TSUKUBA J. MATH.
Vol. 37 No. 2 (2013), 321–337

ON THE WAVELET-GALERKIN METHOD WITH DESLAURIERS–DUBUC INTERPOLATING SCALING FUNCTIONS

By

Naohiro FUKUDA

Abstract. Compactly supported orthonormal wavelets are often used in numerical analysis. However, since these functions have not an explicit formula in the time domain, the difficulty of integrations often occurs. In this paper, we introduce the Galerkin method with Deslauriers–Dubuc interpolating scaling functions, and we use the biorthogonality of the wavelets to overcome the difficulties of integration. We present numerical results that show the efficiency and accuracy of this method.

1 Introduction

The Galerkin method is a powerful tool for calculating numerical solutions of differential equations. In particular, lower-degree polynomials are often used for the basis and test functions since the resulting coefficient matrices of the Galerkin equations have simpler structures. This method is called the finite element method (FEM). Let us consider the following problem as an example:

$$\begin{cases} -u'' + u = f, & 0 < x < 1, \\ u(0) = u(1) = 0. \end{cases} \tag{1}$$

A weak formula that corresponds to the problem is given by

$$a(u, v) = \langle f, v \rangle_{L^2} \quad \text{for all } v \in H_0^1(0, 1) \tag{2}$$

with a bilinear form

$$a(u, v) = \int_0^1 uv \, dx + \int_0^1 u'v' \, dx.$$

2000 *Mathematics Subject Classification*: 65N30, 65T60.
Key words and phrases: wavelet-Galerkin method, Deslauriers–Dubuc interpolating wavelets.
Received January 16, 2013.
Revised May 29, 2013.

Here we denote the Sobolev space $H^1(0,1) = \{u \in L^2(0,1) \mid u' \in L^2(0,1)\}$ and $H_0^1(0,1) = \{u \in H^1(0,1) \mid u(0) = u(1) = 0\}$ is its subspace. A solution of (2) is called a weak solution.

The Galerkin method constructs an approximate solution as the weak solution. Let $V_n \subset H_0^1$ be an n-dimensional subspace, and let $\varphi_1, \ldots, \varphi_n$ be a basis of V_n. By substituting $u_n \in V_n$ for u and $v_n \in V_n$ for v we obtain

$$a(u_n, v_n) = \langle f, v_n \rangle_{L^2} \quad \text{for all } v_n \in V_n. \tag{3}$$

We consider the approximate solution $u_J \in V_J$ of the form

$$u_n(x) = \sum_{j=1}^{n} U_j \varphi_j(x).$$

Taking $v_n = \varphi_j$ $(j = 1, 2, \ldots, n)$ in (3) we obtain a Galerkin equation:

$$MU = F,$$

where $M = \{a(\varphi_i, \varphi_j)\}_{i,j=1,\ldots,n}$ is a coefficient matrix, $F = {}^t\{(f, \varphi_j)\}_{j=1,\ldots,n}$ is a vector generated by the inner products of f and the test functions, and U is a unknown vector $U = {}^t\{U_1, \ldots, U_n\}$. The coefficients $\{U_j\}_j$ are thus obtained as the solution of the equation $U = M^{-1}F$.

Classical FEM employees the hat function $B_2(x) = \max\{1 - |x|, 0\}$ as the basis and test functions. If we put $\{\varphi_i(x) = v_i(x) = B_2(x/h - i)\}_{i=1}^{1/h-1} \subset H_0^1(0,1)$, then we can easily see that the components of the stiffness and mass matrices are given, respectively, by

$$a_{i,j} = \langle \varphi_i', v_j' \rangle_{L^2} = \frac{1}{h} \times \begin{cases} 2, & j = i, \\ -1, & j = i \pm 1, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$c_{i,j} = \langle \varphi_i, v_j \rangle_{L^2} = \frac{h}{6} \times \begin{cases} 4, & j = i, \\ 1, & j = i \pm 1, \\ 0, & \text{otherwise}. \end{cases}$$

Thus the coefficient matrix M is a tridiagonal matrix, and its components are given by $M_{i,i} = 2/h + 2h/3$, $M_{i,i\pm1} = -1/h + h/6$, and $M_{i,j} = 0$ otherwise. The sparsity of this matrix results in decreased computing costs.

Wavelet theory has been developing rapidly in several fields since its inception in the 1980's, and many wavelets has been introduced. The application of

wavelets to the Galerkin method is an interesting topic, and the flexibility of wavelet functions provides many options for approximation spaces. Especially, compactly supported orthogonal wavelets or scaling functions give sparse matrices, including the stiffness matrix, because of their locality and orthogonality. Among these, the Daubechies scaling function [5], which is well known as a compactly supported orthogonal function, is commonly used for numerical analysis. But the Daubechies wavelets and scaling functions do not have explicit expressions in the time domain, so if we try to compute the inner product on a wavelet $\langle f, \psi \rangle_{L^2}$ or a scaling function $\langle f, \varphi \rangle_{L^2}$ with high-dimensional accuracy, it is computationally expensive. Therefore, in some cases inner products with scaling functions are simply approximated by its sampling, i.e., $\langle f, 2^{j/2}\varphi(2^j \cdot -k) \rangle_{L^2} \approx f(2^{-j}k)$, but the accuracy of these approximations depends on the smoothness of f, and getting high-precision analysis results requires evaluation of the integrals. To overcome this difficulty with integrations, many methods using wavelets and scaling functions have been introduced [1, 3, 6, 8, 16].

When we use the orthogonal functions as basis and test functions, resulting mass matrix becomes a diagonal matrix, but in almost all cases, the highest derivative of the original equation is a leading term. Thus, in the above case, the structure of the stiffness matrix plays an important role. Fukuda et al. [12] introduced a uniform approach that generates Riesz bases such that the associated stiffness matrices become tridiagonal. This method is highly accurate, but the difficulty with the integral remains unsolved. In this paper, we further develop the method of [12] and use the propeties of the biorthogonality of the wavelets to overcome the difficulty with the integrals of the test functions. The Deslauriers–Dubuc interpolating scaling functions [9, 10] are used as basis functions.

Section 2 contains the definitions and properties of the Deslauriers–Dubuc interpolating functions and the biorthogonal B-spline wavelets that will be to be needed in later sections. In Section 3, we construct a Galerkin equation with our method and explains how to apply the results derived in [12] to the interpolating functions. In the last section, we present some numerical results which prove the efficiency and accuracy of this method.

2 Interpolating Schemes

2.1 Deslauriers–Dubuc Interpolating Wavelet

Deslauriers and Dubuc [9] and Dubuc [10] introduced an interpolation scheme [14] that constructs a function on $\mathbf{R}$ from an initial value $\{f(k)\}_{k \in \mathbf{Z}}$. The

functions obtained from the initial value $\{\delta_{k,0}\}_{k\in\mathbf{Z}}$ are called the fundamental functions. We denote the Deslauriers–Dubuc fundamental functions of order $D = 2L+1$ $(L = 0, 1, \ldots)$ by F_D. F_D satisfies the refinement relation

$$F_D(x) = \sum_{k\in\mathbf{N}} F_D(k/2)F_D(2x-k)$$

and supp $F_D = [-D, D]$. The smoothness of F_D increases as D increases [9].

F_D is known as a scaling functions of the interpolating wavelet function. In general, an interpolating scaling function φ has some useful properties. First, $\varphi(k) = \delta_{k,0}$ for $k \in \mathbf{Z}$, which is useful in terms of the approximation. Second, the two scale equation is given by $\varphi(x) = \sum_{k\in\mathbf{Z}} \varphi(k/2)\varphi(2x-k)$, which means that the filter coefficients $\{h_k\}_k$ are equal to the half values $\varphi(k/2)$. Moreover, the associate wavelet function is simply $\psi(x) = \varphi(2x-1)$.

In the case of Deslauriers–Dubuc scaling functions, the filter coefficients $\{h_k\}_k$ are easily calculated from the Lagrange polynomial: If

$$L_k(x) = \prod_{\substack{i=-N \\ i\neq k}}^{N+1} \frac{x-i}{k-i}, \quad k = -N, -N+1, \ldots, N+1, \tag{4}$$

then

$$h_{2k} = \delta_{k,0},$$

$$h_{2k+1} = \begin{cases} L_{-k}(1/2), & k = -N-1, \ldots, N, \\ 0, & \text{otherwise.} \end{cases}$$

For example,

$$\{h_{-1}, h_0, h_1\} = \left\{\frac{1}{2}, 1, \frac{1}{2}\right\}$$

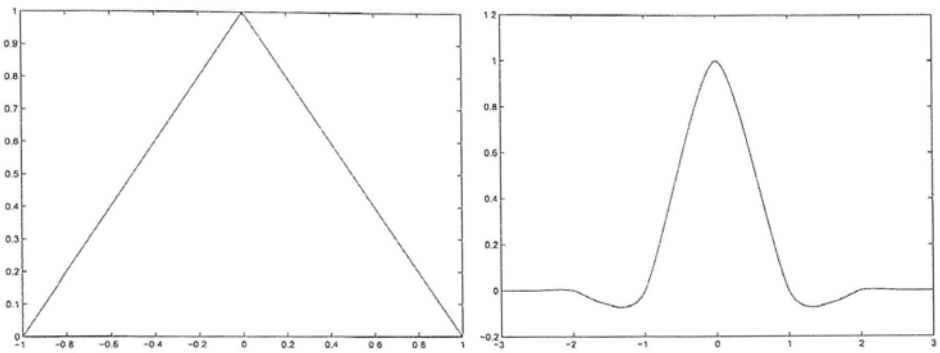
when $D = 1$, and

$$\{h_{-3}, h_{-2}, h_{-1}, h_0, h_1, h_2, h_3\} = \left\{-\frac{1}{16}, 0, \frac{9}{16}, 1, \frac{9}{16}, 0, -\frac{1}{16}\right\}$$

when $D = 3$.

2.2 Average Interpolation

Donoho [11] and Harten [13] generalized the Deslauriers–Dubuc interpolation scheme and also introduced a scheme called average interpolation. The fundamental functions of the average interpolation scheme A_D of order $D = 2L$



(a) F_1 (b) F_3

Figure 1: Deslauriers–Dubuc fundamental functions

$(L = 1, 2, \ldots)$ still have compact supports supp $A_D \subset [-D, D+1]$ and satisfy the two scale equation

$$A_D(x) = \sum_{k \in \mathbf{Z}} c_k A_D(2x - k).$$

For example,

$$\{c_{-2}, c_{-1}, c_0, c_1, c_2, c_3\} = \left\{-\frac{1}{8}, \frac{1}{8}, 1, 1, \frac{1}{8}, -\frac{1}{8}\right\}$$

when $D = 2$,

$$\{c_{-3}, c_{-2}, c_{-1}, c_0, c_1, c_2, c_3, c_4\} = \left\{\frac{3}{128}, -\frac{3}{128}, -\frac{11}{64}, \frac{11}{64}, 1, 1, \frac{11}{64}, -\frac{11}{64}, -\frac{3}{128}, \frac{3}{128}\right\}$$

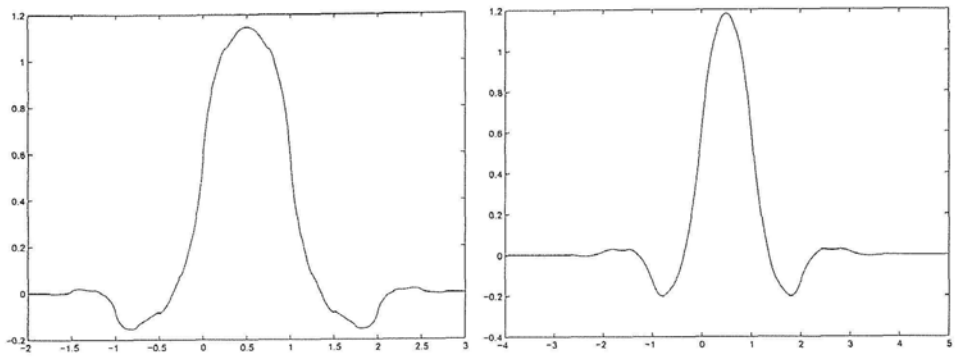
when $D = 4$.

The fundamental functions A_D and F_D have a strong relationship. If we set $\varphi = A_D$ and $\phi = F_{D+1}$, then it holds that

$$\phi'(x) = \varphi(x+1) - \varphi(x). \tag{5}$$

Since

$$\begin{aligned}\varphi(x+1) - \varphi(x) &= \frac{d}{dx}\int_x^{x+1} \varphi(y)\, dy \\ &= \frac{d}{dx}\int_{\mathbf{R}} N_1(y - x)\varphi(y)\, dy \\ &= \frac{d}{dx}\varphi * N_1^{\vee}(x),\end{aligned}$$



(a) A_2 (b) A_4

Figure 2: Fundamental functions of the average interpolation scheme

(5) is equivalent to

$$\phi = \varphi * N_1^{\vee}, \tag{6}$$

where $f^{\vee}(x) = f(-x)$ and N_m is the m-th order B-spline, i.e., $N_1 = \chi_{[0,1)}$ and $N_m = N_{m-1} * N_1$ $(m \geq 2)$. For the construction of Riesz bases, this means that ϕ is an elevation of φ with elevator N_1 ([12]). In terms of the low-pass filters $m^{DD}(\xi) = \sum_k h_k^{DD} e^{-ik\xi}$ and $m^{A}(\xi) = \sum_k h_k^{A} e^{-ik\xi}$, it is denoted as $m^{DD} = m^{A} m$, where $m(\xi) = (1 + e^{i\xi})/2$, or simply, $\{h^{DD}\} = \{h^{A}\} * \{1/2, 1/2\}$.

REMARK 1. Deslauriers–Dubuc fundamental functions also have a special relationship to Daubechies scaling functions [5]. Let Φ_N^D be a Daubechies scaling function of order N. Then Beylkin and Saito [15] proved the following equation:

$$\int_R \Phi_N^D(x)\Phi_N^D(x - y)\, dx = F_{2N-1}(y). \tag{7}$$

Therefore F_{2N+1} is called the autocorrelation function of Φ_N^D.

Orthogonal wavelets lose several properties due to strong restrictions, but we can construct many wavelets by discarding the orthogonality. Cohen, Daubechies and Feaubeau [7] constructed biorthogonal spline wavelets, whose primal and dual functions both have compact support.

Generally, the biorthogonal B-spline wavelets are specified with two parameters. Let φ_p and $\tilde{\varphi}_{p,\tilde{p}}$ be the primal and dual scaling functions of the biorthogonal B-spline wavelet, then the associated low-pass filters m_0 and $\tilde{m}_0$ are

given by

$$m_0(\xi) = e^{-i\varepsilon\xi/2} \cos^p\left(\frac{\xi}{2}\right)$$

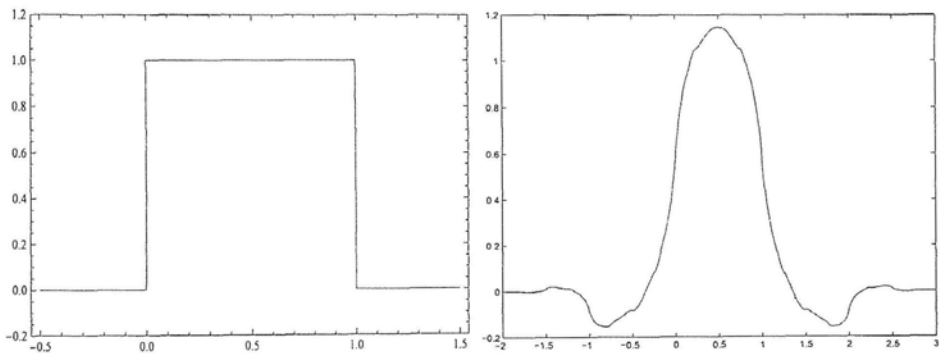
and

$$\tilde{m}_0(\xi) = e^{-i\varepsilon\xi/2} \cos^{\tilde{p}}\left(\frac{\xi}{2}\right) \sum_{k=0}^{(p+\tilde{p})/2-1} \binom{(p+\tilde{p})/2-1+k}{k} \sin^{2k}\left(\frac{\xi}{2}\right),$$

where $p + \tilde{p}$ is an even integer, $\varepsilon = 0$ when p is even, and $\varepsilon = 1$ when p is odd.

For $p = 1$, we note that $m_0(\xi) = e^{-i\xi/2} \cos(\xi/2)$ is just the low-pass filter of the Haar wavelet. Thus, in this case, $\varphi_1 = N_1(x)$. Moreover, Donoho [11] showed that the dual scaling functions are equal to the fundamental functions: more precisely, for $D = 2, 4, \ldots$, it holds that

$$\tilde{\varphi}_{1,D+1} = \Lambda_D. \tag{8}$$



(a) φ_1 (b) $\tilde{\varphi}_{1,3}$

Figure 3: Biorthogonal B-spline functions

3 Wavelet-Galerkin Method with Biorthogonal Functions

In this section, we introduce a way to construct approximate solutions for certain differential equations by using Deslauriers–Dubuc fundamental functions. As mentioned above, these functions have compact support, are symmetric, and satisfy $F_D(k) = \delta_{k,0}$; the Daubechies functions do not have these properties.

With $\varphi = F_3$, $h = 1/(n+1)$ and $n \geq 5$, we seek a numerical solution

$$u_n(x) = \sum_{k=3}^{n-2} U_k \varphi(x/h - k) \tag{9}$$

for the Dirichlet boundary value problem (1). The standard Galerkin method leads to

$$a(u_n, \varphi_k) = \langle f, \varphi_k \rangle_{L^2}, \quad k = 3, 4, \ldots, n-2. \tag{10}$$

From this, we obtain the Galerkin equation

$$MU = F, \tag{11}$$

where $M = \{\int_{\mathbf{R}} \varphi_i' \varphi_j' \, dx + \int_{\mathbf{R}} \varphi_i \varphi_j \, dx\}_{i,j}$ is a coefficient matrix; $F = {}^t\{\langle f, \varphi_j \rangle_{L^2}\}_{j=1,\ldots,n}$; and U is a unknown vector $U = {}^t\{U_1, \ldots, U_n\}$. This equation can be solved to obtain the coefficients U_k.

In this case, the stiffness matrix is a heptadiagonal matrix, which is relatively full compared with the one for classical FEM. Moreover, as in the case of the Daubechies function, Deslauriers–Dubuc fundamental function φ_k does not have an explicit formula; the difficulty of the integral on the right-hand side of (10) thus remains.

To deal with this problem, we replace the φ_k by the hat functions $v_k = B_2(\cdot/h - k)$ and consider

$$a(u_n, v_k) = \langle f, v_k \rangle_{L^2}, \quad k = 3, 4, \ldots, n-2.$$

This leads to a new Galerkin equation:

$$\tilde{M}U = \tilde{F}, \tag{12}$$

with $\tilde{M} = \{\int_{\mathbf{R}} \varphi_i' v_j' \, dx + \int_{\mathbf{R}} \varphi_i v_j \, dx\}_{i,j}$; $\tilde{F} = {}^t\{\langle f, v_j \rangle_{L^2}\}_{j=1,\ldots,n}$; and $U = {}^t\{U_1, \ldots, U_n\}$. Equation (12) is more convenient and manageable than (11) for the following reasons:

(i) Both F_3 and B_2 are elevated functions of pair of biorthogonal functions with elevator $N_1^{\vee}$, thus the resulting stiffness matrix is a tridiagonal matrix, which is sparse compared with the one of (11).

(ii) Both F_3 and B_2 are refinement functions; therefore we can explicitly calculate the mass matrix.

(iii) Compared to (10), the integrals on the right-hand side of (12) are simpler, and they can be processed more quickly by computer. Thus our scheme quickly obtains the solution u once f has been set.

Let us more fully consider the advantages stated in (i), above. Fukuda et al. [12] proved that if φ is orthogonal, i.e., $\langle \varphi, \varphi(\cdot - k) \rangle_{L^2} = \delta_{k,0}$, then the stiffness matrix generated by its elevated function $\Phi = \varphi * N_1$ is a tridiagonal matrix, i.e., $\langle \Phi', \Phi'(\cdot - k) \rangle_{L^2} = 2\delta_{k,0} - \delta_{|k|,1}$. We can easily see that this is also true for a pair

of biorthogonal functions, i.e., if $\langle \varphi_1, \varphi_2(\cdot - k)\rangle_{L^2} = \delta_{k,0}$, then

$$\langle (\varphi_1 * N_1)', (\varphi_2 * N_1)'(\cdot - k)\rangle_{L^2} = 2\delta_{k,0} - \delta_{|k|,1}. \tag{13}$$

Since F_3 and B_2 are elevated functions of the pair of biorthogonal functions $A_2 = \tilde{\varphi}_{1,3}$ and $N_1 = \varphi_1$ with elevator $N_1^{\vee}$ (see (5) and (8)), the resulting stiffness matrix is a tridiagonal matrix.

REMARK 2. One may expect that there exists an elevator $\mathscr{E}$ such that the stiffness matrix become a diagonal matrix, i.e., $\langle \varphi * \mathscr{E}, \varphi * \mathscr{E}(\cdot - k)\rangle = \delta_{k,0}$ with an orthogonal function φ. But this means that $\mathscr{E}$ is the sign function, and the resulting elevated function is thus non compactly supported. We therefore can not use this function for the Galerkin finite element method.

Now let us consider (ii), above. Let f and g be compactly supported refinable functions. Then, $I_k = \int_{\mathbf{R}} f(x)g(x-k)\,dx = f * g^{\vee}(k)$. Here we remark that $p^{\vee}(x) = p(-x)$ is also refinable when p is refinable. Since the convolution of refinable functions is refinable [2], it can be given as a solution of an eigenvalue problem.

In the case $f = F_3$ and $g = B_2$, the above is summarized as follows:

THEOREM 3. *Set* $M_k = \langle F_3, B_2(\cdot - k)\rangle_{L^2}$ *and* $S_k = \langle F_3', B_2'(\cdot - k)\rangle_{L^2}$. *Then we obtain*

$$M_k = \begin{cases} 131/180 & \text{if } k = 0, \\ 37/240 & \text{if } k = \pm 1, \\ -11/600 & \text{if } k = \pm 2, \\ 1/3600 & \text{if } k = \pm 3, \\ 0 & \text{otherwise}, \end{cases} \tag{14}$$

and

$$S_k = \begin{cases} 2 & \text{if } k = 0, \\ -1 & \text{if } k = \pm 1, \\ 0 & \text{otherwise}. \end{cases} \tag{15}$$

PROOF. Equation (15) is easily seen from (13), so let us prove (14). Set $f = F_3 * B_2^{\vee} = F_3 * B_2$. Then f is a refinable function with filter coefficients

$$\{h_k\}_{k=-4}^{4} = \frac{1}{2}\left\{-\frac{1}{16}, 0, \frac{9}{16}, 1, \frac{9}{16}, 0, -\frac{1}{16}\right\} * \left\{\frac{1}{2}, 1, \frac{1}{2}\right\}$$

$$= \left\{-\frac{1}{64}, -\frac{1}{32}, \frac{1}{8}, \frac{17}{32}, \frac{25}{32}, \frac{17}{32}, \frac{1}{8}, -\frac{1}{32}, -\frac{1}{64}\right\}.$$

From the two-scale equation, we get $M_k = f(k) = \sum_m h_m f(2k-m) = \sum_m h_{2k-m} f(m) = \sum_m h_{2k-m} M_m$, and we can obtain the M_k as the eigenvector of

$$\begin{pmatrix} M_{-3} \\ M_{-2} \\ M_{-1} \\ M_0 \\ M_1 \\ M_2 \\ M_3 \end{pmatrix} \begin{pmatrix} h_{-3} & h_{-4} & 0 & 0 & 0 & 0 & 0 \\ h_{-1} & h_{-2} & h_{-3} & h_{-4} & 0 & 0 & 0 \\ h_1 & h_0 & h_{-1} & h_{-2} & h_{-3} & h_{-4} & 0 \\ h_3 & h_2 & h_1 & h_0 & h_{-1} & h_{-2} & h_{-3} \\ 0 & h_4 & h_3 & h_2 & h_1 & h_0 & h_{-1} \\ 0 & 0 & 0 & h_4 & h_3 & h_2 & h_1 \\ 0 & 0 & 0 & 0 & 0 & h_4 & h_3 \end{pmatrix} = \begin{pmatrix} M_{-3} \\ M_{-2} \\ M_{-1} \\ M_0 \\ M_1 \\ M_2 \\ M_3 \end{pmatrix} \tag{16}$$

under the normalization $\sum_k M_k = 1$. □

REMARK 4. In [12] and [17], with Φ_2^D and elevator N_1, a Riesz basis $\varphi_2^D = \Phi_2^D * N_1$ was constructed. Since Φ_2^D is orthogonal, $\langle \varphi_2^{D\prime}, \varphi_2^{D\prime}(\cdot - k) \rangle$ corresponds to S_k in Theorem 1. Moreover, $\langle \varphi_2^D, \varphi_2^D(\cdot - k) \rangle$ also corresponds to M_k in the theorem. Although this may seem strange, it is justified by the autocorrelation property (7); from $\hat{F}_3 = \mathscr{F}[\Phi_2^D * \Phi_2^{D\vee}] = |\hat{\Phi}_2^D|^2$, we have

$$\begin{aligned} \langle F_3, B_2(\cdot - k) \rangle_{L^2} &= \frac{1}{2\pi} \int_{\mathbf{R}} \hat{F}_3(\xi) \hat{B}_2(\xi) e^{ik\xi} \, d\xi \\ &= \frac{1}{2\pi} \int_{\mathbf{R}} |\hat{\Phi}_2^D(\xi)|^2 \hat{N}_1(\xi) \overline{\hat{N}_1(\xi)} e^{ik\xi} \, d\xi \\ &= \langle \varphi_2^D, \varphi_2^D(\cdot - k) \rangle_{L^2}. \end{aligned}$$

4 Numerical Results

In this section we present some numerical results to showing the efficacy of our method. Let us illustrate some numerical examples. All computations were carried out with a Mac OS X, Intel Core i7, 3.4GHz, and by using Mathematica ver. 8.0.1.0.

We consider the following Dirichlet boundary value problem:

$$\begin{cases} -u'' + u = f, & 0 < x < 1, \\ u(0) = u(1) = 0. \end{cases}$$

In classical FEM, the hat function B_2 is used to represent an approximate solution, and in [12], an elevated Daubechies scaling function $\varphi_2^D = \Phi_2^D * N_1$ was

used. To compare these two, we calculated the approximate solutions using the Galerkin method:

$$\tilde{u}(x) = \sum_{n=1}^{2^j-1} u_n B_2(2^j x - k),$$

$$\tilde{u}(x) = \sum_{n=0}^{2^j-4} u_n \varphi_2^D(2^j x - k),$$

$$\tilde{u}(x) = \sum_{n=3}^{2^j-3} u_n F_3(2^j x - k),$$

with test functions $B_2(2^j x - k)$ $(k = 1, \ldots, 2^j - 1)$; $\varphi_2^D(2^j x - k)$ $(k = 0, 1, \ldots, 2^j - 4)$; and $B_2(2^j x - k)$ $(k = 3, 3, \ldots, 2^j - 3)$, respectively. The error was estimated by the relative ℓ^2-error:

$$e_j = \frac{\sqrt{\sum_{k=0}^{2^j} (u(k/2^j) - \tilde{u}(k/2^j))^2}}{\|u\|_{L^2}}. \tag{17}$$

The results with various choices of u are presented as follows:

Table 1: The case of $u(x) = x^5(1-x)^5$

	B_2-B_2		φ_2^D-φ_2^D		F_3-B_2	
2^j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j
6	1.50×10^{-4}	—	2.87×10^{-4}	—	3.65×10^{-4}	—
7	5.30×10^{-5}	2.83	4.66×10^{-5}	6.16	1.76×10^{-5}	20.7
8	1.88×10^{-5}	2.83	9.49×10^{-6}	4.91	8.13×10^{-7}	21.7
9	6.63×10^{-6}	2.83	3.33×10^{-6}	2.85	3.68×10^{-8}	22.1
10	2.35×10^{-6}	2.83	3.15×10^{-6}	1.06	1.20×10^{-9}	30.7

Table 2: The case of $u(x) = N_5(5x)$

	B_2-B_2		φ_2^D-φ_2^D		F_3-B_2	
2^j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j
6	1.51×10^{-4}	—	4.31×10^{-4}	—	6.18×10^{-4}	—
7	5.33×10^{-5}	2.83	6.76×10^{-5}	6.38	5.50×10^{-5}	11.2
8	1.88×10^{-5}	2.83	1.28×10^{-5}	5.29	4.88×10^{-6}	11.3
9	6.66×10^{-6}	2.83	3.80×10^{-6}	3.36	4.32×10^{-7}	11.3
10	2.36×10^{-6}	2.83	3.14×10^{-6}	1.21	3.77×10^{-8}	11.4

Table 3: The case of $u(x) = N_3(3x)$

	B_2-B_2		φ_2^D-φ_2^D		F_3-B_2	
2^j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j
6	1.51×10^{-4}	—	3.64×10^{-2}	—	7.50×10^{-2}	—
7	5.22×10^{-5}	2.90	1.31×10^{-2}	2.78	2.67×10^{-2}	2.81
8	1.87×10^{-5}	2.79	4.66×10^{-3}	2.81	9.46×10^{-3}	2.82
9	6.57×10^{-6}	2.85	1.65×10^{-3}	2.82	3.35×10^{-3}	2.82
10	2.33×10^{-6}	2.82	5.84×10^{-4}	2.83	1.19×10^{-3}	2.83

Table 4: The case of $u(x) = N_3(10x/3 - 1/6)$ (supp $u = [1/20, 19/20]$)

	B_2-B_2		φ_2^D-φ_2^D		F_3-B_2	
2^j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j
6	1.51×10^{-4}	—	8.24×10^{-4}	—	2.01×10^{-6}	—
7	5.21×10^{-5}	2.91	2.13×10^{-4}	3.87	7.80×10^{-7}	2.55
8	1.86×10^{-5}	2.80	5.17×10^{-5}	4.12	6.39×10^{-8}	12.4
9	6.63×10^{-6}	2.81	1.36×10^{-5}	3.79	2.61×10^{-8}	2.45
10	2.34×10^{-6}	2.84	4.47×10^{-6}	3.05	2.57×10^{-9}	10.1

Table 5: The case of $u(x) = \sin^2(2\pi x)$

	B_2-B_2		φ_2^D-φ_2^D		F_3-B_2	
2^j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j	e_j	e_{j-1}/e_j
6	1.53×10^{-4}	—	8.24×10^{-4}	—	2.01×10^{-6}	—
7	5.41×10^{-5}	2.82	2.13×10^{-4}	3.87	7.80×10^{-7}	2.55
8	1.91×10^{-5}	2.83	5.17×10^{-5}	4.12	6.39×10^{-8}	12.4
9	6.77×10^{-6}	2.83	1.36×10^{-5}	3.79	2.61×10^{-8}	2.45
10	2.39×10^{-6}	2.83	4.47×10^{-6}	3.05	2.57×10^{-9}	10.1

Figures 4–7 shows the CPU time required to calculate the integrals of F, i.e., the inner products of f and the test functions versus the error (17).

From these results, we can conclude that our method obtain smoother approximate solutions within the time required to perform classical FEM. In particular, we note that when an exact solution rapidly decays to zero near the boundaries of the domain, our method is more effective. When the decay is not rapid, there is a slight loss of accuracy, which is presumably due to the shape of the basis F_3. Since F_3 is nearly zero at the endpoints of its support, non zero values of the exact solution cannot be represented well in this region. However, this weakness can be easily eliminated. Recall that our proposed method denotes

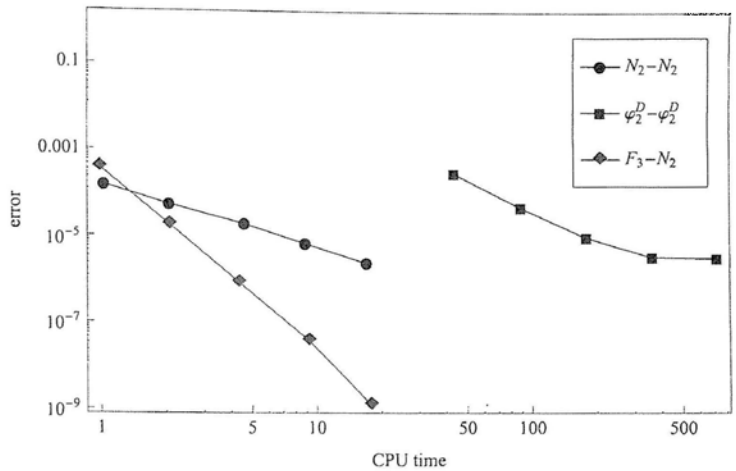


Figure 4: case 1; $u(x) = x^5(1-x)^5$

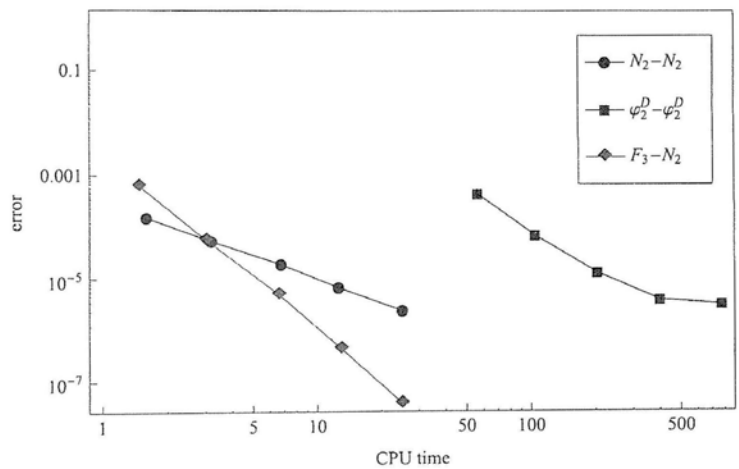


Figure 5: case 2; $u(x) = N_5(5x)$

an approximation solution using F_3 as

$$\tilde{u}(x) = \sum_{n=3}^{2^j-3} u_n F_3(2^j x - n). \tag{18}$$

To capture the behavior of u near the boundary of the domain, we denote the approximate solution using F_3 and B_2 as

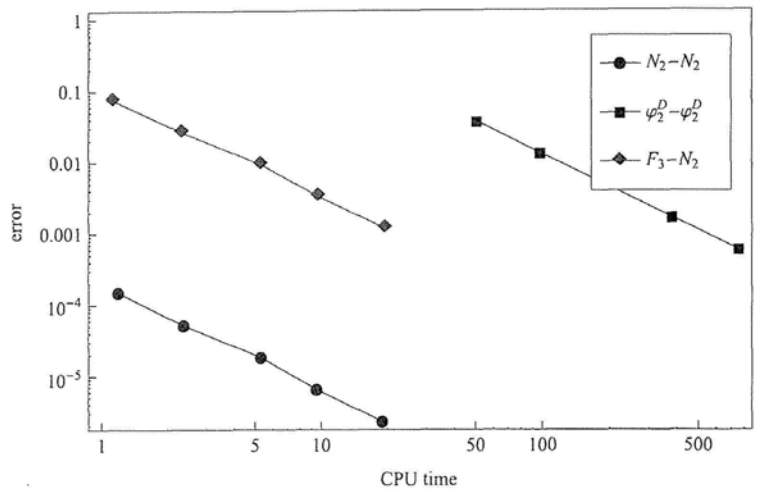


Figure 6: case 3; $u(x) = N_3(3x)$

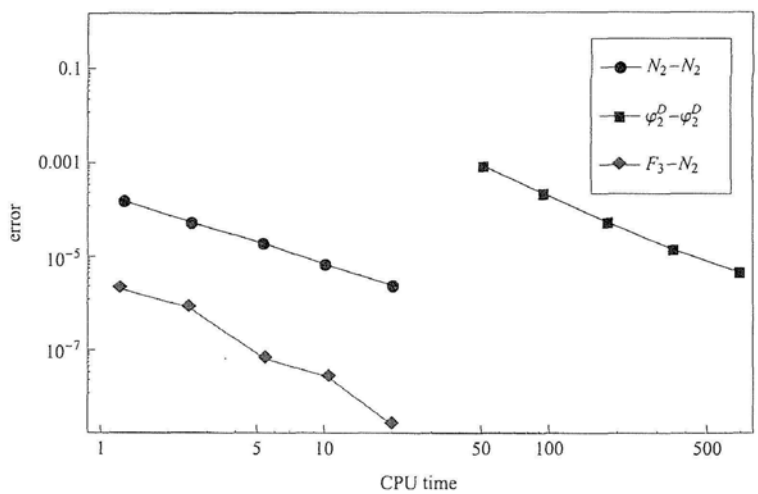


Figure 7: case 4; $u(x) = N_3(10x/3 - 1/6)$ (supp $u = [1/20, 19/20]$)

$$\tilde{u}(x) = \sum_{n=3}^{2^j-3} u_n F_3(2^j x - n) + \sum_{n \in \{1, 2, 2^j-2, 2^j-1\}} u_n B_2(2^j x - n). \tag{19}$$

Figure 8 illustrates the basis and test functions of (18) and (19). This modification increases the size of the coefficient matrix from $2^j - 5$ to $2^j - 1$, but the form of the stiffness matrix does not change. In Figure 9 we show that the computational

cost of the modification is comparable to the unmodified form and that the efficiency of the modification.

Appendix. Cases of Higher-Order Equations

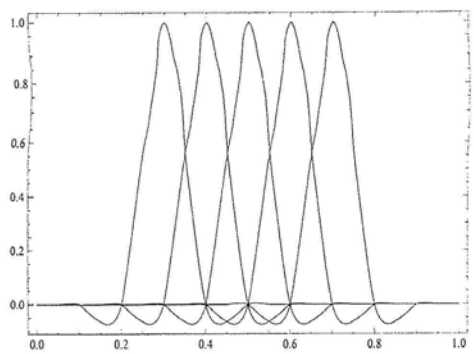
In the previous sections, we considered only second-order differential equations, but higher-order differential equations, such as the beam equation, are also important. In the Galerkin equation, a $2m$-th order differential appears in the coefficient matrix form as $\langle \varphi^{(m)}, \varphi^{(m)}(\cdot - k)\rangle_{L^2}$. For an orthogonal function Φ, Fukuda et al. [12] proved using the formula of inverse matrix of Vandermonde type [4] that

$$\langle \varphi^{(m)}, \varphi^{(m)}(\cdot - k)\rangle_{L^2} = \begin{cases} (-1)^0 {}_{2m}C_m & \text{if } k = 0, \\ (-1)^1 {}_{2m}C_{m-1} & \text{if } k = \pm 1, \\ \vdots & \\ (-1)^m {}_{2m}C_{m-m} & \text{if } k = \pm m, \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

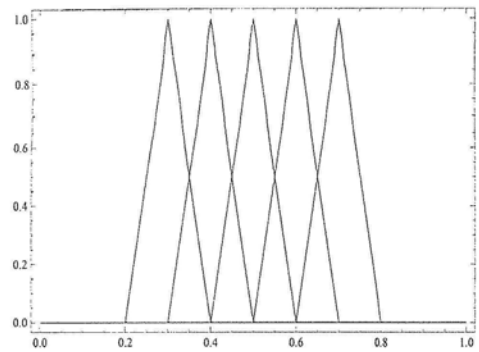
where $\varphi = \Phi * N_m$. Applying this approach to the biorthogonal function, we obtain a pair of basis and test functions $\varphi = F_3 * N_{m-1}$ and $\tilde{\varphi} = N_{m+1}$. There is no difficulty with the integral of the test function N_{m+1}, since N_m is a piecewise polynomial that has an explicit expression in the time-domain, similarly to the previous case of $m = 1$. Moreover, φ is a refinement function, so if we consider the differential equation $u^{(2m)} + u = f$, we can obtain an explicit coefficient matrix. However, the basis function $F_3 * N_{m-1}$ is no longer an interpolation, so it needs a little adjustment. From the Parseval theorem we get

$$\begin{aligned} \langle F_3 * N_{m-1}, N_{m+1}\rangle_{L^2} &= \frac{1}{2\pi}\langle \hat{F}_3 \hat{N}_{m-1}, \hat{N}_{m+1}\rangle_{L^2} \\ &= \frac{1}{2\pi}\langle \hat{F}_3, \hat{N}_{m+1}\overline{\hat{N}_{m-1}}\rangle_{L^2} \\ &= \frac{1}{2\pi}\langle \hat{F}_3, \hat{N}_{m+1}\hat{N}_{m-1}e^{im\cdot}\rangle_{L^2} \\ &= \frac{1}{2\pi}\langle \hat{F}_3, \hat{N}_{2m}e^{im\cdot}\rangle_{L^2} \\ &= \langle F_3, N_{2m}(\cdot + m)\rangle_{L^2}. \end{aligned}$$

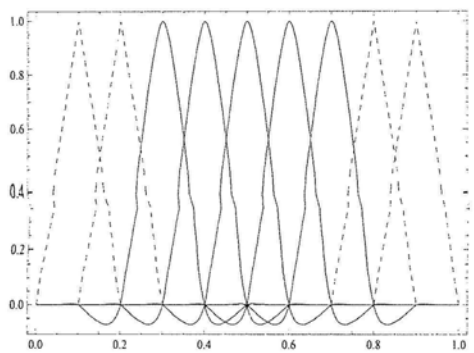
This means that by using F_3 and N_{2m} we can construct an effective scheme for numerical computations.



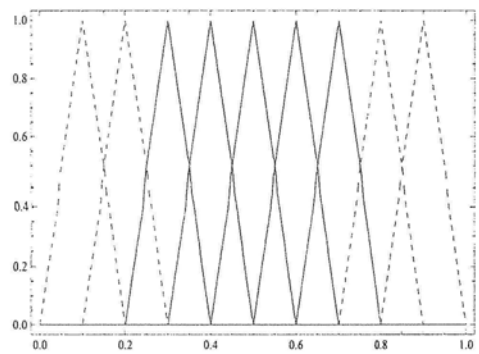
(a) basis functions F_3 for (18)



(b) test functions B_2 for (18)



(c) basis functions F_3, B_2 for (19)



(d) test functions B_2 for (19)

Figure 8: basis and test functions for (18) and (19)

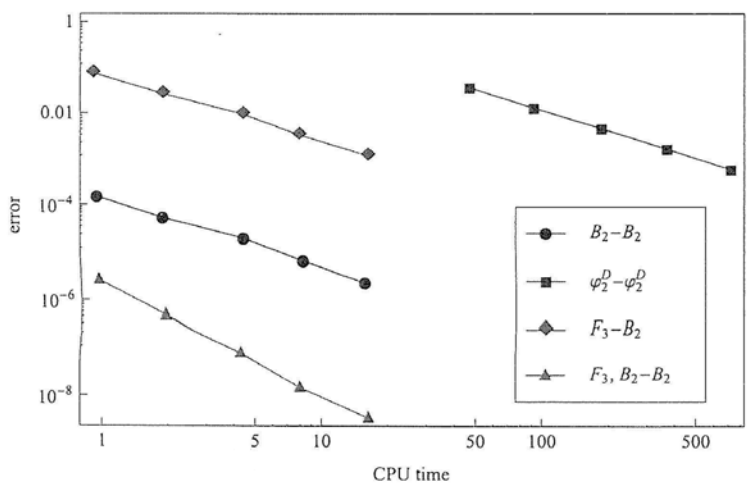


Figure 9: case 3; $u(x) = N_3(3x)$

Acknowledgements

The author thanks to professors T. Kinoshita and T. Kubo for their valuable suggestions and discussions. The author is supported by Grant in Aid for JSPS Fellows.

References

[1] M. Q. Chen, C. Hwang, Y. P. Shih, The computation of wavelet-Galerkin approximation on a bounded interval, Int. J. Numer. Methods Eng. **39** (1996), 2921–2944.
[2] A. Cohen, Numerical Analysis of Wavelet Methods, Studies in Mathematics and its Applications 32. North-Holland, Amsterdam, 2003.
[3] A. Cohen and A. Ezzine, Quadratures singulières et fonctions d'échelle, CRAS Paris **323** (1996), Série I, 829–834.
[4] F. Colombini and T. Kinoshita, On the Gevrey wellposedness of the Cauchy problem for weakly hyperbolic equations of higher order, J. Differential Equations **186** (2002), 394–419.
[5] I. Daubechis, Ten lectures on wavelets, SIAM, Philadelphia, 1992.
[6] W. Danmen, C. A. Micchelli, Using the refinement equation for evaluating integrals of wavelets, SIAM J. Numer. Anal. **30** (1993), 507–537.
[7] A. Cohen, I. Daubechies, J. C. Feauveau, Biorthogonal bases of compactly supported wavelets, Comm. Pure. Appl. Math. **45** (1992), 485–560.
[8] W. Dahmen, A. Kunoth and R. Schneider, Operator Equations, Multiscale Concepts and Complexity, Lectures in Applied Mathematics 32, 225–261 (1996).
[9] G. Deslauriers and S. Dubuc, Symmetric iterative interpolation processes, Constructive approximation **5** (1989), 49–68.
[10] S. Dubuc, Interpolation through an iterative scheme, J. Math. Anal. Appl. **114** (1986), 185–204.
[11] D. L. Donoho, Smooth wavelet decomposition with blocky coefficient kernels, Recent Advances in Wavelet Analysis, (L. Schumaker and F. Ward, eds.), Academic Press, Boston, 1993.
[12] N. Fukuda, T. Kinoshita, T. Kubo, On the Galerkin-wavelet method for higher order differential equations, to appear in the Bulletin of the Korean Mathematical Society.
[13] A. Harten, Multiresolution representation of cell-averaged data, Technical Report, UCLA CAM Report, 1994.
[14] C. Micchelli, Interpolatory subdivision schemes and wavelets, J. Approx. Theory **86** (1996), 41–71.
[15] N. Saito, G. Beylkin, Multiresolution Representations Using the Auto-Correlation Functions of Compactly Supported Wavelets, IEEE Trans. Signal Process. **41** (1993), 3584–3590.
[16] W. Sweldens and R. Piessens, Quadrature formulae and asymptotic error expansions for wavelets approximations of smooth functions, SIAM J. Numer. Anal. **31** (1994), 1240–1264.
[17] J. C. Xu and W. C. Shann, Galerkin-wavelet methods for two-point boundary value problems, Numer. Math. **63** (1992), 123–144.

Institute of Mathematics
University of Tsukuba
Tsukuba, Ibaraki 305-8571, Japan
E-mail: naohiro-f@math.tsukuba.ac.jp