Notes on Wavelet-like Basis Matrices

Fu-Rong Lin*

May 24, 1999

Abstract

In this paper, we improve the algorithms for the construction of the wavelet-like basis matrix introduced by B. Alpert, *et. al.* [3]. It has been shown in [3] that the $n \times n$ wavelet-like basis matrix is of the form $U = U_l U_{l-1} \cdots U_1$, where $n = k2^l$ is the number of quadrature points and U_j, $j = 1, \cdots, l$ are sparse orthogonal matrices. In this paper, we prove that each U_j $(1 \le j \le l)$ can be represented by a $2k \times 2k$ matrix. It follows that the storage requirement for all matrices U_j is $4lk^2$. We also show that the cost of the construction of all matrices U_j can be reduced to $O(lk^3) = O(\log n \cdot k^3)$. We recall that in [3], the storage requirement and the construction cost of the matrix U are $4nk$ and $O(nk^2)$ respectively.

Key words. Wavelet-like bases, wavelet-like basis matrix

AMS subject classification. 65F99, 65R20

1 Background

In [3], a class of wavelet-like bases was constructed. In these bases, the dense matrices resulting from the discretization of the second-kind integral equations are transformed into sparse matrices. More precisely, the $n \times n$ matrices resulting from an n-point discretization are transformed into matrices with $O(n \log n)$ nonzero elements (to arbitrary finite precision). The inverse matrices are also sparse and are obtained in order $O(n \log^2 n)$ operations by Schulz method [5].

Let $n = k2^l$, where k and l are positive integers. Let $\{x_1, x_2, \cdots, x_n\} \subset \mathbb{R}$ be a set of n distinct points with $x_1 < x_2 < \cdots < x_n$. The wavelet-like basis defined on $\{x_1, x_2, \cdots, x_n\}$ has two fundamental properties:

1. all but k basis vectors have k vanishing moments; and
2. the basis vectors are nonzero on different scales.

*Department of Mathematics, Shantou University, Shantou 515063, Guangdong, People's Republic of China. Research was carried out during the visit to the Institute of Mathematical Sciences of the Chinese University of Hong Kong. Supported in part by Guangdong Provincial Natural Science Foundation of China No. 974007, HKRGC grant CUHK4207/97P and CUHK DAG grant 2060143.

As an illustration, we show a matrix of basis vectors for $n = 128$ and $k = 4$ in Figure 1. In Figure 1, each row represents one basis vector, with the dots depicting nonzero elements. The first k basis vectors are nonzero on $x_1, \cdots, x_{2k}$, the next k are nonzero on $x_{2k+1}, \cdots, x_{4k}$, and so forth. In all, one-half of the basis vectors are nonzero on $2k$ points from $\{x_1, x_2, \cdots, x_n\}$, one-fourth are nonzero on $4k$ points, one-eighth are nonzero on $8k$ points, etc. Each of these $n/2 + n/4 + \cdots + k = n - k$ basis vectors has k zero moments, i.e., if $b = (b_1, \cdots, b_n)$ is one of these vectors, then

$$\sum_{i=1}^{n} b_i x_i^j = 0, \qquad j = 0, 1, \cdots, k-1.$$

The final k vectors result from the orthogonalization of the moments $(x_1^j, x_2^j, \cdots, x_n^j)$ for $j = 0, 1, \cdots, k-1$.

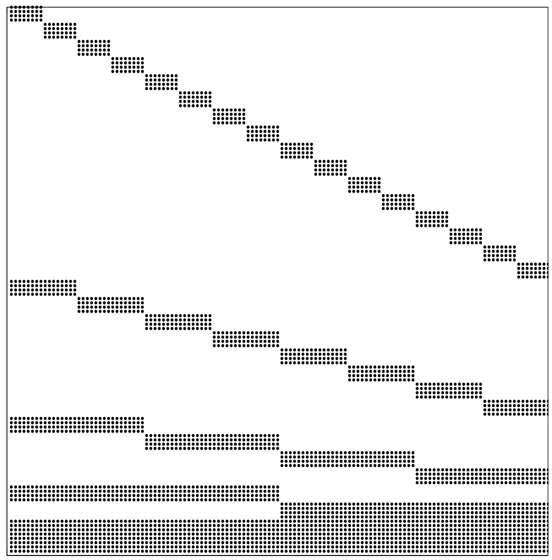

Figure 1. An example of wavelet-like basis matrix ($k = 4$, $n = 128$)

These properties of local support and vanishing moments lead to efficient representations of functions that are smooth except at a finite set of singularities, see [2]. The projection of such a function on an element of this basis will be negligible unless the element is nonzero near one of the singularities.

1.1 The Construction Procedure

In [3], it has been shown that the wavelet-like basis matrix has the form

$$U = U_l U_{l-1} \cdots U_1,$$

where the matrices U_j, $j = 1, 2, \cdots, l$ are sparse orthogonal matrices. Before we state the algorithm to construct the matrices $U_1, \cdots, U_l$, let us give some additional notation and show the sparse structure of U_j, $j = 1, 2, \cdots, l$.

(i) Suppose that V is a matrix whose columns $v_1, \cdots, v_{2k}$ are linearly independent. We define $W = Orth(V)$ to be the matrix that results from the column-by-column Gram-Schmidt orthogonalization of V. Namely, denoting the columns of W by $w_1, \cdots, w_{2k}$, we have

$$\text{span}\{w_1, \cdots, w_i\} = \text{span}\{v_1, \cdots, v_i\} \quad \text{and} \quad w_i^T w_j = \delta_{i,j},\ i, j = 1, 2, \cdots, 2k. \tag{1}$$

Here $\delta_{i,j}$ denotes the Kronecker symbol.

(ii) For a $p \times q$ matrix V, we let $V(i_1 : i_2, j_1 : j_2)$ denote the submatrix of V defined by

$$V(i_1 : i_2, j_1 : j_2) = \begin{pmatrix} v_{i_1,j_1} & v_{i_1,j_1+1} & \cdots & v_{i_1,j_2} \\ v_{i_1+1,j_1} & v_{i_1+1,j_1+1} & \cdots & v_{i_1+1,j+2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{i_2,j_1} & v_{i_2,j_1+1} & \cdots & v_{i_2,j_2} \end{pmatrix},$$

where $1 \leq i_1 \leq i_2 \leq p$ and $1 \leq j_1 \leq j_2 \leq q$. In particular, for a $2k \times 2k$ matrix V, we let $V(1 : k, :)$ and $V(k+1 : 2k, :)$ denote two $k \times 2k$ matrices, with $V(1 : k, :)$ consisting of the upper k rows and $V(k+1 : 2k, :)$ the lower k rows of V.

(iii) For a pair of numbers $(\mu, \sigma) \in \mathbb{R} \times (\mathbb{R} \backslash \{0\})$ we define a $2k \times 2k$ upper-triangular matrix $S(\mu, \sigma)$ whose (i, j)th element is the binomial term

$$S(\mu, \sigma)_{i,j} = \begin{pmatrix} j-1 \\ i-1 \end{pmatrix} \frac{(-\mu)^{j-i}}{\sigma^{j-1}} \tag{2}$$

for $i \leq j$ and $S(\mu, \sigma)_{i,j} = 0$ otherwise.

Now we show the sparse structure of the orthogonal matrices $U_1, \cdots, U_l$. The matrix U_1 is given by the formula

$$U_1 = \begin{pmatrix} U_{1,1}(k+1:2k,:) & & & \\ & U_{1,2}(k+1:2k,:) & & \\ & & \ddots & \\ & & & U_{1,n/(2k)}(k+1:2k,:) \\ U_{1,1}(1:k,:) & & & \\ & U_{1,2}(1:k,:) & & \\ & & \ddots & \\ & & & U_{1,n/(2k)}(1:k,:) \end{pmatrix},$$

where $U_{1,i}$, $i = 1, 2, \cdots, n/(2k)$ are $2k \times 2k$ orthogonal matrices. In general, for $j = 2, , \cdots, l$, we have

$$U_j = \begin{pmatrix} I_{n-n/2^{j-1}} & \\ & U_j' \end{pmatrix},$$

where I_p is the $p \times p$ identity matrix and

$$U'_j = \begin{pmatrix} U_{j,1}(k+1:2k,:) & & & \\ & U_{j,2}(k+1:2k,:) & & \\ & & \ddots & \\ & & & U_{j,n/(k2^j)}(k+1:2k,:) \\ U_{j,1}(1:k,:) & & & \\ & U_{j,2}(1:k,:) & & \\ & & \ddots & \\ & & & U_{j,n/(k2^j)}(1:k,:) \end{pmatrix}.$$

Here $U_{j,i}$ for $j = 2, \cdots, l$, $i = 1, 2, \cdots, n/(k2^j)$ are $2k \times 2k$ orthogonal matrices. The orthogonal matrices $U_{j,i}$ are obtained by the following Algorithm 1. For the details of the derivation of the algorithm, we refer the readers to [1, 3].

Algorithm 1: Computation of $U_{j,i}$, $j = 1, 2, \cdots, l$, $i = 1, 2, \cdots, n/(k2^j)$.

<u>Step 1</u>

For $i = 1, \cdots, n/(2k)$, compute $M_{1,i}$ by

$$M_{1,i} = \begin{pmatrix} 1 & \frac{x_{(i-1)\cdot 2k+1} - \mu_{1,i}}{\sigma_{1,i}} & \cdots & \left(\frac{x_{(i-1)\cdot 2k+1} - \mu_{1,i}}{\sigma_{1,i}}\right)^{2k-1} \\ 1 & \frac{x_{(i-1)\cdot 2k+2} - \mu_{1,i}}{\sigma_{1,i}} & \cdots & \left(\frac{x_{(i-1)\cdot 2k+2} - \mu_{1,i}}{\sigma_{1,i}}\right)^{2k-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \frac{x_{(i-1)\cdot 2k+2k} - \mu_{1,i}}{\sigma_{1,i}} & \cdots & \left(\frac{x_{(i-1)\cdot 2k+2k} - \mu_{1,i}}{\sigma_{1,i}}\right)^{2k-1} \end{pmatrix}, \tag{3}$$

where $\mu_{1,i} = (x_{(i-1)\cdot 2k+1} + x_{i\cdot 2k})/2$ and $\sigma_{1,i} = (x_{i\cdot 2k} - x_{(i-1)\cdot 2k+1})/2$.

<u>Step 2</u>

For $i = 1, \cdots, n/(2k)$, compute $U_{1,i}$ from $M_{1,i}$ by using the column-by-column Gram-Schmidt orthogonalization (1):

$$U_{1,i}^T = Orth(M_{1,i}). \tag{4}$$

<u>Step 3</u>

Compute $M_{j,i}$ and $U_{j,i}$ for $j = 2, 3, \cdots, l$ and $i = 1, 2, \cdots, n/(2^j k)$.
Do $j = 2, 3, \cdots, l$
Do $i = 1, 2, \cdots, n/(2^j k)$

3.1 Compute $\mu_{j,i}$ and $\sigma_{j,i}$ by

$$\mu_{j,i} = (x_{1+(i-1)k2^j} + x_{ik2^j})/2 \tag{5}$$

and

$$\sigma_{j,i} = (x_{ik2^j} - x_{1+(i-1)k2^j})/2. \tag{6}$$

Compute

$$\alpha_1 = \frac{\mu_{j,i} - \mu_{j-1,2i-1}}{\sigma_{j-1,2i-1}}, \quad \beta_1 = \frac{\sigma_{j,i}}{\sigma_{j-1,2i-1}}$$

and

$$\alpha_2 = \frac{\mu_{j,i} - \mu_{j-1,2i}}{\sigma_{j-1,2i}} \quad \beta_2 = \frac{\sigma_{j,i}}{\sigma_{j-1,2i}}.$$

3.2 Compute $S^1_{j,i} = S(\alpha_1, \beta_1)$ and $S^2_{j,i} = S(\alpha_2, \beta_2)$ respectively, where $S(\mu, \sigma)$ is defined as in (2).

3.3 Compute

$$M_{j,i} = \begin{pmatrix} U_{j-1,2i-1}(1:k,:) M_{j-1,2i-1} S^1_{j,i} \\ U_{j-1,2i}(1:k,:) M_{j-1,2i} S^2_{j,i} \end{pmatrix}. \tag{7}$$

3.4 Compute $U_{j,i}$ from $M_{j,i}$ by using the formula (1):

$$U^T_{j,i} = Orth(M_{j,i}). \tag{8}$$

Enddo

Enddo

The outline of this paper is as follows. In §2, we discuss the operation count of Algorithm 1. In §3, we first show that we can reduce at least 25% of the operations and then prove that each U_j, $j = 1, 2, \cdots, l$ can be represented by a $2k \times 2k$ orthogonal matrix.

2 The Construction Cost

In order to measure the improvement we can make, we discuss in detail the complexity of Algorithm 1. We only give the number of multiplications. We first discuss the cost of the column-by-column Gram-Schmidt orthogonalization (1), the costs of the computation of the matrix $S(\mu, \sigma)$ which is defined in (2) and the matrix $M_{j,i}$ ($j \geq 2$) which is defined in (7).

(1) *The cost of the column-by-column Gram-Schmidt orthogonalization.* In computing w_1, we require $2k+1$ multiplications to get $||v_1||_2$ and another $2k$ multiplications to get $v_1/||v_1||$. Thus, the cost for obtaining w_1 is $4k+1$. It is not difficult to see that the costs for getting w_2, w_3, $\cdots$ and w_{2k} are $2 \cdot 4k + 1$, $3 \cdot 4k + 1$, $\cdots$ and $2k \cdot 4k + 1$ respectively. Therefore, the total cost is

$$\sum_{i=1}^{2k} (4ik + 1) = 8k^3 + 4k^2 + 2k.$$

(2) *The cost of the construction of $S(\mu, \sigma)$.* We first compute three arrays $x = (0!, 1!, \cdots, (2k-1)!)$, $y = (\sigma^0, \sigma^1, \cdots, \sigma^{2k-1})$ and $z = ((-\mu)^0, (-\mu)^1, \cdots, (-\mu)^{2k-1})$. These three arrays

require $6k$ multiplications to obtain. We then compute the (i,j)th entry $(1 \le i \le j \le 2k)$ of the $2k \times 2k$ upper-triangular matrix $S(\mu,\sigma)$ in 4 multiplications by the formula

$$S(\mu,\sigma)_{i,j} = \frac{(j-1)!(-u)^{j-1}}{(j-i)!(i-1)!\sigma^{j-1}} = \frac{x(j)z(j-i+1)}{x(j-i+1)x(i)y(j)}.$$

Therefore, the total cost is

$$6k + 4 \cdot k(2k+1) = 8k^2 + 10k.$$

(3) *The cost of the computation of* $M_{j,i}$. The product of $k \times 2k$ matrix with $2k \times 2k$ matrix requires $4k^3$ multiplications. The product of $k \times 2k$ matrix with $2k \times 2k$ upper-triangular matrix requires $k(1+2+\cdots+2k) = 2k^3 + k^2$. Therefore, the total cost is

$$2(4k^3 + 2k^3 + k^2) = 12k^3 + 2k^2.$$

In Table 1, we provide the operation counts for each step of Algorithm 1.

Table 1. Computational Cost of Algorithm 1

Step	Complexity	Explanation
1	$\frac{n}{2k}(4k^2 - 2k + 2)$	There are $n/(2k)$ matrices $M_{1,i}$ and each $M_{1,i}$ requires $4k^2 - 2k + 2$ multiplications to construct.
2	$\frac{n}{2k}(8k^3 + 4k^2 + 2k)$	The column-by-column Gram-Schmidt orthogonalization of each $M_{1,i}$ requires $8k^3 + 4k^2 + 2k$ multiplications.
3.1	$6(\frac{n}{2k} - 1)$	There are $n/(4k) + n/(8k) + \cdots + 1 = n/(2k) - 1$ pairs of (j,i) $(j = 2, 3, \cdots, l,\ i = 1, 2, \cdots, n/(2^j k))$. Each pair requires 6 multiplications to get the parameters.
3.2	$(\frac{n}{k} - 2)(8k^2 + 10k)$	There are $2(n/(2k) - 1) = n/k - 2$ matrices $S(\mu,\sigma)$ and each $S(\mu,\sigma)$ requires $8k^2 + 10k$ multiplications.
3.3	$(\frac{n}{2k} - 1)(12k^3 + 2k^2)$	Each matrix $M_{j,i}$ requires $12k^3 + 2k^2$ multiplications to obtain.
3.4	$(\frac{n}{2k} - 1)(8k^3 + 4k^2 + 2k)$	The column-by-column Gram-Schmidt orthogonalization of each $M_{j,i}$ requires $8k^3+4k^2+2k$ multiplications.
total	$n(14k^2 + 15k + 11 + 4/k)$	Neglect low order term: $-(20k^3 + 22k^2 + 22k + 6)$.

3 Improvement

In this section, we show that we can save at least 25% of the work in the construction of the wavelet-like basis matrix U by simplifying the computation of $M_{j,i}$ for $j = 2, \cdots, l,\ i =$

$1, \cdots, n/(k2^j)$. We then prove that each U_j, $j = 1, 2, \cdots, l$ can be represented by a $2k \times 2k$ matrix.

We first note that the column-by-column Gram-Schmidt orthogonalization (1) is equivalent to the QR factorization. Let the QR factorization of V be given by $V = QR$, where Q is an orthogonal matrix and R is an upper-triangular matrix. We have

$$(v_1, v_2, \cdots, v_{2k}) = (q_1, q_2, \cdots, q_{2k}) \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,2k} \\ & r_{2,2} & \cdots & r_{2,2k} \\ & & \ddots & \\ & & & r_{2k,2k} \end{pmatrix},$$

where v_j and q_j are the jth-column of matrices V and Q respectively. It follows that

$$\begin{cases} v_1 = r_{1,1} q_1, \\ v_2 = r_{1,2} q_1 + r_{2,2} q_2, \\ \vdots \\ v_{2k} = r_{1,2k} q_1 + r_{2,2k} q_2 + \cdots + r_{2k,2k} q_{2k}. \end{cases}$$

That is,

$$\mathrm{span}\{q_1, \cdots, q_i\} = \mathrm{span}\{v_1, \cdots, v_i\} \quad \text{and} \quad q_i^T q_j = \delta_{i,j}, \quad i, j = 1, 2, \cdots, 2k.$$

Therefore, the QR factorization is equivalent to the column-by-column Gram-Schmidt orthogonalization. Hence, from (4) and (8) we have

$$M_{j,i} = U_{j,i}^T R_{j,i}, \quad j = 1, 2, \cdots, l, \quad i = 1, \cdots, n/(2^j k),$$

where $U_{j,i}$ and $R_{j,i}$ are orthogonal and upper-triangular matrices respectively. It follows that $U_{j,i} M_{j,i} = R_{j,i}$. In particular, we have $U_{j,i}(1:k,:) M_{j,i} = R(1:k,:)$. Thus by (7), we have

$$\begin{aligned} M_{j,i} &= \begin{pmatrix} U_{j-1,2i-1}(1:k,:) M_{j-1,2i-1} S_{j,i}^1 \\ U_{j-1,2i}(1:k,:) M_{j-1,2i} S_{j,i}^2 \end{pmatrix} \\ &= \begin{pmatrix} R_{j-1,2i-1}(1:k,:) S_{j,i}^1 \\ R_{j-1,2i}(1:k,:) S_{j,i}^2 \end{pmatrix}. \end{aligned} \tag{9}$$

We note that the matrices $R_{j-1,2i-1}(1:k, 1:k)$ and $S_{j,i}^p (p = 1, 2)$ are $k \times k$ and $2k \times 2k$ upper-triangular matrices respectively. It is not difficult to show that by using the formula (9), the multiplications required for constructing each $M_{j,i}$ are $7/3k^3 + 3k^2 + 2/3k$. Therefore, the multiplications required in Step 3.3 is reduced from $(n/(2k) - 1)(12k^3 + 2k^2)$ to $(n/(2k) - 1)(7/3k^3 + 3k^2 + 2/3k)$, i.e., we can save about $n(4.8k^2 - 0.5k - 0.3)$ multiplications, cf. Table 1. Since that the total multiplications in Algorithm 1 are less than $n(14k^2 + 15k + 11 + 4/k)$, we can save at least 25% of multiplications by using (9) for $k \geq 4$. We recall that the integer k is the degree of approximation polynomials and in general it is required that $k \geq 4$ to get an accurate approximation.

In the following, we assume that

$$x_{2ik+i'} = x_{i'} + ih, \quad i' = 1, \cdots, 2k \quad \text{and} \quad i = 1, 2, \cdots, n/(2k) - 1, \tag{10}$$

where $h = 2k/n$. We note that this assumption is quite general. For example, the quadrature points of compound quadrature rules such as the compound Newton-Cotes and the compound Gaussian rules satisfy (10). We will prove that for $j = 1, 2, \cdots, l$, the matrices $M_{j,i}$ only depend on the first index j. We first prove that the matrices $S^1_{j,i}$ and $S^2_{j,i}$ (cf. Step 3.1 and 3.2 of Algorithm 1) are independent of i.

Lemma 1 *Let the points $\{x_1, x_2, \cdots, x_n\}$ satisfy (10). Then we have*

$$\sigma_{j,i} = \frac{x_{2k} - x_1 + (2^{j-1} - 1)h}{2}$$

and

$$\mu_{j,i} - \mu_{j-1,2i-1} = 2^{j-3}h, \quad \mu_{j,i} - \mu_{j-1,2i} = -2^{j-3}h.$$

It follows that $S^1_{j,i}$ and $S^2_{j,i}$ are independent of i.

Proof. By the assumption on $\{x_1, x_2, \cdots, x_n\}$, we have

$$x_{1+(i-1)k2^j} = x_1 + (i-1)2^{j-1}h.$$

Note that $ik2^j = 2k + (i \cdot 2^{j-1} - 1)2k = 2k + [(2^{j-1} - 1) + (i-1) \cdot 2^{j-1}]2k$, we have

$$x_{ik2^j} = x_{2k} + (2^{j-1} - 1)h + (i-1)2^{j-1}h.$$

Therefore, by (5) and (6) we have

$$\mu_{j,i} = \frac{(x_{1+(i-1)k2^j} + x_{ik2^j})}{2} = \frac{x_1 + x_{2k} + (2^{j-1} - 1)h}{2} + (i-1)2^{j-1}h \tag{11}$$

and

$$\sigma_{j,i} = \frac{x_{ik2^j} - x_{1+(i-1)k2^j}}{2} = \frac{x_{2k} - x_1 + (2^{j-1} - 1)h}{2}. \tag{12}$$

Furthermore,

$$\begin{aligned}
& \mu_{j,i} - \mu_{j-1,2i-1} \\
= \; & \frac{x_1 + x_{2k} + (2^{j-1} - 1)h}{2} + (i-1)2^{j-1}h - \frac{x_1 + x_{2k} + (2^{j-2} - 1)h}{2} - (2i-2)2^{j-2}h \\
= \; & 2^{j-3}h.
\end{aligned}$$

Similarly,

$$\mu_{j,i} - \mu_{j-1,2i} = -2^{j-3}h.$$

Thus, $\sigma_{j,i}$, $\mu_{j,i} - \mu_{j-1,2i-1}$ and $\mu_{j,i} - \mu_{j-1,2i}$ are independent of i. It follows that the matrices $S^1_{j,i}$ and $S^2_{j,i}$ are independent of i. $\square$

Let μ_j and σ_j be defined by

$$\mu_j = \frac{\mu_{j,i} - \mu_{j-1,2i-1}}{\sigma_{j-1,2i-1}} = \frac{2^{j-2}h}{x_{2k} - x_1 + (2^{j-2} - 1)h} \tag{13}$$

and

$$\sigma_j = \frac{\sigma_{j,i}}{\sigma_{j-1,2i-1}} = \frac{x_{2k} - x_1 + (2^{j-1} - 1)h}{x_{2k} - x_1 + (2^{j-2} - 1)h} \tag{14}$$

respectively. We have $S^1_{j,i} = S(\mu_j, \sigma_j)$ and $S^2_{j,i} = S(-\mu_j, \sigma_j)$, i.e. they are independent of the index i. In the following, we will denote $S^1_{j,i}$ and $S^2_{j,i}$ by S^1_j and S^2_j respectively.

Theorem 1 *Let the points $\{x_1, x_2, \cdots, x_n\}$ satisfy (10). Then for each j, the matrices $M_{j,i}$ are the same for $i = 1, 2, \cdots, n/(2^j k)$. It follows that for each j, the matrices $U_{j,i}$ are the same for $i = 1, 2, \cdots, n/(2^j k)$.*

Proof. From (3) we have that

$$M_{1,i} = \begin{pmatrix} 1 & u_{i,1} & \cdots & u_{i,1}^{2k-1} \\ 1 & u_{i,2} & \cdots & u_{i,2}^{2k-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & u_{i,2k} & \cdots & u_{i,2k}^{2k-1} \end{pmatrix},$$

where $u_{i,i'} = (x_{(i-1)\cdot 2k+i'} - \mu_{1,i})/\sigma_{1,i}$. By (11) and (12), we have

$$u_{i,i'} = \frac{x_{i'} + (i-1)h - (x_1 + x_{2k})/2 - (i-1)h}{(x_{2k} - x_1)/2} = \frac{2x_{i'} - x_1 - x_{2k}}{x_{2k} - x_1}, \quad i' = 1, \cdots, 2k.$$

Obviously, $u_{i,i'}$ are independent of i. Therefore, all matrices $M_{1,i}$, $i = 1, 2, \cdots, n/(2k)$ are the same. We can denote them by M_1.

Let the QR factorization of $M_{1,i}$ be given by $M_{1,i} = U^T_{1,i} R_{1,i}$. We have that the matrices $U_{1,i}$ and $R_{1,i}$ are independent of i and can be denoted by V_1 and R_1 respectively. Note that the matrices $S^1_{j,i}$ and $S^2_{j,i}$ are independent of i, we see from (9) that the matrices $M_{2,i}$ are given by

$$M_{2,i} = \begin{pmatrix} R_{1,2i-1}(1:k,:)S^1_{1,i} \\ R_{1,2i}(1:k,:)S^2_{1,i} \end{pmatrix} = \begin{pmatrix} R_1(1:k,:)S^1_1 \\ R_1(1:k,:)S^2_1 \end{pmatrix}.$$

Thus the matrices $M_{2,i}$ are the same for $i = 1, \cdots, n/(4k)$ and can be denoted by M_2. It follows that $U_{2,i}$ and $R_{2,i}$, $i = 1, \cdots, n/(4k)$, can be denoted by V_2 and R_2 respectively.

Similarly, we have that

$$M_{3,i} = \begin{pmatrix} R_{2,2i-1}(1:k,:)S^1_{2,i} \\ R_{2,2i}(1:k,:)S^2_{2,i} \end{pmatrix} = \begin{pmatrix} R_2(1:k,:)S^1_2 \\ R_2(1:k,:)S^2_2 \end{pmatrix}$$

are independent of i. In general, for $j = 1, 2, \cdots, l$, the matrices $M_{j,i}$ are independent of i. $\Box$

From Theorem 1, we see that for each $j = 1, 2, \cdots, l$, we only require to compute a $2k \times 2k$ orthogonal matrix $V_j = U_{j,i},\ i = 1, 2, \cdots, n/(k2^j)$. Thus we have come up the following revised algorithm:

Algorithm 2: Computation of V_j, $j = 1, 2, \cdots, l$.

<u>Step 1</u>

Compute the matrix M_1 by

$$M_1 = \begin{pmatrix} 1 & u_1 & u_1^2 & \cdots & u_1^{2k-1} \\ 1 & u_2 & u_2^2 & \cdots & u_2^{2k-1} \\ \vdots & \vdots & & \vdots & \\ 1 & u_{2k} & u_{2k}^2 & \cdots & u_{2k}^{2k-1} \end{pmatrix},$$

where $u_i = (2x_i - x_{2k} - x_1)/(x_{2k} - x_1)$ for $i = 1, 2, \cdots, 2k$.

<u>Step 2</u>

Compute the QR factorization of M_1:

$$M_1 = V_1^T R_1;$$

<u>Step 3</u>

For $j = 2, 3, \cdots, l$,
Do

Compute μ_j and σ_j as defined in (13) and (14) respectively.

Compute $S_j^1 = S(\mu_j, \sigma_j)$ and $S_j^2 = S(-\mu_j, \sigma_j)$.

Compute the matrix M_j by the formula $M_j = \begin{pmatrix} R_{j-1}(1:k,:)S_j^1 \\ R_{j-1}(1:k,:)S_j^2 \end{pmatrix}$.

Compute the QR factorization of M_j: $M_j = V_j^T R_j$.

Enddo

Our algorithm 2 avoids redundant work and therefore the construction cost is minimized. By using the fact that $U_{j,i} = V_j$ for each $j = 1, 2, \cdots, l$, we see that the construction cost is reduced by $4n/(3l)$ times, cf. Tables 1–2 and the storage requirement is reduced from $4nk$ to $4lk^2$.

Table 2. Computational Cost

Item	Complexity	Explanation
M_1	$4k^2 - 2k$	The matrix M_1 requires $4k^2 - 2k$ multiplications to construct.
V_1 and R_1	$8k^3 + 4k^2 + 2k$	The column-by-column Gram-Schmidt orthogonalization of M_1 requires $8k^3 + 4k^2 + 2k$ multiplications.
S_j^1 and S_j^2	$2(l-1)(8k^2 + 10k)$	There are $2(l-1)$ matrices $S(\mu, \sigma)$ and each $S(\mu, \sigma)$ requires $8k^2 + 10k$ multiplications to obtain.
M_j	$(l-1)(\frac{7}{3}k^3 + 3k^2 + \frac{1}{3}k)$	Each matrix M_j requires $(7/3k^3 + 3k^2 + 1/3k)$ multiplications.
V_j and R_j	$(l-1)(8k^3 + 4k^2 + 2k)$	The column-by-column Gram-Schmidt orthogonalization of each M_j requires $8k^3 + 4k^2 + 2k$ multiplications.
total	$\leq l(31k^3 + 45k^2 + 37k)/3$	

References

[1] B. Alpert, *Wavelets and Other Bases for Fast Numerical Linear Algebra*, In Wavelet–A Tutorial in Theory and Applications, C.K. Chui (ed.), Academic Press, Inc., Boston, 1992, 181–216.

[2] B. Alpert, *A Class of Bases in L^2 for the Sparse Representation of Integral Operators*, SIAM. J. Math. Anal., 24(1993), 246–262.

[3] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, *Wavelet-like Bases for the Fast Solution of Second-Kind Integral Equations*, SIAM J. Sci. Comput., 14 (1993), 159–184.

[4] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., John Hopkins University Press, Baltimore, 1989.

[5] G. Schulz, *Iterative Berechnung der Reziproken Matrix*, Z. Angew. Math. Mech., 13 (1933), 57–59.