2

NASA Contractor Report 189721

ICASE Report No. 92-55

AD-A257 888

ICASE

FAST MULTIRESOLUTION ALGORITHMS FOR MATRIX-VECTOR MULTIPLICATION

DTIC ELECTE DEC 0 4 1992 S A D

Ami Harten

Itai Yad-Shalom

Contract Nos. NAS1-18605 and NAS1-19480

October 1992

Institute for Computer Applications in Science and Engineering

NASA Langley Research Center

Hampton, Virginia 23681-0001

Operated by the Universities Space Research Association

92-30810

NASA

National Aeronautics and Space Administration

Langley Research Center

Hampton, Virginia 23665-5225

92 12 03 058

FAST MULTIRESOLUTION ALGORITHMS FOR MATRIX-VECTOR MULTIPLICATION

Ami Harten[1]
School of Mathematical Sciences
Tel-Aviv University - ISRAEL
and
Department of Mathematics, UCLA
and
Itai Yad-Shalom
School of Mathematical Sciences
Tel-Aviv University - ISRAEL

ABSTRACT

In this paper we present a class of multiresolution algorithms for fast application of structured dense matrices to arbitrary vectors, which includes the fast wavelet transform of Beylkin, Coifman and Rokhlin and the multilevel matrix multiplication of Brandt and Lubrecht. In designing these algorithms we first apply data compression techniques to the matrix and then show how to compute the desired matrix-vector multiplication from the compressed form of the matrix. In describing this class we pay special attention to an algorithm which is based on discretization by cell-averages as it seems to be suitable for discretization of integral transforms with integrably singular kernels.

[1]This research was supported by the National Aeronautics and Space Administration under NASA Contract Nos. NAS1-18605 and NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001. Research was also supported by ONR Grant #N00014-91-J-1034 and NSF Grant #DMS91-03104.

1. Introduction.

In this paper we present a class of multiresolution algorithms for rapid application of dense matrices to vectors. A direct application of an arbitrary $N \times N$ dense matrix to a vector requires N^2 operations. However, when the matrix-vector multiplication stems from a discretization of an integral transform

$$u(x) = \int\int K(x,y)v(y)dy, \tag{1.1}$$

where the kernel $K(x,y)$ is smooth except possibly along curves, this product can be performed to any prescribed accuracy with only $O(N)$ operations.

In [1] Beylkin, Coifman and Rokhlin (BCR) present a wavelet based algorithm (referred to as the "nonstandard form"), in which the matrix-vector multiplication is performed by successive contributions from different scales. It starts with an initial blurred (low resolution) output vector for u in (1.1), which is then upgraded successively to higher and higher resolution, in much the same way as the pyramid scheme in image comrpession.

In [2] Brandt and Lubrecht (BL) describe a multilevel matrix-vector multiplication which is viewed as performing part of the integration in (1.1) on coarser grids. This is possible wherever the local smoothness of the kernel $K(x,y)$ enables the replacement of its fine grid values by sufficiently accurate interpolation from coarser grids.

In [7] we have presented a class of multiresolution algorithms for data compression. In the present paper we apply these data compression algorithms to matrices as a tensor product of one-dimensional oeprators to obtain a multiresolution representation of the matrix. Using this representation we derive a class of $O(N)$ matrix-vector multiplication algorithms, which includes the BCR algorithm [1] and the BL algorithm [2] as particular cases. In describing this class we also pay special attention to the algorithm which is based on discretization by cell-averages, because it seems to be particularly suitable to kernels with integrable singularity.

2. Discretization and Reconstruction.

In this section we describe a class of discretizations of a function and the approximate inverse of these discretizations, namely the approximate recovery of a function from its given discrete values; we refer to the process of recovery as reconstruction.

Let $\{x_j^0\}$, $x_j^0 = j \cdot h^0$ be a partition of the real line into uniform intervals $\{I_j\}$, $I_j = [x_{j-1}^0, x_j^0]$, of size h_0. Let $\varphi(x)$ be a function which is concentrated around $x = 0$ and satisfies

$$\text{(2.1a)} \qquad \int \varphi(x)dx = 1,$$

and define its scaled translates

$$\text{(2.1b)} \qquad \varphi_j^0(x) = \frac{1}{h_0}\varphi\left(\frac{x}{h_0} - j\right).$$

Given a function $f(x)$ we discretize it by

$$\text{(2.2)} \qquad \bar{f}_j^0 = \langle f, \varphi_j^0 \rangle = \int f(x)\varphi_j^0(x)dx.$$

Next let us introduce an approximate recovery of the function $f(x)$ from its given values $\bar{f}^0 = \{\bar{f}_j^0\}$ which we refer to as reconstruction and denote by $\mathcal{R}(x; \bar{f}^0)$. We say that the reconstruction is r-th order accurate if

$$\text{(2.3a)} \qquad \mathcal{R}(x; \bar{f}^0) = f(x) + O((h_0)^r), \qquad \text{(accuracy)}$$

provided that $f(x)$ is sufficiently smooth. We assume that the reconstruction is conservative in the sense that

$$\text{(2.3b)} \qquad \langle \mathcal{R}(\cdot; \bar{f}^0),\ \varphi_j^0 \rangle = \bar{f}_j^0 \qquad \text{(conservation)}.$$

Unlike the setup in [7], where the goal is to obtain maximal data compression, in the present application to matrix-vector multiplication we want minimal number of operations. Therefore we assume that $\mathcal{R}(\cdot; \bar{f}^0)$ is a linear functional of $\bar{f}^0$ and that $\varphi(x)$ satisfies a dilation equation

$$\text{(2.4a)} \qquad \varphi(x) = 2\sum_{\ell} \alpha_\ell \varphi(2x - \ell),$$

where the coefficients $\{\alpha_\ell\}$ satisfy

$$\sum \alpha_\ell = 1 \tag{2.4b}$$

$$\sum \alpha_\ell \alpha_{\ell+2m} = 0 \quad \text{for } m \neq 0. \tag{2.4c}$$

We note that relation (2.4b) is just a consistency condition. Given a set of $\{\alpha_\ell\}$, $\sum \alpha_\ell = 1$, it is shown in [4] and [8] that $\varphi(x)$ is determined by the dilation equation (2.4a) up to a multiplicative constant. Hence $\varphi(x)$ is determined uniquely by adding the normalization (2.1a) to (2.4a)-(2.4b). In Appendix A we show that condition (2.4c) implies orthogonality of some matrices and thus reduces the number of operations in our algorithm. In order for the set of functions $\{\varphi_j^0(x)\}$ to be orthogonal we have to add another consistency relation (see [8])

$$\sum_\ell \alpha_\ell^2 = \frac{1}{2}, \tag{2.5a}$$

in which case

$$\langle \varphi_i^0, \varphi_j^0 \rangle = \frac{\|\varphi\|^2}{h_0} \delta_{i,j} \tag{2.5b}$$

where $\delta_{i,j}$ is the Krönecker-δ; i.e. $\delta_{i,i} = 1$, $\delta_{ij} = 0$ for $i \neq j$.

In this paper we highlight the following three cases:

Case 1. **Pointvalues.**

$$\varphi(x) = \delta(x) \tag{2.6a}$$

where $\delta(x)$ is Dirac's distribution. As pointed out by Strang [8] it satisfies the dilation equation

$$\delta(x) = 2\delta(2x) \tag{2.6b}$$

and thus

$$\alpha_0 = 1, \quad \alpha_\ell = 0 \text{ for } \ell \neq 0. \tag{2.6c}$$

Note that the coefficients (2.6c) trivially satisfy the orthogonality relation (2.4c). However

$$\sum \alpha_\ell^2 = 1$$

and thus (2.5) is not valid in this case.

The discretization (2.2) becomes

$$\bar{f}_j^0 = \int f(x)\delta\left(\frac{x}{h_0} - j\right)\frac{dx}{h_0} = f(x_j^0), \tag{2.7a}$$

i.e. the function $f(x)$ is discretized by taking its value at the grid points $\{x_j^0\}$. The conservation property (2.3b) becomes

$$\mathcal{R}(x_j^0; \bar{f}^0) = \bar{f}_j^0, \tag{2.7b}$$

i.e. the reconstruction is an interpolation of the values $\{\bar{f}_j^0\}$ at the grid points $\{x_j^0\}$.

Case 2. **Cell-averages.**

$$\varphi(x) = \mathcal{X}_{[-1,0)}(x) = \begin{cases} 1 & -1 \le x < 0 \\ 0 & \text{otherwise,} \end{cases} \tag{2.8a}$$

satisfies the dilation equation

$$\varphi(x) = \varphi(2x) + \varphi(2x+1) \tag{2.8b}$$

and thus

$$\alpha_0 = \alpha_{-1} = \frac{1}{2}, \quad \alpha_\ell = 0 \text{ for } \ell \neq -1, 0. \tag{2.8c}$$

The discretization of $f(x)$ in (2.2) becomes

$$\bar{f}_j^0 = \int f(x)\mathcal{X}_{[-1,0)}\left(\frac{x}{h_0} - j\right)\frac{dx}{h_0} = \frac{1}{h_0}\int_{x_{j-1}^0}^{x_j^0} f(x)dx, \tag{2.9a}$$

i.e. $f(x)$ is discretized by taking $\bar{f}_j^0$ to be its average in the interval I_j^0. The conservation requirement (2.3b) becomes

$$\frac{1}{h_0}\int_{x_{j-1}^0}^{x_j^0} \mathcal{R}(x;\bar{f}^0)dx = \bar{f}_j^0. \tag{2.9b}$$

Let us denote by $F(x)$

$$F(x) = \int_0^x f(\xi)d\xi, \tag{2.10a}$$

the primitive function of $f(x)$

$$\frac{d}{dx}F(x) = f(x) \tag{2.10b}$$

and observe that

$$F(x_j^0) = h_0 \sum_{i=1}^{j} \bar{f}_i^0. \tag{2.10c}$$

It is easy to see that

$$\mathcal{R}(x;\bar{f}^0) = \frac{d}{dx}I(x;F^0), \tag{2.11}$$

where $I(x;F^0)$ is any interpolation of the values $F_j^0 = F(x_j^0)$ (2.10c), satisfies the conservation requirement (2.9b). This reconstruction procedure is r-th order accurate (2.3a) if the interpolation technique in (2.11) satisfies

$$\frac{d}{dx}I(x;F^0) = \frac{d}{dx}F(x) + O((h_0)^r) = f(x) + O((h_0)^r) \tag{2.12}$$

for sufficiently smooth $f(x)$.

Case 3. **Orthogonal Wavelets.**

Let $\varphi(x)$ be a function which is determined by the dilation equation (2.4a), with coefficients that satisfy (2.4b)-(2.4c) and (2.5a). Thus we assume orthogonality of the set $\{\varphi_j^0\}$ (2.5b). In the context of this paper it is most natural to describe wavelets by first specifying the reconstruction to be a linear combination of $\{\varphi_j^0\}$, i.e.

$$\mathcal{R}(x;\bar{f}^0) = \sum_i a_i \varphi_i^0(x) \tag{2.13a}$$

and to leave the discretization (2.2) to be determined later. The conservation requirement (2.3b) becomes

$$\langle \mathcal{R}(\cdot;\bar{f}^0),\ \varphi_j^0\rangle = \sum_i a_i \langle \varphi_i^0, \varphi_j^0\rangle = \bar{f}_j^0. \tag{2.13b}$$

Using the orthogonality (2.5b) we get

$$a_i = \frac{h_0}{\|\varphi\|^2}\bar{f}_i^0. \tag{2.13c}$$

Thus

$$\mathcal{R}(x;\bar{f}^0) = \frac{h_0}{\|\varphi\|^2}\sum_i \bar{f}_i^0 \varphi_i^0(x). \tag{2.14}$$

Using the theory of approximation by translates Strang [8] shows that in order for the reconstruction (2.14) to be r-th order accurate (2.3a) we have to impose the following condition on the coefficients $\{\alpha_\ell\}$,

$$\sum_\ell (-1)^\ell \ell^m \alpha_\ell = 0 \qquad \text{for } m = 0, 1, \dots, r-1. \tag{2.15}$$

Daubechies [4] showed that in order to satisfy the conditions on $\{\alpha_\ell\}$ listed so far, one needs at least $2r$ nonzero coefficients, and that the set of exactly $2r$ nonzero coefficients is unique. For $r = 1$ this solution is given by (2.8c), i.e. $\varphi(x)$ is the box function (2.8a). For $r \geq 2$ the resulting $\varphi(x)$ is necessarily nonsymmetric; the smoothness of $\varphi(x)$ increases with r, but only by half a derivative (approximately) each time. Beylkin, Coifman and Rokhlin in [1] impose an additional set of requirements on $\{\alpha_\ell\}$, namely that there exists an integer τ_r so that

$$\int \varphi(x+\tau_r)x^m dx = 0 \qquad \text{for } m = 1, 2, \dots, r-1; \tag{2.16a}$$

This implies

$$\bar{f}_j^0 = \langle f, \varphi_j^0 \rangle = f(x_j^0 + \tau_r h_0) + O((h_0)^r), \tag{2.16b}$$

which shows that the integration in (2.16b) can be approximated to r-th order accuracy by a single point quadrature. They show that there is a solution to the extended set of conditions with $3r$ nonzero coefficients $\{\alpha_\ell\}$.

We remark that for large r, the discretization implied by (2.16b) is close to that of pointvalues.

3. Multiresolution Algorithms for Data Compression

In this section we consider a situation where we are given N_0 values

$$\bar{f}^0 = \{\bar{f}_j^0\}_{j=1}^{N_0}, \quad N_0 = 2^{n_0}, \quad n_0 \text{ integer}, \tag{3.1a}$$

which represent a discretization (2.2) of some function $f(x)$ corresponding to a uniform partition of $[0,1]$,

$$x_j^0 = j \cdot h_0, \quad 0 \le j \le N_0, \quad h_0 = 1/N_0. \tag{3.1b}$$

To simplify our presentation we assume for the time being that $f(x)$ is periodic with period 1, so that values outside $[0,1]$ are known by periodic extension.

We consider the set of nested grids

$$\{x_j^k\}_{j=1}^{N_k}, \quad x_j^k = j \cdot h_k, \quad h_k = 1/N_k, \quad N_k = 2^{-k} N_0; \tag{3.2a}$$

for $0 \le k \le L$, where $k = 0$, the original grid, is the finest in the hierarchy and $k = L$, $L < n_0$, is the coarsest. The coarser $(k+1)$-th grid is formed form the k-th grid by removing the grid points $\{x_{2j-1}^k\}_{j=1}^{N_k}$; thus

$$x_j^{k+1} = x_{2j}^k, \quad 0 \le j \le N_{k+1}, \quad N_{k+1} = N_k/2. \tag{3.2b}$$

To each of the nested grids we associate a discretization

$$\bar{f}^k = \{\bar{f}_j^k\}_{j=1}^{N_k}, \quad \bar{f}_j^k = \langle f, \varphi_j^k \rangle, \tag{3.3a}$$

where φ_j^k is properly scaled

$$\varphi_j^k(x) = \frac{1}{h_k}\varphi\left(\frac{x}{h_k} - j\right). \tag{3.3b}$$

It follows from the dilation relation (2.4a) that

$$\varphi_j^k(x) = \sum_\ell \alpha_\ell \varphi_{2j+\ell}^{k-1}(x), \tag{3.3c}$$

and consequently

$$\bar{f}_j^k = \sum_\ell \alpha_\ell \bar{f}_{2j+\ell}^{k-1} = \sum_\ell \alpha_{\ell-2j} \bar{f}_\ell^{k-1}, \quad 1 \le j \le N_k. \tag{3.3d}$$

We rewrite (3.3d) in the matrix form

$$\bar{f}^k = H\bar{f}^{k-1}, H_{ij} = \alpha_{j-2i}, \quad H_{N_k} \times {}_{2N_k}. \tag{3.4}$$

Given $\bar{f}^0$ we use (3.4) to successively compute $\bar{f}^1, \dots, \bar{f}^L$. Observe that these values are not computed from the definition (3.3a) but from the dilation relation (3.3d); thus no explicit knowledge of $f(x)$ is required.

Given $\bar{f}^k$ we can use the reconstruction $\mathcal{R}(x; \bar{f}^k)$ in order to get an approximation $\tilde{f}^{k-1}$ to the discrete values $\bar{f}^{k-1}$ of the finer level by

$$\tilde{f}_j^{k-1} = \langle \mathcal{R}(\cdot; \bar{f}^k), \varphi_j^{k-1} \rangle, \quad 1 \le j \le 2N_k = N_{k-1}. \tag{3.5a}$$

As we have mentioned earlier, in this paper we take $\mathcal{R}(\cdot; \bar{f})$ to be a linear functional of $\bar{f}$. Hence (3.5a) can be epxressed in the matrix form

$$\tilde{f}^{k-1} = R\bar{f}^k \tag{3.5b}$$

where R is an $2N_k \times N_k$ matrix. Because of (3.3c) and the conservation property of the reconstruction (2.3b) we get that

$$\begin{aligned} \sum_\ell \alpha_\ell \tilde{f}_{2j+\ell}^{k-1} &= \sum_\ell \alpha_\ell \langle \mathcal{R}(\cdot; \bar{f}^k),\ \varphi_{2j+\ell}^{k-1} \rangle \\ &= \langle \mathcal{R}(\cdot; \bar{f}^k), \sum_\ell \alpha_\ell \varphi_{2j+\ell}^{k-1} \rangle = \langle \mathcal{R}(\cdot; \bar{f}^k), \varphi_j^k \rangle = \bar{f}_j^k, \end{aligned} \tag{3.6a}$$

or in matrix form

$$H\tilde{f}^{k-1} = \bar{f}^{k}. \tag{3.6b}$$

It follows then from (3.5b) and (3.6b) that for any vector $\bar{f}^k$

$$\bar{f}^k = H\tilde{f}^{k-1} = HR\bar{f}^k,$$

which shows that

$$HR = I, \tag{3.7a}$$

and consequently

$$H(I - RH) = H - (HR)H = H - H = 0. \tag{3.7b}$$

We turn now to examine the error e^{k-1} in the prediction $\tilde{f}^{k-1}$ (3.5)

$$e^{k-1} = \bar{f}^{k-1} - \tilde{f}^{k-1} = \bar{f}^{k-1} - R\bar{f}^k = (I - RH)\bar{f}^{k-1}. \tag{3.8a}$$

¿From (3.7b) it follows that

$$He^{k-1} = 0, \tag{3.8b}$$

which shows that only N_k out of the $2N_k$ components of e^{k-1} are independent quantities. In order to get rid of this redundancy in e^{k-1} we introduce the $N_k \times 2N_k$ matrix G

$$G_{ij} = (-1)^{j+1}\alpha_{2i-1-j}, \qquad (G)_{N_k \times 2N_k} \tag{3.9a}$$

which satisfies

$$HG^* = 0. \tag{3.9b}$$

In Appendix A we show that it follows from the orthogonality condition (2.4c) that

$$HH^* = GG^* = |\alpha|^2 \cdot I, \tag{3.10a}$$

$$H^*H + G^*G = |\alpha|^2 \cdot I, \tag{3.10b}$$

$$|\alpha|^2 = \sum_\ell \alpha_\ell^2. \tag{3.10c}$$

Using (3.10b) and (3.8b) we now get that

$$e^{k-1} = \frac{1}{|\alpha|^2}(H^*H + G^*G)e^{k-1} = \frac{1}{|\alpha|^2}G^*(Ge^{k-1}) = \frac{1}{|\alpha|^2}G^*d^k, \tag{3.11a}$$

where

$$d^k = Ge^{k-1} \tag{3.11b}$$

is a vector of N_k components. Combining (3.11) with (3.8a) we get

$$\bar{f}^{k-1} = \tilde{f}^{k-1} + e^{k-1} = R\bar{f}^k + \frac{1}{|\alpha|^2}G^*d^k, \tag{3.12}$$

which is the basis for the following data compression algorithm:

Given a sequence of N_0 numbers $u = \{u_i\}_{i=1}^{N_0}$, we set

$$\bar{f}^0 = u \tag{3.13a}$$

and execute

$$\begin{cases} \text{Do for } k = 1, 2, \dots, L \\ \bar{f}^k = H\bar{f}^{k-1} \\ e^{k-1} = \bar{f}^{k-1} - R\bar{f}^k \\ d^k = Ge^{k-1} \end{cases}, \tag{3.13b}$$

thus arriving at the multiresolution representation u^{MR} of u

$$u^{MR} = \{\bar{f}^L, (d^L, \dots, d^1)\}. \tag{3.13c}$$

Starting from the multiresolution representation we recover u by (3.12), i.e.

$$\text{(3.13d)} \qquad \begin{cases} \text{Do for } k = L, L-1, \dots, 1 \\ \bar{f}^{k-1} = R\bar{f}^k + \frac{1}{|\alpha|^2} G^* d^k, \end{cases}$$

$$\text{(3.13e)} \qquad u = \bar{f}^0.$$

The number of quantities in the multiresolution representation u^{MR} (3.13c) is N_0 as in the original vector u (3.13a). The difference is that the quantities $\{d_i^k\}$ are expected to be small in absolute value wherever the underlying function $f(x)$ is adequately resolved on the k-th grid. Thus data compression can be achieved by setting to zero elements of d^k which fall below some tolerance ε_k. See [7] for more details.

In Appendix A we present the form of the data compression algorithm (3.13) when we do not assume the orthoginality condition (2.4c).

In the following we present the details for the three cases that we highlight in this paper.

Case 1. **Pointvalues.**

$\mathcal{R}(x; \bar{f}^k)$ is the interpolation (2.7b). For reasons of symmetry we consider even order of accuracy $r = 2s$ and take $\mathcal{R}(x; \bar{f}^k)$ in $[x_{j-1}^k, x_j^k]$ to be the unique polynomial of degree $(2s-1)$ that interpolates $\bar{f}^k$ at the gridpoints $\{x_{j-s}^k, \dots, x_{j+s-1}^k\}$. In (3.5) we get for $1 \le i \le N_k$

$$\text{(3.14a)} \qquad \bar{f}_{2i}^{k-1} = (R\bar{f}^k)_{2i} = \bar{f}_i^k$$

$$\text{(3.14b)} \qquad \bar{f}_{2i-1}^{k-1} = (R\bar{f}^k)_{2i-1} = \sum_{\ell=1}^{s} \beta_\ell (\bar{f}_{i+\ell-1}^k + \bar{f}_{i-\ell}^k)$$

where

$$\text{(3.14c)} \qquad \begin{cases} r = 2 \Rightarrow \beta_1 = \frac{1}{2} \\ r = 4 \Rightarrow \beta_1 = \frac{9}{16},\ \beta_2 = -\frac{1}{16} \\ r = 6 \Rightarrow \beta_1 = \frac{150}{256},\ \beta_2 = \frac{-25}{256},\ \beta_2 = \frac{3}{256}. \end{cases}$$

In (3.4) and (3.9a) we get

$$H_{ij} = \delta_{2i,j}, \quad G_{ij} = \delta_{2i-1,j}. \tag{3.15}$$

The multiresolution representation (3.13c) is obtained by:

Set

$$\bar{f}^0 = u \tag{3.16a}$$

$$\begin{cases} \text{Do for } k = 1, 2, \ldots, L \\ \bar{f}_i^k = \bar{f}_{2i}^{k-1}, \ 1 \le i \le N_k, \\ d_i^k = \bar{f}_{2i-1}^{k-1} - \sum_{\ell=1}^{s} \beta_\ell (\bar{f}_{i+\ell-1}^k + \bar{f}_{i-\ell}^k), \qquad 1 \le i \le N_k. \end{cases} \tag{3.16b}$$

u is recovered from the multiresolution representation u^{MR} by

$$\begin{cases} \text{Do for } k = L, L-1, \ldots, 1 \\ \bar{f}_{2i}^{k-1} = \bar{f}_i^k, \ \ 1 \le i \le N_k \\ \bar{f}_{2i-1}^{k-1} = \sum_{\ell=1}^{s} \beta_\ell (\bar{f}_{i+\ell-1}^k + \bar{f}_{i-\ell}^k) + d_i^k, \ \ 1 \le i \le N_k, \end{cases} \tag{3.16c}$$

$$u = \bar{f}^0 \tag{3.16d}$$

Case 2. **Cell-averages.**

Using interpolation of order $(2s+2)$ as above for the primitive function in (2.11) we obtain a reconstruction of order $r = 2s+1$. In (3.5) we get for $1 \le i \le N_k$

$$\bar{f}_{2i-1}^{k-1} = (R\bar{f}^k)_{2i-1} = \bar{f}_i^k + z_i^k \tag{3.17a}$$

$$\bar{f}_{2i}^{k-1} = (R\bar{f}^k)_{2i} = \bar{f}_i^k - z_i^k \tag{3.17b}$$

where

$$z_i^k = \sum_{\ell=1}^{s} \gamma_\ell (\bar{f}_{i+\ell}^k - \bar{f}_{i-\ell}^k) \tag{3.17c}$$

and

$$\begin{cases} r = 3 \Rightarrow \gamma_1 = -\frac{1}{8} \\ r = 5 \Rightarrow \gamma_1 = -\frac{22}{128}, \ \gamma_2 = \frac{3}{128}; \end{cases} \tag{3.17d}$$

note that $z_i^k \equiv 0$ for $r = 1$.

In (3.4) and (3.9a) we get

$$H_{ij} = \frac{1}{2}(\delta_{2i,j} + \delta_{2i-1,j}), \quad G_{ij} = \frac{1}{2}(\delta_{2i-1,j} - \delta_{2i,j}). \tag{3.18}$$

The multiresolution representation (3.15c) is obtained by:

Set

$$\bar{f}^0 = u \tag{3.19a}$$

$$\begin{cases} \text{Do for } k = 1, 2, \dots, L \\ \bar{f}_i^k = \frac{1}{2}(\bar{f}_{2i-1}^{k-1} + \bar{f}_{2i}^{k-1}), \quad 1 \le i \le N_k \\ d_i^k = \bar{f}_{2i-1}^{k-1} - \bar{f}_i^k - \sum_{\ell=1}^{s} \gamma_\ell(\bar{f}_{i+\ell}^k - \bar{f}_{i-\ell}^k), \quad 1 \le i \le N_k \end{cases} \tag{3.19b}$$

u is recovered from the multiresolution representation u^{MR} (3.13c) by

$$\begin{cases} \text{Do for } k = L, L-1, \dots, 1 \\ \text{Do for } i = 1, 2, \dots, N_k \\ \Delta = \sum_{\ell=1}^{s} \gamma_\ell(\bar{f}_{i+\ell}^k - \bar{f}_{i-\ell}^k) + d_i^k \\ \bar{f}_{2i-1}^{k-1} = \bar{f}_i^k + \Delta, \quad \bar{f}_{2i}^{k-1} = \bar{f}_i^k - \Delta, \end{cases} \tag{3.19c}$$

$$u = \bar{f}^0. \tag{3.19d}$$

Case 3. **Orthogonal Wavelets**

The reconstruction (2.14) for the k-th level is

$$\mathcal{R}(x; \bar{f}^k) = \frac{h_k}{\|\varphi\|^2} \sum_{j=1}^{N_k} \bar{f}_j^k \varphi_j^k(x), \tag{3.20a}$$

and (3.5) becomes

$$\bar{f}_i^{k-1} = (R\bar{f}^k)_i = \langle \mathcal{R}(\cdot; \bar{f}^k), \varphi_i^{k-1} \rangle = \frac{h_k}{\|\varphi\|^2} \sum_{j=1}^{N_k} \bar{f}_j^k \langle \varphi_j^k, \varphi_i^{k-1} \rangle.$$

Using (3.3c) and (2.5b) we get that

$$\langle \varphi_j^k, \varphi_i^{k-1} \rangle = \sum_\ell \alpha_\ell \langle \varphi_{2j+\ell}^{k-1}, \varphi_i^{k-1} \rangle = \frac{\|\varphi\|^2}{h_{k-1}} \alpha_{i-2j} = \frac{\|\varphi\|^2}{h_{k-1}} H_{ij}^*.$$

Recalling that $h_k = 2h_{k-1}$ we get

$$R = 2H^*. \tag{3.20b}$$

Using (3.10) with (2.5a) to express the error in (3.8a) we get that

$$e^{k-1} = (I - RH)\bar{f}^{k-1} = (I - 2H^*H)\bar{f}^{k-1} = 2G^*G\bar{f}^{k-1} = 2G^*d^k \tag{3.21a}$$

$$d^k = G\bar{f}^{k-1}. \tag{3.21b}$$

The coefficients $\{\sqrt{2}\cdot\alpha_\ell\}$, $1 \le \ell \le 2r$ of Daubechies [4] are given in the following table:

Table 1.

	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$
$\sqrt{2}\alpha_1$	.482962913145	.332670552950	.230377813309	.160102397974	.111540743350
$\sqrt{2}\alpha_2$	.836516303738	.806891509311	.714856570553	.603829269797	.494623890398
$\sqrt{2}\alpha_3$	.224143868042	.459877502118	.630880767930	.724308528438	.751133908021
$\sqrt{2}\alpha_4$	-.129409522551	-.135011020010	-.027983769417	.138428145901	.315250351709
$\sqrt{2}\alpha_5$		-.085441273882	-.187034811719	-.242294887066	-.226264693965
$\sqrt{2}\alpha_6$		.035226291882	.030841381836	-.032244869585	-.129766867567
$\sqrt{2}\alpha_7$			.032883011667	.077571493840	.097501605587
$\sqrt{2}\alpha_8$			-.010597401785	-.006241490213	.027522865530
$\sqrt{2}\alpha_9$				-.012580751999	-.031582039318
$\sqrt{2}\alpha_{10}$				.003335725285	.000553842201
$\sqrt{2}\alpha_{11}$					.004777257511
$\sqrt{2}\alpha_{12}$					-.001077301085

The multiresolution representation (3.15c) is obtained by:

Set

(3.22a) $$\bar{f}^0 = u$$

(3.22b) $$\begin{cases} \text{Do for } k = 1, 2, \dots, L \\ \text{Do for } i = 1, 2, \dots, N_k \\ \bar{f}_i^k = \sum_{\ell=1}^{2r} \alpha_\ell \bar{f}_{2i+\ell}^{k-1} \\ d_i^k = \sum_{\ell=1}^{2r} (-1)^\ell \alpha_\ell \bar{f}_{2i-1-\ell}^{k-1} \end{cases}$$

u is recovered from the multiresolution representation u^{MR} by:

(3.22c) $$\begin{cases} \text{Do for } k = L, L-1, \dots, 1 \\ \text{Do for } i = 1, 2, \dots, N_k \\ \bar{f}_{2i-1}^{k-1} = 2\sum_{\ell=1}^{r} [\alpha_{2\ell-1} \bar{f}_{i-\ell}^k + \alpha_{2\ell} d_{i+\ell}^k] \\ \bar{f}_{2i}^{k-1} = 2\sum_{\ell=1}^{r} [\alpha_{2\ell} \bar{f}_{i-\ell}^k - \alpha_{2\ell-1} d_{i+\ell}^k], \end{cases}$$

(3.22c) $$u = \bar{f}^0.$$

4. Matrix-vector multiplication.

In this section we describe a multiresolution algorithm for the multiplication of an N_0-vector b by an $N_0 \times N_0$ matrix A, which is based on the data compression of A; we denote the result of this product by the N_0-vector c,

(4.1) $$Ab = c.$$

We start by presenting a tensor-product extension of the one-dimensional data compression algorithm (3.13) to the matrix case, in which each column and row of the matrix are treated as one-dimensional vectors. Let us set

(4.2a) $$\bar{A}^0 = A$$

and define the $N_k \times N_k$ matrix $\bar{A}^k$ by

(4.2b) $$\bar{A}^k = H\bar{A}^{k-1}H^*, \quad k = 1, \dots, L,$$

where H is the $N_k \times 2N_k$ matrix defined by (3.4).

Given $\bar{A}^k$ we form the prediction $\tilde{A}^{k-1}$ by

$$\tilde{A}^{k-1} = R\bar{A}^k R^*, \tag{4.3a}$$

where R is the $2N_k \times N_k$ matrix in (3.5). It follows from (4.2b) and (3.7a) that the error in this prediction E^{k-1},

$$E^{k-1} = \bar{A}^{k-1} - \tilde{A}^{k-1} = \bar{A}^{k-1} - R\bar{A}^k R^* \tag{4.3b}$$

satisfies

$$HE^{k-1}H^* = H\bar{A}^{k-1}H^* - (HR)\bar{A}^k(HR)^* = \bar{A}^k - \bar{A}^k = 0. \tag{4.3c}$$

Consequently, using (3.10b) and (4.3c) we get

$$\begin{aligned} E^{k-1} &= \frac{1}{|\alpha|^4}(H^*H + G^*G)E^{k-1}(H^*H + G^*G) \\ &= \frac{1}{|\alpha|^4}(G^*D_1^kG + GD_2^kH + H^*D_3^kG), \end{aligned} \tag{4.4a}$$

where the $N_k \times N_k$ matrices $\{D_i^k\}_{i=1}^3$ denote

$$D_1^k = GE^{k-1}G^*, \; D_2^k = GE^{k-1}H^*, \; D_3^k = HE^{k-1}G^*. \tag{4.4b}$$

Thus

$$\begin{aligned} \bar{A}^{k-1} &= \tilde{A}^{k-1} + E^{k-1} \\ &= R\bar{A}^k R^* + \frac{1}{|\alpha|^4}[G^*(D_1^kG + D_2^kH) + H^*D_3^kG], \end{aligned} \tag{4.5}$$

and we get the following data compression algorithm for the $N_0 \times N_0$ matrix A:

Set

$$\bar{A}^0 = A \tag{4.6a}$$

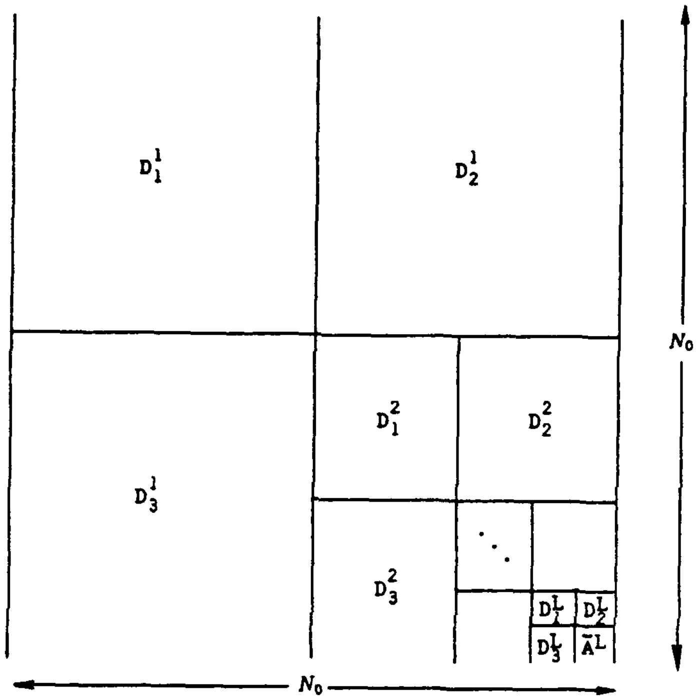
$$\left\{ \begin{aligned} &\text{Do for } k = 1, 2, \ldots, L \\ &\bar{A}^k = H\bar{A}^{k-1}H^* \\ &E^{k-1} = \bar{A}^{k-1} - R\bar{A}^k R^* \\ &D_1^k = GE^{k-1}G^*, \; D_2^k = GE^{k-1}H^*, \; D_3^k = HE^{k-1}G^*. \end{aligned} \right. \tag{4.6b}$$

The multiresolution representation A^{MR} of A is

$$A^{MR} = \{\bar{A}^L, (\{D_i^L\}_{i=1}^3, \dots, \{D_i^1\}_{i=1}^3)\}. \tag{4.7a}$$

It is convenient to store A^{MR} in the form

(4.7b) $A^{MR} =$



which also shows that the number of elements in A^{MR} is $(N_0)^2$, as in the original matrix $A = \bar{A}^0$.

Starting from the multiresolution representation A^{MR} (4.7), we recover the original matrix A by (4.5), i.e.

$$\begin{cases} \text{Do for } k = L, L-1, \dots, 1 \\ \bar{A}^{k-1} = R\bar{A}^k R^* + \frac{1}{|\alpha|^4}[G^*(D_1^k G + D_2^k H) + H^* D_3^k G], \end{cases} \tag{4.8a}$$

$$A = \bar{A}^0. \tag{4.8b}$$

The elements of $\{D_i^k\}_{i=1}^3$ are proportional to the local error in predicting $\bar{A}^{k-1}$ from the k-th level of resolution (4.3b). Therefore these elements are small wherever the discretized function is properly resolved on the k-th grid. Data compression can

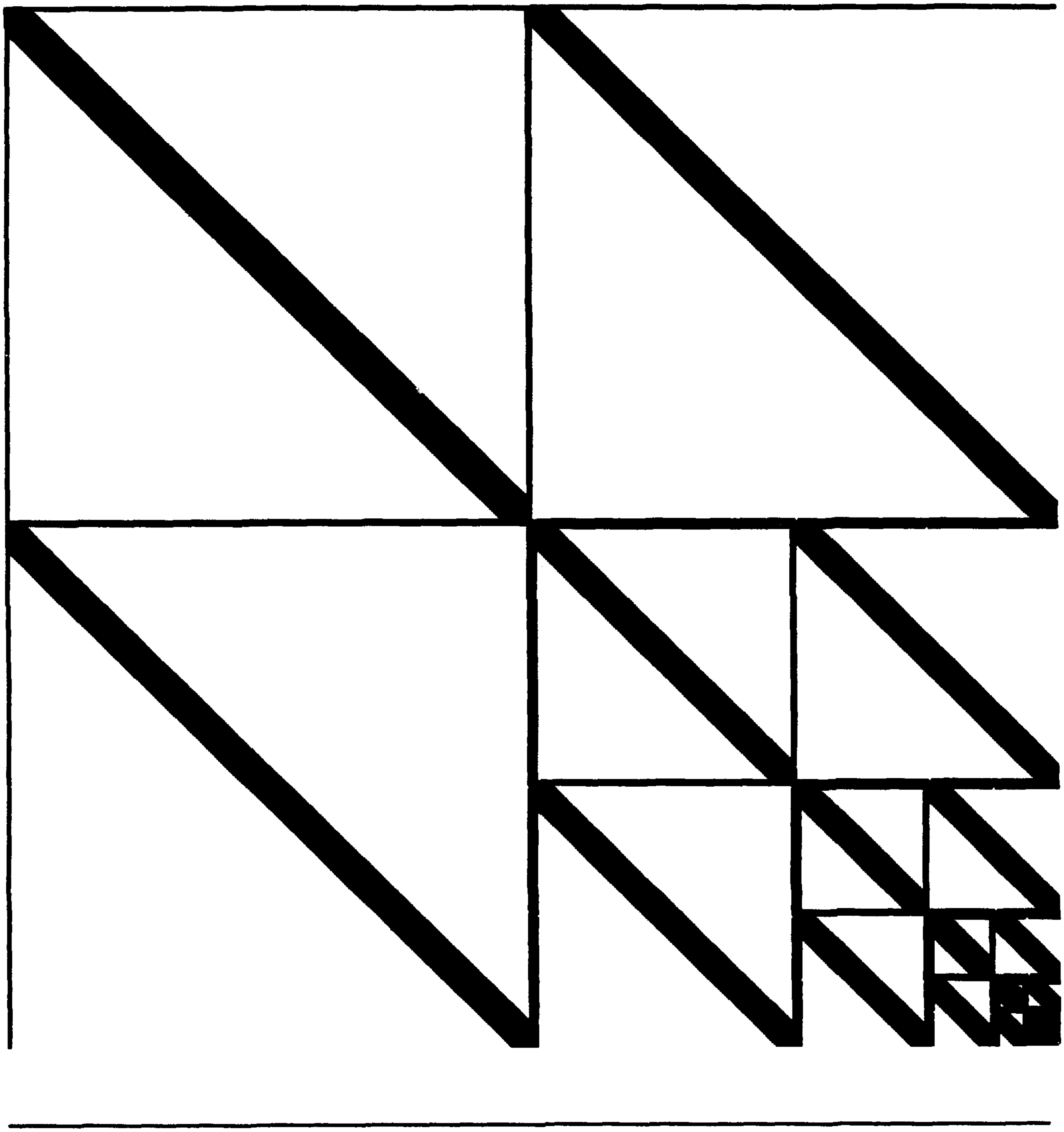


Figure 1a

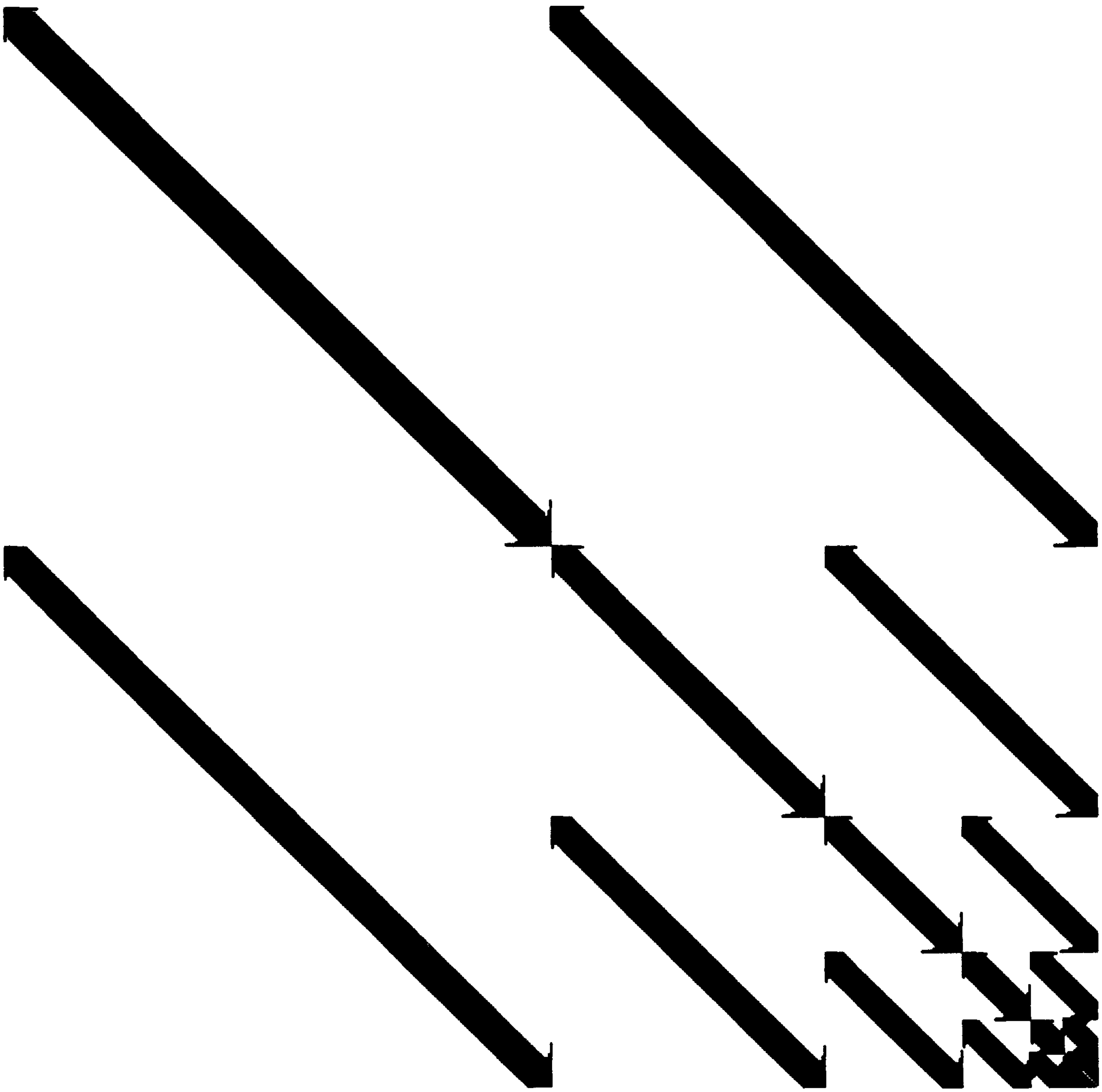


Figure 1b

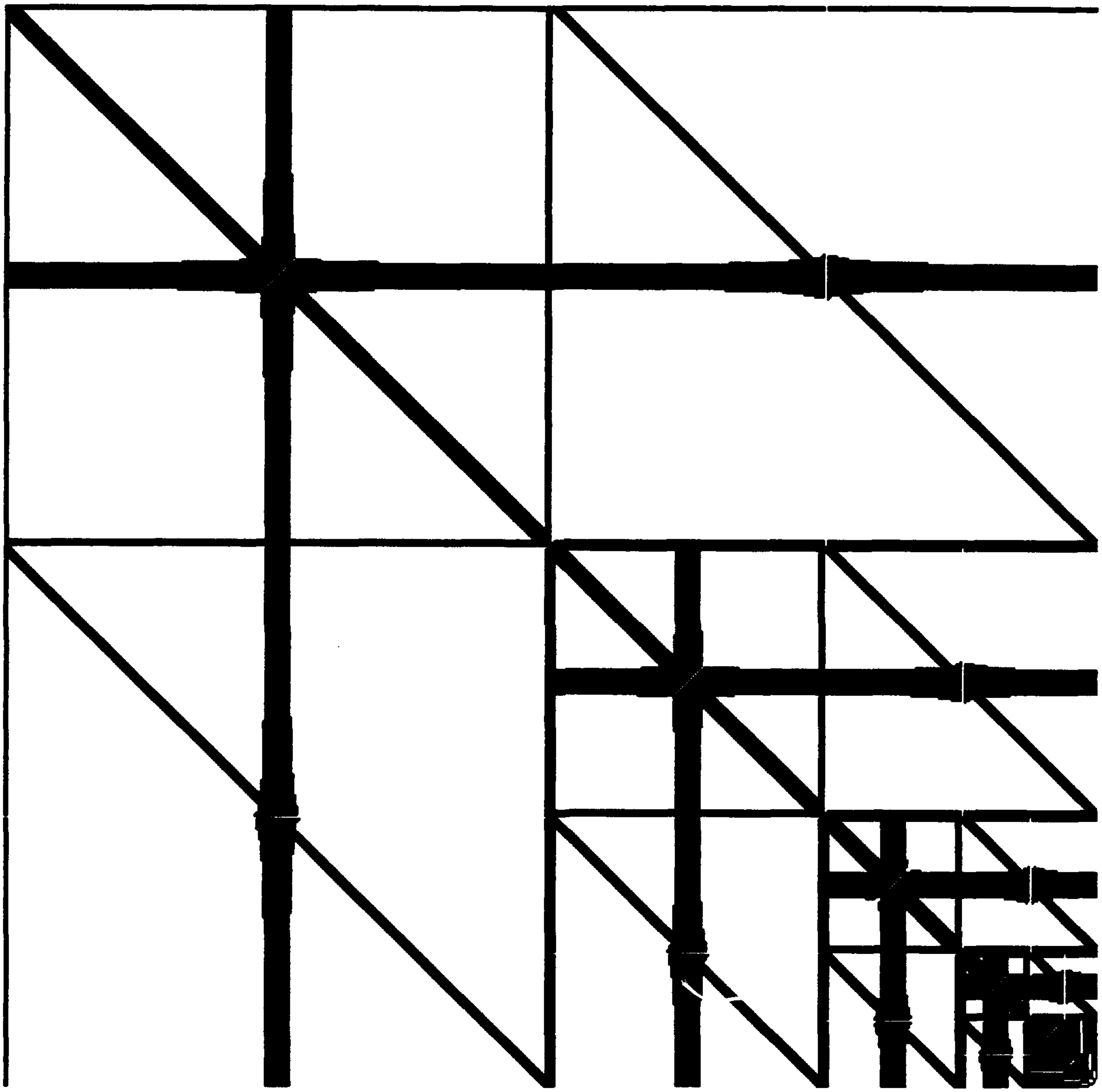


Figure 2a

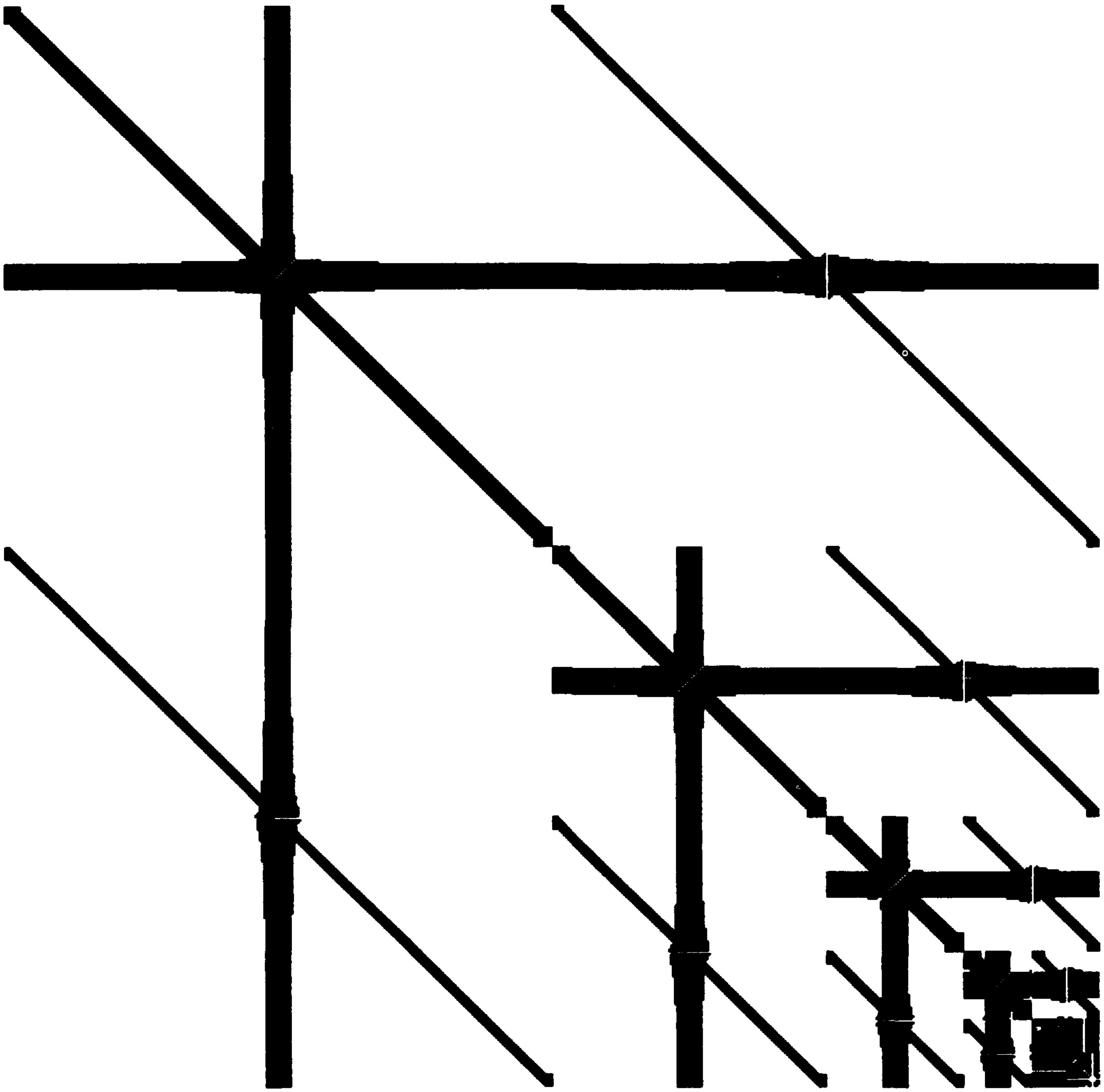


Figure 2b

be achieved by setting to zero elements of $\{D_i^k\}_{i=1}^3$ which are smaller in absolute value than some tolerance ε_k.

In Figures 1a,b and 2a,b we show results of data compression of two matrices which are the first two examples in the BCR paper [1]. In Figures 1a,b we show the multiresolution representation A^{MR} (4.7b) of the matrix

$$A_{ij} = \begin{cases} \frac{1}{i-j} & i \neq j \\ 0 & i = j \end{cases} \tag{4.9}$$

with $N_0 = 512$. The discretization in this calculation is assumed to be by pointvalues, i.e. H and G are (3.15) and the reconstruction is by interpolation. We take R to be (3.14) with $r = 6$. Entries of $\{D_i^k\}_{i=1}^3$ which are larger in absolute value than $\varepsilon_k = 10^{-7}$ are marked in black. The calculations in Figures 1a and 1b differ in the treatment of boundaries: In Figure 1a we use periodic boundary conditions while in Figure 1b we use one-sided interpolation near the boundaries. The compression rate (ratio between $(N_0)^2$ to the number of entries that are larger in absolute value than 10^{-7}) is 6.72 for the periodic case in Fig. 1a and 8.57 for the one-sided interpolation at boundaries in Fig. 1b; the compression rate for the wavelet based algorithm in [1] is 7.33.

In Figures 2a,b we repeat the calculations of Figures 1a,b for the matrix

$$A_{ij} = \begin{cases} \frac{\log|i-N_0/2| - \log|j-N_0/2|}{i-j} & \text{for } i \neq j, i \neq N_0/2, j \neq N_0/2 \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

Here the compression rates are 6.11 in Fig. 2a and 7.60 for Fig. 2b; the corresponding BCR result is 7.50.

We remark that the "BCR results" above are quoted from [1] in which a different normalization in (3.3) is used. These results show that the BCR compression rates are of the same order as the ones in Figures 1 and 2.

We turn now to describe how to compute the product $Ab = c$ (4.1) from the multiresolution representation A^{MR} (4.7b) of A. Multiplying (4.5) by a vector b^{k-1}

of N_{k-1} components we get

$$\bar{A}^{k-1}b^{k-1} = R\bar{A}^k(R^*b^{k-1}) + \frac{1}{|\alpha|^4}\{G^*[D_1^k(Gb^{k-1}) + D_2^k(Hb^{k-1})] + H^*D_3^k(Gb^{k-1})\}, \tag{4.11a}$$

from which we see that if for all k we define

$$b^k = R^*b^{k-1} \tag{4.11b}$$

$$c^k = \bar{A}^k b^k, \tag{4.11c}$$

then (4.11a) becomes

$$c^{k-1} = Rc^k + \frac{1}{|\alpha|^4}\{G^*[D_1^k(Gb^{k-1}) + D_2^k(Hb^{k-1})] + H^*D_3^k(Gb^{k-1})\}. \tag{4.12}$$

It follows therefore that given the (compressed) multiresolution representation A^{MR} (4.7b) of A we can calculate $c = Ab$ by:

Set

$$b^0 = b, \tag{4.13a}$$

$$\begin{cases} \text{Do for } k = 1, 2, \ldots, L \\ s^k = \frac{1}{|\alpha|^2}Hb^{k-1}, t^k = \frac{1}{|\alpha|^2}Gb^{k-1}, \\ b^k = R^*b^{k-1} \end{cases} \tag{4.13b}$$

evaluate by direct multiplication

$$c^L = \bar{A}^L b^L, \tag{4.13c}$$

and execute

$$\begin{cases} \text{Do for } k = L, L-1, \ldots, 1 \\ c^{k-1} = Rc^k + \frac{1}{|\alpha|^2}[G^*(D_1^k t^k + D_2^k s^k) + H^*(D_3^k t^k)], \end{cases} \tag{4.13d}$$

(4.13e) $$c = c^0.$$

Relation (4.11b) can be thought of as stating the proper scaling of the input vector as we go to a coarser grid. After preparing the values of b^k for all the levels (4.13b), we start the computation of $c = Ab$ by calculating its lowest resolution version $c^L = \bar{A}^L b^L$ in (4.13c). Then we proceed in (4.13d) to successively upgrade c^k by first using the reconstruction technique to predict the value $\tilde{c}^{k-1} = Rc^k$ for the finer grid and then correct this prediction wherever needed by the term in the curved brackets in the RHS of (4.12).

If the number of elements in $\{D_i^k\}_{i=1}^3$ that are larger in absolute value than the tolerance ε_k is $O(N_k)$, and the matrices H, G and R are banded (with constant width), then the number of operations for each k in (4.13b) and (4.13d) is $O(N_k)$, and consequently the number of operations in the multiplication algorithm (4.13) is $O(N_0)$.

It is important to observe that due to the tensor-product nature of this algorithm, the operations on the rows are independent of the operations on the columns. This enables us to use H_x, G_x and R_x on the left and different H_y, G_y and R_y on the right. Modifying the relations (4.2b), (4.3b) and (4.4b) to be

(4.2b)′ $$\bar{A}^k = H_x \bar{A}^{k-1} H_y^*,$$

(4.3b)′ $$E^{k-1} = \bar{A}^{k-1} - R_x \bar{A}^k R_y^*,$$

(4.4b)′ $$D_1^k = G_x E^{k-1} G_y^*,\ D_2^k = G_x E^{k-1} H_y^*,\ D_3^k = H_x E^{k-1} G_y^*,$$

we now get the following multiplication algorithm:

Set

$$(4.13a)' \qquad b^0 = b,$$

$$(4.13b)' \qquad \begin{cases} \text{Do for } k = 1, 2, \dots, L \\ s^k = \frac{1}{|\alpha|^2} H_y b^{k-1}, \; t^k = \frac{1}{|\alpha|^2} G_y b^{k-1}, \\ b^k = R_y^* b^{k-1} \end{cases}$$

$$(4.13c)' \qquad c^L = \bar{A}^L b^L,$$

$$(4.13d)' \qquad \begin{cases} \text{Do for } k = L, L-1, \dots, 1 \\ c^{k-1} = R_x c^k + \frac{1}{|\alpha|^2}[G_x^*(D_1^k t^k + D_2^k s^k) + H_x^*(D_3^k t^k)], \end{cases}$$

$$(4.13e) \qquad c = c^0.$$

This extra freedom in algorithm $(4.13)'$ can be utilized for example to discretize the integral transform (1.1) by pointvalues in x and cell-averages in y.

Next we present details for the three cases that we highlight in this paper.

Case 1. **Pointvalues.**

It follows form the definitions of H and G in (3.15) that (4.2b) and (4.3b) become

$$(4.14a) \qquad \bar{A}^k_{i,j} = \bar{A}^{k-1}_{2i,2j}, \;\; 1 \le i, j \le N_k,$$

$$(4.14b) \qquad (D_1^k)_{i,j} = E^{k-1}_{2i-1,2j-1}, \; (D_2^k)_{i,j} = E^{k-1}_{2i-1,2j}, \; (D_3^k)_{i,j} = E^{k-1}_{2i,2j}, \quad 1 \le i, j \le N_k.$$

Using the definition (3.14) of R in (4.13b) we get

$$b_i^k = (R^* b^{k-1})_i = b_{2i}^{k-1} + \sum_{\ell=1}^{s} \beta_\ell (b^{k-1}_{2(i+\ell)-1} + b^{k-1}_{2(i-\ell)-1}), \; 1 \le i \le N_k.$$

Algorithm (4.13) can be expressed in this case by:

Set

$$b^0 = b \tag{4.15a}$$

$$\begin{cases} \text{Do for } k = 1, 2, \dots, L \\ s_i^k = b_{2i}^{k-1}, \ t_i^k = b_{2i-1}^{k-1}, \quad 1 \le i \le N_k, \\ b_i^k = s_i^k + \sum_{\ell=1}^{s} \beta_\ell (t_{i+\ell}^k + t_{i-\ell}^k), \quad 1 \le i \le N_k, \end{cases} \tag{4.15b}$$

$$c^L = \bar{A}^L b^L, \tag{4.15c}$$

$$\begin{cases} \text{Do for } k = L, L-1, \dots, 1 \\ \text{Do for } i = 1, 2, \dots, N_k \\ c_{2i-1}^{k-1} = \sum_{\ell=1}^{s} \beta_\ell (c_{i+\ell}^k - c_{i-\ell}^k) + (D_1^k t^k + D_2^k s^k)_i \\ c_{2i}^{k-1} = c_i^k + (D_3^k t^k)_i, \end{cases} \tag{4.15d}$$

$$c = c^0; \tag{4.15e}$$

here $r = 2s$.

While writing this paper we found out that algorithm (4.15), although derived differently, had already been published in [2]. Moreover, it was extended further in [3] to integral transforms with an oscillatory kernel and to many-body problems.

Case 2. **Cell-averages.**

It is convenient to introduce the operators μ and δ,

$$\mu v_i = \frac{1}{2}(v_i + v_{i-1}), \quad \delta v_i = \frac{1}{2}(v_i - v_{i-1}), \tag{4.16}$$

and use the convention that, when applied to two-dimensional arrays, superscripts x and y denote operation on the first and the second index, respectively. It follows from the definition of H and G in (3.18) that (4.2b) and (4.3b) become

$$\bar{A}_{i,j}^k = \mu^x \mu^y \bar{A}_{2i,2j}^{k-1}, \quad 1 \le i, j \le N_k, \tag{4.17a}$$

$$(D_1^k)_{i,j} = \delta^x \delta^y E_{i,j}^{k-1}, \ (D_2^k)_{i,j} = \mu^y \delta^x \bar{E}_{i,j}^{k-1}, \ (D_3^k)_{i,j} = \mu^x \delta^y \bar{E}_{i,j}^{k-1}, \quad 1 \le i, j \le N_k. \tag{4.17b}$$

Using the definition (3.17) of R in (4.13b) we get

$$b_i^k = (R^* b^{k-1})_i = 2[\mu b_{2i}^{k-1} + \sum_{\ell=1}^{s} \gamma_\ell (\delta b_{2(i+\ell)}^{k-1} - \delta b_{2(i-\ell)}^{k-1})]. \tag{4.18}$$

Algorithm (4.13) can be expressed in this case by:

Set

$$b^0 = b \tag{4.19a}$$

$$\begin{cases} \text{Do for } k = 1, 2, \dots, L \\ s_i^k = 2\mu b_{2i}^{k-1}, \; t_i^k = 2\delta b_{2i}^{k-1}, \quad 1 \le i \le N_k, \\ b_i^k = s_i^k + \sum_{\ell=1}^{s} \gamma_\ell (t_{i+\ell}^k - t_{i-\ell}^k), \quad 1 \le i \le N_k, \end{cases} \tag{4.19b}$$

$$c^L = \bar{A}^L b^L, \tag{4.19c}$$

$$\begin{cases} \text{Do for } k = L, L-1, \dots, 1 \\ \text{Do for } i = 1, 2, \dots, N_k \\ w = c_i^k + (D_3^k t^k)_i \\ z = \sum_{\ell=1}^{s} \gamma_\ell (c_{i+\ell}^k - c_{i-\ell}^k) - (D_1^k t^k + D_2^k s^k)_i \\ c_{2i-1}^{k-1} = w + z \\ c_{2i}^{k-1} = w - z, \end{cases} \tag{4.19b}$$

$$c = c^0; \tag{4.19e}$$

here $r = 2s + 1$.

Case 3. **Orthogonal wavelets.**

In this case H and G are defined by (3.4) and (3.9a) and the Daubechies coefficients (see Table 1). Since $R = 2H^*$ (3.20b) and $HG^* = 0$ (3.9b) we get in (4.4b)

$$D_1^k = G\bar{A}^{k-1}G^*, \; D_2^k = G\bar{A}^{k-1}H^*, \; D_3^k = H\bar{A}^{k-1}G^*; \tag{4.20}$$

thus for $1 \leq i,j \leq N_k$

$$\bar{A}^k_{i,j} = \sum_{\ell=1}^{2r} \sum_{m=1}^{2r} \alpha_\ell \alpha_m \bar{A}^{k-1}_{2i+\ell,2j+m}, \tag{4.21a}$$

$$(D^k_1)_{i,j} = \sum_{\ell=1}^{2r} \sum_{m=1}^{2r} (-1)^{\ell+m} \alpha_\ell \alpha_m \bar{A}^{k-1}_{2i-1-\ell,2j-1-m}, \tag{4.21b}$$

$$(D^k_2)_{i,j} = \sum_{\ell=1}^{2r} (-1)^\ell \alpha_\ell \sum_{m=1}^{2r} \alpha_m \bar{A}^{k-1}_{2i-1-\ell,2j+m}, \tag{4.21c}$$

$$(D^k_3)_{i,j} = \sum_{m=1}^{2r} (-1)^m \alpha_m \sum_{\ell=1}^{2r} \alpha_\ell \bar{A}^{k-1}_{2i+\ell,2j-1-m}. \tag{4.21d}$$

Using $R = 2H^*$ in (4.13b) we get that

$$b^k = 2Hb^{k-1}, \quad s^k = b^k. \tag{4.22}$$

Algorithm (4.13) can be expressed in this case by:

Set

(4.23a) $b^0 = b$,

$$\text{(4.23b)} \begin{cases} \text{Do for } k = 1,2,\dots,L \\ \text{Do for } i = 1,2,\dots,N_k \\ b^k_i = 2\sum_{\ell=1}^{2r} \alpha_\ell b^{k-1}_{2i+\ell} \\ t^k_i = 2\sum_{\ell=1}^{2r} (-1)^\ell \alpha_\ell b^{k-1}_{2i-1-\ell}, \end{cases}$$

(4.23c) $c^L = \bar{A}^L b^L$,

$$\text{(4.23d)} \begin{cases} \text{Do for } k = L, L-1,\dots,1 \\ \text{Do for } i = 1,2,\dots,N_k \\ c^{k-1}_{2i-1} = 2\sum_{\ell=1}^{r} \{\alpha_{2\ell-1}[c^k_{i-\ell} + (D^k_3 t^k)_{i-\ell}] + \alpha_{2\ell}(D^k_1 t^k + D^k_2 b^k)_{i+\ell}\}, \\ c^k_{2i} = 2\sum_{\ell=1}^{r} \{\alpha_{2\ell}[c^k_{i-\ell} + (D^k_3 t^k)_{i-\ell}] - \alpha_{2\ell-1}(D^k_1 t^k + D^k_2 b^k)_{i+\ell}\}, \end{cases}$$

(4.23e) $c = c^0$.

Algorithm (4.23) is identical to the BCR algorithm (the "nonstandard form") in [1].

5. Stability and Efficiency.

In this section we examine the stability of the data compression algorithm (4.6)-(4.8) and discuss the efficiency of the matrix-vector multiplication algorithm (4.13).

From (4.3b) we get

$$\begin{aligned} \bar{A}^0 &= E^0 + R\bar{A}^1R^* = E^0 + RE^1R^* + R^2\bar{A}^2(R^2)^* = \cdots \\ &= E^0 + \sum_{k=1}^{L-1} R^kE^k(R^k)^* + R^L\bar{A}^L(R^L)^*. \end{aligned} \tag{5.1}$$

Applying data compression to A^{MR} (4.7) we get truncated matrices $\hat{E}^k$ which result in $\hat{A}^0$ in (5.1). Denoting

$$\mathcal{E}^k = \hat{E}^k - E^k \tag{5.2a}$$

we thus get

$$\hat{A}^0 - \bar{A}^0 = \mathcal{E}^0 + \sum_{k=1}^{L-1} R^k\mathcal{E}^k(R^k)^*, \tag{5.2b}$$

which shows that each column and row in $\mathcal{E}^k$ are amplified by R^k. For discretization in $[0,1]$ R^k in (5.1)-(5.2) should be interpreted as

$$R^k = R_1 \cdot R_2 \cdots R_k \tag{5.3a}$$

where R_m is the $2N_m \times N_m$ matrix in (3.5); for discretization in $(-\infty, \infty)$ R is an infinite matrix and R^k should be interpreted as the k-th power of R, i.e.

$$R^k = (R)^k. \tag{5.3b}$$

Let e denote the unit sequence corresponding to a partition of the real line into intervals of size 1 with integer endpoints,

$$e_\ell = \delta_{\ell,0}, \qquad -\infty < \ell < \infty,$$

and consider successive applications $R^k e$, $k \to \infty$. For example when R is the piecewise linear interpolation (3.14) with $r = 2$ we get

Table 2.

			$x=-1$								$x=0$								$x=1$				
e			0								1								0				
Re			0				$\frac{1}{2}$				1				$\frac{1}{2}$				0				0
R^2e	0		0		$\frac{1}{4}$		$\frac{2}{4}$		$\frac{3}{4}$		1		$\frac{3}{4}$		$\frac{2}{4}$		$\frac{1}{4}$		0		0		0
R^3e	0	0	0	$\frac{1}{8}$	$\frac{2}{8}$	$\frac{3}{8}$	$\frac{4}{8}$	$\frac{5}{8}$	$\frac{6}{8}$	$\frac{7}{8}$	1	$\frac{7}{8}$	$\frac{6}{8}$	$\frac{5}{8}$	$\frac{4}{8}$	$\frac{3}{8}$	$\frac{2}{8}$	$\frac{1}{8}$	0	0	0	0	0
$\vdots$																							

Clearly here

$$(R^k e)_j = \eta(2^{-k} j), \tag{5.4a}$$

$$\eta(x) = \begin{cases} 1 - |x| & |x| \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.4b}$$

We observe that $\eta(x)$, the "hat function", is the solution of the dilation equation

$$\eta(x) = \tfrac{1}{2}\eta(2x-1) + \eta(2x) + \tfrac{1}{2}\eta(2x+1), \tag{5.4c}$$

the coefficients of which are given by $a_\ell = (Re)_\ell$.

The limiting process $R^k e$, $k \to \infty$, has been studied by Deslauriers and Duboc [5] and Dyn, Gregory and Levin [6] for interpolating R, and by Daubechies [4] for orthonormal wavelets, $R = 2H^*$ (3.20b). As in the example above they found that the limiting process is convergent in the sense that

$$\lim_{k\to\infty} \sum_{j=-\infty}^{\infty} (R^k e)_j \mathcal{X}_{[0,1)}(2^k x - j) = \eta(x), \tag{5.5a}$$

where $\mathcal{X}_{[0,1)}$ is the characteristic function of $[0, 1)$ and the convergence is uniform in x. The limit $\eta(x)$ is a continuous function of compact support which satisfies the dilation equation

$$\eta(x) = \sum_{\ell} a_\ell \eta(2x - \ell), \qquad a_\ell = (Re)_\ell, \tag{5.5b}$$

and

$$\bar{\eta}_j^{-k} = \langle \eta, 2^k \varphi(2^k \cdot - j) \rangle = (R^k e)_j. \tag{5.5c}$$

Since $\eta(x)$ is continuous and of compact support it follows from (5.5a) that

$$\sup_k \{2^{-k} \sum_{j=-\infty}^{\infty} |(R^k e)_j|\} \le \text{const}, \qquad \sup_{j,k} |(R^k e)_j| \le \text{const}. \tag{5.6a}$$

Consequently we get for the matrix norms

$$\|R^k\|_\infty \le C_\infty, \quad \|R^k\|_1 \le 2^k \cdot C_1. \tag{5.6b}$$

We return now to the stability analysis (5.2) of the data compression algorithm. Setting to zero elements of D_i^k (4.7) which fall below the tolerance ε_k we get

$$|\mathcal{E}_{ij}^k| \le \text{const} \cdot \varepsilon_{k+1}, \tag{5.7a}$$

$$\|\mathcal{E}^k\|_p \le \hat{C} \cdot N_k \cdot \varepsilon_{k+1}, \quad p = 1, \infty. \tag{5.7b}$$

For each term in (5.2b) we now get for both the L_1 and L_∞ norms that

$$\begin{aligned} \|R^k \mathcal{E}^k (R^k)^*\| &\le \|R^k\|_\infty \|R^k\|_1 \cdot \hat{C} \cdot N_k \cdot \varepsilon_{k+1} \le \hat{C} C_\infty C_1 \cdot N_0 \cdot \varepsilon_{k+1} \\ &\equiv C \cdot N_0 \cdot \varepsilon_{k+1}, \end{aligned} \tag{5.8a}$$

and consequently

$$\frac{1}{N_0} \|\hat{A}^0 - \bar{A}^0\| \le C \cdot \sum_{k=1}^{L} \varepsilon_k; \tag{5.8b}$$

this shows the stability of the data compression algorithm.

In the numerical experiments shown in Figures 1a,b and 2a,b for the matrices (4.9) and (4.10) we have used $\varepsilon_k = \varepsilon = 10^{-7}$ (here $h_0 = 1$) and computed

$$\hat{\nu}_p(\varepsilon) = \|(\hat{A}^0 - A)b\|_p / \|b\|_p, \quad p = 1, \infty \tag{5.9}$$

for a randomly generated vector b; for purposes of comparison we also computed $\hat{\nu}_p(0)$ which corresponds to running the program with $\varepsilon = 0$ and thus shows the effect of round-off error. In Table 3 we show the results for the case where R is the interpolation (3.14) with $r = 6$.

Table 3.

case	boundary	ratio	$\hat{\nu}_1(10^{-7})$	$\hat{\nu}_\infty(10^{-7})$	$\hat{\nu}_1(0)$	$\hat{\nu}_\infty(0)$
(4.9)	periodic	6.72	6.95×10^{-6}	4.96×10^{-6}	1.09×10^{-7}	1.33×10^{-7}
	one-sided	8.57	7.52×10^{-6}	4.41×10^{-5}	9.34×10^{-7}	2.77×10^{-5}
(4.10)	periodic	6.11	1.62×10^{-6}	1.82×10^{-6}	4.76×10^{-8}	9.15×10^{-8}
	one-sided	7.60	1.46×10^{-6}	2.04×10^{-6}	6.46×10^{-7}	8.64×10^{-6}

It seems to us that the convergence of $R^k e$ to a continuous $\eta(x)$ stems from the conservation property $HR = I$ and the accuracy requirement (2.3a) with $r \geq 2$. Therefore we expect the reconstruction from cell-averages (3.17) to also satisfy the relations (5.5), (5.6). In Appendix B we prove convergence of the limiting process (5.5a) for reconstruction from cell-averages under the assumption that the corresponding limiting function for the interpolation in (2.11) is continuously differentiable.

In Table 4 we repeat the calculation of Table 3 for the reconstruction from cell-averages (3.17) with $r = 5$.

Table 4.

case	boundary	ratio	$\hat{\nu}_1(10^{-7})$	$\hat{\nu}_\infty(10^{-7})$	$\hat{\nu}_1(0)$	$\hat{\nu}_\infty(0)$
(4.9)	periodic	5.71	6.03×10^{-7}	7.83×10^{-7}	5.66×10^{-7}	4.39×10^{-7}
	one-sided	6.71	1.03×10^{-6}	1.55×10^{-6}	9.87×10^{-7}	9.52×10^{-7}
(4.10)	periodic	6.29	4.00×10^{-7}	5.97×10^{-7}	3.50×10^{-7}	3.06×10^{-7}
	one-sided	7.53	2.76×10^{-7}	6.09×10^{-7}	1.73×10^{-7}	3.06×10^{-7}

We turn now to discuss the question of efficiency. If $a(x, y)$ is a function that has isolated regions of large variation then its discretization on a uniform grid results in a matrix A which is actually over-resolved in most of the computational domain. In this case it pays to use multiresolution algorithms as they offer the efficiency of an adaptive grid method without the complicated logics that is associated with such a calculation. In applying multiresolution algorithms to matrix-vector multiplication there is another important consideration: The computational effort of preparing the representation A^{MR} (4.7) may be greater than a direct application of the matrix A to a single input vector b. Therefore it makes sense to use algorithm (4.13) only when the computational task calls for an application of the same matrix to many input vectors and/or there is apriori knowledge of the location of regions of large variation.

An important class of applications is the calculation of integral transforms (1.1)

$$u(x) = \int_0^1 K(x, y) v(y) dy, \tag{5.10}$$

where the kernel $K(x, y)$ is smooth except for curves $y_s(x)$ at which it has integrable singularity. To each grid of size h_k we associate a finite-dimensional approximation $K^k(x, y)$ to the kernel $K(x, y)$

$$K^k(x, y) = \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \bar{K}_{ij}^k \eta_i^k(x) \eta_j^k(y), \tag{5.11a}$$

$$\bar{K}_{ij}^k = \int_0^1 \int_0^1 K(x, y) \varphi_i^k(x) \varphi_j^k(y) dx dy, \tag{5.11b}$$

and a finite-dimensional approximation $u^k(x)$ to the output $u(x)$

$$u^k(x) = \int_0^1 K^k(x,y)v(y)dy = \sum_{i=1}^{N_k}(h_k \sum_{j=1}^{N_k} \bar{K}^k_{ij}\bar{v}^k_j)\eta^k_i(x), \tag{5.12a}$$

$$\bar{v}^k_j = \frac{1}{h_k}\int_0^1 v(y)\eta^k_j(y)dy. \tag{5.12b}$$

Here $\eta(x)$ is the limiting function in (5.5) and

$$\eta^k_\ell(z) = \eta(\frac{z}{h_k} - \ell). \tag{5.13a}$$

¿From (5.5c) with $k = 0$ we get that

$$\int \eta(x)\varphi(x-j)dx = \delta_{0,j} \tag{5.13b}$$

and thus by scaling

$$\langle \eta^k_i, \varphi^k_j \rangle = \delta_{i,j}. \tag{5.13c}$$

Using (5.13c) in (5.12a) we get

$$\bar{u}^k_i = h_k \sum_{j=1}^{N_k} \bar{K}^k_{ij}\bar{v}^k_j \qquad 1 \le i \le N_k, \tag{5.14a}$$

$$\bar{u}^k_i = \langle u^k, \varphi^k_i \rangle. \tag{5.14b}$$

Applying the data compression algorithm to the matrix $\bar{K}^0$ and setting to zero elements of $\{D^k_i\}$ that fall below ε_k we get from (5.8b) that

$$h_0\|\bar{K}^0 - \hat{K}^0\| \le C\sum_{k=1}^{L} \varepsilon_k. \tag{5.15a}$$

It follows therefore that

$$\|\bar{u}^0 - \hat{u}^0\| = \|h_0(\bar{K}^0 - \hat{K}^0)\bar{v}^0\| \le C(\sum_{k=1}^{L}\varepsilon_k)\|\bar{v}^0\|. \tag{5.15b}$$

The prediction error E_{ij}^{k-1} (4.3b) can be estimated by

$$|E_{ij}^{k-1}| \leq C_r(h_k)^r \kappa_{ij}^{(r)} = \frac{C_r}{B^r}[(Bh_k)^r \kappa_{ij}^{(r)}] \tag{5.16a}$$

where

$$\kappa_{ij}^{(r)} = (|\frac{\partial^r K}{\partial x^r}| + |\frac{\partial^r K}{\partial y^r}|)_{ij}. \tag{5.16b}$$

If the kernel $K(x, y)$ is such that $\kappa_{ij}^{(r)}$ at a distance Δ from the singularity can be bounded by

$$\kappa_{ij}^{(r)} \leq \frac{C_K}{\Delta^r} \tag{5.17a}$$

then it follows from (5.16) that except for a band of width B around the singularity, all the elements of E^{k-1} satisfy

$$|E_{ij}^{k-1}| \leq \frac{C_r C_K}{B^r}. \tag{5.17b}$$

Choosing $\varepsilon_k = \varepsilon$ in (5.15) and

$$B = (\frac{C_r C_K}{\varepsilon})^{\frac{1}{r}} \tag{5.18a}$$

we get that the number of nonzero elements in the compressed $\{\hat{D}_i^k\}_{i=1}^3$ (4.7) is proportional to $2BN_k$, and the cumulative error (5.15b) is

$$\|\bar{u}^0 - \hat{u}^0\| \leq C\|\bar{v}^0\|\varepsilon \log_2 N_0. \tag{5.18b}$$

This error can be made arbitrarily small by taking a wider band B in (5.18a).

Beylkin, Coifman and Rokhlin [1] point out that the estimate (5.17a) is satisfied by the kernels of Calderon-Zygmund operators and pseudo-differential operators. In this case, taking into account the actual decay of the prediction error away from the singularity to sharpen the estimates in (5.7), the $\log_2 N_0$ factor in the RHS of (5.18b) can be removed.

6. Summary and Conclusions.

In this paper we have presented a class of multiresolution algorithms for data compression and matrix-vector multiplication. In constructing this class we have introduced subclasses of different discretizations. Each subclass corresponds to a particular choice of $\varphi(x)$ in (2.2); $\varphi(x)$ is assumed to be a solution of a dilation equation and to satisfy the orthogonality condition (2.4c). Members of each subclass of discretization correspond to different reconstruction procedures $\mathcal{R}(x;\bar{f})$; the reconstruction is assumed to be conservative (2.3b) and to depend linearly on the discrete data $\bar{f}$.

We have paid special attention to the subclasses of discretization corresponding to pointvalues and cell-averages because of their simplicity. The wavelet based algorithms [1] are also included in this class but in a "diagonal" fashion: In each subclass of discretization corresponding to a $\varphi(x)$ which satisfies the moment condition (2.15), there is a wavelet based algorithm corresponding to the reconstruction $R = 2H^*$ (3.20b). For example the wavelet based algorithm for $r = 1$ (Haar basis) is in the subclass of cell-averages.

The rate of compression and the stability properties are about the same for all algorithms of this class with the same order of accuracy. What matters therefore in choosing an algorithm is simplicity, operational count and suitability to the particular application; under simplicity we also include handling of boundaries. Comparing wavelet based algorithms to those of pointvalues and cell-averages of the same order of accuracy r, we find the wavelet based algorithm to be considerably more expensive because of the larger support ($2r$) and lack of symmetry and that the handling of boundaries is not as simple. In comparing cell-averages to pointvalues we find cell-averages to be more suitable for discretization of kernels with integrable singularity.

References

[1] D. Beylkin, R. Coifman and V. Rokhlin, "Fast wavelet transforms and numerical algorithms. I", Comm. Pure Appl. Math., Vol. 44, pp. 141-183, 1991.

[2] A. Brandt and A.A. Lubrecht, "Multilevel matrix multiplication and fast solution of integral equations", J. Compu. Phys., Vol. 90, pp. 348-370, 1990.

[3] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernels", Computer Phys. Comm., Vol. 65, pp. 24-38, 1991.

[4] I. Daubechies, "Orthonormal bases of compactly supported wavelets", Comm. Pure Appl. Math., Vol. 41, pp. 909-996, 1988.

[5] G. Deslauriers and S. Duboc, "Symmetric iterative interpolation scheme", Constructive Approximation, Vol. 5, pp. 49-68, 1989.

[6] N. Dyn, J.A. Gregory and D. Levin, "Analysis of linear binary subdivision schemes for curve design", Constructive Approximation, Vol. 7, pp. 127-147, 1991.

[7] A. Harten, "Discrete multiresolution analysis and generalized wavelets", UCLA Computational and Applied Mathematics Report 92-08, February 1992.

[8] G. Strang, "Wavelets and dilation equations: A brief introduction", MIT Numerical Analysis Report 89-9, August 1989.

Appendix A.

Let $\mathbf{P}$ denote the symmetric matrix

$$(A.1a) \qquad \mathbf{P} = H^*H + G^*G$$

where H and G are (3.4) and (3.9) respectively. A direct calculation shows that

$$(A.1b) \qquad \mathbf{P}_{ij} = p(|i-j|)$$

where for m integer

$$(A.1c) \qquad \begin{cases} p(2m-1) = 0 \\ p(2m) = \sum_k \alpha_k \alpha_{k+2m} . \end{cases}$$

Let us assume now that $\mathbf{P}$ is an invertible matrix. It follows then from (3.8b) that

$$(A.2a) \qquad e^{k-1} = \mathbf{P}^{-1}\mathbf{P}e^{k-1} = \mathbf{P}^{-1}(H^*H + G^*G)e^{k-1} = \mathbf{P}^{-1}G^*Ge^{k-1}$$

$$= \mathbf{P}^{-1}G^*d^k$$

where

$$(A.2b) \qquad d^k = Ge^{k-1} .$$

Replacing relation (3.11) by (A.2) we get that the encoding part (3.13b) of the data compression algorithm (3.13) remains the same, but the decoding part (3.13d) becomes

$$(A.3) \qquad \begin{cases} \text{Do for } k = L, L-1, .., 1 \\ \overline{f}^{k-1} = R\overline{f}^{k} + \mathbf{P}^{-1}G^{*}d^{k} . \end{cases}$$

The orthogonality condition (2.4c) implies that

$$(A.4) \qquad \mathbf{P} = p(0)I = |\alpha|^2 I$$

which brings us back to (3.13d).

As an example for the nonorthogonal case let us consider the "hat function" $\varphi(x)$

$$(A.5a) \qquad \varphi(x) = \begin{cases} 1-|x| & 0 \le |x| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

which satisfies the dilation equation

$$(A.5b) \qquad \varphi(x) = \frac{1}{2}[\varphi(2x-1) + 2\varphi(2x) + \varphi(2x+1)].$$

In this case the only nonzero elements of $\mathbf{P}$ are

$$(A.6) \qquad \mathbf{P}_{i,i} = \frac{3}{8}, \quad \mathbf{P}_{i,i\pm 2} = \frac{1}{16}.$$

Thus $\mathbf{P}$ is diagonally dominant and hence invertible.

Appendix B.

In this appendix we use the interpolation results of [5] and [6] in order to prove convergence of the limiting process (5.5a) for cell-averages with the symmetric reconstruction (3.17).

Let $\tilde{R}$ denote the matrix (3.5) corresponding to the central interpolation (3.14) and let $\tilde{\eta}(x)$ denote the limit function in (5.5a). $\tilde{\eta}(x)$ has its support in $|x| \leq r-1$ where r is the order of accuracy of the interpolation. For $r = 2$, $\tilde{\eta}(x)$ is the "hat-function" (5.4b) which is only Lipschitz-continuous; for $r = 4, 6$, $\tilde{\eta}(x)$ is continuously differentiable.

Let S^m denote the "step-sequence"

$$S_j^m = \begin{cases} 0 & j \leq m-1 \\ 1 & j \geq m \,. \end{cases}$$

The limiting process corresponding to $\tilde{R}^k S^0$ is also convergent and we denote its limit by $\zeta(x)$. Since

$$e = S^0 - S^1$$

we get that

$$\tilde{\eta}(x) = \zeta(x) - \zeta(x-1) \,. \tag{B.1a}$$

It is easy to see that

$$\zeta(x) = \begin{cases} 0 & x \leq -r+1 \\ \sum_{l=0}^{2r-3} \tilde{\eta}(x-l) & -r+1 \leq x \leq r-2 \\ 1 & r-2 \leq x \end{cases} \tag{B.1b}$$

and thus $\zeta(x)$ has at least the same smoothness as $\tilde{\eta}(x)$.

We turn now to express the limiting process $R^k e$ for the reconstruction from cell-averages (3.17) in terms of $\zeta(x)$. From (5.5c) and (2.11) we get that

$$(B.2a) \qquad (R^k e)_j = \frac{\zeta(j2^{-k}) - \zeta((j-1)2^{-k})}{2^{-k}}.$$

Since $\zeta'(x)$ is continuous and of compact support we get that

$$(B.2b) \qquad \eta(x) = \lim_{k\to\infty} \sum_j (R^k e)_j \, \chi_{[(j-1)2^{-k},\, j2^{-k}]}(x) = \zeta'(x)$$

and that the convergence is uniform in x. From (B.1b) it follows that $\eta(x)$ has its support in $-r+1 \le x \le r-2$; from (B.1a) and (B.2b) we get that $\eta(x)$ is related to $\tilde{\eta}(x)$ by

$$(B.3) \qquad \tilde{\eta}'(x) = \eta(x) - \eta(x-1).$$

We remark that for $r = 2$ we get for all k that

$$\sum_j (R^k e)_j \, \chi_{[(j-1)2^{-k},\, j2^{-k}]}(x) \equiv \varphi(x)$$

where $\varphi(x)$ is the "box-function" (2.8a) (note that the order of accuracy of the reconstruction from the cell averages is $r-1$). Thus $\eta(x) = \varphi(x)$ and we get formal pointwise convergence of (B.2b) although $\eta(x)$ is discontinuous.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1992	Contractor Report

4. TITLE AND SUBTITLE
FAST MULTIRESOLUTION ALGORITHMS FOR MATRIX-VECTOR MULTIPLICATION

5. FUNDING NUMBERS
C NAS1-18605
C NAS1-19480

WU 505-90-52-01

6. AUTHOR(S)
Ami Harten
Itai Yad-Shalom

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 92-55

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-189721
ICASE Report No. 92-55

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Michael F. Card
Final Report

Submitted to SIAM Journal of Numerical Analysis

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited

Subject Category 64

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)
In this paper we present a class of multiresolution algorithms for fast application of structured dense matrices to arbitrary vectors, which includes the fast wavelet transform of Beylkin, Coifman and Rokhlin and the multilevel matrix multiplication of Brandt and Lubrecht. In designing these algorithms we first apply data compression techniques to the matrix and then show how to compute the desired matrix-vector multiplication from the compressed form of the matrix. In describing this class we pay special attention to an algorithm which is based on discretization by cell-averages as it seems to be suitable for discretization of integral transforms with integrably singular kernels.

14. SUBJECT TERMS
multiresolution analysis; fast matrix vector multiplication

15. NUMBER OF PAGES
43

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		