

An Improved Wavelet Based Preconditioner for Sparse Linear Problems

Arikera Padmanabha Reddy, Nagendrapp M. Bujurke

Department of Mathematics, Karnatak University, Dharwad, India

E-mail: bujurke@yahoo.com, paddu7_math@rediffmail.com

Received August 9, 2010; revised September 11, 2010; accepted September 14, 2010

Abstract

In this paper, we present the construction of purely algebraic Daubechies wavelet based preconditioners for Krylov subspace iterative methods to solve linear sparse system of equations. Effective preconditioners are designed with DWTPerMod algorithm by knowing size of the matrix and the order of Daubechies wavelet. A notable feature of this algorithm is that it enables wavelet level to be chosen automatically making it more robust than other wavelet based preconditioners and avoids user choosing a level of transform. We demonstrate the efficiency of these preconditioners by applying them to several matrices from Tim Davis collection of sparse matrices for restarted GMRES.

Keywords: Discrete Wavelet Transform, Preconditioners, Sparse Matrices, Krylov Subspace Iterative Methods

1. Introduction

The linear system of algebraic equations

$$Ax = b \quad (1)$$

where A is $n \times n$ non-singular matrix and b is vector of size n arise while discretising differential equations using finite difference or finite element schemes. For the solution of (1), we have a choice between direct and iterative methods. While direct methods, such as Gaussian elimination, can be implemented if n is small, for very large values of n (from Tim Davis collection of matrices [1]) it is often necessary to resort to iteration methods such as relaxation schemes or Krylov subspace iterative methods (Conjugate Gradient (CG), Generalised Minimum Residual (GMRES) and their variants etc). Each iterative method is aimed at providing a way of solving a particular class of linear system, but none can be relied upon to perform well for all problems. Usually it is necessary to make use of a preconditioner to accelerate the convergence of iteration. This means that the original system of equations is transformed into equivalent system, which is more amenable to solution by a chosen iterative method (Ford [2]). The combination of preconditioning and Krylov subspace iterations could provide efficient and simple general-purpose procedures that compete with direct solvers (Saad [3]). A preconditioner

is a non-singular matrix $P (\approx A)$ with $P^{-1}A \approx I$. The philosophy is that the preconditioner has to be close to A and still allow fast converging iteration. The broad class of these preconditioners are mainly obtained by splitting A , say, in the form $A = P + Q$ and writing (1) in the form

$$(I + P^{-1}Q)x = P^{-1}b. \quad (2)$$

If P is the diagonal of A then the iteration scheme is due to Jacobi (P is not close to A) and the other extreme is $P = A$ (too close to A). In between these two extremities, we look for P as band matrix constructed by setting to zero all entries of the matrix outside a chosen diagonal bandwidth [4].

Discrete Wavelet Transform (DWT) is a big boon especially in signal and Image processing where in smooth data can be compressed sufficiently without losing primary features of the data. Compressed data either thresholding or cutting is of much use if we are not using it in further process such as using it as preconditioner. DWT results in matrices with finger pattern. If DWT is followed by the permutation of the rows and columns of the matrix then it centres/brings the finger pattern about the leading diagonal. This strategy is termed as DWTPer an elegant analysis presented by Chen [5,6] for large dense matrices which enables in predicting the width of

band of matrix. Later, an approximate form of this can be formed and taken as preconditioner, which controls fill-in whenever schemes like LU decomposition is used. Similar criteria is adopted in transforming non-smooth parts, if any, which are horizontal/vertical bands and are shifted to the bottom and right-hand edges of the matrix after applying DWTPer. This procedure is termed as DWTPerMod [7] and takes care of other cases where non-smooth parts are located in the matrix. This scheme is more effective in incorporating missing finer details such as fixing of precise bandwidth and automatic selection of choice of transform level. Using DWTPerMod algorithm Ford [7] has presented its salient features by applying it to standard dense matrices arising in various disciplines/fields of interest.

Kumar and Mehra [8] use iDWT, iDWTPer (implementation of DWT by zero padding [9]) and demonstrate their efficiency as preconditioner for ill conditioned sparse matrices and iDWTPer to be more robust preconditioners for sparse systems. Motivated by this work ([5-8]), we have successfully used DWTPerMod algorithm in selecting the level of wavelet transform automatically by knowing the size of the matrix, order of wavelet used and construct effective preconditioner for *sparse unsymmetric linear systems*.

The structure of the paper is organised as follows: In Section 2 we explain briefly about orthogonal wavelets. Section 3 contains practical computation details of transforms required. Here, we present the main results of wavelet based preconditioners in the form of Theorem and Algorithms. In Section 4 we illustrate seven typical test problems selected from Tim Davis collection to emphasize the potential usefulness of preconditioners designed using various orders of Daubechies wavelet. Finally, in Section 5 we summarise our conclusion and give future plan of work.

2. Wavelet Preliminaries

The effective introduction to wavelets via multiresolution analysis (MRA) is broad based compared with conventional/classical way through scaling function concept and as such the following basics are given below.

2.1. Multiresolution Analysis

A multiresolution analysis [9-11] on R is a sequence of closed subspaces $\{V_j\}_{j \in \mathbb{Z}}$ of $L^2(R)$ satisfying the following conditions

- 1) nestedness: $V_j \subset V_{j+1}$
- 2) completeness: closure of $\bigcup V_j = L^2(R)$, nullity:

$$\bigcap V_j = \{0\}$$

- 3) scale invariance: $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$
- 4) shift invariance: $f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0, k \in \mathbb{Z}$
- 5) shift invariant basis: there exists function ϕ in V_0 whose integer translates $\{\phi(\bullet-k)\}$ are an orthonormal basis for V_0 .

Let W_0 be a complement of V_0 in V_1 , i.e., $V_1 = V_0 \oplus W_0$. This implies that there exists a function $\psi \in W_0$ such that its translates $\{\psi(\bullet-k)\}_{k \in \mathbb{Z}}$ are an orthonormal basis for W_0 . For each $j, k \in \mathbb{Z}$, define

$$\begin{aligned} \phi_{j,k}(x) &= 2^{j/2} \phi(2^j x - k) \\ \psi_{j,k}(x) &= 2^{j/2} \psi(2^j x - k) \end{aligned}$$

then these form orthonormal bases for V_j and W_j respectively and $\{\psi_{n,k} / n, k \in \mathbb{Z}\}$ forms an orthonormal basis of $L^2(R)$ and

$$V_{j+1} = V_j \oplus W_j \tag{3}$$

Since V_0 and W_0 are contained in V_1 , we have

$$\phi(x) = \sqrt{2} \sum_k h_k \phi(2x-k), \quad \psi(x) = \sqrt{2} \sum_k g_k \phi(2x-k)$$

for some $\{h_k\}, \{g_k\}$ in $l^2(R)$. Here $\{h_k\}, \{g_k\}$ are called lowpass and highpass filter coefficients respectively. These filter coefficients can be generated using factorization scheme of Daubechies polynomials [11,12] and the functions ϕ and ψ are called scaling and wavelet functions respectively. We define orthogonal projections P_j, Q_j from $L^2(R)$ onto V_j, W_j by

$$P_j f = \sum_{k \in \mathbb{Z}} \langle f, \phi_{j,k} \rangle \phi_{j,k} \tag{4}$$

$$Q_j f = P_{j+1} f - P_j f = \sum_{k \in \mathbb{Z}} \langle f, \psi_{j,k} \rangle \psi_{j,k}.$$

P_j and Q_j are also called approximation and detail operators on f and for any $f \in L^2(R)$, $P_j f \rightarrow f$ in L^2 as $j \rightarrow \infty$.

3. Practical Computations

Suppose that a vector (signal) s from a vector space R^n is given. One may construct it as an infinite sequence by extending the signal by periodic form/extension [9,11] and use this extended signal as a sequence of scaling coefficients for some underlying function f_L in some fixed space V_L of L^2 (from (4)) as

$$f_L(x) = \sum_k s_{L,k} \phi_{L,k}(x). \tag{5}$$

From (3), we have $V_L = V_r \oplus W_r \oplus W_{r+1} \oplus \dots \oplus W_{L-1}$, this implies that

$$f_L(x) = \sum_k s_{r,k} \phi_{r,k}(x) + \sum_{l=r}^{L-1} \sum_i d_{l,i} \psi_{l,i}(x). \tag{6}$$

perform the wavelet transform independently for each dimension. For two dimensional cases, let A be a $n \times n$ matrix then its wavelet transform is

$$\tilde{A} = WAW^T. \tag{15}$$

Here \tilde{A} contains four types of coefficients/subbands [13]: LL-lowpass in both the horizontal and vertical directions (approximation/average coefficients), LH-lowpass in the vertical, highpass in the horizontal direction, HL and HH (detail coefficients). When iterated on the approximation coefficients, the result is multiresolution decomposition as shown in the **Figure 1**. This LL subband contains lowpass information, which is essentially a low resolution and represents a coarser version of the original matrix A very much.

If A is smooth, the transformed matrix \tilde{A} will have a large part of small coefficients, corresponding to detail coefficients (after thresholding). Singularity features within A give rise to additional large entries in \tilde{A} . For a matrix that is smooth apart from the diagonal, the large entries form a finger pattern, as shown in left-side of **Figure 2**. This sparsity pattern is not convenient for preconditioning purposes, because of large amount of fill-in that occurs under LU factorisation. One way of avoiding the finger pattern is to permute the rows and columns of \tilde{A} so as to bring the detail coefficients into a diagonal band. The sparsity pattern of this DWTPer transform is shown in the centre of **Figure 2**. For a matrix A of order $n \times n$, the DWTPer would give $\hat{A} = \hat{W}A\hat{W}^T$. To relate \hat{A} to \tilde{A} from a standard DWT or relate \hat{W} to W , in [6], it is proved that $\hat{A} = R\tilde{A}R^T$, where $R = P_1^T P_2^T \dots P_k^T$ and each P_i is a permutation matrix as defined in (11).

3.2. Banded Wavelet Based Preconditioner

To solve $Ax = b$, using wavelet based preconditioner following algorithm is considered [5-8].

Algorithm 3.1:

- 1) Apply DWTPer to $Ax = b$ to obtain $\hat{A}u = z$
- 2) Select a suitable band form \hat{M} of \hat{A} (Theorem 3.1)
- 3) Use \hat{M} as a preconditioner to solve $\hat{A}u = z$ iteratively.

- 4) Apply Inverse Wavelet Transform on u to get required solution.

The strategy that we take in preconditioning step is split the given matrix $A = P + Q$, where P is a Band (α, β) for some $\alpha, \beta \in \mathbb{N}$. Band (α, β) means a matrix whose lower bandwidth is α and upper bandwidth is β . First apply DWTPer with k levels to give

$$\hat{A}u = (\hat{P} + \hat{Q})u = z.$$

Here \hat{P} is at most of Band (λ_1, λ_2) , λ_1, λ_2 are first given by Chen [6] and further tightened by Ford [7], which are given in the following Theorem. Select a matrix \hat{M} as preconditioner of Band (α_1, β_1) part of the matrix \hat{A} such that $\alpha_1 \ll \lambda_1$ and $\beta_1 \ll \lambda_2$.

Theorem 3.1: When an order D , level k DWTPer is applied to a band matrix A with lower bandwidth α and upper bandwidth β , the resulting \hat{A} which is at most a Band (λ_1, λ_2) matrix with $\lambda_1 - \alpha = \lambda_2 - \beta = (D - 1)(2^k - 1) + 2^{k-1}$.

By using Algorithm 3.1, Kumar and Mehra [8] developed wavelet based preconditioners for Krylov subspace iterative methods for ill conditioned sparse matrices and shown that these preconditioners are more effective compared with that of classical preconditioners by applying them to several test matrices. Performance of Algorithm 3.1 depends on the level of wavelet transform used, which must be decided in advance by user. The following impulsive results due to Ford [7] overcome this limitation. In Subsection 3.3 it is briefly summarised.

3.3. Border Block Preconditioner

After applying k level of DWT to matrix A , detail coefficients in \tilde{A} are brought to make diagonal band by

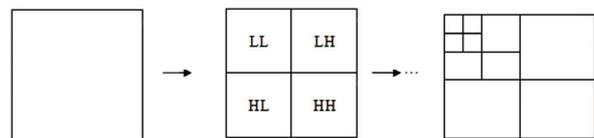


Figure 1. Two-dimensional wavelet transform: iteration on the LL subbands (average coefficients).

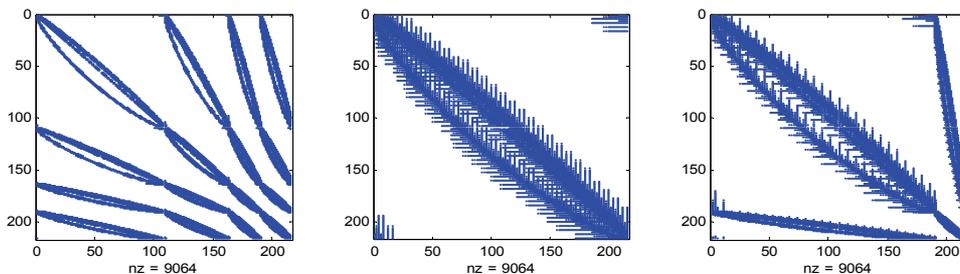


Figure 2. Sparsity pattern of Gre_216a matrix under Daub-4, left: Standard DWT; centre: DWTPer; right: DWTPerMod.

permutation matrices (thus obtained is \hat{A}). Now permute the rows and columns so that average coefficients in \hat{A} are confined within bands of width $\lfloor n/2^k \rfloor$, at the bottom (horizontal) and to the right-hand (vertical) edges of \hat{A} . Then preconditioner is constructed by setting to zero all entries that fall outside the diagonal band and these two edge bands (right side of **Figure 2**). This modification is termed as DWTPerMod [7].

Once we determine the diagonal bandwidth (using Theorem 3.1), we can estimate the cost of applying the factorised block preconditioner by forward and backward substitution, based on widths of diagonal, horizontal and vertical bands. The cost of forward and backward substitution is proportional to the number $N_z(k)$ of non-zero entries in LU factors of P . Hence we choose k such that $N_z(k)$ is minimum. A matrix of size n with borders of width r and a diagonal band with lower and upper bandwidth p can be factored into LU factors such that $N_z(k) \approx n(3p + 2r)$ [14]. We summarise our new preconditioning method as Algorithm 3.2.

Algorithm 3.2 (DWTPerMod Preconditioner). Given a sparse matrix A of size n and a DWT of order D , compute DWTPerMod preconditioner as follows:

- 1) for $i = 1, 2, \dots, \lfloor \log_2^{(n/D-1)} \rfloor$, compute $p(i)$ using Theorem 3.1, $r(i) = n/2^i$, $N_z(i) = n(3p(i) + 2r(i))$.
 - 2) Choose k such that $N_z(k) = \min_i \{N_z(i)\}$.
 - 3) Apply a level k DWTPer to A to obtain \hat{A} .
 - 4) Permute the rows and columns of \hat{A} so that the average coefficients lie in bands at the bottom and right hand edges.
 - 5) Form a border block preconditioner \bar{M} by setting to zero entries in \hat{A} outside of diagonal band of width $p_1(k) \ll p(k)$ and borders of width $r_1(k) \ll r(k)$.
- In the above Algorithm 3.2, $\lfloor x \rfloor$: rounds x to the nearest integer towards minus infinity. For simplicity we took the upper and lower bandwidths equal ($\alpha = \beta$). However, this preconditioning strategy is equally applicable when bandwidths are different [7].

4. Numerical Experiments

To test the robustness of above explained wavelet based preconditioners, we have considered various problems given in **Table 1**. The right hand side of linear system was computed from the solution vector x of all ones. This choice is suggested by Zlatev [15]. We have implemented the proposed algorithms using Matlab-7.5 and Mathematica-7. The initial guess is always $x_0 = 0$ and stopping criteria is relative residual is less than or equal

to 10^{-6} (i.e., $\|b - Ax\|_2 \leq \|b\|_2 10^{-6}$) and the Krylov Subspace iterative Method adapted is GMRES (25) [3]. In **Table 1** n, nnz and $k(A)$ represent the size, number of nonzero entries and condition number of the corresponding matrix A . The symbol Ω stands for slow convergence/no convergence in the tables and numbers in the table represent number of iterations required for convergence. Last column of **Table 1** is obtained without using preconditioner for GMRES (25). Daubechies wavelets of order two, four and six are considered in our numerical experiments as preconditioners for GMRES (25). **Tables 2-4** are obtained for $\alpha = \beta = 5$. **Table 5** is obtained for various values of α and β .

By applying Algorithm 3.2, we can easily fix the level of transform for a matrix of given size and prescribed Daubechies wavelet in developing wavelet based preconditioners for Krylov subspace iterative methods. To illustrate further salient features of DWTPerMod based preconditioner the computed details are shown in **Table 5** for Gre_512 matrix. It is of interest to note that DWT based preconditioner fail to yield convergence with Daubechies wavelet of order two, four and six, where as DWTPerMod based preconditioner with Daub-4 and Daub-6 result into faster convergence of iterative schemes.

Table 1. Sparse unsymmetric matrices from Tim Davis collection [1].

| Matrix Name | Size (n) | nnz | $k(A)$ | $P = I$ |
|-------------|--------------|-------|--------------------------|---------|
| Bcsstk02 | 66 × 66 | 4356 | 1.2900 × 10 ⁴ | 165 |
| Gre_216a | 216 × 216 | 812 | 3.0499 × 10 ² | Ω |
| Qc324 | 324 × 324 | 26730 | 7.3835 × 10 ⁴ | Ω |
| Ck400 | 400 × 400 | 2860 | 2.1654 × 10 ⁶ | Ω |
| Gre_512 | 512 × 512 | 1976 | 3.8479 × 10 ² | Ω |
| Epb0 | 1794 × 1794 | 7764 | 1.4719 × 10 ⁵ | Ω |
| Dw8192 | 8192 × 8192 | 41746 | 1.5001 × 10 ⁷ | Ω |

Table 2. Convergence details using Daub-2 based preconditioners.

| Matrix Name | DWT | DWTPer | DWTPerMod |
|-------------|-----|--------|-----------|
| Bcsstk02 | 93 | 91 | 29 |
| Gre_216a | Ω | Ω | 267 |
| Qc324 | Ω | 12 | 11 |
| Ck400 | 172 | 22 | 23 |
| Epb0 | Ω | 1042 | 1247 |
| Dw8192 | Ω | 17 | 16 |

Table 3. Convergence details using Daub-4 based preconditioners.

| Matrix Name | DWT | DWTPer | DWTPerMod |
|-------------|----------|--------|-----------|
| Bcsstk02 | 113 | 67 | 16 |
| Gre_216a | Ω | 71 | 21 |
| Qc324 | Ω | 15 | 15 |
| Ck400 | 443 | 64 | 51 |
| Epb0 | Ω | 684 | 214 |
| Dw8192 | Ω | 23 | 25 |

Table 4. Convergence details using Daub-6 based preconditioners.

| Matrix Name | DWT | DWTPer | DWTPerMod |
|-------------|----------|--------|-----------|
| Bcsstk02 | 66 | 24 | 16 |
| Gre_216a | Ω | 765 | 65 |
| Qc324 | 394 | 16 | 17 |
| Ck400 | Ω | 69 | 119 |
| Epb0 | Ω | 712 | 194 |
| Dw8192 | Ω | 36 | 50 |

Table 5. Convergence details using various orders of wavelet based preconditioners for Gre_512 matrix.

| Wavelet of order | DWT | DWTPer | DWTPerMod |
|--|----------|----------|-----------|
| Daub-2 with $\alpha = \beta (\leq 30)$ | Ω | Ω | Ω |
| Daub-4 with $\alpha = \beta = 30$ | Ω | Ω | 43 |
| Daub-6 with $\alpha = \beta = 20$ | Ω | 751 | 17 |

5. Conclusions and Future Work

DWTPerMod algorithm *improves* on other preconditioners providing 1) tighter bounds on the bandwidth for DWTPer band preconditioning, resulting into preconditioning to be done at lower cost; 2) it removes uncertainty about choosing an appropriate bandwidth, wavelet level and results into more robust scheme.

Convergence details are presented in **Tables 2-5**. It is remarkable to observe the rapid convergence of GMRES (25) with preconditioners designed using DWTPerMod compared with other schemes. Preconditioners developed here can also be used for other Krylov subspace iterative methods and no user intervention is required to choose an appropriate transform level for each example. This is a significant advancement towards the development of purely algebraic wavelet based preconditioning strategy for sparse matrices.

We are developing non orthogonal [16,17] wavelet

based preconditioners and compare their efficiency with the existing Daubechies orthogonal wavelet based preconditioners for various matrices.

6. Acknowledgements

This research work is supported by DST (SR/S4/MS: 281/05). N. M. Bujurke acknowledges the financial support of INSA, New Delhi.

7. References

- [1] T. A. Davis, "University of Florida Sparse Matrix Collection," *NA Digest*, Vol. 97, No. 23, 1997, p. 7. <http://www.cise.ufl.edu/research/sparse/matrices/>
- [2] J. M. Ford, "A Black Box at the End of the Rainbow: Searching for the Perfect Preconditioner," *Philosophical Transactions of Royal Society London A*, Vol. 361, No. 1813, 2003, pp. 2665-2680.
- [3] Y. Saad, "Iterative Methods for Sparse Linear Systems," SIAM, Philadelphia, 2003.
- [4] M. Benzi, "Preconditioning Techniques for Large Linear Systems: A Survey," *Journal of Computational Physics*, Vol. 182, No. 2, 1 November 2002, pp. 418-477.
- [5] K. Chen, "Matrix Preconditioning Techniques and Applications," Cambridge University Press, Cambridge, 2005.
- [6] K. Chen, "Discrete Wavelet Transforms Accelerated Sparse Preconditioners for Dense Boundary Element Systems," *Electronic Transactions on Numerical Analysis*, Vol. 8, 1999, pp. 138-153. <http://etna.mcs.kent.edu/vol.8.1999/pp138-153.dir/pp138-153.pdf>
- [7] J. M. Ford, "An Improved Discrete Wavelet Transform Preconditioner for Dense Matrix Problems," *SIAM Journal on Matrix Analysis and Applications*, Vol. 25, No. 3, 2003, pp. 642-661.
- [8] B. V. R. Kumar and M. Mehra, "Wavelet-Based Preconditioners for Sparse Linear Systems," *Applied Mathematics and Computation*, Vol. 171, No. 1, 1 December 2005, pp. 203-224.
- [9] D. F. Walnut, "An Introduction to Wavelet Analysis," Birkhauser, Boston-Basel-Berlin, 2002.
- [10] S. Mallat, "A Wavelet Tour of Signal Processing," Academic Press, Elsevier, California, 1999.
- [11] G. Strang and T. Nguyen, "Wavelets and Filter Banks," Wellesley-Cambridge Press, Wellesley, 1997.
- [12] I. Daubechies, "Ten Lectures on Wavelets," SIAM, Philadelphia, 1992.
- [13] G. Uytterhoeven, "Wavelets: Software and Applications," Ph.D. Dissertation, Katholieke Universiteit Leuven, Belgium, 1999.
- [14] G. H. Goulab and C. F. van Loan, "Matrix Computations," Hindustan Book Agency, Hindustan, 2007.
- [15] Z. Zlatev, "Computational Methods for General Sparse

- Matrices,” Kluwer, Dordrecht, the Netherlands, 1991.
- [16] A. Cohen, I. Daubechies and J. C. Feauveau, “Bior-
thogonal Bases of Compactly Supported Wavelets,”
Communications on Pure and Applied Mathematics, Vol.
45, No. 5, June 1992, pp. 485-560.
- [17] W. Sweldens, “The Lifting Scheme: A Construction of
Second Generation Wavelets,” *SIAM Journal on Mathe-
matical Analysis*, Vol. 29, No. 2, March 1998, pp. 511-
546.