes:*

1. *for $n_{dwt} = 2$, $K_2 = \bigcup_{k=0}^{L} p_k$ with $p_0 = \{1\}$ and $p_k = \{n/2^k + 1\}$.*

2. *for $n_{dwt} = 4$, $K_4 = \bigcup_{k=0}^{L} p_k$ with $p_0 = \{1,\ n\}$ and $p_k = \{n/2^k - 1,\ n/2^k,\ n/2^k + 1\}$.*

Proof. By induction. We only need to illustrate the second case. Note that the overall transform [16] is $W = W_L W_{L-1} \cdots W_1$ with

$$W_\nu = \begin{bmatrix} X & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & X & & \\ X_b & & & & X_a & \\ Y & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & Y & & \\ Y_b & & & & Y_a & \\ & & & & & \mathbf{I}_\nu \end{bmatrix}_{n\times n},$$

where $X = [X_a\ X_b] = [c_0\ c_1\ c_2\ c_3]$ and $X = [X_a\ X_b] = [d_0\ d_1\ d_2\ d_3]$ are 1×4 blocks, $\mathbf{I}_\nu$ is an identity matrix. Then at step 1 with W_1, we get the index set $\{1,\ n/2,\ n/2+1,\ n\} = p_0 \bigcup \{n/2,\ n/2+1\}$ from

$$z^{(1)} = W_1 x = [z_1^{(1)}\ 0\ \ldots\ 0\ z_{n/2}^{(1)}\ z_{n/2+1}^{(1)}\ 0\ \ldots\ 0\ z_n^{(1)}]^\top$$

and at step 2 the set $\{1,\ n/2^2-1,\ n/2^2,\ n/2^2+1,\ n/2-1,\ n/2,\ n/2+1,\ n\} = p_0 \bigcup p_1 \bigcup p_2$ from

$$z^{(2)} = W_2 W_1 x = [z_1^{(2)}\ 0\ \ldots\ 0\ z_{n/2^2-1}^{(2)}\ z_{n/2^2}^{(2)}\ z_{n/2^2+1}^{(2)}\ 0\ \ldots\ 0\ z_{n/2-1}^{(2)}\ z_{n/2}^{(2)}\ z_{n/2+1}^{(2)}\ 0\ \ldots\ 0\ z_n^{(2)}]^\top.$$

Proceeding this way will complete the proof. ■

Consider the above example again with

$$x = [30\ 7\ 6\ 5\ 4\ 3\ 2\ 1]^\top.$$

With $n_{dwt} = 2$ and $L = 3$, Theorem 2 gives the index set $K = \{1,\ n/2+1,\ n/2^2+1,\ n/2^3+1\} = \{1\ 2\ 3\ 5\}$. As $\{1\ 2\ 3\}$ are already within bandwidth 3, we have

$$\begin{aligned} \widetilde{x} &= \begin{bmatrix} 20.5061 & -13.4350 & -13 & -2 & -16.2635 & -0.7071 & -0.7071 & -0.7071 \end{bmatrix}^\top, \\ \widetilde{y} &= \begin{bmatrix} 20.5061 & -13.4350 & -13 & 0 & -16.2635 & 0 & 0 & 0 \end{bmatrix}^\top, \\ \widetilde{c} &= \begin{bmatrix} 0 & 0 & 0 & -2 & 0 & -0.7071 & -0.7071 & -0.7071 \end{bmatrix}^\top. \end{aligned}$$

The new approximation error is much smaller: $\|\widetilde{c}\|_F^2/\|\widetilde{x}\|_F^2 = 5.5/1040 = 0.0053$.

2.3 Preconditioning the problem

Assume that there is a strong singularity and the idea is, by preconditioning, to split the given vector into a unit vector plus a smooth vector before applying the wavelet transform. This preconditioning will ensure that the smooth vector is well compressed by a wavelet transform while the transformed unit vector, that is really a column of the orthogonal wavelet matrix, is included exactly. Let α be a suitable scalar and e_1 be the usual unit vector. Then

$$\alpha^{-1}x = e_1 + x_\alpha = e_1 + \mathbf{WB}_\mu(x_\alpha) + \mathbf{WO}_\mu(x_\alpha), \quad W\alpha^{-1}x = We_1 + \widetilde{x}_\alpha. \tag{13}$$

For vectors, including We_1 in a banded approximation in the wavelet space is equivalent to the exact inclusion approach of the previous subsection. However for matrices, a transformed identity matrix is still an identity and thus the exact inclusion is fulfilled implicitly without extra work. This preconditioning step corresponds to the stage 1 preconditioning discussed later.

For our main vector example, we may consider these two splittings $x = \alpha_1 e_1 + x_{\alpha_1}$ and $x = \alpha_2 e_1 + x_{\alpha_2}$ with $\alpha_1 = 22$, $\alpha_2 = 24$, and

$$\begin{aligned} x_{\alpha_1} &= \begin{bmatrix} 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}^\top, \\ x_{\alpha_2} &= \begin{bmatrix} 6 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}^\top, \end{aligned}$$

$$\begin{aligned} \alpha_1^{-1}x_{\alpha_1} &= \begin{bmatrix} 0.3636 & 0.3182 & 0.2727 & 0.2273 & 0.1818 & 0.1364 & 0.0909 & 0.0455 \end{bmatrix}^\top, \\ \alpha_2^{-1}x_{\alpha_2} &= \begin{bmatrix} 0.2500 & 0.2917 & 0.2500 & 0.2083 & 0.1667 & 0.1250 & 0.0833 & 0.0417 \end{bmatrix}^\top. \end{aligned}$$

Clearly the scaled vectors, apart from adding a unit vector, are smooth and can be compressed well. Further as expected, the resulting approximation errors for the two preconditioned cases are both small, respectively, 0.005 and 0.03; in particular $\alpha_1^{-1}x_{\alpha_1}, \alpha_2^{-1}x_{\alpha_2} \in \mathcal{WB}(3, 0.073)$. This idea is explored below for matrix problems.

2.4 The matrix case

Based on previous observations of the vector case, we now apply the same ideas to matrices, leading to our new preconditioning algorithms. Firstly we illustrate existing problems with the present WSPAI algorithm [5].

For the discrete operator $A \in \mathbf{R}^{n\times n}$, consider the splitting

$$A = D + C, \tag{14}$$

where D is a diagonal matrix (not necessarily the diagonal of A) selected so that C is smooth and we assume the singularity of A is along D (refer to (12) and the example below). If such a singularity is strong and D is not a constant diagonal matrix, then $D \notin \mathcal{WB}(\mu, \epsilon)$ and $C \in \mathcal{WB}(\mu, \epsilon)$ (for some interested μ and ϵ) and we claim that WSPAI will produce a poor preconditioner.

To explain in details, for a given bandwidth μ, we shall use Definition 1 and consider the wavelet band splitting

$$A = \mathbf{WB}_\mu(A) + \mathbf{WO}_\mu(A).$$

The wavelet transformed matrix is

$$\widetilde{A} = WDW^\top + WCW^\top = W\mathbf{WB}_\mu(A)W^\top + W\mathbf{WO}_\mu(A)W^\top,$$

and the WSPAI selects the block diagonal part of $\widetilde{A}$, or equivalently to:

$$M = W\mathbf{WB}_\mu(A)W^\top = (WAW^\top)_{band},$$

for the WSPAI preconditioner M^{-1}. To see a relationship between M and D and C, consider

$$D = \mathbf{WB}_\mu(D) + \mathbf{WO}_\mu(D) \quad \text{and} \quad C = \mathbf{WB}_\mu(C) + \mathbf{WO}_\mu(C).$$

Then one can verify that

$$\mathbf{WB}_\mu(A) = \mathbf{WB}_\mu(D) + \mathbf{WB}_\mu(C), \quad \mathbf{WO}_\mu(A) = \mathbf{WO}_\mu(D) + \mathbf{WO}_\mu(C).$$

That is,

$$A = \underbrace{\underbrace{\mathbf{WB}_\mu(D)}_{\text{large}} + \underbrace{\mathbf{WB}_\mu(C)}_{\text{large}}}_{\text{preconditioner } M} + \underbrace{\underbrace{\mathbf{WO}_\mu(D)}_{\text{large}} + \underbrace{\mathbf{WO}_\mu(C)}_{\text{small}}}_{\text{off-band dropped}}, \qquad (15)$$

where the first part is used to construct a preconditioner and $\mathbf{WO}_\mu(D)$ is usually large but is zero when D is a constant diagonal matrix.

Such a WSPAI preconditioner M^{-1}, generated from $\mathbf{WB}_\mu(A)$, contains all the dominating information $\mathbf{WB}_\mu(C)$ from the smooth part of A and one non-smooth part $\mathbf{WB}_\mu(D)$ of D but not the other non-smooth part $\mathbf{WO}_\mu(D)$. Because

$$M^{-1}\widetilde{A} = W\mathbf{WB}_\mu(A)^{-1}AW^\top = W[\mathbf{WB}_\mu(D) + \mathbf{WB}_\mu(C)]^{-1}AW^\top$$

and W is orthogonal, the preconditioned matrix $M^{-1}\widetilde{A}$ has the same spectra as $[\mathbf{WB}_\mu(D) + \mathbf{WB}_\mu(C)]^{-1}A$. This implies that this WSPAI preconditioner of [5]

- is very effective whenever $\mathbf{WO}_\mu(D)$ is small; e.g. for Laplace' equation where A is relatively smooth along the diagonal.

- may not be effective whenever $\mathbf{WO}_\mu(D)$ is not small or dominates $\mathbf{WB}_\mu(D)$. In such cases, for the same reason, WSPAI may not be better than the simple diagonal preconditioner – this depends on a balance of dominance between $\mathbf{WB}_\mu(C)$ and $\mathbf{WO}_\mu(D)$.

In the next subsection we show one example of wavelet band splittings for a 2 × 2 matrix.

To verify our observations on WSPAI, we solve the following two linear systems: $Ax = b$ with

Example I

$$A = \begin{bmatrix} 2.01 & -1 & & & & -1 \\ -1 & 2.01 & -1 & & & \\ & -1 & 2.01 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2.01 & -1 \\ -1 & & & & -1 & 2.01 \end{bmatrix}. \tag{16}$$

Example II

$$A = \begin{bmatrix} (n+4)^2 & 81 & & & & 81 \\ 81 & (n+3)^2 & 81 & & & \\ & 81 & (n+2)^2 & 81 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 81 & 4^2 & 81 \\ 81 & & & & 81 & 5^2 \end{bmatrix}. \tag{17}$$

We use the iterative solver GMRES(10) [15] and compare the WSPAI preconditioner with the simple diagonal preconditioner. Here we take $n = 256$, $n_{dwt} = L = 4$, $\mu = 5$ (bandwidth). The convergence results are shown in Figures 3-4 respectively for the two examples, where WSPAI is shown as 'x' and the diagonal preconditioner as 'o'. We can see that for the smooth Example I, a diagonal preconditioner alone is not as good as WSPAI but for the nonsmooth (near the diagonal) Example II the reverse is true — confirming our claim. We now seek improvements to WSPAI.

Remark. For sparse matrices, wavelet band splittings can be dense. Thus wavelet band splitting provides a tool to study preconditioners in the original matrix space instead of the wavelet space. This brings out an interesting fact about wavelet transforms. They could help to design a dense preconditioner to approximate the original sparse matrix more closely but can be implemented in a cheap way. This fact seems trivial for dense matrices but is an interesting way to justify the use of sparse preconditioners (associated with wavelet band splittings) [8]. However the wavelet band space determines the effectiveness of such preconditioners — a stage 1 preconditioner enables a matrix to change its wavelet band space (Section 3).

2.5 Algorithm 1 – exact inclusion

The previous two examples have demonstrated that the strength of diagonal singularity is directly related to wavelet compression of matrices into sparse band forms. Following the idea of exact inclusion in §2.2, we propose a method that combines wavelet compression and sparse approximation.

Consider the matrix splittings in (14) and (15). The new idea is simply to include

Figure 3: Convergence history for Example I: GMRES(10) with the diagonal preconditioner ('o'), the present WSPAI ('x') and the new Algorithm 1 ('*'). This example shows that when WSAI works it out performs the simple diagonal preconditioner but Algorithm 1 is the best.

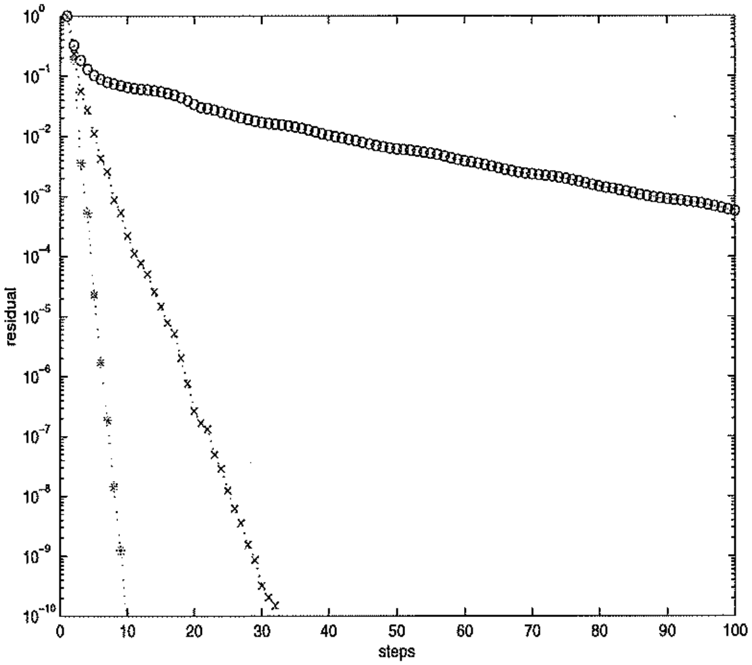
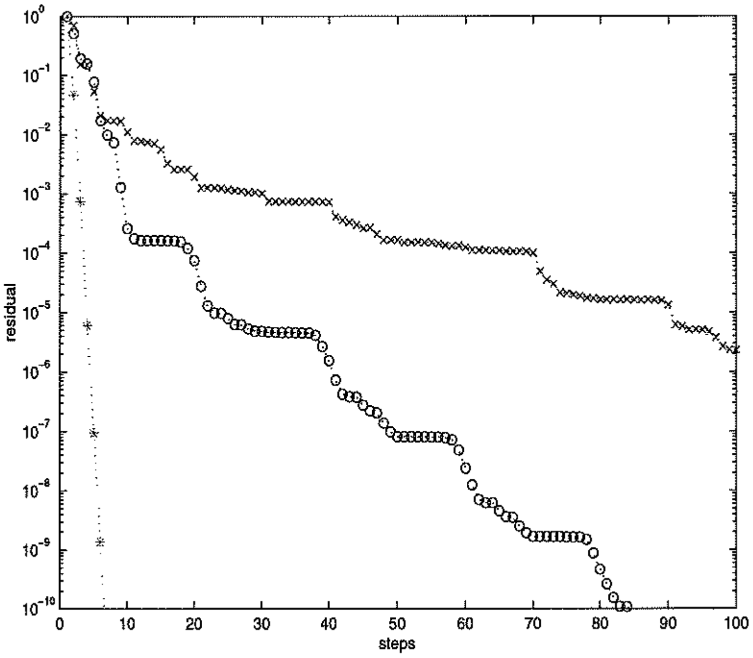


Figure 4: Convergence history for Example II: GMRES(10) with the diagonal preconditioner ('o'), the present WSPAI ('x') and the new Algorithm 1 ('*'). This example shows that when WSAI does not work it is worse than the simple diagonal preconditioner but Algorithm 1 is again the best.



$\mathbf{WO}_\mu(D)$ in the preconditioner:

$$A = \underbrace{\mathbf{WB}_\mu(D)}_{\text{large}} + \underbrace{\mathbf{WB}_\mu(C)}_{\text{large}} + \underbrace{\mathbf{WO}_\mu(D)}_{\text{large}} + \underbrace{\mathbf{WO}_\mu(C)}_{\text{small}} = \underbrace{D + \mathbf{WB}_\mu(C)}_{M} + \underbrace{\mathbf{WO}_\mu(C)}_{\text{dropped}}. \quad (18)$$

That is,

$$\begin{aligned} \widetilde{A} &= WDW^\top + WCW^\top \\ &= \left[WDW^\top + W\mathbf{WB}_\mu(C)W^\top\right] + W\mathbf{WO}_\mu(C)W^\top, \end{aligned} \quad (19)$$

and we select the new preconditioner as

$$\widetilde{M}^{-1} = WDW^\top + W\mathbf{WB}_\mu(C)W^\top$$

for $\widetilde{A}$. This can be illustrated in Figure 5, where $A1 = \widetilde{A}$ is the wavelet transform matrix, $D1 = WDW^\top$ and $M1 = D1 + W\mathbf{WB}_\mu(C)W^\top = \widetilde{M}$. To relate the new wavelet SPAI preconditioner to the preconditioning of the original problem, consider

$$\begin{aligned} \widetilde{M}^{-1}\widetilde{A} &= \left[WDW^\top + W\mathbf{WB}_\mu(C)W^\top\right]^{-1}\widetilde{A} \\ &= W\left[D + \mathbf{WB}_\mu(C)\right]^{-1} AW^\top. \end{aligned}$$

As C is a smooth matrix, we expect that in the wavelet band splitting $\mathbf{WB}_\mu(C)$ dominates. So $[D + \mathbf{WB}_\mu(C)]$ can approximate A well and thus $\widetilde{M}^{-1}$ is a good preconditioner.

Although $[D + \mathbf{WB}_\mu(C)]$ and $\widetilde{M}$ are equivalent in the spectral sense and both can be computed efficiently, however, the former matrix in the original matrix space is not easy to invert so preconditioning using $\widetilde{M}$ will be done in the wavelet space. The procedure can be summarised as follows:

Algorithm 1

1. *Perform a DWT to $Ax = b$ to get: $\widetilde{A}x_1 = b_1$;*

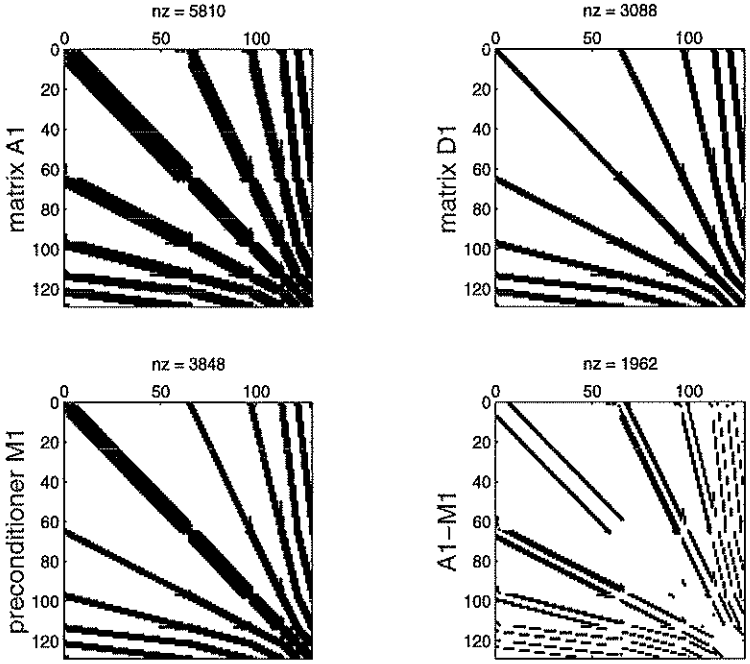
2. *Perform a symbolic DWT to a diagonal matrix D to get a boolean matrix $D1$;*

3. *Select the sparse preconditioner $\widetilde{M}^{-1}$ from $\widetilde{A}$ such that it has the sparse pattern as $D1$ plus a banded matrix of width μ;*

4. *Solve the preconditioned system: $\widetilde{M}^{-1}\widetilde{A}_1x_1 = \widetilde{M}^{-1}\widetilde{b}_1$.*

The effectiveness of this algorithm is illustrated by solving Examples I and II again; in Figures 3-4 we plot the results of Algorithm 1 using the symbol '*'. We can see that Algorithm 1 is robust and effective. The only problem with this algorithm is that the preconditioner can be expensive to invert because it has the same unpleasant sparse pattern as matrix $\widetilde{A}$. A solution to alleviate this problem is to find a suitable thresholding ϵ to drop small elements of $\widetilde{D} = WDW^\top$ in the symbolic transform stage; this idea is pursued in [10]. Instead we consider the alternative method of preconditioning as in §2.3.

Figure 5: New exact inclusion preconditioner $\widetilde{M} = M1$. Observe that the preconditioner $M1$ follows the sparsity pattern of the transformed matrix $A1$.



3 Two stage sparse preconditioners

Following §2.3, for a given matrix $A \notin \mathcal{WB}(\mu, \epsilon)$, we wish to construct a preconditioner M_1^{-1}

$$M_1^{-1} A = I + S, \tag{20}$$

such that $S \in \mathcal{WB}(\mu, \epsilon)$ (and we know that $I \in \mathcal{WB}(\mu, 0)$). Matrix M_1^{-1} is called a stage 1 preconditioner. Then it is not difficult to see that a wavelet transform can compress the preconditioned matrix $M_1^{-1}A$ well and lead to an efficient preconditioner M_2^{-1} with

$$M_2 = I + W\mathbf{WB}_\mu(S)W^\top.$$

3.1 Algorithm 2

One way to choose M_1 is to use D in the simpler splitting (14) and do the *stage* 1 *preconditioning* as follows

$$A_1 = D^{-1}A = I + D^{-1}C \tag{21}$$

where we may identify D with M_1^{-1} and $D^{-1}C$ with S in (20). The wavelet transformed matrix is

$$\widetilde{A}_1 = I + WD^{-1}CW^\top = I + W\mathbf{WB}_\mu(S)W^\top + W\mathbf{WO}_\mu(S)W^\top, \tag{22}$$

giving rise to the *stage* 2 *preconditioning*

$$\widetilde{M_2}^{-1} = \left[I + W\mathbf{WB}_\mu(S)W^\top\right]^{-1}.$$

Thus our new algorithm can be stated as follows

Algorithm 2 (2-stages)

Stage 1:

1. *Find on a suitable operator splitting $A = D + C$ such that C is smooth and D can be easily inverted;*
2. *Select the first preconditioner as $M_1^{-1} = D^{-1}$;*
3. *Precondition the original problem: $M_1^{-1}Ax = M_1^{-1}b$;*

Stage 2:

4. *Perform a DWT to the scaled system and get $\tilde{A}_1 x_1 = \tilde{b}_1$;*
5. *Select the WSPAI preconditioner $\widetilde{M_2}^{-1}$ from a band part of matrix $\tilde{A}_1$;*
6. *Solve the preconditioned system: $\widetilde{M_2}^{-1}\tilde{A}_1 x_1 = \widetilde{M_2}^{-1}\tilde{b}_1$.*

Here we assumed A is unsymmetric because we are primarily interested in unsymmetric linear systems; it is possible to develop a symmetric version where both stages of preconditioning must be done symmetrically.

It is of interest to combine both preconditioning steps and show how preconditioning is precisely done.

Theorem 3 (equivalence of 2-stage preconditioning) *Cast both preconditioning stages (21) and (22) into one step, the proposed 2-stage preconditioning is equivalent to preconditioning the original matrix $A = D + D\mathbf{WB}_\mu(D^{-1}C) + D\mathbf{WO}_\mu(D^{-1}C)$ by*

$$\left[D + D\mathbf{WB}_\mu(D^{-1}C)\right]^{-1}.$$

Proof. Since $S = D^{-1}C$, a simple substitution shows that

$$\begin{aligned} \widetilde{M_2}^{-1}\tilde{A}_1 &= \left[I + W\mathbf{WB}_\mu(S)W^\top\right]^{-1} WD^{-1}AW^\top \\ &= W\left[D + D\mathbf{WB}_\mu(S)\right]^{-1} AW^\top. \end{aligned} \qquad (23)$$

Note that $A = D + D\mathbf{WB}_\mu(S) + D\mathbf{WO}_\mu(S) = D + D\left[\mathbf{WB}_\mu(S) + \mathbf{WO}_\mu(S)\right] = D + D * S = D + D * D^{-1}C = D + C$. It is evident that the proposed 2-stage preconditioning is equivalent to using $M^{-1} = [D + D\mathbf{WB}_\mu(S)]^{-1}$ to precondition the original matrix A. ■

Therefore the two stage preconditioner will be effective provided that $\mathbf{WO}_\mu(S)$ as well as $\mathbf{WO}_\mu(C)$ are small i.e. $S \in \mathcal{WB}(\mu, \epsilon)$ if $C \in \mathcal{WB}(\mu, \epsilon)$, as expected. This will be established next after showing a simple example.

3.2 Summary and example

To demonstrate the differences between Algorithms 1, 2 and WSPAI of [5] in terms of spectral analysis and wavelet band splittings, we may use the following table (note $D = \mathbf{WB}_\mu(D) + \mathbf{WO}_\mu(D)$):

Method	Equivalent splitting of A (induced by wavelet band splittings)	
	preconditioner M	$A - M$
WPSAI	$\mathbf{WB}_\mu(D) + \mathbf{WB}_\mu(C)$	$\mathbf{WO}_\mu(D) + \mathbf{WO}_\mu(C)$
Algorithm 1	$D + \mathbf{WB}_\mu(C)$	$\mathbf{WO}_\mu(C)$
Algorithm 2	$D + D\mathbf{WB}_\mu(D^{-1}C)$	$D\mathbf{WO}_\mu(D^{-1}C)$

Now consider a simple 2×2 matrix A to illustrate;

$$A = \begin{bmatrix} 17 & 2 \\ 1 & 3 \end{bmatrix} = D + C = \begin{bmatrix} 16 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}.$$

Using $N_{dwt} = 2$, $L = 1$, $\mu = 0$, we find

$$\mathbf{WB}_0(D) = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}, \mathbf{WB}_0(C) = \begin{bmatrix} 1 & 1.5 \\ 1.5 & 1 \end{bmatrix},$$

$$\mathbf{WO}_0(D) = \begin{bmatrix} 7 & 0 \\ 0 & -7 \end{bmatrix}, \mathbf{WO}_0(C) = \begin{bmatrix} 0 & 0.5 \\ -0.5 & 0 \end{bmatrix}.$$

Because $C \in \mathcal{WB}(0, 0.28)$, $D \in \mathcal{WB}(0, 0.78)$, we may say C is smoother than D.

The WSPAI algorithm [5] gives the following splitting

$$A = \begin{bmatrix} 10 & 1.5 \\ 1.5 & 10 \end{bmatrix} + \begin{bmatrix} 7 & 0.5 \\ -0.5 & -7 \end{bmatrix},$$

$$\widetilde{M}^{-1}\widetilde{A} = \begin{bmatrix} 11.5 & 0 \\ 0 & 8.5 \end{bmatrix}^{-1} \begin{bmatrix} 11.5 & -6.5 \\ -7.5 & 8.5 \end{bmatrix} = \begin{bmatrix} 1 & -0.5652 \\ -0.8824 & 1 \end{bmatrix}$$

with eigenvalues $\lambda(\widetilde{M}^{-1}\widetilde{A}) = [0.2938 \ 1.7062]$, singular values $\sigma(\widetilde{M}^{-1}\widetilde{A}) = [0.2887 \ 1.7363]$ and $\|\widetilde{A} - \widetilde{M}\|_F^2 / \|\widetilde{M}\|_F^2 = 0.69$ because $A \in \mathcal{WB}(0, 0.69)$. In comparison, Algorithm 1 gives

$$A = \begin{bmatrix} 17.0 & 1.5 \\ 1.5 & 3.0 \end{bmatrix} + \begin{bmatrix} 0 & 0.5 \\ -0.5 & 0 \end{bmatrix},$$

$$\widetilde{M}^{-1}\widetilde{A} = \begin{bmatrix} 11.5 & -7.0 \\ -7.0 & 8.5 \end{bmatrix}^{-1} \begin{bmatrix} 11.5 & -6.5 \\ -7.5 & 8.5 \end{bmatrix} = \begin{bmatrix} 0.9282 & 0.0872 \\ -0.1179 & 1.0718 \end{bmatrix}$$

with better distributed eigenvalues $\lambda(\widetilde{M}^{-1}\widetilde{A}) = [1 - 0.0716i \ \ 1 + 0.0716i]$, singular values $\sigma(\widetilde{M}^{-1}\widetilde{A}) = [0.9318 \ 1.0787]$ and $\|\widetilde{A} - \widetilde{M}\|_F^2 / \|\widetilde{M}\|_F^2 = 0.04$. Finally Algorithm 2 gives

$$A = \begin{bmatrix} 20.5 & 5 \\ 0.6250 & 2.5625 \end{bmatrix} + \begin{bmatrix} -3.5 & -3 \\ 0.375 & 0.4375 \end{bmatrix},$$

$$\begin{aligned} \widetilde{M_2}^{-1}\widetilde{A_1} &= \begin{bmatrix} 1.5938 & 0 \\ 0 & 0.9688 \end{bmatrix}^{-1} \begin{bmatrix} 1.5938 & 0.0313 \\ 0.4063 & 0.9688 \end{bmatrix} \\ = \ \widetilde{M}^{-1}\widetilde{A} &= \begin{bmatrix} 14.3438 & -6.7812 \\ -11.1563 & 8.7188 \end{bmatrix}^{-1} \begin{bmatrix} 11.5 & -6.5 \\ -7.5 & 8.5 \end{bmatrix} = \begin{bmatrix} 1 & 0.0196 \\ 0.4194 & 1 \end{bmatrix} \end{aligned}$$

with these eigenvalues $\lambda(\widetilde{M}^{-1}\widetilde{A}) = [0.9093\ 1.0907]$, singular values $\sigma(\widetilde{M}^{-1}\widetilde{A}) = [0.8003\ 1.2393]$ and $\|\widetilde{A} - \widetilde{M}\|_F^2 / \|\widetilde{M}\|_F^2 = 0.22$ while $D^{-1}C \in \mathcal{WB}(0, 0.22)$. Both the eigenvalues and singular values suggest that Algorithms 1 and 2 will yield more effective preconditioners than WSPAI [5]. The singular values in particular imply that the conjugate gradients normal equation method (CGN) may be used for the preconditioned systems [14].

The assumption of a smooth C means that in a wavelet band splitting of C, $\mathbf{WO}_\mu(C)$ is small in F-norm i.e. $C \in \mathcal{WB}(\mu, \epsilon)$. In the spectral analysis of (23), we assume that $\mathbf{WO}_\mu(S) = \mathbf{WO}_\mu(D^{-1}C)$ is small if $\mathbf{WO}_\mu(C)$ is small i.e. $D^{-1}C \in \mathcal{WB}(\mu, \epsilon)$ if $C \in \mathcal{WB}(\mu, \epsilon)$. Below we shall make this statement more precise by considering several simple cases.

3.3 The diagonal scaling

A diagonal stage 1 preconditioner is the easiest matrix to invert. We study its effectiveness. As before assume that $A = D + C$ and $C \in \mathcal{WB}(\mu, \epsilon)$ is a smooth matrix such that $\mathbf{WO}_\mu(C)$ is small. We hope to establish that $\mathbf{WO}_\mu(S) = \mathbf{WO}_\mu(D^{-1}C)$ is also small. Inevitably we need to compare the elements of $\widetilde{C} = WCW^\top$ with $\widetilde{S} = WSW^\top = WCD^{-1}W^\top$. This requires the following elementary lemma.

Lemma 1 *Given an integer n_{dwt}, real numbers $w_1, w_2, \ldots, w_{n_{dwt}}$ and positive real numbers $D_1, D_2, \ldots, D_{n_{dwt}}$, we have*

$$T = \sum_{j=1}^{n_{dwt}} \frac{w_j}{D_j} = \frac{1}{D_*} \sum_{j=1}^{n_{dwt}} w_j,$$

where D_* lies in the interval $[\min D_j, \max D_j]$.

Proof. The case where all w_j are of one sign is trivial. We assume the first n_1 numbers of w_j are positive and the rest negative. Let $a = \sum_{j=1}^{n_1} w_j > 0$ and $b = \sum_{j=n_1+1}^{n_{dwt}} w_j < 0$. Then because each partial sum contains terms of a single sign, there exist two harmonic means D_a, $D_b \in [\min D_j, \max D_j]$ such that $a/D_a = \sum_{j=1}^{n_1} \frac{w_j}{D_j}$ and $b/D_b = \sum_{j=n_1+1}^{n_{dwt}} \frac{w_j}{D_j}$. Now consider $T = a/D_a + b/D_b$ and two separate cases: $D_a \geq D_b$ or $D_a < D_b$. It follows that there exists $D_* \in [\min D_j, \max D_j]$ such that $T = (a+b)/D_*$. ■

Remark. The above quantity D_* computed by

$$D_* = \sum_{j=1}^{n_{dwt}} w_j \Bigg/ \sum_{j=1}^{n_{dwt}} \frac{w_j}{D_j}$$

is a weighted harmonic mean of D_j's and it is the exact harmonic mean of D_j's if $w_j = 1$ for all j.

Our main result can be stated as follows.

Theorem 4 (diagonal scaling) *Let $D, C \in R^{n\times n}$ with $D = diag(D_j)$ a diagonal matrix and n_{dwt} be the order of Daubechies' orthogonal wavelets. Assume we have the wavelet band splittings: $C = \mathbf{WB}_\mu(C) + \mathbf{WO}_\mu(C)$ and $D^{-1}C = \mathbf{WB}_\mu(D^{-1}C) + \mathbf{WO}_\mu(D^{-1}C)$. Then the following results hold*

1. *$WD^{-1}CW^\top = \hat{H} \cdot WCW^\top$, where '$\cdot$' means a pointwise product and element $\hat{H}_{ij} = 1/H_{ij}$ with H_{ij} a weighted harmonic mean of D_j lying inside $[\min D_j, \max D_j]$;*
2. $\dfrac{\min D_j}{\max D_j}\|\mathbf{WO}_\mu(C)\|_F \;\le\; \|D\mathbf{WO}_\mu(D^{-1}C)\|_F \;\le\; \dfrac{\max D_j}{\min D_j}\|\mathbf{WO}_\mu(C)\|_F.$

Proof. (1). Let $S = D^{-1}C, \widetilde{S} = WSW^\top, U = \widetilde{C} = WCW^\top$. To compare $\widetilde{S}$ with $\widetilde{C}$, define $B = CW^\top$. For simplicity, consider $n_{dwt} = 4$ and $L = 1$ first (using the notation of Theorem 2). Then a direct calculation shows that

$$\widetilde{S}_{ij} = \begin{cases} \displaystyle\sum_{k=1}^{4} c_{k-1}B_{i_k,j}/D_{i_k} & \text{if } i \le n/2 \\ \displaystyle\sum_{k=1}^{4} d_{k-1}B_{i_k,j}/D_{i_k} & \text{if } i > n/2 \end{cases}$$

where the index $i_k = (2i-1) + (k-1)$ for $i \le n/2$ and $i_k = 2[(i-n/2)-1] + (k-1)$ for $i > n/2$. Using Lemma 1, there exists a weighted harmonic mean H_{ij} such that

$$\widetilde{S}_{ij} = H_{ij}^{-1}U_{ij}.$$

For $L > 1$, the wavelet transform operates with dimension $n/2^\ell$ with $\ell \ge 1$ and Lemma 1 can be applied repeatedly. This completes the proof of this part.
(2). Define $Y = DV$ with $V = \mathbf{WO}_\mu(S) = W^\top(WSW^\top)_{off}W$. Because

$$\|Y\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} Y_{ij}^2 = \sum_{i=1}^{n}\left(D_i^2 \sum_{j=1}^{n} V_{ij}^2\right)$$

and

$$\sum_{i=1}^{n}\sum_{j=1}^{n} V_{ij}^2 = \|W^\top(WSW^\top)_{off}W\|_F^2 = \|(WSW^\top)_{off}\|_F^2,$$

therefore

$$\min D_j^2\|(WSW^\top)_{off}\|_F^2 \ge \|D\mathbf{WO}_\mu(S)\|_F^2 \le \max D_j^2\|(WSW^\top)_{off}\|_F^2.$$

We now try to bound the last term on the right hand side. We find:

$$\begin{aligned}\|(WSW^\top)_{off}\|_F^2 = \sum_{|i-j|>\mu} \widetilde{S}_{ij}^2 &= \sum_{|i-j|>\mu} \frac{U_{ij}^2}{H_{ij}^2} \\ &\le \max\frac{1}{H_j^2}\sum_{|i-j|>\mu} U_{ij}^2 \\ &= \frac{1}{\min D_j^2}\|(WCW^\top)_{off}\|_F^2 \\ &= \frac{1}{\min D_j^2}\|\mathbf{WO}_\mu(C)\|_F^2.\end{aligned}$$

Similarly,

$$\|(WSW^\top)_{off}\|_F^2 \geq \frac{1}{\max D_j^2}\|\mathbf{WO}_\mu(C)\|_F^2.$$

Combining with above norm bounds gives

$$\frac{\min D_j^2}{\max D_j^2}\|\mathbf{WO}_\mu(C)\|_F^2 \leq \|D\mathbf{WO}_\mu(D^{-1}C)\|_F^2 \leq \frac{\max D_j^2}{\min D_j^2}\|\mathbf{WO}_\mu(C)\|_F^2.$$

Taking the squares root yields the required result. ■

Remark. When one of the diagonal entry D_k is negative, the results of Lemma 1 and Theorem 3 are not true. However, this case can be fixed because we may multiply -1 to rows that have a negative diagonal first before applying a scaling preconditioner.

Corollary 1

1. *If D is a constant diagonal matrix, then*

$$\|D\mathbf{WO}_\mu(D^{-1}C)\|_F^2 = \|\mathbf{WO}_\mu(C)\|_F^2$$

and Algorithm 2 *is identical to Algorithm* 1.

2. *If C is a matrix of identical rows, following on Lemma* 1 *and Theorem* 3*, a simpler result holds*

$$WD^{-1}CW^\top = \bar{H}\cdot WCW^\top$$

where $\bar{H}_{ii} = 1/H_{ii}$ is a diagonal matrix with H_{ii} a weighted harmonic mean of D_j's and inside $[\min D_j, \max D_j]$. Then we have $\mathbf{WB}_\mu(D^{-1}C) = W^\top(\bar{H}\widetilde{C})_{band}W$ and $\mathbf{WO}_\mu(D^{-1}C) = W^\top(\bar{H}\widetilde{C})_{off}W$. That is, approximately, $D^{-1}C \in \mathcal{WB}(\mu,\epsilon)$ if $C \in \mathcal{WB}(\mu,\epsilon)$.

3. *More generally, Algorithm* 2 *is efficient whenever C is a smooth matrix and the condition number of D is not large.*

3.4 Selection of the best diagonal scaling preconditioner

We now discuss how to partition the matrix $A = D + C$ so that the error term or the upper error bound in Theorem 4 is minimised, hence elaborating on Corollary 1(3). Clearly the minimization of the single term $\|\mathbf{WO}_\mu(C)\|_F^2$, not including the other term $\max D_j/\min D_j$, is easy to do. We shall investigate the general problem of minimising the whole terms and propose a solution method for simple cases.

To motivate the ideas, consider the following problem for $x = [x_1, \cdots, x_n]^\top \in \mathbf{R}^n$ (similar to (11) and (13)):

$$\min_\beta \|\mathbf{WO}_\mu(x^\beta)\|_F^2,$$

where $x = x^\alpha + x^\beta$, $x_1 = \alpha + \beta$ and

$$\begin{cases} x^\alpha &= [\alpha,\ 0, \cdots,\ 0]^\top, \\ x^\beta &= [\beta,\ x_2,\ \cdots,\ x_n]^\top. \end{cases}$$

As in §2.1, we take bandwidth $\mu = 3$ and consider, for simplicity, the Haar wavelets $n_{dwt} = 2$. Let $n = 2^k$ and a full wavelet transform of k levels for x^β is $\widetilde{x}^\beta = W_k W_{k-1} \ldots W_1 x^\beta$.

Then a direct calculation shows that:

$$W_1 x^\beta = \left[\frac{1}{\sqrt{2}}(x_2 + \beta),\ \frac{1}{\sqrt{2}}(x_4 + x_3),\ \cdots,\ \frac{1}{\sqrt{2}}(x_n + x_{n-1}), \right.$$
$$\left. \frac{1}{\sqrt{2}}(x_2 - \beta),\ \frac{1}{\sqrt{2}}(x_4 - x_3),\ \cdots,\ \frac{1}{\sqrt{2}}(x_n - x_{n-1}) \right]^\top.$$

The off-band squared error at level $\nu = 1$ induced by β is $\frac{1}{2}(x_2 - \beta)^2$. Similar calculations for $W_2 W_1 x^\beta$ gives the extra error at level $\nu = 2$ is $\frac{1}{4}(x_4 + x_3 - x_2 - \beta)^2$. Thus the overall error induced by β component is:

$$E_\beta = \frac{1}{2}(x_2 - \beta)^2 + \frac{1}{2^2}(x_4 + x_3 - x_2 - \beta)^2 + \cdots + \frac{1}{2^k}(x_n + x_{n-1} + \cdots + x_{n/2+1} - x_{n/2} - \cdots - x_2 - \beta)^2$$

with

$$\frac{\partial E_\beta}{\partial \beta} == \frac{n-1}{2^{k-1}}\beta - \frac{1}{2^{k-1}} \sum_{j=2}^{n} x_j.$$

Clearly

$$\min_\beta E_\beta = \min_\beta \|\mathbf{WO}_\mu(x^\beta)\|_F^2,$$

whose solution is

$$\beta = \frac{1}{n-1} \sum_{j=2}^{n} x_j.$$

For the example (11) with $x_1 = 30$, in §2.3, we have intuitively tried $\alpha_1 = 22, \beta_1 = 8$, and $\alpha_2 = 24, \beta_2 = 6$. From the above analysis, $\beta_1 = 8$ is a reasonable choice because the level $\mu = 2$ error is zero but $\beta = 6$ is not a good choice. We could try $\beta = x_2 = 7$ which makes the level $\mu = 1$ error zero. However the best choice is $\beta_* = \sum_{j=2}^{8} x_j / 7 = 28/7 = 4$.

Now consider the matrix case. Let $A = D + C$ with D a diagonal matrix and C have n unknown parameters: $C_{1,1},\ \cdots,\ C_{nn}$. Then following Theorem 4, our problem is to solve either

$$\min_{C_{1,1},\ \cdots,\ C_{n,n}} \|\mathbf{WO}_\mu(C)\|_F \frac{\max(A_{j,j} - C_{j,j})}{\min(A_{j,j} - C_{j,j})} \qquad s.t. \qquad C_{j,j} < A_{j,j}, \tag{24}$$

or

$$\min_{C_{1,1},\ \cdots,\ C_{n,n}} \|D\mathbf{WO}_\mu(D^{-1}C)\|_F \qquad s.t. \qquad C_{j,j} < A_{j,j}. \tag{25}$$

Here our assumption is that A has positive diagonal entries (see the previous Remark) and the constraint is to ensure D is positive. The general optimal solution for either problem is hard to find. We shall consider an iterative solution for minimising the object functional in a subspace of 4 parameters for simplicity; it is analogous to minimize a different number of parameters (say 1 or 8).

To demonstrate how to decouple the problem and reduce it into a local minimization, we consider $n_{dwt} = 2$ with only 2 levels of wavelet transform. As with vectors, we perform direct transforms and compute the squared errors $\mathbf{WO}_\mu(C)$ and $\mathbf{WO}_\mu(D^{-1}C)$ of off-band elements involving $C_{j,j}$. We consider the 2 cases of (24) and (25) separately.

<u>**Problem (24)**</u>. For level $\nu = 1$, we have (showing elements involving $C_{j,j}$):

$$W_1 C W_1^{\mathrm{T}} = \left[\begin{array}{cccccccc}
C_1^{ss} & & & & C_1^{sd} & & & \\
& C_2^{ss} & & & & C_2^{sd} & & \\
& & \ddots & & & & \ddots & \\
& & & C_{n/2}^{ss} & & & & C_{n/2}^{sd} \\
C_1^{ds} & & & & C_1^{dd} & & & \\
& C_2^{ds} & & & & C_2^{dd} & & \\
& & \ddots & & & & \ddots & \\
& & & C_{n/2}^{ds} & & & & C_{n/2}^{dd}
\end{array}\right]_{n\times n},$$

where 's' stands for sum (average) and 'd' for difference, and

$$C_j^{ss} = \frac{1}{2}(C_{2j,2j} + C_{2j-1,2j} + C_{2j,2j-1} + C_{2j-1,2j-1}),$$

$$C_j^{sd} = \frac{1}{2}(C_{2j,2j} + C_{2j-1,2j} - C_{2j,2j-1} - C_{2j-1,2j-1}),$$

$$C_j^{ds} = \frac{1}{2}(C_{2j,2j} - C_{2j-1,2j} + C_{2j,2j-1} - C_{2j-1,2j-1}),$$

$$C_j^{dd} = \frac{1}{2}(C_{2j,2j} - C_{2j-1,2j} - C_{2j,2j-1} + C_{2j-1,2j-1}).$$

The quantity of interest is the level $\nu = 1$ squares error:

$$\sum_{j=1}^{n/2}\left[C_j^{sd^2} + C_j^{ds^2}\right] = \frac{1}{2}\sum_{j=1}^{n/2}\left[\underbrace{(C_{2j,2j} - C_{2j-1,2j-1})^2}_{\textit{unknown}} + \underbrace{(C_{2j-1,2j} - C_{2j,2j-1})^2}_{\textit{known}}\right].$$

We may use the notation:

$$\|\mathbf{WO}_\mu^{(1)}(C)\|_F^2 = \frac{1}{2}\sum_{j=1}^{n/2}(C_{2j,2j} - C_{2j-1,2j-1})^2$$

to represent the level $\nu = 1$ error.

To work out level $\nu = 2$ error, note that due to orthogonality, the level $\nu = 1$ error of off-band elements remains unchanged at fine levels. Thus we are only required to look into a smaller diagonal block for each new level with:

$$(W_2W_1CW_1^{\top}W_2^{\top})_{n/2\times n/2}=\left[\begin{array}{cccccccc} C_1^{ssss} & & & & C_1^{sdsd} & & & \\ & C_2^{ssss} & & & & C_2^{sdsd} & & \\ & & \ddots & & & & \ddots & \\ & & & C_{n/2}^{ssss} & & & & C_{n/2}^{sdsd} \\ C_1^{dsds} & & & & C_1^{dddd} & & & \\ & C_2^{dsds} & & & & C_2^{dddd} & & \\ & & \ddots & & & & \ddots & \\ & & & C_{n/2}^{dsds} & & & & C_{n/2}^{dddd} \end{array}\right],$$

where similarly

$$C_j^{ssss}=\frac{1}{2}(C_{2j}^{ss}+C_{2j-1,2j}^{ss}+C_{2j,2j-1}^{ss}+C_{2j-1}^{ss})$$

and so on. The extra level $\nu=2$ error is:

$$\begin{aligned}&\frac{1}{16}\sum_{j=1}^{n/4}\left[(C_{2j}^{ss}-C_{2j-1}^{ss})^2+(C_{2j-1,2j}^{ss}-C_{2j,2j-1}^{ss})^2\right]\\ &=\frac{1}{8}\sum_{j=1}^{n/4}[(C_{4j-1,4j-1}+C_{4j-1,4j}+C_{4j,4j-1}+C_{4j,4j})-\\ &\qquad(C_{4j-3,4j-3}+C_{4j-3,4j-2}+C_{4j-2,4j-3}+C_{4j-2,4j-2})]^2+(\cdot)^2,\end{aligned}$$

where $(\cdot)^2$ denotes a second term not involving the unknown diagonal entries. That is,

$$\begin{aligned}\|\mathbf{WO}_{\mu}^{(2)}(C)\|_F^2=\frac{1}{8}\sum_{j=1}^{n/4}&[(C_{4j-1,4j-1}+C_{4j-1,4j}+C_{4j,4j-1}+C_{4j,4j})-\\ &(C_{4j-3,4j-3}+C_{4j-3,4j-2}+C_{4j-2,4j-3}+C_{4j-2,4j-2})]^2\end{aligned}$$

is the level $\nu=2$ error. Here the pattern to observe is that each level error involves a difference of large diagonal blocks: 1×1 for $\nu=1$, 2×2 for $\nu=2$ and 4×4 for $\nu=3$ and so on.

Therefore to construct a smooth C matrix, we need to select $C_{j,j}$ in such a way that the difference of (at least fixed) diagonal blocks is minimal. To illustrate this idea, we have tried several simple matrices and found that the following C matrices are desirable in terms of good smoothness:

1. A constant diagonal matrix — the off-band errors are zero because the difference of any sized diagonal blocks is zero.

2. A block diagonal matrix of constant and symmetric 2×2 blocks — level 1 errors are zero.

3. A block tri-diagonal matrix of constant and symmetric 2×2 blocks (not necessarily with global symmetry).

4. A block matrix of 4×4 blocks which are symmetric (and especially have constant diagonals) — the difference of blocks are predictable.

The first and simplest choice of $C_{j,j} = 0$ and $D_{j,j} = A_{j,j}$ for all j is a reasonable one because level 1 error $\|\mathbf{WO}_\mu^{(1)}(C)\|_F^2$ is zero — this is the well known case of a diagonal preconditioner: $D = diag(A)$. In this case, term 1 in (24) gains a minimised solution while term 2 inherits the condition of the diagonal entries of A.

The other extreme choice $D = const\ I$ is absolutely not useful because C has the smoothness of A as the off-band errors remain the same. In this case, term 1 in (24) inherits the smoothness/non-smoothness of A while term 2 gains the optimal condition of 1! Thus a good strategy would be to minimise the first term in (24) after specifying a bound for the second term as our aim is to produce a smooth C.

From the off-band error formulae, we can see that the minimization of term 1 in (24) can be easily decoupled while term 2 may be localised to sub-problems. Thus we consider a simple iterative method for solving (24) in an alternating minimization:
for $j = 1, 5, \cdots, n-3$:

$$\min_{C_{j,j},\cdots,C_{j+3,j+3}} \left[\|\mathbf{WO}_\mu^{(1)}(C)\|_F^2 + \mathbf{WO}_\mu^{(2)}(C)\|_F^2\right] \frac{\max_k D_{k,k}^2}{\min_k D_{k,k}^2} \tag{26}$$

$$s.t. \quad D_{\ell+j,\ell+j} > 0, \text{ with } \ell = 0, 1, 2, 3.$$

For more clarity, define $d_k = A_{j+k-1,j+k-1} - C_{j+k-1,j+k-1}$ and rewrite (26) as

$$\min_{d_1,d_2,d_3,d_4} f(d_1, d_2, d_3, d_4), \qquad s.t. \quad d_k > 0, \tag{27}$$

where the objective function f is defined as

$$\begin{aligned}
&\left[\|\mathbf{WO}_\mu^{(1)}(A-D)\|_F^2 + \mathbf{WO}_\mu^{(2)}(A-D)\|_F^2\right] \frac{\max_k D_{k,k}^2}{\min_k D_{k,k}^2} \\
=& \sum_{\substack{\ell=1 \\ i\,=\,4(\ell-1)+1}}^{n/4} \Big[(A_{i+1,i+1} - A_{i,i} - D_{i+1} + D_i)^2 + (A_{i+3,i+3} - A_{i+2,i+2} - D_{i+3} + D_{i+2})^2 \\
&\qquad + (\delta_i + D_{i+3} + D_{i+2} - D_{i+1} - D_i)^2\Big] \frac{\max_k D_{k,k}^2}{\min_k D_{k,k}^2} \\
=& \left[S_j + (a_{12} + d_2 - d_1)^2 + (a_{34} + d_4 - d_3)^2 + (\delta + d_3 + d_4 - d_1 - d_2)^2\right] \frac{\max[d_k^2, d_a^2]}{\min[d_k^2, d_b^2]},
\end{aligned}$$

where $a_{12} = A_{j+1,j+1} - A_{j,j}$, $a_{34} = A_{j+3,j+3} - A_{j+2,j+2}$, $\delta = \delta_j$ is the difference between off-diagonal elements of the 2 consecutive blocks, d_a, d_b denote, respectively, the maximum and the minimum of $|D_{i,i}|$ for $i \neq j, j+1, j+2, j+3$ and S_j is the known sum of all terms in $\mathbf{WO}_\mu^{(1)}(C) + \mathbf{WO}_\mu^{(2)}(C)$ except when $i = j$. An initial guess for the solution of (27) may be taken as $d_k^{(0)} = A_{j+k-1,j+k-1}$ — the diagonal elements of A.

The above nonlinear problem (*outer iterations* for the whole matrix) is still hard to solve and a further simplification is the following. We propose the iterative method (*inner* *iterations* within each block) for $i = 1, 2, \cdots$

$$\left.\begin{array}{l} \min\limits_{d_1^{(i)},d_2^{(i)}} f_{12}(d_1^{(i)}, d_2^{(i)}) = \min\limits_{d_1^{(i)},d_2^{(i)}} f(d_1^{(i)}, d_2^{(i)}, d_3^{(i-1)}, d_4^{(i-1)}), \\ \min\limits_{d_3^{(i)},d_4^{(i)}} f_{34}(d_3^{(i)}, d_4^{(i)}) = \min\limits_{d_3^{(i)},d_4^{(i)}} f(d_1^{(i)}, d_2^{(i)}, d_3^{(i)}, d_4^{(i)}), \end{array}\right\} \tag{28}$$

in a Gauss-Seidel fashion. All solutions will be subject to the constraint: $d_k > 0$ for $k = 1, 2, 3, 4$. In solving equation (28), one must convert the term $\frac{\max}{\min}$ (inside f) into simple and elementary functions. This can be done by considering for each sub-problem of (28) seven possible cases; for example the first sub-problem for d_1, d_2 considers:

$$\left.\begin{array}{ll} 1. & \dfrac{\max_k[d_k^2, d_a^2]}{\min_k[d_k^2, d_b^2]} = \dfrac{\max[d_3^2, d_4^2, d_a^2]}{\min[d_3^2, d_4^2, d_b^2]} = c; \\ 2. & \dfrac{\max_k[d_k^2, d_a^2]}{\min_k[d_k^2, d_b^2]} = \dfrac{d_1^2}{d_2^2}; \\ 3. & \dfrac{\max_k[d_k^2, d_a^2]}{\min_k[d_k^2, d_b^2]} = \dfrac{d_2^2}{d_1^2}; \\ 4. & \dfrac{\max_k[d_k^2, d_a^2]}{\min_k[d_k^2, d_b^2]} = \dfrac{d_1^2}{\min[d_3^2, d_4^2, d_b^2]} = cd_1^2; \\ 5. & \dfrac{\max_k[d_k^2, d_a^2]}{\min_k[d_k^2, d_b^2]} = \dfrac{d_2^2}{\min[d_3^2, d_4^2, d_b^2]} = cd_2^2; \\ 6. & \dfrac{\max_k[d_k^2, d_a^2]}{\min_k[d_k^2, d_b^2]} = \dfrac{\max[d_3^2, d_4^2, d_a^2]}{d_1^2} = c/d_1^2; \\ 7. & \dfrac{\max_k[d_k^2, d_a^2]}{\min_k[d_k^2, d_b^2]} = \dfrac{\max[d_3^2, d_4^2, d_a^2]}{d_2^2} = c/d_2^2, \end{array}\right\} \tag{29}$$

where c is a generic constant independent of the solution of minimization problem (28). All solutions to individual cases will be compared by using the objective function $f = f(d_1, d_2, d_3, d_4)$ to select the minimum for that iteration. Although (28) is an iterative scheme, in practice, we only iterate a few steps to get an improved solution. Similarly the outer iterations (27) will be compared to the initial guess by using the same objective function for problem (24). In practice, we found that it is more effective to monitor the objective function formulated for problem (25), i.e. (31), which is discussed next.

In summary, we have obtained a simple optimization algorithm in system (28) for $j = 1, 5, 9, \ldots, n-3$, that solves problem (24), and gives a minimal solution to the complicated problem (24). The method can be stated as follows:

Algorithm 3 (selection of D)

1. *Compute the objective function $f = f_0$ on setting $D_{k,k} = A_{k,k}$.*

   Start of Outer iterations: *Repeatedly for $j = 1, 5, 9, \cdots, n-3$*

2. *Set an initial guess for d_k from diag(A) or the previous outer iteration;*

3. *Compute $\delta = \delta_j$, S_j, a_{12} and a_{34}.*

```
    Start of Inner iterations: Repeatedly
4. Solve equation 1 of (28) for new $d_1, d_2$;
5. Solve equation 2 of (28) for new $d_3, d_4$;
    End of Inner iterations
6. Set $D_{j,j} = d_1$, $D_{j+1,j+1} = d_2$, $D_{j+2,j+2} = d_3$, $D_{j+3,j+3} = d_4$.
7. Each time after $j = n - 3$, monitor the reduction of the same function $f$ over $f_0$.
    End of Outer iterations
```

Here it should be understood that as soon as a value is obtained for $D_{k,k}$, the corresponding $C_{k,k}$ is also set. If the solution has to be reset to the initial guess for a certain j in this Algorithm (due to negative d_j), then this particular 4 by 4 block uses the simple diagonal preconditioner. In any case, the matrix C obtained should be smooth.

Problem (25). As we are interested in reducing the off-band error of a scaled matrix, it is more natural to seek a solution of problem (25) than (24). However, it turns out that both the formulation and the solution are more difficult to proceed. As we work with $n_{dwt} = 2$ and 2 levels of transform, we now formulate this problem by assuming A is a block diagonal matrix of blocks sized 4×4. For the first diagonal block, the off-band wavelet error can be found to be:

$$\begin{aligned} f_1 &= \|D\mathbf{WO}_\mu(D^{-1}C)\|_F^2 = f_1(d_1, d_2, d_3, d_4) \\ &= d_1^2\left[(H1 + h1)^2 + (G1 - g1)^2 + (a1 - b1)^2 + (a2 - b2)^2\right] + \\ &\ d_2^2\left[(H1 - h1)^2 + (G1 + g1)^2 + (a1 - b1)^2 + (a2 - b2)^2\right] + \\ &\ d_3^2\left[(H1 - h2)^2 + (G1 - g2)^2 + (a1 + b1)^2 + (a2 + b2)^2\right] + \\ &\ d_4^2\left[(H1 + h2)^2 + (G1 + g2)^2 + (a1 + b1)^2 + (a2 + b2)^2\right], \end{aligned} \tag{30}$$

where

$$
\begin{array}{ll}
H1 = (-CSS - FSS + BSS + CTS)/2; & \\
G1 = (-CSS + FSS - BSS + CTS)/2; & \\
h1 = (C1D + F1D)/\sqrt{(2)}; & g1 = (-C1D + F1D)/\sqrt{(2)}; \\
h2 = (B1D + C2D)/\sqrt{(2)}; & g2 = (-B1D + C2D)/\sqrt{(2)}; \\
a1 = (C1S + B1S)/\sqrt{(2)}; & b1 = (-C1S + B1S)/\sqrt{(2)}; \\
a2 = (F1S + C2S)/\sqrt{(2)}; & b2 = (-F1S + C2S)/\sqrt{(2)}; \\
CSS = (\frac{A_{1,1}-d1}{d1} + \frac{A_{1,2}}{d1} + \frac{A_{2,1}}{d2} + \frac{A_{2,2}-d2}{d2})/2; & \\
C1D = (-\frac{A_{1,1}-d1}{d1} - \frac{A_{1,2}}{d1} + \frac{A_{2,1}}{d2} + \frac{A_{2,2}-d2}{d2})/2; & \\
C1S = (-\frac{A_{1,1}-d1}{d1} + \frac{A_{1,2}}{d1} - \frac{A_{2,1}}{d2} + \frac{A_{2,2}-d2}{d2})/2; & \\
BSS = (\frac{A_{3,1}}{d3} + \frac{A_{3,2}}{d3} + \frac{A_{4,1}}{d4} + \frac{A_{4,2}}{d4})/2; & \\
B1D = (-\frac{A_{3,1}}{d3} - \frac{A_{3,2}}{d3} + \frac{A_{4,1}}{d4} + \frac{A_{4,2}}{d4})/2; & \\
B1S = (-\frac{A_{3,1}}{d3} + \frac{A_{3,2}}{d3} - \frac{A_{4,1}}{d4} + \frac{A_{4,2}}{d4})/2; & \\
FSS = (\frac{A_{1,3}}{d1} + \frac{A_{1,4}}{d1} + \frac{A_{2,3}}{d2} + \frac{A_{2,4}}{d2})/2; & \\
F1D = (-\frac{A_{1,3}}{d1} - \frac{A_{1,4}}{d1} + \frac{A_{2,3}}{d2} + \frac{A_{2,4}}{d2})/2; & \\
F1S = (-\frac{A_{1,3}}{d1} + \frac{A_{1,4}}{d1} - \frac{A_{2,3}}{d2} + \frac{A_{2,4}}{d2})/2; & \\
CTS = (\frac{A_{3,3}-d3}{d3} + \frac{A_{3,4}}{d3} + \frac{A_{4,3}}{d4} + \frac{A_{4,4}-d4}{d4})/2; & \\
C2D = (-\frac{A_{3,3}-d3}{d3} - \frac{A_{3,4}}{d3} + \frac{A_{4,3}}{d4} + \frac{A_{4,4}-d4}{d4})/2; & \\
C2S = (-\frac{A_{3,3}-d3}{d3} + \frac{A_{3,4}}{d3} - \frac{A_{4,3}}{d4} + \frac{A_{4,4}-d4}{d4})/2. &
\end{array}
$$

Unfortunately, both $\min_{d1,d_2,d_3,d_4} f_1$ and $\min_{d_k} f_1$ for any particular k are difficult to solve as it is not possible to find an analytical formula.

However, once a new set of d_1, d_2, d_3, d_4 values are known, we may use f_1 to check whether they produce a smaller value than a previous iterate. Furthermore, the overall error for the whole matrix is:

$$f^*(D) = \|D\mathbf{WO}_\mu(D^{-1}C)\|_F^2 = \sum_{\ell=1}^{n/4} f_\ell. \tag{31}$$

This new objective function is a better quantity to minimise and monitor than f in (27) and can be used in step 7 of Algorithm 3.

To summarise, we have proposed an iterative method for solving the minimization problem (25), with a highly nonlinear (difficult) objective function f^*, by minimising its (easier) upper bound function f in (24) while using f^* to monitor the solution process. For example, taking $D = diag(A)$ for the following matrix:

$$A = \begin{bmatrix}
22 & 11 & & & & & & \\
-11 & 2 & 11 & & & & & \\
 & -11 & 3 & 1 & & & & \\
 & & -1 & 4 & 50 & & & \\
 & & & 50 & 88 & -1 & & \\
 & & & & -3 & 99 & -1 & \\
 & & & & & -3 & 0.5 & 0.2 \\
 & & & & & & 0.01 & 0.02
\end{bmatrix},$$

we find that $f = 5.4 \times 10^7$ and $f^* = 3.5 \times 10^{12}$ with $\frac{\max_k d_k^2}{\min_k d_k^2} = 9900$ and $\|\mathbf{WO}_\mu(C)\|_F^2 = 5498$. Now using Algorithm 3 with 1 outer iteration and 10 inner iterations, we get a much better partition $A = D + C$ with

$$D \approx [\ 63 \quad 63 \quad 54 \quad 53 \quad 98 \quad 87 \quad 90 \quad 90\ ]^\top$$

because $f = 1.8 \times 10^4$ and $f^* = 2.7 \times 10^3$ with $\frac{\max_k d_k^2}{\min_k d_k^2} = 1.8$ and $\|\mathbf{WO}_\mu(C)\|_F^2 = 9904$.

To make C even smoother, it may be necessary to vary more than just a diagonal matrix. We could consider more general scaling preconditioners.

3.5 Other scaling preconditioners

To construct suitable scaling preconditioners, consider equations (14) and (22) again. We wish to find different scaling matrices D so that $S = D^{-1}C$ is smoother than C and the transformed matrix $W\mathbf{WB}_\mu(S)W^\top = \left(WSW^\top\right)_{band}$ for the preconditioner dominates the remaining off-diagonal matrix $W\mathbf{WO}_\mu(\mu S)W^\top = \left(WD^{-1}CW^\top\right)_{off}$. That is, as before, $D^{-1}C \in \mathcal{WB}(\mu, \epsilon)$ if $C \in \mathcal{WB}(\mu, \epsilon)$. Because D is no longer a diagonal matrix, we expect $\mathbf{WO}_\mu(C)$ to be much smaller (that is, C much smoother) and a similar form of Theorem 4 to hold.

However, as we work with sparse matrices A, the practical issue is not only that D is (of course) easily and cheaply invertible but also the resulting scaled matrix $D^{-1}A$ must be sparse. Therefore the suitable scaling preconditioners D^{-1} must be a sparse matrix. So the requirement becomes this: find a sparse splitting D that contains the singularity of matrix A and whose inverse D^{-1} is sparse.

Below we consider one simple construction of such a matrix D of near tridiagonal blocks $m \times m$:

$$D = \begin{bmatrix} T & & & \\ & T & & \\ & & \ddots & \\ & & & T \end{bmatrix},$$

$$D^{-1} = \begin{bmatrix} F & & & \\ & F & & \\ & & \ddots & \\ & & & F \end{bmatrix},$$

where T is an $m \times m$ block matrix of a band form (e.g. tridiagonal) and F is a (usually full) matrix of $m \times m$. Although we may allow this block size m to vary in an adaptive fashion, we shall take $m = 2$ in our experiments later.

Determining which scaling preconditioner should be used depends on the nature of the given problem. Intuitively, one should study the underlying matrix A first and ensure that C is very smooth and D will contain the most important features (discontinuous elements) of A while possessing one of these sparse patterns. One should in theory compute a norm of the off-diagonal elements of each case before selecting or specifying the pattern of a suitable scaling preconditioner and for each type of preconditioners try a minimization approach to optimise the construction. More research is still needed to work out efficient ways of implementing these ideas.

4 Complexity of the new WSPAI preconditioner

We now consider the complexity of our main Algorithm 2 and this can be analysed in the two separate stages. In Stage 1, if an $m \times m$ block diagonal preconditioner of size $n = m \times r$ is used (§3.5), the cost of working out $D^{-1}A$ is $O(m^3) * r = O(m^2)n = O(n)$ — increasing as m does although this stage is only performed once. The optimization Algorithm 3 costs $O(n)$ operations for $n_{dwt} = 2$ and 2 levels of wavelet transform in selecting a diagonal matrix D (with $m = 1$). The complexity of Stage 2 is $O(nL)$ with L the number of wavelet levels used. This has been discussed in [5]. Overall the new preconditioning algorithm has a similar operation count to WSPAI of [5].

5 Numerical results

We shall apply the new WSPAI preconditioner with GMRES(20) for solving the following matrix and PDE problems:

P1. A is the perturbed 1024×1024 matrix from the smooth matrix of (4) adding a non-smooth diagonal matrix D with with $D_{ii} = -3$ $(i \leq n/2)$, 3 $(i > n/2)$.

P2. A is the 2D Laplacian operator. This is the second example tested in [5] and we choose this simple test to demonstrate the behaviour of Algorithm 3 for a smooth case.

P3. The oscillatory example from Harwell-Boeing collection [12]: Watt (case 1) with $n = 1856$.

P4. An anisotropic problem in both x and y directions:

$$a(x,y)u_{xx} + b(x,y)u_{yy} = 1,$$

where the coefficients are defined as ([18, Ch.5])

$$a(x,y) = \begin{cases} 100 & (x,y) \in [0,0.5] \times [0,0.5] \text{ or } [0.5,1] \times [0.5,1] \\ 1 & \text{otherwise;} \end{cases}$$

$$b(x,y) = \begin{cases} 100 & (x,y) \in [0,0.5] \times [0.5,1] \text{ or } [0.5,1] \times [0,0.5] \\ 1 & \text{otherwise.} \end{cases}$$

P5. A discontinuous coefficient problem:

$$(a(x,y)u_x)_x + (b(x,y)u_y)_y + u_x + u_y = \sin(\pi xy),$$

where the coefficients are defined as ([18, Ch.3])

$$a(x,y) = b(x,y) = \begin{cases} 10^{-3} & (x,y) \in [0,0.5] \times [0.5,1] \\ 10^3 & (x,y) \in [0.5,1] \times [0,0.5] \\ 1 & \text{otherwise.} \end{cases}$$

Each PDE problem will be discretised by the finite difference method giving rise to a linear system of size $n \times n$ with $n = 1024$. As in [5], we shall mainly use $L = 6$ levels of Daubechies' order $n_{dwt} = 4$ wavelets; however when testing Algorithm 3 we shall use $n_{dwt} = 2$ and $L = 2$ to verify the effectiveness of optimization (see method S5 below). Recall that WSPAI, when it works, can be compared favorably with other SPAI preconditioners and ILU(0) and the conclusion in [5] was that WSPAI will out perform other preconditioners if it does not fail all together. Here in this section we present results using Algorithm 2 to solve the above examples and compare them with WSPAI [5] only.

For easy presentation, we shall use these abbreviation codes to denote different preconditioning methods (with GMRES(20)):

S1 — WSPAI method of [5] with no scaling;

S2 — Simple diagonal preconditioner;

S3 — Algorithm 2 with a simple stage 1 preconditioner $D = diag(A)$;

S4 — Algorithm 2 with a block 2×2 stage 1 preconditioner;

S5 — Algorithm 2 with an optimal stage 1 preconditioner from Algorithm 3.

Then in Table 1, we show the number of accumulated GMRES steps required to reduce the relative residual error to below 10^{-6}. That is to say, 20 steps mean 'one GMRES(20)' step and 40 steps mean '2 GMRES(20) steps'.

Clearly the simple diagonal preconditioner (case S2) does not work well in general (nor does the unpreconditioned case which is not listed here). The method of [5] (case S1) can perform well but appears to be problem dependent. However all cases (S3-S5) of two level preconditioning Algorithm 2 show much better and consistent performance. Note that the stage 1 preconditioner for both S3 and S4 are cheap and trivial to implement. The optimization case S5 uses only $L = 2$ levels but gives the best results. This suggests that it may be worthwhile to develop an algorithm for $L > 2$ and $n_{dwt} \geq 4$ wavelets. For example P5, we display in Figure 6 the convergence behaviour of all five cases of preconditioned GMRES. Again one may conclude that for this discontinuous problem, S1 and S2 will not converge.

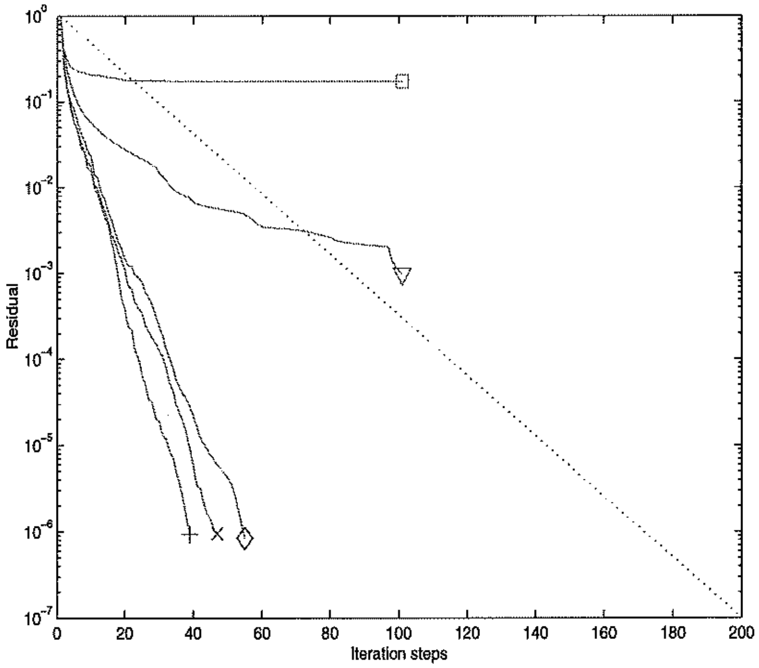
6 Conclusions

We have presented a robust two level sparse preconditioner for Krylov subspace methods, improving our previous work on the subject. The novel idea was to introduce a smoothing

Table 1: Number of GMRES iterations for all test examples

Example/Method	S1	S2	S3	S4	S5
P1	68	58	51	25	45
P2	22	> 100	22	21	19
P3	50	> 100	34	32	29
P4	62	> 100	63	55	22
P5	> 100	> 100	54	46	38

Figure 6: Convergence behaviour of GMRES(20) for problem P5. Symbol $\square$: method S1, ∇: S2, $\diamondsuit$: S3, $\times$: S4, $+$: S5. Observe that all variants (S3-S5) of Algorithm 2 perform better than S1-S2 although the optimized version (S5) is the best.



step for the original matrix prior to applying wavelet transforms — combining the wavelet sparse approximate inverse preconditioner with an extra (scaling) preconditioning stage. We have introduced and used a wavelet band splitting idea to characterise singularities and smoothness in the context of wavelet compression. Simple scaling preconditioners are proposed and analysed. We have developed a minimization approach to select the best diagonal scaling preconditioner for simple cases. The successful experiments suggest that the new two level sparse preconditioner is robust as well as efficient. More general optimization procedures should be investigated in future study for the case of discontinuities not along the main diagonal. A combination with other ideas such as [13] and [10] may also be considered.

Acknowledgement

The authors wish to thank Justin Wan for providing several test programs and data sets for experiments. The second author wishes to thank the Department of Mathematics, UCLA for its hospitality during his visits conducting part of this work.

References

[1] Axelsson O. (1994), *Iterative solution methods*, Cambridge University Press, UK.

[2] Benson M. W. and Frederickson P. O. (1982), *Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems*, Utilitas Math., 22, pp.127-140.

[3] Beylkin G., Coifman R. and Rokhlin V. (1991), *Fast wavelet transforms and numerical algorithms I*, Commu. Pure Appl. Math., **XLIV**, pp.141-183.

[4] Bond D. and Vavasis S. (1994), *Fast wavelet transforms for matrices arising from boundary element methods*, Computer Science Research Report TR-174, Cornell University, USA.

[5] Chan T. F., Tang W. P. and Wan W. L. (1997), *Wavelet sparse approximate inverse preconditioners*, BIT, 37, pp.644-660.

[6] Chen K. (1996), *Preconditioning boundary element equations*, Boundary elements: implementation and analysis of advanced algorithms (W. Hackbusch & G. Wittum, ed.), no. 54, Vieweg, Germany.

[7] ——— (1998), *On a class of preconditioning methods for dense linear systems from boundary elements*, SIAM J. Sci. Comput., 20, 684-698.

[8] ——— (1999), *Discrete wavelet transforms accelerated sparse preconditioners for dense boundary element systems*, Elec. Trans. Numer. Anal., 8, 138-153.

[9] Chow E. (2000), *A priori sparsity patterns for parallel sparse inverse preconditioners*, SIAM J. Sci. Compt., 21, 1804-1822.

[10] Ford, J. M., Chen K. and Scales L. E. (2000), *Wavelet transform preconditioners for matrices arising from elasto-hydrodynamic lubrication*, to appear in: Int. J. Compt. Math..

[11] Greenbaum A. (1997), *Iterative Methods for Solving Linear Systems*, SIAM publications, Philadelphia.

[12] Duff I. S. et al (1992), *User's Guide for the Harwell-Boeing Sparse Matrix Collection*, p.80, available from ftp://ftp.cerfacs.fr/pub/harwell_boeing/userguide.ps.Z. See also *http://math.nist.gov/MatrixMarket/*.

[13] Duff I. S. and Koster J. (1999), *The Design and Use of Algorithms for Permuting Large Entries to the Diagonal of Sparse Matrices*, SIAM J. Matr. Anal. Appl., 20, 889-901. (also RAL-TR-1999-030, http://www.numerical.rl.ac.uk/reports/)

[14] Nachtigal N. M., Reddy S. and Trefethen N. (1992), *How fast are nonsymmetric matrix iterations?*, SIAM J. Matr. Anal. Appl., 13, 778-795.

[15] Saad Y. (1996), *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston.

[16] Strang G. and Nguyen T. (1996), *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA, USA.

[17] Vavasis S. (1992), *Preconditioning for boundary integral equations*, SIAM J. Matr. Anal. Appl., 13, pp. 905–925.

[18] Wan W. L. (1998), *Scalable and multilevel iterative methods*, Ph.D thesis, CAM report 98-29, Dept of Mathematics, UCLA, USA.