Technische Universität Chemnitz
Sonderforschungsbereich 393
Numerische Simulation auf massiv parallelen Rechnern

Zhanlav T.

Some choices of moments of refinable function and applications

Preprint SFB393/03-11

Preprint-Reihe des Chemnitzer SFB 393
ISSN 1619-7178 (Print) **ISSN 1619-7186** (Internet)

SFB393/03-11 **Juni 2003**

Contents

Author's addresses:

Zhanlav T.
National University of Mongolia
P.O Box 46/145
Ulaanbaatar 210646
e-mail address: zhanlav@chinggis.com

Some choices of moments of refinable function and applications *

T.Zhanlav

Abstract

We propose a recursive formula for moments of scaling function and sum rule. It is shown that some quadrature formulae has a higher degree of accuracy under proposed moment condition. On this basis we obtain higher accuracy formula for wavelet expansion coefficients which are needed to start the fast wavelet transform and estimate convergence rate of wavelet approximation and sampling of smooth functions.
We also present a direct algorithm for solving refinement equation.

AMS subject classification: 42C40, 42C15, 41A25, 65D32, 65D15.

Key words: wavelet approximation, quadrature rule, refinable function evaluation.

1 Introduction

This paper concerns the construction of compactly supported orthonormal wavelets.It is well known that the moment condition for kernel is essentially equivalent to good approximation properties.At present there are a numerous necessary and sufficient conditions for kernel moment condition [1,2,3]. In this paper we propose another necessary and sufficient condition for kernel moment condition,which is constructive in sense that using this we find recursively moments of the scaling function.These conditions also allow us to construct efficient quadrature formulas for evaluation of coefficients of the wavelet expansion,which are needed to start the fast wavelet transform and to estimate convergence rate of wavelet and sampling approximations for smooth functions.On the basis of the proposed moment conditions we suggested an unified algorithm for the exact evaluation of the refinable function and its derivatives.We organize our paper as follows.In section 2 is considered the construction of compactly supported scaling function that generates multiresolution analysis of $L_2(\mathbb{R})$. Section 3 deal with the construction of higher accuracy formulas for the wavelet expansion coefficients and sampling approximation.In section 4 is presented an unified and exact algorithm for evaluation of refinable function and its derivatives.Some numerical results are also presented.

*This work has been supported in part by the Deutsche Forschungsgemeinschaft.

2 Choice of moments. Approximation theorem

Let $\varphi(x)$ be a scaling function that generates a multiresolution analysis of $L_2(\mathbb{R})$ and $K(x, y)$ be a periodic projection kernel of the form

$$K(x, y) = \sum_k \varphi(x - k)\overline{\varphi(y - k)}. \qquad (2.1)$$

For a measurable function f define an operator associated with the kernel

$$K_j f(x) = \int_{-\infty}^{\infty} K_j(x, y) f(y) dy, \qquad (2.2)$$

where $K_j(x, y) = \frac{1}{h} K(\frac{x}{h}, \frac{y}{h}), h = 2^{-j}$.
Let us introduce some conditions on kernels used in the sequel [1,2,3]. Let $N \geq 0$ be an integer.

Condition $H(N)$. There exists an integrable function $F(x)$ such that $|K(x, y)| \leq F(x - y), \forall x, y \in \mathbb{R}$ and

$$\int |x|^N F(x) dx < \infty.$$

Condition $M(N)$ (moment condition). Condition $H(N)$ is satisfied and

$$\int K(x, y)(y - x)^s dy = \delta_{0s}, \forall s = 0, 1, ..., N, \forall x \in \mathbb{R}, \qquad (2.3)$$

where δ_{jk} is the Kronecker delta.
Condition $S(N)$. There exists a bounded non increasing function ϕ such that

$$\int \phi(|u|) du < \infty, \quad |\varphi(u)| \leq \phi(|u|)$$

and

$$\int \phi(|u|)|u|^N du < \infty.$$

In sequel we will assumed that condition $S(N)$ is satisfied.
It is well known that, condition $S(N)$ being satisfied, the condition $H(N)$holds as well, and the following quantities are well defined

$$M_n = \int \varphi(x) x^n dx, \qquad (2.4)$$

$$\mu_n(x) = \int K(x, y)(y - x)^n dy,$$

$$c_n(x) = \sum_k \varphi(x - k)(x - k)^n, n = 0, 1, ..., N.$$

Moreover [3] the following relations are equivalent

$$c_n(x) = c_n \quad (a.e), n = 0, 1, ..., N,$$
$$\mu_n(x) = \mu_n \quad (a.e), n = 0, 1, ..., N.$$

Each of these relations implies that

$$c_n = M_n, n = 0, 1, ..., N.$$

We assume that

$$\sum_k \varphi(x - k)(x - k)^n = M_n, n = 0, 1, ..., N. \tag{2.5}$$

The moment condition for the kernel is essentially equivalent to good approximation properties. Next theorem gives a necessary and sufficient condition for the condition $M(N)$.

Theorem 2.1 *The kernel* (2.1) *satisfies moment condition* $M(N)$ *if and only if* (2.5) *holds and*

$$\sum_{m=0}^{s} \begin{pmatrix} s \\ m \end{pmatrix} (-1)^m M_{s-m} M_m = \delta_{0s}, s = 0, 1, ..., N. \tag{2.6}$$

Proof. First, prove that (2.5) and (2.6) are a necessary condition. Let $K(x, y)$ be a kernel satisfying moment condition (2.3).
Then by Proposition 8.5 [3] we have

$$\sum_k \varphi(x - k)(x - k)^s = const = M_s, s = 0, 1, ..., N$$

If we take into account (1.1) and (1.5) the left hand side of (1.3) may be rewritten as

$$\int K(x, y)(y - x)^s dy = \sum_{m=0}^{s} \begin{pmatrix} s \\ m \end{pmatrix} (-1)^m M_{s-m} M_m = \delta_{0s}, s = 0, 1, ..., N.$$

The converse is obvious. $\square$

Obviously the condition (2.6) gives us relationship between moments M_i and using this we can determine all moments. It should be mentioned that the equality (2.6) turns out to identity for odd s. This means that moments with odd indices can be chosen as a free parameters, while the moments with even indices are given by

$$M_0 M_{2e} = -\sum_{m=1}^{2l} \begin{pmatrix} 2l \\ m \end{pmatrix} (-1)^m M_{2l-m} M_m, l = 1, 2, ... \tag{2.7}$$

We denote by m_i the discrete moments of $\{h_k\}$ i.e.,

$$m_i = \sum_k h_k k^i, i = 0, 1, ..., N, \tag{2.8}$$

where h_k is the coefficients of refinement equation

$$\varphi(x) = \sqrt{2} \sum_k h_k \varphi(2x - k). \tag{2.9}$$

To find this discrete moments we use well-known equality

$$M_0 m_l = (2^l - 1)\sqrt(2)M_l - \sum_{i=1}^{l-1} \left(\begin{array}{c} l \\ i \end{array} \right) m_{l-i} M_i, l = 1, 2, ..., N, m_0 = \sqrt(2). \tag{2.10}$$

Using (2.7) and (2.10) we can recursively determine all m_i. Since the moments M_{2l+1}, are free parameters in (2.7) we have possibility to obtain a set of refinable function.
For example, if we choose the odd moments by formula

$$M_{2i+1} = \left(\tfrac{M_1}{M_0} \right)^{2i+1} M_0, i = 0, 1, ...$$

Then from (2.7) and (2.10) immediately we obtain

$$M_i = \left(\tfrac{M_1}{M_0} \right)^i M_0, i = 0, 1, ..., N \tag{2.11}$$

and

$$m_i = \sqrt(2) \left(\tfrac{M_1}{M_0} \right)^i M_0, i = 0, 1, ..., N. \tag{2.12}$$

In generally we seek for m_i and M_i in the form

$$m_i = \sqrt(2) a_i \left(\tfrac{M_1}{M_0} \right)^i M_0, i = 0, 1, ..., N, a_0 = 1 \tag{2.13}$$

and

$$M_i = b_i \left(\tfrac{M_1}{M_0} \right)^i M_0, i = 0, 1, ..., N, b_0 = 1. \tag{2.14}$$

as (2.11) and (2.12) respectively.
Substituting (2.13) and (2.14) into (2.10) and (2.7) we get

$$a_l = 2^l b_l - \sum_{j=1}^{l} \left(\begin{array}{c} l \\ j \end{array} \right) a_{l-j} b_j, l = 1, 2, ..., N \tag{2.15}$$

and

$$b_l = -\sum_{m=1}^{2l} (-1)^l \left(\begin{array}{c} 2l \\ m \end{array} \right) b_m b_{2l-m}, l = 1, ... \tag{2.16}$$

Thus we obtain two recursive relations for determining the coefficients a_i and b_i.
Now we consider the construction of wavelet system.As is known, the moment condition $M(N)$ for kernel (2.1) is equivalent to

$$\int x^s \psi(x) dx = 0, s = 0, 1, ..., N \tag{2.17}$$

or

$$\sum_k \lambda_k k^s = 0, s = 0, 1, ..., N, \tag{2.18}$$

where λ_k are the coefficients of equation

$$\psi(x) = \sqrt(2) \sum_k \lambda_k \varphi(2x - k), \lambda_k = (-1)^{k+1} \overline{h}_{1-k}.$$

The equality (2.18) gives

$$\sum_k (2k)^s h_{2k} = \sum_k (2k+1)^s h_{2k+1}, s = 0, 1, ..., N. \tag{2.19}$$

We collect all necessary equations for determining coefficients h_k:

$$\sum_k \overline{h}_k h_{k+2l} = \delta_{0l}, l \in \mathbb{N}, \tag{2.20}$$

$$\frac{1}{\sqrt(2)} \sum_k h_k = 1, \tag{2.21}$$

$$\sum_k (2k)^i h_{2k} = \sum_k (2k+1)^i h_{2k+1} = \frac{\sqrt(2)}{2} a_i (\frac{M_1}{M_0})^i, i = 0, 1, ..., N. \tag{2.22}$$

Solving nonlinear system of equations (2.20), (2.21) and (2.22) we obtain coefficients of refinement equation (2.9). It should be mentioned that if we choose $M_1 = 0$ in (2.22) then the system of equations $(2.20) - (2.22)$ turns out to ones for a coiflet [6].
Now we consider approximation properties of the wavelet expansion. We denote by P_j the orthogonal projection operator onto V_j i.e.

$$P_j f(x) = \sum_k \alpha_{jk} \varphi_{jk}(x). \tag{2.23}$$

where α_{jk} are given by inner product

$$\alpha_{jk} = (f, \varphi_{jk}). \tag{2.24}$$

Theorem 2.2 *Assume that the conditions* (2.5) *and* (2.6) *hold and* φ *satisfies the condition* $S(N+1)$. *Then for* $f \in C^{N+1}$ *is valid:*

$$\|P_j f(x) - f(x)\| = O(h^{N+1}), h = 2^{-j}. \tag{2.25}$$

Proof. If we use of (2.24) then $P_j f$ can be rewritten as

$$P_j f(x) = 2^j \int f(y) \sum_k \varphi(2^j x - k) \overline{\varphi(2^j - k)} dy = \int K_j (\frac{x}{h}, \frac{y}{h}) f(y) dy. \tag{2.26}$$

Substituting the Taylor expansion for $f(x)$

$$f(y) = \sum_{m=0}^{N} \frac{f^{(m)}(x)}{m!} (y - x)^m + \frac{f^{(N+1)}(\xi)}{(N+1)!} (y - x)^{N+1}$$

into (2.26) we obtain

$P_j f(x) = \sum\limits_{m=0}^{N} \frac{f^{(m)}(x)}{m!} h^m (\int K(x,y)(y-x)^m dy) +$

$\frac{f^{(N+1)}(s)}{(N+1)!} h^{N+1} \int K(x,y)(y-x)^{N+1} dy = \sum\limits_{m=0}^{N} \frac{f^{(m)}(x)}{m!} h^m \delta_{0m} + O(h^{N+1}),$

in which we have used the condition (2.5), (2.6) and condition $S(N+1)$. This completes the proof of the theorem. $\square$

3 Quadrature formula based on the wavelet expansion

We consider a quadrature formula

$$I = \int\limits_{-\infty}^{\infty} f(x)\varphi(x)dx \approx Q[f(x)] = \sum_{k=0}^{r} w_k f(x_k), \tag{3.1}$$

where the weights w_k and abscissae x_k are to be determined. Here $\varphi(x)$ is the father wavelet that generates a multiresolution analysis of L_2. That is compactly supported scaling function $\varphi(x)$ satisfies the equation [1,2]

$$\varphi(x) = \sqrt{2}\sum_k c_k \phi(2x-k), \quad \sum |c_k|^2 < \infty \tag{3.2}$$

with finite number nonzero coefficients c_k and the system of function

$$\varphi_{jk}(x) = 2^{\frac{j}{2}}\varphi(2^j x - k), \quad k \in Z \tag{3.33}$$

forms a orthonormal system in space V_j. If the orthogonal projection operator onto V_j denote by P_j then it can be written as

$$P_J f(x) = \sum_k \alpha_{jk}\varphi_{jk}(x), \quad \alpha_{jk} = (f, \varphi_{jk}). \tag{3.4}$$

Recall that the degree of accuracy of the quadrature formula (3.1) is q if it yields the exact result for every polynomial of degree less than or equal to q.

As mentioned in [4], the quadrature formula is usually constructed by demanding that

$$Q[x^i] = M_i \quad \text{for} \quad O \le i \le q \tag{3.5}$$

which leads to an algebraic system with respect to unknowns w_k and x_k, $k = 0, \ldots, r$. Here M_i are the moments of the scaling function, i.e.,

$$M_i = \int\limits_{-\infty}^{\infty} x^i \varphi(x)dx. \tag{3.6}$$

The scaling function is usually normalized with $M_o = 1$.
Another way to construct a quadrature formula is to use of expansion (3.4), in which we dwell more detail. Let $f(x) \in C^{n+1}$. Then using Taylor expansion of $f(x)$ it is easy to show that

$$\alpha_{jk} = (f, \varphi_{jk}) = \sqrt{h}\left\{\sum_{m=0}^{N} \frac{f^{(m)}(kh)}{m!} h^m \cdot M_m + O(h^{N+1})\right\}, \quad h = 2^{-j}. \tag{3.7}$$

Here and in sequence we assume

$$\sum_k \varphi(x-k)(x-k)^S = M^S, \quad S = 0, 1, \ldots, N. \tag{3.8}$$

We consider next quadrature formula

$$I \approx Q[f(x)] = \frac{1}{\sqrt{2}} \sum_k c_k f\Big(\frac{k+M}{2}\Big) \tag{3.9}$$

with known weights $w_k = \frac{c_k}{\sqrt{2}}$ and abscissae $x_k = \frac{(k+M)}{2}$. We have

Theorem 3.1 *Let $f(x) \in C^{N+1}$ and the condition (3.8) is fulfilled. Then the error of quadrature formula (3.9) is of order $O(h^{N+\frac{3}{2}})$ i.e.,*

$$\int_{-\infty}^{\infty} f(x)\varphi(x)dx - \frac{1}{\sqrt{2}} \sum_k c_k f\Big(\frac{k+M}{2}\Big) = O(h^{N+\frac{3}{2}}), \quad h = 0.5. \tag{3.10}$$

Proof. Taking into account the condition (3.8) we can write M_i in the form.

$$M_i = \sum_k \int_k^{k+1} \varphi(x)x^i dx = \int_0^1 \Big(\sum_k \varphi(x+k)(x-k)^i\Big)dx = M^i, \;\; i = 0, 1, \ldots, N. \tag{3.11}$$

Thereby, from (3.7) we get

$$\alpha_{jk} = \sqrt{h}\left\{\sum_{m=0}^{N} \frac{f^{(m)}(kh)}{m!}(hM)^m + O(h^{N+1})\right\} =$$

$$= \sqrt{h}\{f(k+M)h + O(h^{N+1})\}, \;\; h = \frac{1}{2^j}. \tag{3.12}$$

On the other hand, using scaling equation (3.2) and (3.4) in integral I, we have

$$I = \sqrt{2} \sum_k c_k \int_{-\infty}^{\infty} f(x)\varphi(2x-k)dx = \sum_k c_k \alpha_{1k}. \tag{3.13}$$

Substituting α_{1k} from (3.12) into (3.13) we obtain

$$I = \frac{1}{\sqrt{2}} \sum_k c_k f\Big(\frac{(k+M)}{2}\Big) + O(h^{N+\frac{3}{2}})$$

which completes the proof of theorem. □

Theorem 3.2 *The degree of accuracy of the quadrature formula* (3.9) *equal to* N *under condition* (3.8)*, i.e., the equality* (3.5) *holds for* $i = 0, 1, \ldots, N$.
Proof. Set $f(x) \equiv x^i$, $i = 0, 1, \ldots, N$ in (3.9). Then by definition of moments and assumption (3.8) the left hand side of (3.9) gives (3.11), i.e.,

$$M_i = M^i, \ \ i = 0, 1, \ldots, N.$$

The right hand side of (3.9) gives

$$Q[x^i] \equiv \frac{1}{\sqrt{2}} \sum_k c_k (k+M)^i h^i = \frac{h^i}{\sqrt{2}} \sum_k c_k [k^i + C_i^1 k^{i-1} M + C_i^2 k^{i-2} M^2 + \cdots + C_i^{i-1} k M^{i-1} + M^i].$$

By virtue of lemma [5] we have

$$\sum_k c_k k^i = \sqrt{2} M^i, \ \ i = 0, 1, \ldots, N.$$

Then

$$Q[x^i] = h^i M^i (1 + C_i^1 + C_i^2 + \cdots + C_i^{i-1} + 1) = h^i M^i 2^i = M^i = M_i,$$

which completes the proof of theorem 3.2. □
Note that the proposed quadrature formula (3.9) is stable. Indeed from normalization condition $M_o = 1$ it follows that

$$\sum_k w_k = \frac{1}{\sqrt{2}} \sum_k c_k = 1.$$

Although among these coefficients c_k may be occurred negative ones, the following uniform bound holds:

$$\frac{1}{\sqrt{2}} \sum_k |c_k| < C, \tag{3.14}$$

where C is constant not depending on k, due to (3.2). The inequality (3.14) proves the stability of formula (3.9).
By virtue of (3.4) we have

$$\alpha_{0,0} = (f, \varphi_{0,0}) = \int_{-\infty}^{\infty} f(x)\varphi(x)dx = I. \tag{3.15}$$

Therefore in order to calculate the integral (3.1) or (3.15) with higher accuracy than formula (3.9) we need to use of well-known Mallat algorithm:

$$\alpha_{mo} = \sum_e c_e \alpha_{m+1,e}, \quad m = j, j-1, \ldots, 0. \tag{3.16}$$

At the high level j we can use the approximate formula (3.12), i.e.,

$$\alpha_{j+1,e} \approx 2^{-\frac{j+1}{2}} f((l+M)h), \quad h = 2^{-(j+1)} \tag{3.17}$$

with accuracy $O(h^{N+\frac{3}{2}})$. By this algorithm the integral (3.15) will be evaluated with higher accuracy.

It should be mentioned that in [4] were proposed some quadrature formula with r degree of accuracy. But these algorithms lead to solve nonlinear system of $(r+1)$ unknowns, which is ill-conditioned. Unlike this the formula (3.9) is a more simple one with known weights and abscissae. Moreover we can construct the extension of formula (3.9). Indeed, it is easy to show that

$$\varphi_{jo}(x) = \sum_k c_k \varphi_{j+1,k}(x). \tag{3.18}$$

As above, we can consider

$$I_j = \int_{-\infty}^{\infty} f(x)\varphi_{jo}(x)dx = \sum_k c_k \alpha_{j+1,k} \tag{3.19}$$

By virtue of (3.12) we obtain

$$I_j = \int_{-\infty}^{\infty} f(xh)\varphi(x)dx \approx \frac{1}{\sqrt{2}} \sum_k c_k f\Big(\frac{k+M}{2}h\Big), \quad h = 2^{-j} \tag{3.20a}$$

or

$$\int_{-\infty}^{\infty} f(x)\varphi(2^j x)dx \approx \frac{h}{\sqrt{2}} \sum_k c_k f\Big(\frac{k+M}{2}h\Big). \tag{3.20b}$$

Quadrature formula (3.20) with $j = 0$ leads to (3.9). Obviously, the theorem 3.1 and theorem 3.2 remain true also for (3.20) with $h = 2^{-j}$.

Obviously,the smaller the number of abscissae r ,the more efficient the quadrature formula (3.1) since the number of function evaluations and algebraic operations for one coefficient is proportional to r. Therefore the idea of a one-point quadrature is attractive because of its simplicity. However, its degree of accuracy is limited. More precisely the degree of accuracy of the one-point formula

$$I = \int_{-\infty}^{\infty} f(x)\varphi(x)dx \approx Q\Big[f(x)\Big], x_1 = M_1 \tag{3.21}$$

is two, when φ is an orthogonal scaling function [4].
It is also known, that for the coiflets with N vanishing moments

$$M_p = 0, \qquad 1 \leq p \leq N \tag{3.22}$$

the one-point formula (3.21) with $x_1 = 0$ has a degree of accuracy of N. This conclusion is also true for (3.21) with a scaling function $\varphi(x)$, satisfying the condition (3.8). Indeed, setting $f(x) = x^s$ in (3.21) and using (3.11) we see that the left and right hand sides of (3.21) are equal to each other for $s = 0, 1, ..., N$.
Therefore we can use the one-point quadrature formula (3.21) for an evaluation of the wavelet coefficients

$$\alpha_{j+1,l} = (f, \varphi_{j+1,l}) = 2^{-\frac{j+1}{2}} \int_{-\infty}^{\infty} f\Big(\frac{y+l}{2}\Big)\varphi(y)dy \approx 2^{-\frac{j+1}{2}} f\Big(\frac{M+l}{2^{j+1}}\Big).$$

That is we arrive at formula (3.17). This means that the approximate formula (3.17) may be considered as a result of using one-point quadrature formula (3.21) for evaluation $\alpha_{j+1,l}$.
Let $M_i = c, i = 1, ..., N$. Then from (3.7) it follows

$$\alpha_{jk} = \sqrt(h)[(1-c)f(kh) + cf((k+1)h)] + O(h^{N+\frac{3}{2}}). \tag{3.23}$$

Thus we have a two-point quadrature formula:

$$\alpha_{jk} = \int\limits_{-\infty}^{\infty} f(x)\varphi_{jk}(x)dx \approx \sqrt(h)\sum_{k=1}^{2}\omega_k f(x_k); h = 2^{-j}, \tag{3.24}$$

where $\omega_1 = 1-c; \omega_2 = c, x_1 = kh, x_2 = (k+1)h.$
Using (3.18), (3.19) and (3.23) we easy to seen that

$$I_j = \int\limits_{-\infty}^{\infty} f(xh)\varphi(x)dx \approx \frac{1}{\sqrt(2)}\sum_k((1-c)c_k + cc_{k-1})f(kh), h = 2^{-j} \tag{3.25}$$

or

$$\int\limits_{-\infty}^{\infty} f(x)\varphi(2^j x)dx \approx \frac{h}{\sqrt(2)}\sum_k((1-c)c_k + cc_{k-1})f\Big(\frac{k}{2}h\Big) \tag{3.26}$$

with error $O(2^{-(j+1)(N+\frac{3}{2})})$.
When $j = 0$ the formula (3.26) leads to

$$I = \int\limits_{-\infty}^{\infty} f(x)\varphi(x)dx \approx \frac{1}{\sqrt(2)}\sum_k((1-c)c_k + cc_{k-1})f\Big(\frac{k}{2}\Big) \tag{3.27}$$

with error $O(2^{-(N+\frac{3}{2})})$.
Now we consider the trapezoidal rule

$$I = \int\limits_{-\infty}^{\infty} f(x)\varphi(x)dx \approx Q[f(x)] = \sum_k f(k)\varphi(k) \tag{3.28}$$

It is easy to show that the degree of accuracy of the rule (3.28) is equal to N, if the scaling function satisfies condition (2.5).
Then we use this rule for evaluating the coefficients of the wavelet expansion

$$\alpha_{jk} = 2^{\frac{j}{2}} \int\limits_{-\infty}^{\infty} f(x)\varphi(2^j x - k)dy = 2^{-\frac{j}{2}} \int\limits_{-\infty}^{\infty} f((y+k)h)\varphi(y)dy \approx 2^{-\frac{j}{2}} \sum_l f((l+k)h)\varphi(l).$$

Thus we have approximate expression for coefficients:

$$\widehat{\alpha}_{jk} = 2^{-\frac{j}{2}} \sum_l f((l+k)h)\varphi(l), \quad h = 2^{-j}. \tag{3.29}$$

This means that $\widehat{\alpha}_{jk}$ represents a discrete convolution of f and φ. Now we are interested in error estimate of this approximate expression (3.29). Assume that $f \in C^{N+1}$. Then using Taylor expansion

$$f((k+l)h) = \sum_m \frac{f^{(m)}(kh)}{m!}(lh)^m + O(h^{N+1})$$

and (2.10) we obtain

$$\widehat{\alpha}_{jk} = 2^{-\frac{j}{2}} \sum_{m=0}^{N} \frac{f^{(m)}(kh)}{m!}(h)^m M_m + O(h^{N+\frac{3}{2}}). \tag{3.30}$$

Comparing this with (3.7) we conclude that

$$\widehat{\alpha}_{jk} = \alpha_{jk} + O(h^{N+\frac{3}{2}}), h = 2^{-j}. \tag{3.31}$$

In the case of $M_i = (\frac{M_1}{M_0})^i M_0$ for $i = 0, 1, ..., N$ the formula (3.30) yields

$$\widehat{\alpha}_{jk} = \sqrt(h) M_0 f((k + \frac{M_1}{M_0})h) + O(h^{N+\frac{3}{2}}). \tag{3.32}$$

Thus in this case we can use a simple formula (3.32) instead of (3.29). It is easy to seen that the direct application of formula (3.29) to the evaluation of I gives

$$I = \int\limits_{-\infty}^{\infty} f(x)\varphi(x)dx = \sum_k c_k \alpha_{1k} \approx \sum_k c_k \frac{1}{\sqrt(2)} \sum_l f\Big(\frac{k+l}{2}\Big)\varphi(l). \tag{3.33}$$

Of course, the degree of accuracy of this formula (3.33) is N and

$$I - \frac{1}{\sqrt(2)} \sum_k c_k \sum_l f\Big(\frac{k+l}{2}\Big)\varphi(l) = O(h^{N+\frac{3}{2}}) = O(2^{-(N+\frac{3}{2})}).$$

As before, if the accuracy of this formula is not satisfactory then we can use the Mallat algorithm:

$$\alpha_{jk} = \sum c_{m-2k}\alpha_{j+1,m}, j = j-1, ..., 0$$

and using formula (3.29) at higher level $j+1$.
As a result we obtain integral I with higher accuracy $O(2^{-j(N+\frac{3}{2})})$.

Let $R(x)$ be an error of wavelet expansion (2.23) ,i.e.,

$$R(x) = f(x) - \sum_k \alpha_{jk}\varphi_{jk}(x). \tag{3.34}$$

We consider two method for determining coefficients of the decomposition.

(i) Interpolation.

Setting $x = (l+n)h, \quad h = 2^{-j}$ in (3.34) we require that

$$R((l+n)h) = 0, l \in \mathbb{Z}, \tag{3.35}$$

which is a system of equations

$$2^{\frac{j}{2}} \sum_k \alpha_{jk}\varphi(l+n-k) = f((l+n)h), l \in \mathbb{Z}. \tag{3.36}$$

Since $\varphi(x)$ is a scaling function with compacted support then the last system turns out to a finite system of linear equations with respect to coefficients α_{jk} . The system (3.36) can be solved by elimination method.

(ii) Discrete Galerkin method

We require that

$$\sum_l R((l+n)h)\varphi(l) = 0. \tag{3.37}$$

This is a discrete version of Galerkin method.
The last system can be rewritten as

$$\sum_k \alpha_{jk}(\sum_l \varphi(l+n-k)\varphi(l)) = 2^{-\frac{j}{2}} \sum_l \varphi(l) f((l+n)h). \tag{3.38}$$

Again we obtain a system of equations w.r.t α_{jk}.
The solution of this system can be found explicitly by formula

$$\alpha_{jk} = 2^{-\frac{j}{2}} \sum_l \varphi(l) f((l+n)h) \tag{3.39}$$

if

$$\sum_{l} \varphi(l+l')\varphi(l) = \delta_{0l'}. \tag{3.40}$$

The last condition can be considered as a discrete version of the orthogonality condition for $\varphi(x)$. Thus we again arrive at formula (3.29). But in this case we have an exact formula (3.39).
Now we are interested in approximation properties of sampling approximation

$$Sf(x) = \sum_{k} \widehat{\alpha}_{jk}\varphi_{jk}(x), \tag{3.41}$$

where the coefficients of this expression are given by (3.29). From (3.41) it clear the sampling operator S is local when $\varphi(x)$is compactly supported and it is easy to implement.
As before, we assume that $f \in C^{N+1}$ and the conditions $(2.5), (2.6)$ hold and $\varphi(x)$ satisfies the condition $S(N+1)$ .Using Taylor expansion for $f(x)$ and condition $(2.5), (2.6)$ in (3.41) it is easy to show that

$$\|Sf(x) - f(x)\| = O(h^{N+1}), h = 2^{-j}. \tag{3.42}$$

Obviously the estimate (3.42) is also valid for (3.41) with coefficients given by (3.17) and (3.23). From the estimate (3.42) it is clear that the sampling operator (3.41) is a quasi-interpolating one [4]. That is it produces every polynomials of degree less or equal to N.
Note that estimate of type (3.42) for a coiflet was proven by J.Tian and R.O. Wells,Jr [6,10,11] and for generalized coiflet by E-B Lin and X Zhou [10,11].
If $f(x)$ is continuous then from (3.29) it follows that

$$2^{\frac{j}{2}}\alpha_{jk} \to f(kh) \quad \text{as} \quad j \to \infty.$$

Then it is easy to show that

$$\| \sum_{k} f(kh)\varphi(2^j x - k) - f(x)\| = O(h^{\alpha)} \tag{3.43}$$

provided that φ satisfies condition $S(1)$ and f is Hölder continuous with exponent α.

4 Evaluation of the values of scaling function and its derivatives

As is known the wavelet basis in the multiresolution analysis is defined by translation and dilations of the scaling function $\phi(x)$ and wavelet function $\psi(x)$. So it is often needed the values of the function $\phi(x)$ at any points. However no explicit expression of the scaling function is available. There are many algorithms for numerical evaluating of values of the scaling function. (see, for example, [1], [2]). Another method for computing the values $\phi(x)$ at integer points was

given by A. Garba.
In [5,8] the required values of $\phi(x)$ are obtained as a solution of eigensystem. The disadvantage of this method is that it has a time consuming for the higher dimension case of the eigensystem. In this section we present an exact method for computing values of scaling function at integer as well as at dyadic points.

Assume that the scaling function has a support in the interval $[0, \quad N-1]$, where N is positive integer, and satisfies the relation

$$\phi(x) = \sum_{k=0}^{N-1} c_k \phi(2x-k), \;\; c_k = \sqrt{2} h_k. \tag{4.1}$$

Equation (4.1) is called the refinement equation. Since $supp(\phi) \subseteq [0, N-1]$, we only need to compute the values ϕ at the integer points $i = 1, 2, \ldots, N-2$.
This is also true for the derivatives because they are also compactly supported, as can be seen by deriving the scaling equation (4.1). If we use the denotation $\Phi^{(n)} = (\phi_1^{(n)}, \phi_2^{(n)}, \ldots, \phi_{N-2}^{(n)})^T$ then $\Phi^{(n)}$ satisfies an eigensystem [5].

$$H\Phi^{(n)} = \lambda \Phi^n, \quad \lambda = 2^{-n}, \tag{4.2}$$

where the $(N-2)$ by $(N-2)$ matrix H has a form:

$$H = \begin{pmatrix} c_1 & c_0 & 0 & 0 & \ldots & 0 & 0 \\ c_3 & c_2 & c_1 & c_0 & & & \\ \vdots & \vdots & c_3 & c_2 & & & \\ c_{N-1} & c_{N-2} & \vdots & \vdots & \ldots & 0 & 0 \\ 0 & 0 & c_{N-1} & c_{N-2} & \ldots & c_1 & c_0 \\ \vdots & \vdots & 0 & 0 & \ldots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & c_{N-1} & c_{N-2} \end{pmatrix} \tag{4.3}$$

The vector $\Phi^{(n)}$ is an eigenvector corresponding to the known eigenvalue $\lambda = 2^{-n}$. It is understood that the derivative of order 0 corresponds to the scaling function $\phi(x)$. The coefficients h_k are given by Daubechies in [1].
In order to find vector $\Phi^{(n)}$ we use, in addition Eq.(4.2), a partition of unity satisfied by ϕ, i.e.,

$$\sum_k \phi(x-k) \equiv 1 \qquad \text{for} \qquad \forall x \in \mathbb{R}. \tag{4.4}$$

Differentiating both side of (4.4) and setting $x = 2i$, for $i = 1, 2, \ldots, N-2$ in the obtained equation, we have

$$\sum_{i=1}^{N-2} \phi_i^{(n)} = \delta_{on}, \quad n = 0, 1 \ldots, \tag{4.5}$$

where δ_{on} is a Kronecker symbol. The eigensystem (4.2) together with (4.5), as in [5], may be solved by numerical methods. Since the eigenvalue of this system is known, it is useful to consider this as an system of linear algebraic equations. Dividing both side of Eq.(4.2) by $2^{-n}\phi_1^{(n)} \neq 0$ and ignoring one of these equations involving all the unknowns we obtain the nonhomogeneous linear system of $(N-3)$ equations with $(N-3)$ unknowns $\bar{\phi}^{(n)} = \frac{\phi_i^{(n)}}{\phi_1^{(n)}},\ i = 2, 3, \ldots, N-2$. This system can be solved, for instance, by Gaussian elimination method.
After this, substituting

$$\phi_i^{(n)} = \phi_1^{(n)} \cdot \bar{\phi}_i^n,\ \ i = 2, \ldots, N-2 \tag{4.6}$$

in equations (4.5), we have

$$\phi_1^{(n)}(1 + \bar{\phi}_2^{(n)} + \bar{\phi}_3^{(n)} + \cdots + \bar{\phi}_{N-2}^{(n)}) \equiv \delta_{on}. \tag{4.7}$$

From this we find

$$\phi_1 = \frac{1}{1 + \bar{\phi}_2 + \bar{\phi}_3 + \cdots + \phi_{N-2}}, \quad \text{when} \quad n = 0 \tag{4.8}$$

The remainder values of ϕ_i are defined by formula (4.6) for $n = 0$.
Thus for $n = 0$ the values of $\phi(x)$ at integer points determined completely.
Substituting $x_k = kh,\ \ h = 2^{-j}$ in equation (4.1), it is easy to show that

$$\phi(\frac{k}{2^j}) = \sum_{m=0}^{N-1} c_m \phi(\frac{k}{2^{j-1}} - m),\ \ j = 0, 1, \ldots \tag{4.9}$$

$$\phi(\frac{k}{2^j} + \frac{1}{2^{j+1}}) = \sum_{m=0}^{N-1} c_m \phi(\frac{k}{2^{j-1}} + \frac{1}{2^j} - m),\ \ j = 0, 1, \ldots. \tag{4.10}$$

Thus the values of ϕ at the dyadic points are founded recursively by relations $(4.9), (4.10)$. For examples, we have

$$\phi(k + \frac{l}{2^{j+1}}) = \sum_{m=0}^{N-1} c_m \phi(2k + \frac{l}{2^j} - m),\ \ l = 1, 2, \ldots, 2^{j+1} - 1,\ \ j = 0, 1, 2, \ldots \tag{4.11}$$

For $n \geq 1$ from (4.7) it is evident that the eigensystem (4.2) has a nonzero solution if and only if

$$1 + \bar{\phi}_2^{(n)} + \bar{\phi}_3^{(n)} + \cdots + \bar{\phi}_{N-2}^{(n)} = 0. \tag{4.12}$$

In this case the value $\phi_1^{(n)}$ is remains unknown. To find $\phi_1^{(n)}$ we need to use of next lemmas.

Lemma 4.1 *Let ϕ be C^n scaling function satisfying conditions*

$$\sum_k \phi(x-k)(x-k)^S = \int \phi(x) x^S dx = M_S,\ \ s = 0, 1, \ldots, N. \tag{4.13}$$

Then

$$\sum_k \phi^{(m)}(x-k)(x-k)^S = (-1)^m s(s-1)(s-2)\dots(s-m+1)M_{S-m}, \tag{4.14}$$

$$s = 0, 1, \dots, N, \ \ m = 0, 1, 2, \dots, n$$

Proof. We will prove (4.14) by induction. For $m = 0$ the equality (4.14) becomes (4.13). Differentiating (4.13) we get

$$\sum_k \phi'(x-k)(x-k)^S = -s\sum_k \phi(x-k)(x-k)^{S-1}, \ \ s = 0, 1, \dots, N.$$

By virtue of (4.13), we have

$$\sum_k \phi'(x-k)(x-k)^S = -sM_{S-1}, \ \ s = 0, 1, \dots, N.$$

This proved the relation (4.14) for $m = 1$. Suppose the relation (4.14) is valid for $m = 1, 2, \dots, j$. We prove (4.14) for $m = j+1$. To this end differentiating relation (4.13) $(j+1)$-times, we get

$$\sum_k \Big\{ \sum_{l=0}^{j+1} \left(\begin{array}{c} j+1 \\ l \end{array} \right) \phi^{j+1-l}(x-k)((x-k)^S)^{(l)} \Big\} = 0.$$

From this we find

$$\sum_k \phi^{(j+1)}(x-k)(x-k)^S$$

$$= -\sum_k \Big\{ \sum_{l=1}^{j+1} \left(\begin{array}{c} j+1 \\ l \end{array} \right) \phi^{(j+1-l)}(x-k)s(s-1)\dots(s-l+1)(x-k)^{S-l} \Big\}$$

$$= -\sum_{l=1}^{j+1} \left(\begin{array}{c} j+1 \\ l \end{array} \right) s(s-1)\dots(s-l+1)\Big\{ \sum_k \phi^{(j+1-l)}(x-k)(x-k)^{S-l} \Big\}. \tag{4.15}$$

By induction for $l = 1, \dots, j$ holds next relation

$$\sum_k \phi^{(j+1-l)}(x-k)(x-k)^{S-l} = (-1)^{j+1-l}(s-l)(s-l-1)\dots(s-j)M_{S-j-1}. \tag{4.16}$$

Substituting (4.16) into (4.15) we get

$$\sum_k \phi^{(j+1)}(x-k)(x-k)^S = (-1)^{j+1}s(s-1)\dots(s-j)M_{S-j-1}\sum_{l=1}^{j+1} \left(\begin{array}{c} j+1 \\ l \end{array} \right) (-1)^{l+1}.$$

Since $\sum\limits_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) (-1)^k = 0$ we have

$$\sum_{l=1}^{j+1} \left(\begin{array}{c} j+1 \\ l \end{array} \right) (-1)^{l+1} = -\sum_{l=1}^{j+1} \left(\begin{array}{c} j+1 \\ l \end{array} \right) (-1)^{l} \pm 1 = 1 - \sum_{l=0}^{j+1} \left(\begin{array}{c} j+1 \\ l \end{array} \right) (-1)^{l} = 1$$

This completes the proof of Lemma 4.1. $\square$

Lemma 4.2 *Assume the same conditions as in Lemma* 4.1.
Then

$$\phi_1^{(n)} + 2^n\phi_2^{(n)} + 3^n\phi_3^{(n)} + \cdots + (N-2)^n\phi_{N-2}^{(n)} = (-1)^n n!. \tag{4.17}$$

Proof. Take $m = s = n$ in (4.14). Then we have

$$\sum_k \phi^{(n)}(x-k)(x-k)^n = (-1)^n n!. \tag{4.18}$$

Since $supp\phi^{(n)} \subseteq [0, N-1]$ as $\phi(x)$ the summation on k in (4.18) is run from $k = i - N + 2$ to $\dot{k} = i - 1$. Setting $x = i$ in (4.18) we get (4.17). $\square$
From (4.17) we find

$$\phi_1^{(n)} = \frac{(-1)^n n!}{1 + 2^n\bar{\phi}_2^n + 3^n\bar{\phi}_3^n + \cdots + (N-2)^n\bar{\phi}_{N-2}^n}. \tag{4.19}$$

For $n = 0$ the formula (4.19) coincides with (4.8).
Remark 4.1 It is easy to show that the formula (4.19) is also valid for Daubechies scaling function for $n = 0, 1$ and for $n = 2$ since $M_2 = M_1^2$.
Remark 4.2 If $supp\phi \subseteq [N_0, N_1]$ $(\phi(N_0) = \phi(N_1) = 0)$ then it is easy to show that formulae (4.17) and (4.19) lead to

$$\phi_{N_0+1}^{(n)}(N_0+1)^n + \phi_{N_0+2}^{(n)}(N_0+2)^n + \cdots + \phi_{N_1-1}^{(n)}(N_1-1)^n = (-1)^n n!$$

and

$$\phi_{N_0+1}^{(n)} = \frac{(-1)^n n!}{(N_0+1)^n + (N_0+2)^n\bar{\phi}_{N_0+2}^{(n)} + \cdots + (N_1-1)^n\bar{\phi}_{N_1-1}^{(n)}},$$

where $\bar{\phi}_i^{(n)} = \frac{\phi_i^{(n)}}{\phi_{N_0+1}^{(n)}},\ i = N_0+2, ..., N_1-1$, respectively.
The values of $\phi^{(n)}$ at dyadic points are determined by relation

$$\phi^{(n)}(k + \frac{l}{2^{j+1}}) = 2^n \sum_{m=0}^{n-1} c_m \phi^{(n)}(2K + \frac{l}{2^j} - m),\ l = 1, 2, \ldots 2^{j+1} - 1, j = 0, 1, 2, \ldots \tag{4.20}$$

The proposed algorithm for evaluation of the refinable function is immediately extended to the multidimensional cases.For brevity we will restrict ourselves only by two dimensional case ($d =$

2) an $n = 0$.
Let

$$\varphi(x) = \varphi_1(x_1)\varphi_2(x_2) \tag{4.21}$$

be a tensor product of scaling functions φ_1, φ_2 , whose factors fulfill one-dimensional refinement equations with coefficients h_{k1} and $h_{k2},\ k_1, k_2 \in \mathbb{Z}$.
Then φ satisfies the refinement equation [7,8]

$$\varphi(x) = |\det M|^{\frac{1}{2}} \sum_{k\in\mathbb{Z}^2} h_k\varphi(Mx - k),\ \ k = (k_1, k_2). \tag{4.22}$$

Let φ be normalized

$$\int_{\mathbb{R}^2} \varphi(x)dx = 1. \tag{4.23}$$

For simplicity we shall assume that the dilation matrix $M = 2I$. Then the refinement equation becomes as

$$\varphi(x_1, x_2) = \sum_{k_1,k_2} h_{k_1,k_2}\varphi(2x_1 - k_1, 2x_2 - k_2). \tag{4.24}$$

Assume supp$(\varphi_1, \varphi_2) \subset [0, N - 1]$. Then supp$(\varphi) \subset [0, N - 1]^2$ .Setting $x_1 = l, x_2 = j$ in (4.24) and denoting $\phi = (\phi_{l,j})$ then resulting system of equations can be written as

$$(A - I)\phi = 0, \tag{4.25}$$

where A is a matrix with elements h_{k_1,k_2} of system (4.24).
The last system can be written in the extended form

$$\left.\begin{array}{l} \sum\limits_{k=1}^{N-2} a_{1k}\phi_{kj} = \phi_{1j} \\ \sum\limits_{k=1}^{N-2} a_{2k}\phi_{kj} = \phi_{2j} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \sum\limits_{k=1}^{N-2} a_{N-2,k}\phi_{kj} = \phi_{N-2,j} \end{array}\right\} \tag{4.26}$$

for $j = 1, 2, ..., N - 2$. If we denote $\frac{\phi_{kj}}{\phi_{1j}} = \overline{\phi}_{kj}, k = 2, ..., N - 2$ then (4.26) form a non homogeneous system w.r.t $\overline{\phi}_{kj}, k = 2, ..., N - 2$ in which is lacked one of equations that contain all unknowns.
Since each system of (4.26) can be solved independently of one another, all computations are inherently parallel.
Thus solving the system (4.26) for all $j = 1, ..., N - 2$ we find $\overline{\phi}_{kj}, k = 2, ..., N - 1, j = 1, ..., N - 2$ .It remains to find only $\phi_{1j}, j = 1, ..., N - 2$. To this end we use the partition of unity [9].

$$\sum_{i,j} \phi_{ij} \equiv 1. \tag{4.27}$$

Obviously,the equality (4.27) holds ,if (for example)

$$\sum_{i=1}^{N-2} \phi_{ij} = \varphi_{2j},\ j = 1, ..., N-2 \tag{4.28}$$

because of

$$\sum_{j=1}^{N-2} \varphi_{2j} \equiv 1.$$

The system (4.28) can be rewritten as

$$\phi_{1j}(1 + \overline{\phi}_{2j} + ... + \overline{\phi}_{N-2,j}) = \varphi_{2j},\ j = 1, .., N-2.$$

From this we find

$$\phi_{1j} = \frac{\varphi_{2j}}{1 + \overline{\phi}_{2j} + ... + \overline{\phi}_{N-2,j}}. \tag{4.29}$$

Using (4.29) we obtain

$$\phi_{kj} = \phi_{1j}\overline{\phi}_{kj}, \quad k = 2, ..., N-2, j = 1, ..., N-2. \tag{4.30}$$

Thus the solution of the system (4.25) is given by $(4.29), (4.30)$.
In conclusion we present numerical results. We were solved the system of equations $(4.2), (4.5)$ for Daubechies scaling function and $N = 4(2)20$ and for $n = 0, 1, 2, \dots$. The results coincide completely with those given by Garba in [5].
In [5] was present the same values for the first and second derivatives of the Daubechies sealing function for $N = 6$. Perhaps, this is a technical mistake. For correctness here we present only these values for $N = 6$.
. Table 1.

i	$\Phi^{(0)}$	$\Phi^{(1)}$	$\Phi^{(2)}$
1	$1.28633506942606E+0000$	1.6384523408819E+0000	9.04249540651108E-0001
2	-3.8583691046152E-0001	-2.23275819046064E+0000	-1.71274862195548E+0000
3	9.52675460038108E-0002	5.50159358273368E-0001	7.12748621955989E-0001
4	4.2343461639624E-0003	4.41464913049908E-0002	9.57504593479706E-0002

References

[1] Daubechies, I. *Orthonormal bases of compactly supported wavelets.* Commun. Pure Appl. Math. 41, No.7, 909-996 (1988).

[2] Daubechies, I. *Ten lectures on wavelets.* CBMS-NSF Regional Conference Series in Applied Mathematics. 61. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics., 357 p. (1992).

[3] Härdle, W. and ets. *Wavelets, approximation, and statistical applications.* Lecture Notes in Statistics (Springer), 265 p. (1998).

[4] Sweldens, W.; Piessens, R. *Quadrature formulae and asymptotic error expansions for wavelet approximations of smooth functions.* SIAM J. Numer. Anal. 31, No.4, 1240-1264 (1994).

[5] Garba, A. *Wavelet collocation approximation of differential operators.* IC/96/212. Miramare-Triesta.

[6] Resnikoff, H.L.; Wells, R.O. *Wavelet analysis.* The scalable structure of information. New York, NY: Springer. 435 p. (1998).

[7] Lin, E-B.; Zhou, X. *Coiflet interpolation and approximate solutions of elliptic partial differential equations.* Numer. Methods Partial Differ. Equations 13, No.4, 303-320 (1997).

[8] Lin, E-B.; Xiao, Z. *Multi-scaling function interpolation and approximation.* Aldroubi, Akram (ed.) et al., Wavelets, multiwavelets, and their applications. AMS special session, January 1997, San Diego, CA, USA. Providence, RI: American Mathematical Society. Contemp. Math. 216, 129-147 (1998).

[9] Dahmen, W. *Wavelet and multiscale methods for operator equations.* Iserles, A. (ed.), Acta Numerica Vol. 6, 1997. Cambridge: Cambridge University Press. 55-228 (1997).

[10] Louis, A.K.; Maa, P.; Rieder, A. *Wavelets. Theory and applications.* Pure and Applied Mathematics. A Wiley-Interscience Series of Texts, Monographs, 324 p. (1997).

[11] Lawton, W.; Lee, S.L.; Shen, Z. *Convergence of multidimensional cascade algorithm.* Numer. Math. 78, No.3, 427-438 (1998).