IMM
DEPARTMENT OF MATHEMATICAL MODELLING

Technical University of Denmark
DK-2800 Lyngby – Denmark

J. No. omn1
1997-5-5
omn

FAST 2D WAVELET TRANSFORM OF CIRCULANT MATRICES

Ole Møller Nielsen

TECHNICAL REPORT

IMM-REP-1997-8

IMM

FAST 2D WAVELET TRANSFORM OF CIRCULANT MATRICES

Ole Møller Nielsen

TECHNICAL REPORT

IMM-REP-1997-8

IMM

Contents

1 Introduction

Circulant matrices arise from a number of problems in numerical analysis such as partial differential equations with periodic boundary conditions and problems involving periodic convolution. It has been suggested by [CP96, EBP0 , PW96, GL94], for example, that one may benefit from treating such problems using wavelets. See e.g. [SN96, Wal94, Dau92, Mey93, Mal89] for the underlying wavelet theory.

In this report we will describe the 2D wavelet transform of a circulant matrix A. This is a transform $A \to H$ of the form

$$H = WAW^T$$

where W is an orthogonal matrix. We will show that this can be done in a computationally efficient way and that H can be represented using very few elements in a suitable data structure. Using this structure we describe an efficient algorithm for computing the matrix-vector product

$$Hx$$

The remainder of our material is organized as follows: In section 2 we define the basic algorithms for the 1D and the 2D case. In section 3 we define circulant and shift-circulant matrices. In section 4 it is shown how H may be computed and stored economically when A is circulant and a formula for the storage requirement is given. The approach follows ideas proposed by Philippe Charton [Cha96]. However, the data structure and many of the details are, to our knowledge, presented here for the first time. In section 5 the algorithm is optimized for banded matrices and the sparsity pattern is analyzed in detail. Finally, in section 6 an algorithm for computing $y = Hx$ is described and the required computational work is calculated.

2 The wavelet transform

2.1 The 1D wavelet transform

Let N be a power of two, i.e. $N = 2^J$, $J \in \mathbb{N}$. The **wavelet transform** is a linear mapping of $\mathbb{R}^N$ onto $\mathbb{R}^N$ defined recursively by a sequence of fundamental linear mappings. We denote this number of mappings by λ and will refer to this as the **depth** of the wavelet transform. Figure 1 illustrates how a vector is transformed.

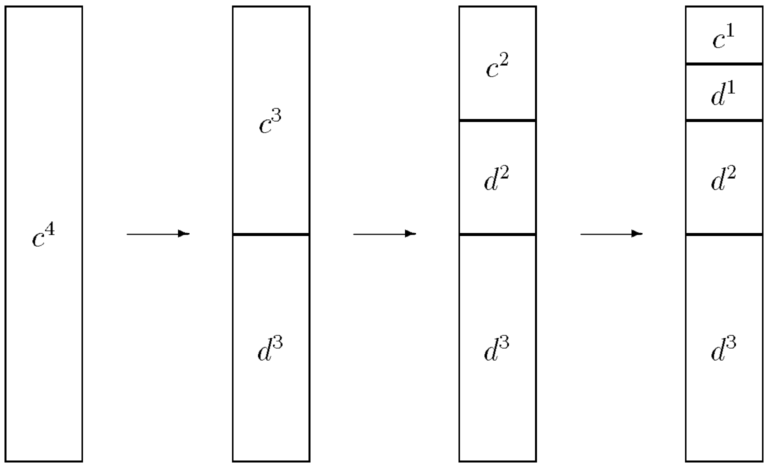

Figure 1: The steps of a 1D wavelet transform for $\lambda = 3$ and $J = 4$.

Let c^J be a given vector with elements $\{c^J_m\}_{m=0,1,\ldots,2^J-1}$. The recurrence formulas for the wavelet transform are

$$c^{j-1}_m = \sum_{k=0}^{D-1} a_k c^j_{\langle k+2m \rangle_{2^j}} \tag{1}$$

$$d^{j-1}_m = \sum_{k=0}^{D-1} b_k c^j_{\langle k+2m \rangle_{2^j}} \tag{2}$$

for $m = 0, 1, \ldots, 2^{j-1} - 1$ and $j = J, J-1, \ldots, J-\lambda+1$. The symbol $\langle k + 2m \rangle_{2^j}$ denotes the modulus operator defined such that $\langle k + 2m \rangle_{2^j} \in [0, 2^j - 1]$ for all values of $k \in \mathbb{Z}$. See appendix A for the details. D is an even positive integer denoting the wavelet genus and the numbers a_k, b_k for $k = 0, 1, \ldots, D-1$ are the wavelet filter coefficients. Each step of the wavelet transform splits a vector into two blocks as illustrated in figure 2.

The choice of D and the filter coefficients comes from the wavelet theory which is beyond the scope of this paper. Here it suffices to mention that the filters characterize the underlying wavelet uniquely and that they are designed such that elements

$$
\begin{array}{ccccccccc}
\mathbf{c}_0^4 & \overset{j:\,4\to3}{\to} & \mathbf{c}_0^3 & \overset{j:\,3\to2}{\to} & \mathbf{c}_0^2 & \overset{j:\,2\to1}{\to} & \mathbf{c}_0^1 & \overset{j:\,1\to0}{\to} & \mathbf{c}_0^0 \\ \cline{9-9}
\mathbf{c}_1^4 & & \mathbf{c}_1^3 & & \mathbf{c}_1^2 & & \mathbf{c}_1^1 & & d_1^0 \\ \cline{7-7}
\mathbf{c}_2^4 & & \mathbf{c}_2^3 & & \mathbf{c}_2^2 & & d_0^1 & & d_0^1 \\
\mathbf{c}_3^4 & & \mathbf{c}_3^3 & & \mathbf{c}_3^2 & & d_1^1 & & d_1^1 \\ \cline{5-5}
\mathbf{c}_4^4 & & \mathbf{c}_4^3 & & d_0^2 & & d_0^2 & & d_0^2 \\
\mathbf{c}_5^4 & & \mathbf{c}_5^3 & & d_1^2 & & d_1^2 & & d_1^2 \\
\mathbf{c}_6^4 & & \mathbf{c}_6^3 & & d_2^2 & & d_2^2 & & d_2^2 \\
\mathbf{c}_7^4 & & \mathbf{c}_7^3 & & d_3^2 & & d_3^2 & & d_3^2 \\ \cline{3-3}
\mathbf{c}_8^4 & & d_0^3 & & d_0^3 & & d_0^3 & & d_0^3 \\
\mathbf{c}_9^4 & & d_1^3 & & d_1^3 & & d_1^3 & & d_1^3 \\
\mathbf{c}_{10}^4 & & d_2^3 & & d_2^3 & & d_2^3 & & d_2^3 \\
\mathbf{c}_{11}^4 & & d_3^3 & & d_3^3 & & d_3^3 & & d_3^3 \\
\mathbf{c}_{12}^4 & & d_4^3 & & d_4^3 & & d_4^3 & & d_4^3 \\
\mathbf{c}_{13}^4 & & d_5^3 & & d_5^3 & & d_5^3 & & d_5^3 \\
\mathbf{c}_{14}^4 & & d_6^3 & & d_6^3 & & d_6^3 & & d_6^3 \\
\mathbf{c}_{15}^4 & & d_7^3 & & d_7^3 & & d_7^3 & & d_7^3
\end{array}
$$

Figure 2: The Wavelet transform. The differences d_m^j obtained at each level remain the same at subsequent levels. Here the level is taken to be as large as possible, i.e. $\lambda = J = 4$.

in the vector c^{j-1} are weighted averages of elements of c^j while d^{j-1} represents the details lost in the mapping from c^j to c^{j-1}. Furthermore, the filter coefficients satisfy the conditions:

$$
\begin{aligned}
\sum_{k=0}^{D-1} a_k &= \sqrt{2} \\
\sum_{k=0}^{D-1-2n} a_k a_{k+2n} &= \delta_{0n} \\
\sum_{k=0}^{D-1-2n} a_k b_{k+2n} &= 0, \quad n \in \mathbf{Z}
\end{aligned}
\tag{3}
$$

The recursion can be carried out to the coarsest level, i.e. $\lambda = J$ where the overall average is represented by the single coefficient c_0^0. However, in practical applications λ is typically chosen to be no larger than 5–7. The number of operations needed to perform the 1D wavelet transform is given by

$$
4DN\left(1 - \left(\frac{1}{2}\right)^{\lambda}\right)
$$

Since, the wavelet transform is a linear mapping of $\mathbb{R}^N$ onto $\mathbb{R}^N$, it can be represented by an $N \times N$ matrix W. If we let $v = c^J$ and $\tilde{v}$ be the result of the recursion after λ steps, i.e.

$$\tilde{v} = \begin{bmatrix} c^{J-\lambda} \\ d^{J-\lambda} \\ d^{J-\lambda+1} \\ \vdots \\ d^{J-1} \end{bmatrix}$$

then we can express the wavelet transform as

$$\tilde{v} = Wv \tag{4}$$

The vector $\tilde{v}$ is denoted the **wavelet spectrum** of v [PPB95]. Because of (3) it can be shown that W is orthogonal so the **inverse wavelet transform** is

$$v = W^T \tilde{v}$$

Note that for $\lambda = 0$ we have $\tilde{v} = v$ and $W = I$.

2.2 The 2D wavelet transform

In this section we will describe the two-dimensional wavelet transform $A \to H$ defined by

$$H = WAW^T \tag{5}$$

where A, H, W are $N \times N$ matrices with $N = 2^J$ and W defined as in the previous section. The operation $(WA)W^T$ corresponds to a 1D wavelet transform of the columns of A followed by a 1D wavelet transform of the rows of (WA). Equation (5) like (4) is a simple representation of a recursive formulation. The 2D recurrence formulas are straightforward generalizations of the 1D formulas, and they decompose a square matrix analogously to the way the 1D recurrence formulas decompose a vector.

The recursion is initialized by assigning the matrix A to the initial block which we denote $CC^{J,J}$. This block is then successively split into smaller blocks denoted $CC^{i,j}$, $DC^{i,j}$, $CD^{i,j}$, and $DD^{i,j}$ for $i, j = J-1, J-2, \ldots, J-\lambda$. The block dimensions are determined by the superscripts i, j: A block with indices i, j has 2^i rows and 2^j columns. The steps of the 2D wavelet transform for $\lambda = 3$ are shown in figure 3 with H being the aggregation of all blocks after the final step shown in the upper right corner in the figure. (Comparison with the 1D wavelet transform shown in figure (1) can be helpful for understanding this scheme). Note that each step of the transform produces blocks that have attained their final values, namely those of the type $DD^{i,j}$, and subsequent steps work on blocks of the type $CC^{j,j}$, $CD^{i,j}$, and $DC^{i,j}$. The formulas for these three types of blocks given below.

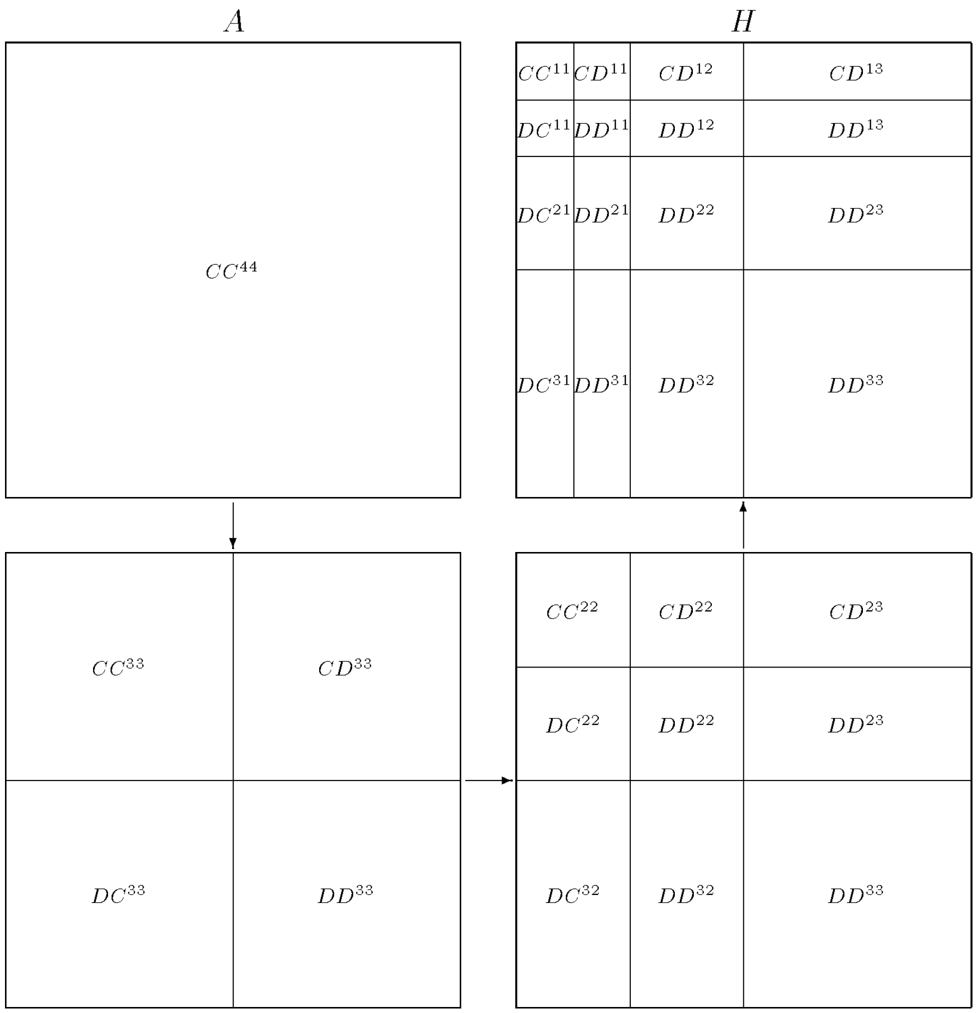

Figure 3: The steps of a 2D wavelet transform for $\lambda = 3$ and $J = 4$. The resulting matrix H has the characteristic "arrow-shaped" block structure.

Blocks on the diagonal

Consider the square blocks $CC^{j,j}$ in figure 3. Each step of the 2D wavelet transform splits such a block into four new blocks denoted $CC^{j-1,j-1}$, $CD^{j-1,j-1}$, $DC^{j-1,j-1}$, and $DD^{j-1,j-1}$. Figure 4 illustrates the decomposition of this type.

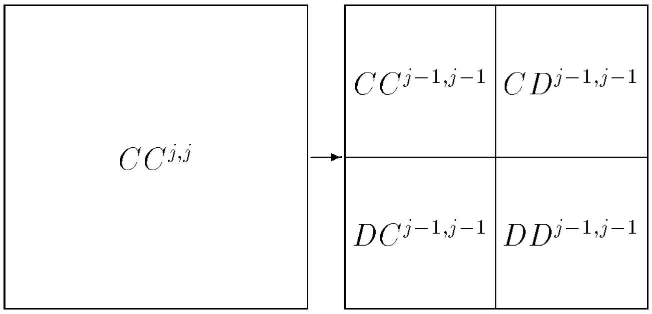

Figure 4: The transform of a square block on the diagonal yields four new square blocks. This operation corresponds to equations (6) to (9).

The recurrence formulas for this operation are as follows:

$$CC^{j-1,j-1}_{m,n} = \sum_{k=0}^{D-1}\sum_{l=0}^{D-1} a_k a_l \, CC^{jj}_{\langle k+2m\rangle_{2^j},\langle l+2n\rangle_{2^j}} \tag{6}$$

$$CD^{j-1,j-1}_{m,n} = \sum_{k=0}^{D-1}\sum_{l=0}^{D-1} a_k b_l \, CC^{jj}_{\langle k+2m\rangle_{2^j},\langle l+2n\rangle_{2^j}} \tag{7}$$

$$DC^{j-1,j-1}_{m,n} = \sum_{k=0}^{D-1}\sum_{l=0}^{D-1} b_k a_l \, CC^{jj}_{\langle k+2m\rangle_{2^j},\langle l+2n\rangle_{2^j}} \tag{8}$$

$$DD^{j-1,j-1}_{m,n} = \sum_{k=0}^{D-1}\sum_{l=0}^{D-1} b_k b_l \, CC^{jj}_{\langle k+2m\rangle_{2^j},\langle l+2n\rangle_{2^j}} \tag{9}$$

for $m, n = 0, 1, \ldots, 2^{j-1} - 1$. The block denoted $DD^{j-1,j-1}$ has attained its final state and will not be modified again.

Blocks below the diagonal

Blocks of the type $DC^{i,j}$ $(i \geq j)$ are split in *one* direction only as indicated in figure 5.

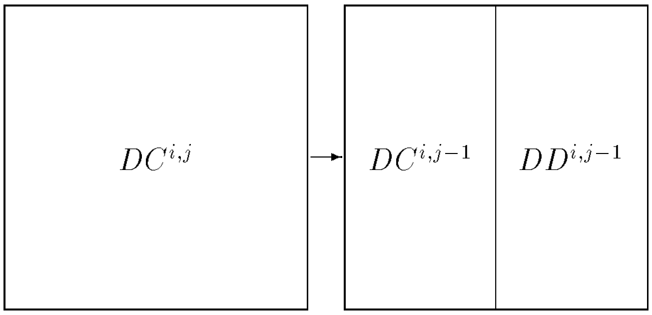

Figure 5: Transform of a block below the diagonal yields two new rectangular blocks. This operation corresponds to the row transforms given in equations (10) and (11).

The recurrence formulas are the 1D formulas applied to each row of the block $DC^{i,j}$:

$$DC^{i,j-1}_{m,n} = \sum_{l=0}^{D-1} a_l \, DC^{i,j}_{m,\langle l+2n \rangle_{2^j}} \qquad (10)$$

$$DD^{i,j-1}_{m,n} = \sum_{l=0}^{D-1} b_l \, DC^{i,j}_{m,\langle l+2n \rangle_{2^j}} \qquad (11)$$

for $m = 0, 1, \ldots, 2^i - 1$ and $n = 0, 1, \ldots, 2^{j-1} - 1$.

Blocks above the diagonal

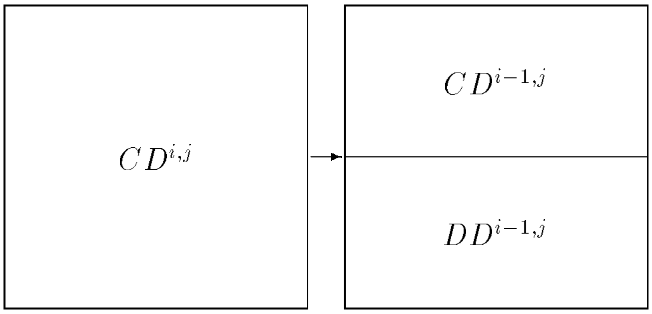

Figure 6: Transform of a block above the diagonal yields two new rectangular blocks. This operation corresponds to the column transforms given in equations (12) and (13).

For blocks $CD^{i,j}$ with $j \geq i$ we have a splitting as shown in figure 6. The recurrence formulas are the 1D formulas applied to each column of $CD^{i,j}$:

$$CD^{i-1,j}_{m,n} = \sum_{k=0}^{D-1} a_k \, CD^{i,j}_{\langle k+2m \rangle_{2^i},n} \tag{12}$$

$$DD^{i-1,j}_{m,n} = \sum_{k=0}^{D-1} b_k \, CD^{i,j}_{\langle k+2m \rangle_{2^i},n} \tag{13}$$

for $m = 0, 1, \ldots, 2^{i-1} - 1$ and $n = 0, 1, \ldots, 2^j - 1$.

3 Circulant matrices

We give the definitions of circulant and shift-circulant matrices that will be used in the following sections.

Definition 3.1 (Circulant matrix) *Let A be an $M \times M$ matrix and let $\{a_m\}_{m=0,1,\ldots,M-1}$ be the first column of A. Then A is* **circulant** *if*

$$A_{mn} = a_{\langle m-n \rangle_M}, \quad m, n = 0, 1, \ldots, M-1$$

The symbol $\langle m-n \rangle_M$ denotes the modulus operator as defined in appendix A.

Since all columns are shifted versions of a, it suffices to store the M elements of a instead of the M^2 elements of A.
A circulant 4×4 matrix has the form

$$\begin{pmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{pmatrix}$$

Definition 3.2 (Shift-circulant matrix)

1. *Let A be an $M \times N$ matrix where $M \geq N$ with M divisible by N, and let $\{a_m\}_{m=0,1,\dots,M-1}$ be the first* column *of A. Then A is* **column-shift-circulant** *if*
$$A_{mn} = a_{\langle m-\sigma n\rangle_M}, \quad m = 0,1,\dots,M-1, n = 0,1,\dots,N-1$$
where $\sigma = M/N$.

2. *Let A be an $M \times N$ matrix where $N \geq M$ with N divisible by M, and let $\{a_n\}_{n=0,1,\dots,N-1}$ be the first* row *of A. Then A is* **row-shift-circulant** *if*
$$A_{mn} = a_{\langle n-\sigma m\rangle_N}, \quad m = 0,1,\dots,M-1, n = 0,1,\dots,N-1$$
where $\sigma = N/M$.

The number σ is a positive integer that denotes the amount by which columns or rows are shifted.

A column-shift-circulant 4×2 matrix ($\sigma = 2$) has the form

$$\begin{pmatrix} a_0 & a_2 \\ a_1 & a_3 \\ a_2 & a_0 \\ a_3 & a_1 \end{pmatrix}$$

A row-shift-circulant 2×4 matrix ($\sigma = 2$) has the form

$$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_2 & a_3 & a_0 & a_1 \end{pmatrix}$$

Note that a circulant matrix is both column-shift-circulant and row-shift-circulant with $\sigma = 1$.

4 2D wavelet transform of a circulant matrix

Since we can work with one column only when A is a circulant matrix, the 2D wavelet transform can be done very economically in terms of storage as well as computations.

Let $\{a_n\}_{n=0,1,\dots,N-1}$ be the first column of a circulant matrix A. Using this column vector as a point of departure we will show how to compute a representation of H using only one vector per block. Note that according to the recurrence equations (6) to (13), the operations can be divided into 2D transforms of blocks on the diagonal and 1D row or column transforms of off-diagonal blocks. We will treat these cases separately.

4.1 Blocks on the diagonal

Lemma 4.1 *Let $CC^{j,j}$ be a $2^j \times 2^j$ circulant matrix. Then $CC^{j-1,j-1}$, $CD^{j-1,j-1}$, $DC^{j-1,j-1}$, and $DD^{j-1,j-1}$ defined by (6) to (9), respectively, are circulant matrices.*

Proof: We will prove the lemma for $CD^{j-1,j-1}$ only since the other cases are completely analogous.

By assumption $CC^{j,j}$ is circulant, i.e.

$$CC^{j,j}_{m,n} = cc^{j,j}_{\langle m-n\rangle_{2^j}}$$

where $cc^{j,j}$ is the first column of $CC^{j,j}$. Equation (7) then becomes

$$CD^{j-1,j-1}_{m,n} = \sum_{k=0}^{D-1}\sum_{l=0}^{D-1} a_k b_l \, cc^{j,j}_{\langle\langle k+2m\rangle_{2^j} - \langle l+2n\rangle_{2^j}\rangle_{2^j}} \tag{14}$$

Considering now the index of the typical term of (14) and using lemma A.1 – A.2 derived in appendix A we find that

$$\begin{aligned}\langle\langle k+2m\rangle_{2^j} - \langle l+2n\rangle_{2^j}\rangle_{2^j} &= \langle\langle 2(m-n)\rangle_{2^j} + k - l\rangle_{2^j}\\ &= \langle 2\langle m-n\rangle_{2^{j-1}} + k - l\rangle_{2^j}\end{aligned}$$

This expression does not depend on the individual values of m and n but only on their *difference*. Therefore $CD^{j-1,j-1}_{m,n}$ depends only on $\langle m-n\rangle_{2^{j-1}}$ which proves that $CD^{j-1,j-1}$ is circulant. Hence $CD^{j-1,j-1}_{m,n} = cd^{j-1,j-1}_{\langle m-n\rangle_{2^{j-1}}}$ for some column vector $cd^{j-1,j-1}$. □

Having established that $CD^{j-1,j-1}$ is circulant, we can now give a formula for the vector $cd^{j-1,j-1}$. Putting $n = 0$ in equation (14) and using lemma A.1 we obtain the first column of $CD^{j-1,j-1}$:

$$\begin{aligned}cd^{j-1,j-1}_m = CD^{j-1,j-1}_{m,0} &= \sum_{k=0}^{D-1}\sum_{l=0}^{D-1} a_k b_l \, cc^{jj}_{\langle\langle 2m+k\rangle_{2^j} - \langle l\rangle_{2^j}\rangle_{2^j}}\\ &= \sum_{k=0}^{D-1}\sum_{l=0}^{D-1} a_k b_l \, cc^{jj}_{\langle 2m+k-l\rangle_{2^j}}\end{aligned} \tag{15}$$

where $m = 0, 1, \dots, 2^{j-1} - 1$. Rearranging the terms in (15) so that we sum over the differences in the indices and using (3) we can derive a computationally more efficient expression for $cd^{j-1,j-1}$. More precisely, we split the double sum into terms where $k - l \geq 0$ and terms where $k - l < 0$ and rearrange these separately. Thus let $p = k - l \geq 0$. The terms in (15) with $k - l \geq 0$ are

$$\sum_{k=0}^{D-1}\sum_{l=0}^{k} a_k b_l \, cc^{jj}_{\langle 2m+k-l\rangle_{2^j}} = \sum_{k=0}^{D-1}\sum_{l=0}^{k} a_{l+p} b_l \, cc^{jj}_{\langle 2m+p\rangle_{2^j}} \tag{16}$$

The following table shows p as a function of k and l (for $D = 6$):

$k\backslash l$	0	1	2	3	4	5
0	0					
1	1	0				
2	2	1	0			
3	3	2	1	0		
4	4	3	2	1	0	
5	5	4	3	2	1	0

Summing over the diagonals yields an alternative expression for (16):

$$\sum_{k=0}^{D-1}\sum_{l=0}^{k} a_{l+p}b_l\, cc^{jj}_{\langle 2m+p\rangle_{2^j}} = \sum_{p=0}^{D-1}\left[\sum_{l=0}^{D-1-p} a_{l+p}b_l\right] cc^{jj}_{\langle 2m+p\rangle_{2^j}} \tag{17}$$

Similarly, we take the terms from (15) with $k-l<0$ and set $p=l-k>0$:

$$\sum_{k=0}^{D-1}\sum_{l=k+1}^{D-1} a_k b_{k+p}\, cc^{jj}_{\langle 2m-p\rangle_{2^j}} = \sum_{p=1}^{D-1}\left[\sum_{k=0}^{D-1-p} a_k b_{k+p}\right] cc^{jj}_{\langle 2m-p\rangle_{2^j}} \tag{18}$$

Now let

$$q^p_{ab} = \sum_{k=0}^{D-1-p} a_k b_{k+p},\ p=0,1,\ldots,D-1$$

Then, combining (17) and (18) we can rewrite (15) as

$$cd^{j-1,j-1}_m = \\ q^0_{ab} cc^{jj}_{2m} + \sum_{p=1}^{D-1}\left[q^p_{ab}\, cc^{jj}_{\langle 2m-p\rangle_{2^j}} + q^p_{ba}\, cc^{jj}_{\langle 2m+p\rangle_{2^j}}\right] \tag{19}$$

for $m = 0,1,\ldots,2^{j-1}-1$.

The vectors $cc^{j-1,j-1}$, $dc^{j-1,j-1}$, and $dd^{j-1,j-1}$ are computed similarly but with the filters q^p_{aa}, q^p_{ba}, q^p_{bb}, respectively. The formulas are

$$cc^{j-1,j-1}_m = \\ q^0_{aa} cc^{jj}_{2m} + \sum_{p=1}^{D-1}\left[q^p_{aa}\, cc^{jj}_{\langle 2m-p\rangle_{2^j}} + q^p_{aa}\, cc^{jj}_{\langle 2m+p\rangle_{2^j}}\right] \tag{20}$$

$$dc^{j-1,j-1}_m = \\ q^0_{ba} cc^{jj}_{2m} + \sum_{p=1}^{D-1}\left[q^p_{ba}\, cc^{jj}_{\langle 2m-p\rangle_{2^j}} + q^p_{ab}\, cc^{jj}_{\langle 2m+p\rangle_{2^j}}\right] \tag{21}$$

$$dd^{j-1,j-1}_m = \\ q^0_{bb} cc^{jj}_{2m} + \sum_{p=1}^{D-1}\left[q^p_{bb}\, cc^{jj}_{\langle 2m-p\rangle_{2^j}} + q^p_{bb}\, cc^{jj}_{\langle 2m+p\rangle_{2^j}}\right] \tag{22}$$

It follows from (3) that

$$\begin{aligned} q^0_{aa} = q^0_{bb} &= 1 \\ q^p_{ab} = q^p_{ba} &= 0,\ p \text{ even} \\ q^p_{aa} = q^p_{bb} &= 0,\ p \text{ even}\ , p > 0 \end{aligned}$$

so the computational work in computing (19)–(22) is reduced accordingly.

4.2 Blocks below the diagonal

We now turn to the task of computing the *lower* off-diagonal blocks $DC^{i,j-1}$ and $DD^{i,j-1}$, $i \geq j$ defined by (10) and (11), respectively. The operations differ from those of the diagonal cases by being applied in one dimension only. Therefore blocks tend to become more rectangular which means that they are not necessarily circulant in the ordinary sense. However, as we shall see, the typical block is still represented by a single vector, one which is shifted to match the rectangular shape. The block is then a *column-shift-circulant* matrix in the sense of definition 3.2.

Lemma 4.2 *Let $DC^{i,j}$, $i \geq j$, be a $2^i \times 2^j$ column-shift-circulant matrix. Then $DC^{i,j-1}$ and $DD^{i,j-1}$ defined by (10) and (11), respectively, are column-shift-circulant matrices.*

Proof: We will give the proof for the case of $DC^{i,j}$ only. By assumption $DC^{i,j}$ is column-shift-circulant, i.e.

$$DC^{i,j}_{m,n} = dc^{i,j}_{\langle m - \sigma n\rangle_{2^i}}$$

where $dc^{i,j}$ is the first column of $DC^{i,j}$ and $\sigma = 2^i/2^j$. Equation (10) then becomes

$$\begin{aligned} DC^{i,j-1}_{m,n} &= \sum_{l=0}^{D-1} a_l \, DC^{i,j}_{m,\langle l+2n\rangle_{2^j}} \\ &= \sum_{l=0}^{D-1} a_l \, dc^{i,j}_{\langle m - \sigma\langle l+2n\rangle_{2^j}\rangle_{2^i}} \end{aligned} \tag{23}$$

We consider the index of the typical term of (23) and use lemmas A.2 and A.1 in appendix A to obtain the following:

$$\begin{aligned} \langle m - \sigma\langle l+2n\rangle_{2^j}\rangle_{2^i} &= \langle m - \langle \sigma(l+2n)\rangle_{\sigma 2^j}\rangle_{2^i} \\ &= \langle m - \langle \sigma l + 2\sigma n\rangle_{2^i}\rangle_{2^i} \\ &= \langle m - \sigma l - 2\sigma n\rangle_{2^i} \end{aligned}$$

Therefore

$$DC^{i,j-1}_{m,n} = \sum_{l=0}^{D-1} a_l \, dc^{i,j}_{\langle m - \sigma l - 2\sigma n\rangle_{2^i}} \tag{24}$$

Equation (24) establishes that there exists a vector, $dc^{i,j-1}$ say, such that $DC^{i,j-1}$ has the desired column-shift-circulant form

$$DC^{i,j-1}_{m,n} = dc^{i,j-1}_{\langle m-2\sigma n\rangle_{2^i}}$$

for $m = 0, 1, \ldots, 2^i - 1$ and $n = 0, 1, \ldots, 2^{j-1} - 1$. □

We can now look for an explicit formula for the vector $dc^{i,j-1}$. Taking the first column ($n = 0$) of $DC^{i,j-1}_{m,n}$ in (24) gives the result immediately:

$$dc^{i,j-1}_m = DC^{i,j-1}_{m,0} = \sum_{l=0}^{D-1} a_l\, dc^{i,j}_{\langle m-\sigma l\rangle_{2^i}},\; m = 0, 1, \ldots, 2^i - 1 \tag{25}$$

An analysis similar to the above establishes that $DD^{i,j-1}$ is column-shift-circulant with respect to the vector $dd^{i,j-1}$ given by

$$dd^{i,j-1}_m = DD^{i,j-1}_{m,0} = \sum_{l=0}^{D-1} b_l\, dd^{i,j}_{\langle m-\sigma l\rangle_{2^i}},\; m = 0, 1, \ldots, 2^i - 1 \tag{26}$$

Since the initial block $DC^{j,j}$ is circulant according to lemma 4.1, it is also column-shift-circulant with $\sigma = 2^j/2^j = 1$ and we can use the column vector $dd^{j,j}$ as computed from (21) directly in (25) and (26) for the case $i = j$.

4.3 Blocks above the diagonal

The *upper* off-diagonal blocks $CD^{i-1,j}$ and $DD^{i-1,j}$, $i \leq j$, are computed according to (12) and (13). The situation is completely analogous to (25) and (26) but the blocks are now *row-shift-circulant* matrices represented by row vectors $cd^{i-1,j}$ and $dd^{i-1,j}$, respectively, as stated by lemma 4.3.

Lemma 4.3 *Let $CD^{i,j}$, $j \geq i$, be a $2^i \times 2^j$ row-shift-circulant matrix. Then $CD^{i-1,j}$ and $DD^{i-1,j}$ defined by (12) and (13), respectively, are row-shift-circulant matrices.*

Proof: We will give the proof for the case of $CD^{i,j}$ only. The proof is completely similar to that of lemma 4.3, but the assumption is now that $CD^{i,j}$ is row-shift-circulant, i.e.

$$CD^{i,j}_{m,n} = cd^{i,j}_{\langle n-\sigma m\rangle_{2^j}}$$

where $cd^{i,j}$ is the first row of $CD^{i,j}$ and $\sigma = 2^j/2^i$. Equation (12) then becomes

$$CD^{i-1,j}_{m,n} = \sum_{k=0}^{D-1} a_k\, CD^{i,j}_{\langle k+2m\rangle_{2^i},n}$$

$$= \sum_{k=0}^{D-1} a_k \, cd^{i,j}_{\langle n-\sigma\langle k+2m\rangle_{2^i}\rangle_{2^j}}$$
$$= \sum_{k=0}^{D-1} a_k \, cd^{i,j}_{\langle n-\sigma k-2\sigma m\rangle_{2^j}}$$

Therefore $CD^{i-1,j}$ has the desired row-shift-circulant form

$$CD^{i-1,j}_{m,n} = cd^{i-1,j}_{\langle n-2\sigma m\rangle_{2^j}}$$

for $m = 0, 1, \ldots, 2^{i-1} - 1$ and $n = 0, 1, \ldots, 2^j - 1$. □

The formulas for the row vectors $cd^{i-1,j}$ and $dd^{i-1,j}$ follow by taking the first *rows* ($m = 0$) of $CD^{i-1,j}_{m,n}$ and $DD^{i-1,j}_{m,n}$, respectively:

$$cd^{i-1,j}_n = CC^{i-1,j}_{0,n} = \sum_{k=0}^{D-1} a_k \, cd^{i,j}_{\langle n-\sigma k\rangle_{2^j}}, \; n = 0, 1, \ldots, 2^j - 1 \tag{27}$$

$$dd^{i-1,j}_n = DD^{i-1,j}_{0,n} = \sum_{k=0}^{D-1} b_k \, dd^{i,j}_{\langle n-\sigma k\rangle_{2^j}} \; n = 0, 1, \ldots, 2^j - 1 \tag{28}$$

However, one minor issue remains to be dealt with before we have a viable algorithm: While the initial blocks $CD^{j,j}$ computed according to (7) are circulant and therefore also row-shift-circulant (with $\sigma = 1$), they are represented by column vectors when computed according to equation (19). Therefore we must modify each $cd^{j,j}$ so that it represents the first row of $CD^{j,j}$ instead of the first column.

From definition 3.1 of a circulant matrix we have that

$$CD^{j,j}_{m,n} = cd^{j,j}_{\langle m-n\rangle_{2^j}}, \;\; n = 0, 1, \ldots, 2^j - 1$$

where $cd^{j,j}$ is the first column of $CD^{j,j}$. Putting $m = 0$ then yields the first row:

$$CD^{j,j}_{0,n} = cd^{j,j}_{\langle -n\rangle_{2^j}}, \;\; n = 0, 1, \ldots, 2^j - 1$$

To obtain a row representation for $CD^{j,j}$ we can therefore take the result from equation (19) and convert it as follows

$$cd^{j,j}_n := cd^{j,j}_{\langle -n\rangle_{2^j}}$$

Alternatively, we can modify equation (19) to produce the row vector directly:

$$cd^{j-1,j-1}_n = q^0_{ab} cc^{jj}_{-2n} + \sum_{p=1}^{D-1} \left[q^p_{ab} \, cc^{jj}_{\langle -2n-p\rangle_{2^j}} + q^p_{ba} \, cc^{jj}_{\langle -2n+p\rangle_{2^j}} \right] \tag{29}$$

for $n = 0, 1, \ldots, 2^{j-1} - 1$.

Note that the equations for blocks above the diagonal ((27) and (28)) and equations for blocks below the diagonal ((25) and (26)) now can be computed with the same algorithm.

4.4 The algorithm

We will now state an algorithm for the 2D wavelet transform of a circulant matrix A:

Let cpwt1 be a function that implements 2D splittings of blocks on the diagonal according to equations (20), (21), (22), and (29):

$$[cc^{j-1,j-1}, cd^{j-1,j-1}, dc^{j-1,j-1}, dd^{j-1,j-1}] = \mathsf{cpwt1}(cc^{j,j})$$

Let cpwt2 be a function that implements 1D splittings of the form described in equations (25) and (26). This function can also be used for computations of (27) and (28) as mentioned above.

$$\begin{aligned} [dc^{i,j-1}, dd^{i,j-1}] &= \mathsf{cpwt2}(dc^{i,j}),\ i \geq j \\ [cd^{i-1,j}, dd^{i-1,j}] &= \mathsf{cpwt2}(cd^{i,j}),\ j \geq i \end{aligned}$$

With these functions the example shown in figure 3 can the be computed as follows: Let $cc^{4,4}$ be the column vector representing the initial circulant matrix $CC^{4,4}$. Then

$$\begin{aligned} [cc^{3,3}, cd^{3,3}, dc^{3,3}, dd^{3,3}] &= \mathsf{cpwt1}(cc^{4,4}) \\ \\ [cc^{2,2}, cd^{2,2}, dc^{2,2}, dd^{2,2}] &= \mathsf{cpwt1}(cc^{3,3}) \\ [dc^{3,2}, dd^{3,2}] &= \mathsf{cpwt2}(dc^{3,3}) \\ [cd^{2,3}, dd^{2,3}] &= \mathsf{cpwt2}(cd^{3,3}) \\ \\ [cc^{1,1}, cd^{1,1}, dc^{1,1}, dd^{1,1}] &= \mathsf{cpwt1}(cc^{2,2}) \\ [dc^{2,1}, dd^{2,1}] &= \mathsf{cpwt2}(dc^{2,2}) \\ [cd^{1,2}, dd^{1,2}] &= \mathsf{cpwt2}(cd^{2,2}) \\ [dc^{3,1}, dd^{3,1}] &= \mathsf{cpwt2}(dc^{3,2}) \\ [cd^{1,3}, dd^{1,3}] &= \mathsf{cpwt2}(cd^{2,3}) \end{aligned}$$

In general, the algorithm is

For $j = J$ downto $J - \lambda + 1$
  $[cc^{j-1,j-1}, cd^{j-1,j-1}, dc^{j-1,j-1}, dd^{j-1,j-1}] = \mathsf{cpwt1}(cc^{j,j})$
  For $i = j$ to $J - 1$
    $[dc^{i,j-1}, dd^{i,j-1}] = \mathsf{cpwt2}(dc^{i,j})$
    $[cd^{j-1,i}, dd^{j-1,i}] = \mathsf{cpwt2}(cd^{j,i})$
  end
end

This algorithm describes the process of computing the 2D wavelet transform as described in the previous section. However, it does not distinguish among vectors that should be kept in the final result and vectors that are merely intermediate stages of the transform. The vectors $dc^{i,j}$ and $cd^{j,i}$, for example, are part of the final result for $j = J - \lambda$ only, so all other vectors can be discarded at some point.

We would like to have an algorithm that makes explicit use of a fixed storage area and that does not store unnecessary information. Therefore, we will introduce a modified notation that is suitable for such an algorithm and also convenient for the matrix-vector multiplication which is described in section 6.

4.5 A data structure for the 2D wavelet transform

As described in section 2.2 the result H of a 2D wavelet transform is a block matrix with a characteristic block structure (see figure 3). We now introduce a new notation for these blocks as follows

$$H^{i,j} = \begin{cases} CC^{J-\lambda,J-\lambda} & \text{for } i,j = 0 \\ CD^{J-\lambda,J-1-\lambda+j} & \text{for } i = 0, 1 \leq j \leq \lambda \\ DC^{J-1-\lambda+i,J-\lambda} & \text{for } 1 \leq i \leq \lambda, j = 0 \\ DD^{J-1-\lambda+i,J-1-\lambda+j} & \text{for } 1 \leq i,j \leq \lambda \end{cases} \tag{30}$$

This is illustrated in figure 7 for the case $\lambda = 3$.

We can now use indices from 0 to λ to index the blocks of H. It follows from this definition that each block $H^{i,j}$ is an N^i by N^j matrix with

$$N^k = \begin{cases} 2^{J-\lambda} & = & N/2^{\lambda} & \text{for } k = 0 \\ 2^{J-1-\lambda+k} & = & N/2^{\lambda+1-k} & \text{for } 1 \leq k \leq \lambda \end{cases} \tag{31}$$

Because all the blocks $H^{i,j}$ are either circulant or shift-circulant, we can represent them as vectors $h^{i,j}$ with

$$h^{i,j}_m = \begin{cases} H^{i,j}_{m,0} & \text{for } i \geq j \\ H^{i,j}_{0,m} & \text{for } i < j \end{cases} \tag{32}$$

It follows from (31) that the length of $h^{i,j}$ is

$$\begin{cases} N^i & \text{for } i \geq j \\ N^j & \text{for } i < j \end{cases} \tag{33}$$

and the shift parameter σ is now given according to definition 3.2 as

$$\sigma = \begin{cases} N^i/N^j & \text{for } i \geq j \\ N^j/N^i & \text{for } i < j \end{cases}$$

Equation (32) suggests a data structure consisting of the vector variables $h^{i,j}$ which refer to the actual arrays representing the final stage of the wavelet transform

H^{00}	H^{01}	H^{02}	H^{03}
H^{10}	H^{11}	H^{12}	H^{13}
H^{20}	H^{21}	H^{22}	H^{23}
H^{30}	H^{31}	H^{32}	H^{33}

Figure 7: The new notation for the block structure of H (for $\lambda = 3$) as defined by (30). Compare with H in figure 3.

(e.g. $dd^{J-1-\lambda-i,J-1-\lambda-j}$). This structure can also be used to store the intermediate vectors from the recursion if we allow the variables $h^{i,j}$ to assume different values (and different lengths) during the computation – for example by using pointer variables. This is demonstrated in figure 8.

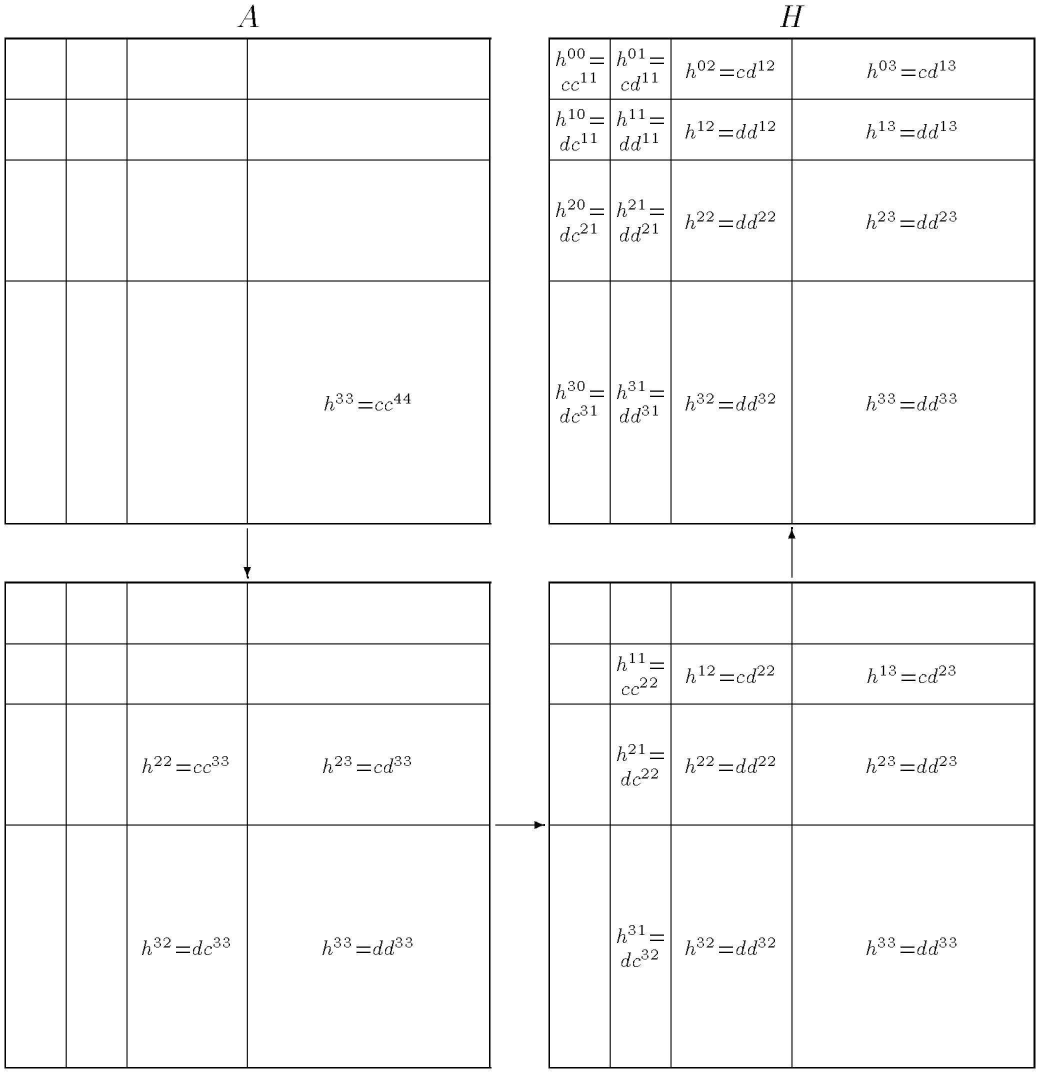

Figure 8: The use of (pointer) variables $h^{i,j}$ to implement the 2D wavelet transform of a circulant matrix. Intermediate results are referenced by variables $h^{i,j}$ before they attain their final values. For example, $h^{3,2}$ stores first $dc^{3,3}$ which is then split further into $dc^{3,2}$ and $dd^{3,2}$. Then $dd^{3,2}$ is stored in $h^{3,2}$ overwriting $dc^{3,3}$. Again, $J = 4$ and $\lambda = 3$.

From figure 8 we arrive at the final formulation of the algorithm in section 4.4.

Algorithm 1 Circulant 2D wavelet transform

For $j = \lambda$ downto 1
    $[h^{j-1,j-1}, h^{j-1,j}, h^{j,j-1}, h^{j,j}] =$ cpwt1$(h^{j,j})$
    For $i = j+1$ to λ
        $[h^{i,j-1}, h^{i,j}] =$ cpwt2$(h^{i,j})$
        $[h^{j-1,i}, h^{j,i}] =$ cpwt2$(h^{j,i})$
    end
end

4.6 Storage

In this section we will investigate the storage requirement as a function of the transform level λ when the proposed structure is used. We will relax the requirement that N is a power of two and express any given N as

$$N = K2^{\lambda_{max}}$$

where $K \in \mathbf{Z}$ and λ_{max} is the largest possible integer. The valid values of λ are thus

$$\lambda = 0, 1, \ldots, \lambda_{max} \leq \log_2 N$$

Lemma 4.4 *The number of elements needed to represent the result of λ steps of the wavelet transform of a circulant $N \times N$ matrix as computed by algorithm 1 is*

$$S_N(\lambda) = N\left(1 + \sum_{k=1}^{\lambda} \frac{k}{2^{\lambda-k}}\right), \ \lambda = 0, 1, \ldots, \lambda_{max} \tag{34}$$

Proof: We will prove (34) by induction:
Induction start: For $\lambda = 0$ there is only one block, namely CC^{JJ} which is represented by the vector cc^{JJ} of length N. Thus, $S_N(0) = N$ and (34) therefore holds for $\lambda = 0$.

Induction step: The induction hypothesis is

$$S_N(\lambda - 1) = N\left(1 + \sum_{k=1}^{\lambda-1} \frac{k}{2^{\lambda-1-k}}\right) \tag{35}$$

The storage differences between a wavelet transform of level $\lambda - 1$ and λ lie in the uppermost and leftmost blocks of the structure as indicated in figure 9. The rest is unaffected.

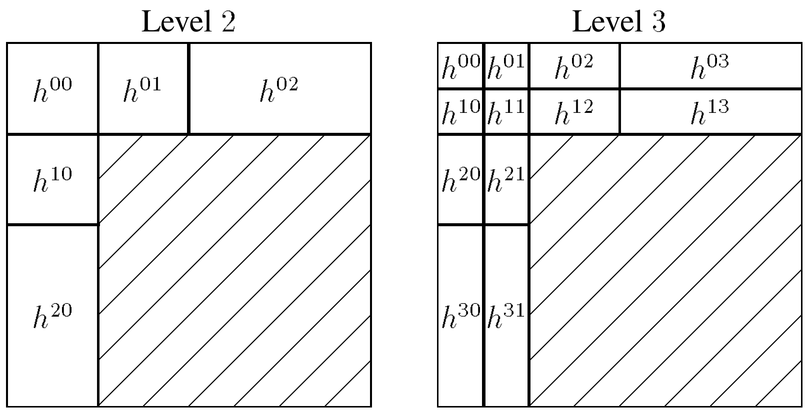

Figure 9: The differences between a wavelet transform of level $\lambda - 1$ and λ (shown for $\lambda = 3$).

Recall that each block is represented by a vector. We will subtract the number of elements in these vectors at level $\lambda - 1$ from (35) and then add the elements from vectors resulting from a level λ transform. The vector lengths for the example $\lambda = 3$ are shown in the following table:

Level 2		Level 3	
Vector	Length	Vector	Length
h^{00}	$N/4$	h^{00}	$N/8$
		h^{01}, h^{10}, h^{11}	$N/8$
h^{01}, h^{10}	$N/4$	$h^{02}, h^{12}, h^{20}, h^{21}$	$N/4$
h^{02}, h^{20}	$N/2$	$h^{03}, h^{13}, h^{30}, h^{31}$	$N/2$

According to (33) the vector h^{00} at level $\lambda - 1$ has the length

$$N/2^{\lambda-1} \tag{36}$$

This is replaced by 4 new vectors ($h^{00}, h^{01}, h^{10}, h^{11}$) at level λ each having the length $N/2^{\lambda}$. We write this number as

$$4N\frac{1}{2^{\lambda}} = N\left(\frac{1}{2^{\lambda-1}} + \frac{1}{2^{\lambda-1}}\right) \tag{37}$$

The blocks above the diagonal that are replaced are those in the upper row. They are represented by a vector of length $N/2$, a vector of length $N/4$ down to a vector of length $N/2^{\lambda-1}$. The number of elements in vectors corresponding to *column* blocks is the same. Hence the number of elements we must subtract from (35) is

$$2\sum_{k=1}^{\lambda-1}\frac{N}{2^k} = 2\sum_{k=1}^{\lambda-1}\frac{N}{2^{\lambda-k}} = \sum_{k=1}^{\lambda-1}\frac{N}{2^{\lambda-1-k}} \tag{38}$$

The splitting of these off-diagonal blocks will create twice as many blocks so the number of elements introduced will be

$$2\sum_{k=1}^{\lambda-1}\frac{N}{2^{\lambda-1-k}} = 2\sum_{k=2}^{\lambda}\frac{N}{2^{\lambda-k}} \tag{39}$$

Now we subtract from (35) the total number of elements that are transformed ((36) and (38)) and add the number of elements introduced ((37) and (39)), i.e.

$$\begin{aligned}
S_N(\lambda) &= S_N(\lambda-1) - \frac{N}{2^{\lambda-1}} - \sum_{k=1}^{\lambda-1}\frac{N}{2^{\lambda-1-k}} + N\left(\frac{1}{2^{\lambda-1}} + \frac{1}{2^{\lambda-1}}\right) + 2\sum_{k=2}^{\lambda}\frac{N}{2^{\lambda-k}} \\
&= N\left(1 + \sum_{k=1}^{\lambda-1}\frac{k-1}{2^{\lambda-1-k}} + \frac{1}{2^{\lambda-1}} + 2\sum_{k=2}^{\lambda}\frac{1}{2^{\lambda-k}}\right) \\
&= N\left(1 + \sum_{k=2}^{\lambda}\frac{k-2}{2^{\lambda-k}} + \frac{1}{2^{\lambda-1}} + 2\sum_{k=2}^{\lambda}\frac{1}{2^{\lambda-k}}\right) \\
&= N\left(1 + \sum_{k=2}^{\lambda}\frac{k}{2^{\lambda-k}} + \frac{1}{2^{\lambda-1}}\right) \\
&= N\left(1 + \sum_{k=1}^{\lambda}\frac{k}{2^{\lambda-k}}\right)
\end{aligned}$$

which is the desired formula. □

We consider now the sum in (34) and define

$$f(\lambda) = \sum_{k=1}^{\lambda}\frac{k}{2^{\lambda-k}},\ \lambda = 0, 1, 2, \ldots$$

With a change of variables we can rewrite f as

$$f(\lambda) = \sum_{k=0}^{\lambda-1}\frac{\lambda-k}{2^k}$$

This is an **arithmetic-geometric series** which has the closed form [Spi93, p. 107]

$$\begin{aligned}
f(\lambda) &= \frac{\lambda(1-(1/2)^{\lambda})}{1-1/2} - \frac{1-\lambda(1/2)^{\lambda-1}+(\lambda-1)(1/2)^{\lambda}}{2(1-1/2)^2} \\
&= 2\lambda(1-(1/2)^{\lambda}) - 2(1-\lambda(1/2)^{\lambda-1}+(\lambda-1)(1/2)^{\lambda}) \\
&= 2\lambda - 2 - 4\lambda(1/2)^{\lambda} + 2\lambda(1/2)^{\lambda-1} + 2(1/2)^{\lambda} \\
&= 2\lambda - 2 + 2(1/2)^{\lambda}
\end{aligned}$$

We observe that $f(\lambda)$ behaves asymptotically as $2\lambda - 2$.
Figure 10 shows a plot of $f(\lambda)$ together with its asymptote.

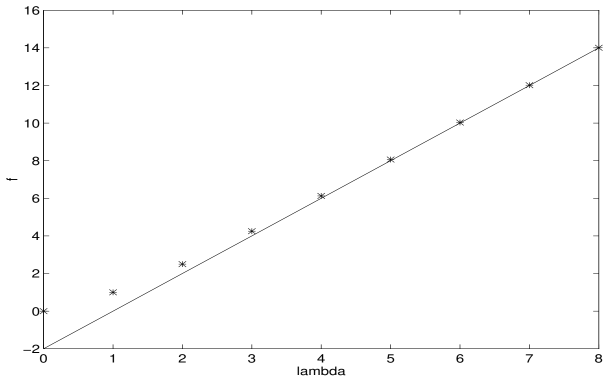

Figure 10: The function $f(\lambda)$ behaves asymptotically like $2\lambda - 2$.

We can now express the storage requirement as

$$\begin{aligned} S_N(\lambda) &= N(1 + f(\lambda)) \\ &= N(2\lambda - 1 + 2(1/2)^{\lambda}),\ \lambda = 0, 1, \ldots, \lambda_{max} \end{aligned}$$

Asymptotically we have for large λ

$$S_N(\lambda) \approx N(2\lambda - 1)$$

5 2D wavelet transform of a circulant, banded matrix

An important special case of a circulant matrix is that A is a *banded* circulant matrix. Here each column of A consists of a piece which is zero and a piece which is regarded as non-zero. In certain columns this non-zero part is wrapped around. Consequently, it is sufficient to store only the non-zero part of the first column along with an index δ determining how it must be aligned in the first column relative to the full-length vector. The length of the non-zero part is called the *bandwidth* of A, and we will denote it by L.

It turns out that each block of the 2D wavelet transform is also banded, so the vector representing it need only include the non-zero part. Therefore the storage requirements can be considerably less than that given by (34). An example of this structure is given in figure 11.

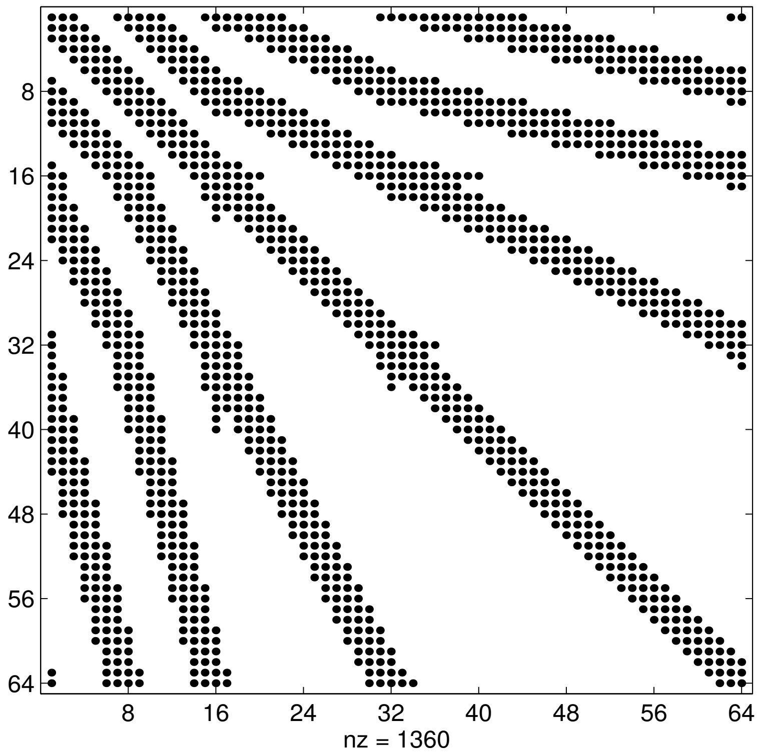

Figure 11: The structure of a wavelet transform of a 64×64 circulant banded matrix. Here $L = 3$, $D = 4$ and $\lambda = 3$. This is the same block structure as in figure 7.

The block $H^{3,1}$, say, has the general structure

$$H^{3,1} = \begin{bmatrix} v_2^{3,1} & & & \\ v_3^{3,1} & & & \\ v_4^{3,1} & v_0^{3,1} & & \\ & v_1^{3,1} & & \\ & v_2^{3,1} & & \\ & v_3^{3,1} & & \\ & v_4^{3,1} & v_0^{3,1} & \\ & & v_1^{3,1} & \\ & & v_2^{3,1} & \\ & & v_3^{3,1} & \\ & & v_4^{3,1} & v_0^{3,1} \\ & & & v_1^{3,1} \\ & & & v_2^{3,1} \\ & & & v_3^{3,1} \\ v_0^{3,1} & & & v_4^{3,1} \\ v_1^{3,1} & & & \end{bmatrix} \tag{40}$$

For each block, we know the non-zero values of the first column (or row in the case of blocks above the diagonal) represented by the vector $v^{i,j} = (v_0^{i,j}, v_1^{i,j}, \ldots, v_{L-1}^{i,j})$, the amount by which it is shifted (σ) and how it is aligned with respect to the first element of the block (δ). In (40), for example, we have $\delta = 3$, $N^i = 16$, $N^j = 4$, $\sigma = 4$ (from definition 3.2), and $L^{i,j} = 5$.

If the matrix is not banded we have the special case $v^{i,j} = h^{i,j}$ and $\delta = 1$ so (34) applies exactly.

5.1 Calculation of bandwidths

Given the bandwidth L of the original matrix A it is possible to derive a formula for the bandwidths of each block of the wavelet transform.

Let $L^{i,j}$ be the bandwidth of block $H^{i,j}$. We can then use the recurrence formulas for the 2D wavelet transform to obtain the desired formulas.

Blocks on the diagonal

We start with the blocks on the diagonal given by equation (6) to (9) and consider again only the cd block as the typical case. Let us recall (19):

$$cd_m^{j-1,j-1} = \\ q_{ab}^0 cc_{2m}^{j,j} + \sum_{p=1}^{D-1} \left[q_{ab}^p \, cc_{\langle 2m-p \rangle_{2^j}}^{j,j} + q_{ba}^p \, cc_{\langle 2m+p \rangle_{2^j}}^{j,j} \right]$$

for $m = 0, 1, \ldots, 2^{j-1} - 1$. Assume that $cc^{j,j}$ is banded, i.e. zero outside a band of length $L^{j,j}$ as shown in figure 12. Then we can use the recurrence formula to compute the bandwidth of $cd^{j-1,j-1}$. Without loss of generality we may assume that the nonzero part is wholly contained in the vector; i.e. there is no wrap-around.

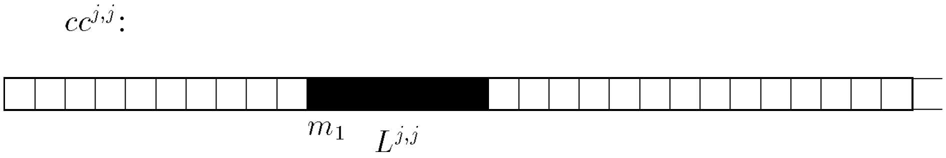

Figure 12: The band of $c^{j,j}$ has length $L^{j,j}$ and starts at index m_1.

Since $cc^{j,j}$ is zero outside the interval starting at m_1 of length $L^{j,j}$, we see from (19) that $cc^{j-1,j-1}$ will be zero only if $2m - p \geq m_1 + L^{j,j}$ and $2m + p < m_1$ for all $p \in [0, D-1]$. This leads immediately to the inequalities

$$\begin{aligned} 2m + (D-1) &< m_1 \\ 2m - (D-1) &\geq m_1 + L^{j,j} \end{aligned}$$

or

$$(m_1 + L^{j,j} + D - 1)/2 \leq m < (m_1 - D + 1)/2$$

The length of this interval is then the bandwidth of $cd^{j-1,j-1}$:

$$\begin{aligned} L^{j-1,j-1} &= \frac{m_1 + L^{jj} + D - 1}{2} - \frac{m_1 - D + 1}{2} \\ &= \frac{L^{jj}}{2} + D - 1 \end{aligned} \tag{41}$$

However, since the bandwidth is an integer the fraction must be rounded either up or down if $L^{j,j}$ is odd. Which of these operations to choose depends on m_1 as illustrated in figure 13.

We have chosen always to round upwards because it ensures that we get an upper bound for the bandwidth. Thus the formula becomes

$$L^{j-1,j-1} = \left\lceil \frac{L^{j,j}}{2} \right\rceil + D - 1 \tag{42}$$

We observe that equation (42) has two fixed points, namely

$$L^{j,j} = \begin{cases} 2D - 2 \\ 2D - 1 \end{cases}$$

and it turns out that there is convergence to one of these values depending on whether the initial $L^{j,j}$ is smaller than $2D-2$ or larger than $2D-1$. However, the important fact is that these fixed points are more related to the wavelet genus D than to the original bandwidth L.

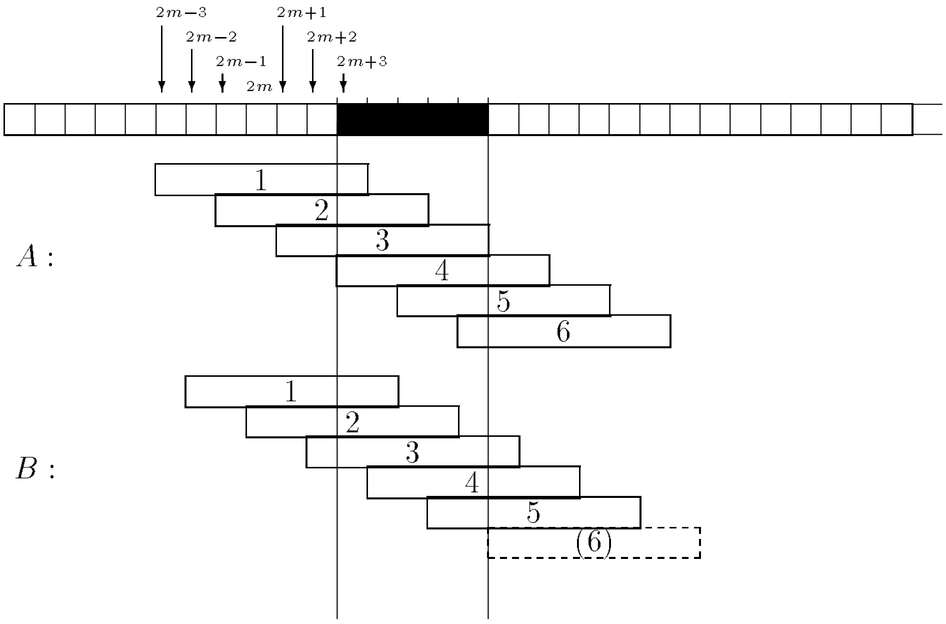

Figure 13: The computation in equation (19) can be viewed as a sliding filter of length $2D-1$ applied to the band of $cc^{j,j}$. The resulting bandwidth depends on how the initial bandwidth is aligned with this sliding filter. In this example $L^{j,j}=5$, $D=4$ so $L^{j-1,j-1}$ is either 5 (case B) or 6 (case A) depending on how the convolutions happen to align with the non-zero block. Thus case A corresponds to rounding up and case B corresponds to rounding down in equation (41).

Blocks below and above the diagonal

The bandwidths of the blocks below the diagonal are found from the recurrence formula

$$dc_m^{i,j-1} = \sum_{l=0}^{D-1} a_l dc^{i,j}_{\langle m-\sigma l\rangle_{2^i}}$$

Again, we disregard wrapping and proceeding as in the case of $cd^{j-1,j-1}$ above we find that $dc^{i,j-1}$ is zero only if $m-\sigma l \geq m_1 + L^{i,j}$ or $m - \sigma l < m_1$ for all $l \in [0, D-1]$. This leads to the inequalities

$$\begin{aligned} m &< m_1 \\ m - \sigma(D-1) &\geq m_1 + L^{i,j} \end{aligned}$$

or $m_1 + L^{i,j} + \sigma(D-1) \leq m < m_1$. The interval length is consequently

$$\begin{aligned} L^{i,j-1} &= m_1 + L^{i,j} + \sigma(D-1) - m_1 \\ &= L^{i,j} + \sigma(D-1) \end{aligned}$$

Performing similar computations for blocks above the diagonal yields the result

$$L^{i,j-1} = L^{j-1,i} = L^{i,j} + \sigma(D-1) \tag{43}$$

Since σ is the ratio between N^i and N^j, equation (43) shows that the bandwidth grows exponentially as the difference between i and j increases. Figures 14 and 15 show examples of the bandwidths for different transform levels.

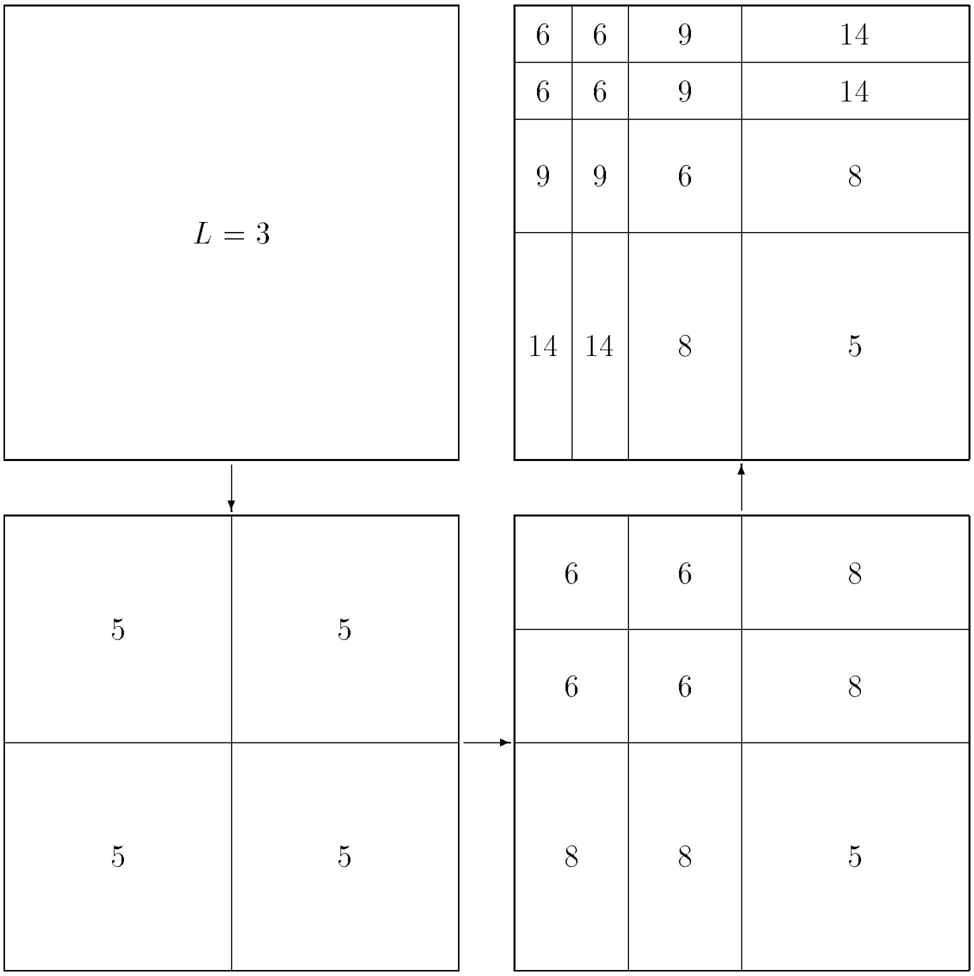

Figure 14: The bandwidths for transform levels $\lambda = 0, 1, 2, 3$. The initial bandwidth is 3 and $D = 4$. These bandwidths are upper bounds since actual bandwidths may be one the less than the predicted values. This can be seen in figure 11.

6	6	9	15	26
6	6	9	15	26
9	9	6	9	14
15	15	9	6	8
26	26	14	8	5

Figure 15: The bandwidths for transform level $\lambda = 4$.

6 Matrix-vector multiplication in a wavelet basis

We now turn to the problem of computing the matrix-vector product $y = Hx$. This system has the form shown in figure 16.

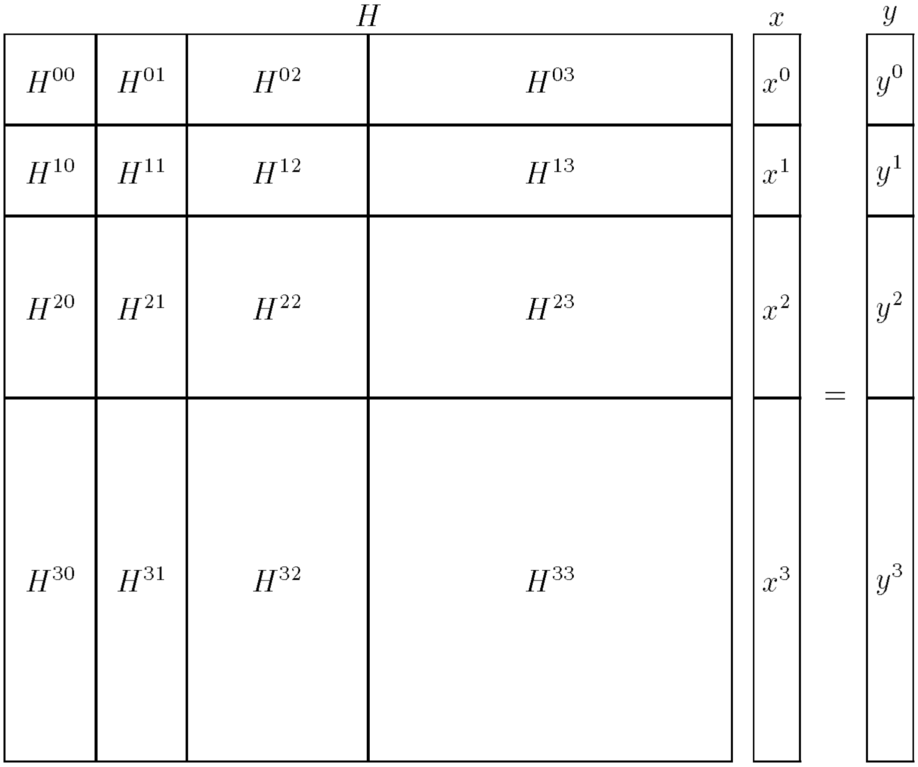

Figure 16: The structure of $y = Hx$ for $\lambda = 3$.

The vector y may be computed block-wise as follows

$$y^i = \sum_{j=0}^{\lambda} H^{i,j} x^j, \quad i = 0, 1, \ldots, \lambda \tag{44}$$

where λ is the depth of the wavelet transform. The symbols x^j, y^i, and $H^{i,j}$ denote the different *blocks* of the x, y, and H as indicated in figure 7. The computation in (44) is thus broken down into the tasks of computing the products

$$y^{i,j} = H^{i,j} x^j, \quad i, j = 0, 1, \ldots, \lambda - 1 \tag{45}$$

In the following we distinguish between blocks on or below the diagonal ($i \geq j$) and blocks above the diagonal ($i < j$).

6.1 Blocks on or below the diagonal

Let $v^{i,j}$, $i \geq j$, be the vector of length $L^{i,j}$ representing the nonzero part of the first column of $H^{i,j}$, i.e.

$$h_m^{i,j} = \begin{cases} v_{\langle m+\delta-1\rangle_{N^i}}^{i,j} & \text{for } \langle m+\delta-1\rangle_{N^i} \in [0, L^{i,j}-1] \\ 0 & \text{otherwise} \end{cases}$$

and

$$H_{m,n}^{i,j} = h_{\langle m-\sigma n\rangle_{N^i}}^{i,j}$$

where $m = 0, 1, \ldots, N^i$, $\sigma = N^i/N^j$, and δ is the offset relative to the upper left element. From equation (45) we see that the typical element of $y^{i,j}$ can be computed column wise as

$$\begin{aligned} y_m^{i,j} &= \sum_{n=0}^{N^j-1} H_{m,n}^{i,j} x_n^j \\ &= \sum_{n=0}^{N^j-1} h_{\langle m-\sigma n\rangle_{N^i}}^{i,j} x_n^j \\ &= \sum_{n=0}^{N^j-1} v_{\langle m-\sigma n+\delta-1\rangle_{N^i}}^{i,j} x_n^j \end{aligned} \tag{46}$$

For each n this computation is only valid for those $m \in [0, N^i-1]$ where $v_{\langle m-\sigma n+\delta-1\rangle_{N^i}}^{i,j}$ is defined, namely where

$$0 \leq \langle m - \sigma n + \delta - 1\rangle_{N^i} \leq L^{i,j} - 1 \tag{47}$$

Let k and l be defined such that $k \leq m \leq l$ whenever (47) is true. Then we can find the first row k from the requirement

$$\begin{aligned} \langle k - \sigma n + \delta - 1\rangle_{N^i} &= 0 \Leftrightarrow \\ k &= \langle \sigma n - \delta + 1\rangle_{N^i} \end{aligned}$$

and the last row as $l = \langle k + L - 1\rangle_{N^i}$. Letting

$$y_{k:l}^{i,j} = (y_k^{i,j}, y_{k+1}^{i,j}, \ldots, y_l^{i,j})$$

then we can write the computation (46) compactly as

$$y_{k:l}^{i,j} = y_{k:l}^{i,j} + x_n^j\, v_{0:L^{i,j}-1}^{i,j}, \quad n = 0, 1, \ldots, N^j - 1 \tag{48}$$

When $k > l$ the band is wrapped and (48) must be modified accordingly.

If the vector x is a wavelet spectrum then many of its elements are normally close to zero (see e.g. [CP96]). Therefore we will design the algorithm to disregard computations involving elements in x^j where

$$\left|x_n^j\right| < \varepsilon$$

The algorithm is given below

Algorithm 2 Block multiplication ($i \geq j$)

For $n = 0$ to $N - 1$
    if $|x_n^j| > \varepsilon$ then
        $k = \langle \sigma n - \delta + 1 \rangle_{N^i}$
        $l = \langle k + L - 1 \rangle_{N^i}$
        if $k < l$ then
            $y_{k:l}^{i,j} = y_{k:l}^{i,j} + x_n^j v^{i,j}$
        else (wrap)
            $y_{0:l}^{i,j} = y_{0:l}^{i,j} + x_n^j v_{L^{i,j}-l:L-1}^{i,j}$
            $y_{k:N^i-1}^{i,j} = y_{k:N^i-1}^{i,j} + x_n v_{0:L^{i,j}-l-1}^{i,j}$
        end
    end
end

6.2 Blocks above the diagonal

Let $v^{i,j}$, $i < j$, be the vector of length $L^{i,j}$ representing the nonzero part of the first *row* of $H^{i,j}$, i.e.

$$h_n^{i,j} = \begin{cases} v_{\langle n+\delta-1\rangle_{N^j}}^{i,j} & \text{for } \langle n + \delta - 1 \rangle_{N^j} \in [0, L^{i,j} - 1] \\ 0 & \text{otherwise} \end{cases} \tag{49}$$

where

$$H_{m,n}^{i,j} = h_{\langle n-\sigma m \rangle_{N^j}}^{i,j} \tag{50}$$

with $\sigma = N^j / N^i$. An example (shown for H^{13}) is

$$\begin{bmatrix} v_2 v_3 & v_4 v_5 v_6 v_7 & v_8 v_9 v_{10} 0 & & & & & & v_0 v_1 \\ & v_0 v_1 v_2 v_3 & v_4 v_5 v_6 v_7 & v_8 v_9 v_{10} 0 & & & & & \\ & & v_0 v_1 v_2 v_3 & v_4 v_5 v_6 v_7 & v_8 v_9 v_{10} 0 & & & & \\ & & & v_0 v_1 v_2 v_3 & v_4 v_5 v_6 v_7 & v_8 v_9 v_{10} 0 & & & \\ & & & & v_0 v_1 v_2 v_3 & v_4 v_5 v_6 v_7 & v_8 v_9 v_{10} 0 & & \\ & & & & & v_0 v_1 v_2 v_3 & v_4 v_5 v_6 v_7 & v_8 v_9 v_{10} 0 & \\ v_{10} 0 & & & & & & v_0 v_1 v_2 v_3 & v_4 v_5 v_6 v_7 & v_8 v_9 \\ v_6 v_7 & v_8 v_9 v_{10} 0 & & & & & & v_0 v_1 v_2 v_3 & v_4 v_5 \end{bmatrix} \tag{51}$$

Here $\sigma = 4, \delta = 3$, $N^i = 8$, $N^j = 32$, $L = 12$ (padded with one zero). The superscripts i, j has been dropped in this example.

In order to be able to disregard small elements in x as in the previous section, we would like to convert the row representation to a column oriented format. As indicated in the example above, the block $H^{i,j}$ can be characterized completely by σ column vectors each with length $L^{i,j}/\sigma$ together with some book-keeping information. We will assume that $L^{i,j}$ is a multiple of σ, possibly obtained through padding with zeros as suggested in the example above.

Now we choose these vectors from the columns

$$L^{i,j}-\delta, L^{i,j}-\delta-1, \ldots, L^{i,j}-\delta-\sigma+1$$

which are the σ last columns where the top row is non-zero. Let these column vectors be denoted $Z^{i,j}_{:,d}$ for $d=0,1,\ldots,\sigma-1$. From (49) and (50) we then get

$$\begin{aligned} Z^{i,j}_{m,d} &= H^{i,j}_{m,L^{i,j}-\delta-d} \\ &= h^{i,j}_{\langle L^{i,j}-\delta-d-\sigma m\rangle_{N^j}} \\ &= v^{i,j}_{\langle L^{i,j}-\delta-d-\sigma m+\delta-1\rangle_{N^j}} \\ &= v^{i,j}_{\langle L^{i,j}-\sigma m-d-1\rangle_{N^j}} \end{aligned} \tag{52}$$

for $m=0,1,\ldots,L^{i,j}/\sigma-1$ and $d=0,1,\ldots,\sigma-1$. In the example above we have

$$Z^{i,j}_{:,0}=\begin{pmatrix}0\\v_7\\v_3\end{pmatrix},\ Z^{i,j}_{:,1}=\begin{pmatrix}v_{10}\\v_6\\v_2\end{pmatrix},\ Z^{i,j}_{:,2}=\begin{pmatrix}v_9\\v_5\\v_1\end{pmatrix},\ Z^{i,j}_{:,3}=\begin{pmatrix}v_8\\v_4\\v_0\end{pmatrix}$$

Equation (45) can then be computed column wise using $Z^{i,j}$ instead of $H^{i,j}$. The typical element of $y^{i,j}$ is

$$\begin{aligned} y^{i,j}_m &= \sum_{n=0}^{N^j-1} H^{i,j}_{m,n} x^j_n \\ &= \sum_{n=0}^{N^j-1} Z^{i,j}_{s,d} x^j_n \end{aligned}$$

Let k and l be defined such that $k\le m\le l$ whenever $0\le s\le L^{i,j}/\sigma-1$ and let n be fixed. Our task is now to determine d together with k and l such that $Z^{i,j}_{s,d}=H^{i,j}_{mn}$. Therefore we put $s=0$ and look for k, d such that

$$Z^{i,j}_{0,d}=H^{i,j}_{kn} \tag{53}$$

In other words we want to know which vector to use and at which row it must be aligned.

Next we insert the definitions (50) and (52) in (53) and obtain the equation

$$v^{i,j}_{\langle L^{i,j}-d-1\rangle_{N^j}} = h^{i,j}_{\langle n-\sigma k\rangle_{N^j}} = v^{i,j}_{\langle n-\sigma k+\delta-1\rangle_{N^j}}$$

which is fulfilled whenever

$$\begin{aligned} \langle L^{i,j}-d-1\rangle_{N^j} &= \langle n-\sigma k+\delta-1\rangle_{N^j} \Leftrightarrow \\ \langle L^{i,j}-d-n+\sigma k-\delta\rangle_{N^j} &= 0 \end{aligned}$$

Let $L = L^{i,j} - n - \delta$. Then we can write the requirement as

$$\langle L - d + \sigma k \rangle_{N^j} = 0 \tag{54}$$

Since we need k in the interval $[0, N^i - 1]$ we rewrite (54) using lemma A.2:

$$\begin{aligned} 0 &= \langle L - d + \sigma k \rangle_{N^j} \\ &= \langle (L - d + \sigma k)/\sigma \rangle_{N^j/\sigma} \\ &= \langle k + (L - d)/\sigma \rangle_{N^i} \end{aligned}$$

from which we get

$$k = \langle (d - L)/\sigma \rangle_{N^i}$$

For this to be well-defined σ must be a divisor in $(d - L)$, i.e. $\langle d - L \rangle_\sigma = 0$ so we must choose

$$d = \langle L \rangle_\sigma$$

Expanding the expression for L we obtain the desired expressions

$$d = \langle L^{i,j} - n - \delta \rangle_\sigma \tag{55}$$

$$k = \langle (d - L^{i,j} + n + \delta)/\sigma \rangle_{N^i} \tag{56}$$

Finally, l is obtained as

$$l = \langle k + L^{i,j} \rangle_{N^i} \tag{57}$$

Let $Z^{i,j}_{s,d}$ be defined by (52). Using (55), (56) and (57) we can formulate the algorithm as

Algorithm 3 Block multiplication ($i < j$)

For $n = 0$ to $N - 1$
    if $|x_n| > \varepsilon$ then
        $d = \langle L^{i,j} - n - \delta \rangle_\sigma$
        $k = \langle (n - L^{i,j} + \delta + d)/\sigma \rangle_{N^i}$
        $l = \langle m + L^{i,j}/\sigma - 1 \rangle_{N^i}$
        if $k < l$ then
            $y^{i,j}_{k:l} = y^{i,j}_{k:l} + x^j_n Z^{i,j}_{:,d}$
        else (wrap)
            $y^{i,j}_{0:l} = y^{i,j}_{0:l} + x^j_n Z^{i,j}_{L^{i,j}/\sigma - l : L^{i,j}/\sigma - 1, d}$
            $y^{i,j}_{k:N^i-1} = y^{i,j}_{k:N^i-1} + x^j_n Z^{i,j}_{0:L^{i,j}/\sigma - l - 1, d}$
        end
    end
end

6.3 Computational work

We are now ready to look at the computational work required for the matrix-vector multiplication $Y = Hx$. We take (44) as the point of departure and start by considering the typical block

$$H^{i,j}x^j$$

The length of x^j is N^j so for blocks on and below the diagonal of H $(i \geq j)$ there will be $L^{i,j}N^j$ multiplications and the same number of additions. Hence the work is

$$2L^{i,j}N^j$$

For blocks above the diagonal $(j > i)$ there will be N^i inner products of length $L^{i,j}$, so the work here is

$$2L^{i,j}N^i$$

The total work can therefore be written as follows

$$\sum_{j=0}^{\lambda} 2L^{j,j}N^j + \sum_{i=1}^{\lambda}\sum_{j=0}^{i-1} 2L^{i,j}N^j + \sum_{j=1}^{\lambda}\sum_{i=0}^{j-1} 2L^{i,j}N^i$$

where the first sum corresponds to blocks on the diagonal, the second to blocks below the diagonal, and the third to blocks above the diagonal. Since $L^{i,j} = L^{j,i}$ we can swap the indices of the last double sum to find the identity

$$\sum_{j=1}^{\lambda}\sum_{i=0}^{j-1} 2L^{i,j}N^i = \sum_{i=1}^{\lambda}\sum_{j=0}^{i-1} 2L^{i,j}N^j$$

Hence we may write the total work as

$$F = \sum_{j=0}^{\lambda} 2L^{j,j}N^j + 2\sum_{i=1}^{\lambda}\sum_{j=0}^{i-1} 2L^{i,j}N^j \tag{58}$$

We recall that

$$N^j = \begin{cases} 2^{J-1-\lambda+j} & \text{for } j \geq 1 \\ 2^{J-\lambda} & \text{for } j = 0 \end{cases}$$

and that $L^{i,j}$ is given by the recurrence formulas

$$\begin{aligned} L^{j-1,j-1} &= \lceil L^{j,j}/2 \rceil + D - 1 \\ L^{i,j-1} &= L^{i,j} + 2^{i-j}(D-1) \\ L^{\lambda+1,\lambda+1} &= L \text{ (the bandwidth of the original matrix } A) \end{aligned}$$

Figures 17 and 18 show tables of F evaluated for various values of L, D and λ.

	$N = 256,\ D = 4$			
λ	$L = 3$	$L = 4$	$L = 5$	$L = 6$
0	1536	2048	2560	3072
1	5120	5120	6144	6144
2	8448	8448	9216	9216
3	11520	11520	12288	12288
4	14592	14592	15360	15360
5	17664	17664	18432	18432

Figure 17: The number of floating point operations F shown for different values of λ and L. Note that $L = 2k$ and $L = 2k + 1$ ($k \in \mathbb{N}$) yield the same values for $\lambda > 0$. This is a direct consequence of the rounding done in (58).

	$N = 256,\ L = 5$			
λ	$D = 2$	$D = 4$	$D = 6$	$D = 8$
0	2560	2560	2560	2560
1	4096	6144	8192	10240
2	5120	9216	13312	17408
3	6272	12288	18560	24576
4	7360	15360	23680	31808
5	8416	18432	28736	38000

Figure 18: The number of floating point operations F shown for different values of λ and D.

The tables suggest that the total computational work grows linearly in L, D, and λ, so the matrix-vector multiplication is most efficient if we take no steps of the wavelet transform ($\lambda = 0$). Consequently, any work reduction must be sought in truncation of the vector x. This can be justified because x will often be a 1D wavelet transform of the same depth (λ) as the matrix H. Therefore, depending on λ and the underlying application, we expect many elements in x to be close to zero so we may be able to discard them in order to reduce the computational work. If we take $N = 256$, $L = 3$, $\lambda = 4$ as an example, the question boils down to whether such a truncation of x can reduce 14592 operations to less than the 1536 operations required for $\lambda = 0$ (no transform). Assuming that the work depends linearly on the number of non-zero elements in x, this means that we must reduce x by a factor of 10 (at least) before any work reduction is obtained. Whether this is possible depends on the application and is beyond the scope of this report.

7 Conclusion

We have derived an algorithm for computing the 2D wavelet transform of a circulant $N \times N$ matrix A. The algorithm works with a data structure that requires $2\lambda N$ elements instead of N^2. If A is also banded the storage requirements are reduced further. Then we have shown how a matrix-vector multiplication using the proposed storage scheme can be implemented and the complexity of this algorithm was analyzed. It was found that the work needed to compute the matrix-vector multiplication grows linearly with the bandwidth (L), the wavelet genus (D), and the depth of the wavelet transform λ. On the other hand, the work is reduced when small elements in x are discarded. Typically, more elements in x will be small as λ grow so there is a trade off that must be considered when the transform parameters are chosen.

A The modulus operator

Let $n \in \mathbb{Z}$ then

$$n = pq + r \tag{59}$$

where $p, q, r \in \mathbb{Z}$.

We denote q the *quotient* of n divided by p and r the *remainder* of that division.

The q and r are not uniquely determined from p and n. Rather, what is uniquely defined is an equivalence class consisting of all values of r as q takes on values from $\mathbb{Z}$. We write

$$r \equiv n \pmod p$$

for any r fulfilling (59) and thus belonging to the equivalence class $n \pmod p$.

However, one representative of this equivalence stands out. It is called the *principal remainder* and it is defined as

$$r = n \bmod p = n - p \left[\frac{n}{p}\right]$$

where $[\cdot]$ denotes the nearest integer towards zero.

This is the way modulus is implemented in many programming languages such as Matlab. While mathematically correct, it has the inherent inconvenience that a negative r is chosen for $n < 0$. In many applications such as periodic convolution we think of r as being the index of an array representing one period of a periodic signal. Therefore, we wish to choose a representative $n \pmod p$ to stay in the interval $r \in [0, p-1]$ for all $n \in \mathbb{Z}$. This can be accomplished by taking

$$r = \langle n \rangle_p = n - p \left\lfloor \frac{n}{p} \right\rfloor \tag{60}$$

where $\lfloor n/p \rfloor$ denotes the nearest integer *below* n/p. We have introduced the notation $\langle n \rangle_p$ in order to avoid confusion, and we note that

$$\langle n \rangle_p \quad = \quad n \bmod p \quad \text{for } n > 0,\ p > 1$$

For practical purposes $\langle n \rangle_p$ should not be implemented as in (60); rather, it should be written using the built-in modulus function modifying the result whenever needed.

The programming language Fortran 90 includes both definitions, so `MOD(n,p)` is the ordinary modulus operator, where `MODULO(n,p)` implements $\langle n \rangle_p$ (see [MR96, p 177]).

A.1 Definitions and formulas

Definition A.1 *Let n be given as in (59). If $r = 0$, we say that p is a divisor in n and write*

$$p \mid n$$

It follows that for all $p, n, r \in \mathbb{Z}$ we have

$$p \mid (n - r)$$

Lemma A.1 *Let $n_1, n_2, q \in \mathbb{Z}$. Then*

$$\begin{aligned} \langle n_1 \pm n_2 \rangle_q &= \langle n_1 \pm \langle n_2 \rangle_q \rangle_q \\ \langle n_1 n_2 \rangle_q &= \langle n_1 \langle n_2 \rangle_q \rangle_q \end{aligned}$$

Proof: We write n_2 as in (59) and find

$$\begin{aligned} \langle n_1 \pm n_2 \rangle_q &= \langle n_1 \pm (p_2 q + r_2) \rangle_q \\ &= \langle n_1 \pm r_2 \rangle_q \\ &= \langle n_1 \pm \langle n_2 \rangle_q \rangle_q \end{aligned}$$

and

$$\begin{aligned} \langle n_1 n_2 \rangle_q &= \langle n_1 (p_2 q + r_2) \rangle_q \\ &= \langle n_1 p_2 q + n_1 r_2 \rangle_q \\ &= \langle n_1 r_2 \rangle_q \\ &= \langle n_1 \langle n_2 \rangle_q \rangle_q \end{aligned}$$

□

Lemma A.2 *Let $k, n, q \in \mathbb{Z}$. Then*

$$k \langle n \rangle_q = \langle kn \rangle_{kq}$$

Proof: We write n as in (59) and get

$$r = \langle n \rangle_q \Leftrightarrow q \mid (n - r)$$

When q is a divisor in $n - r$ then also

$$kq \mid k(n - r)$$

from which we get

$$kr = \langle kn \rangle_{kq}$$

□

References

[Cha96] Philippe Charton. *Produits de matrices rapides en bases d'ondelettes: application á la résolution numérique d'équations aux dérivés partielles*. PhD thesis, Université de Paris Nord, janvier 1996.

[CP96] Philippe Charton and Valerie Perrier. A pseudo-wavelet scheme for the two-dimensional navier-stokes equations. Technical report, Laboratoire de Météorologie Dynamique, Paris, March 1996.

[Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[EBP0] S. Mallat E. Bacry and G. Papanicolaou. A wavelet based space-time adaptive numerical method for partial differential equations. Technical report, Courant Institute of Mathematical Sciences, New York, 1990 ??

[GL94] L. Gagnon and J. M. Lina. Symmetric daubechies' wavelets and numerical solution of nls equations. *J. Phys. A: Math. Gen.*, 27:8207–8230, 1994.

[Mal89] Stephane G. Mallat. Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(R)$. *Trans. am. math. soc*, 315(1), 1989.

[Mey93] Yves Meyer. *Wavelets: Algorithms and Applications*. SIAM, 1993.

[MR96] Michael Metcalf and John Reid. *Fortran 90/95 explained*. Oxford Science Publications, 1996.

[PPB95] Valérie Perrier, Thierry Phillipovitch, and Claude Basdevant. Wavelet spectra compared to fourier spectra. *J. Math. Phys.*, 36(3):1506–1519, March 1995.

[PW96] Iestyn Pierce and Lionel Watkins. Modelling optical pulse propagation in nonlinear media using wavelets. Technical report, School of Electronic Engineering and Computer Systems, University of Wales, Bangor, UK, 1996.

[SN96] Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

[Spi93] Murray R. Spiegel. *Mathematical Handbook of Formulas and Tables*. Schaum's Outline Series, 1993.

[Wal94] Gilbert G. Walter. *Wavelets and Other Orthorgonal Systems*. CRC Press, Inc, 1994.