List of Figures

12

Initialization and Inner Product Computations of Wavelet Transform by Interpolatory Subdivision Scheme

Yu-Ping Wang and Ruibin Qu

This work is supported by the Wavelets Strategic Research Programme funded by the National Science and Technology Board and Ministry of Education of Singapore under the Grant RP960 601/A.

The authors are with the Department of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260. Email: wyp@wavelets.math.nus.edu.sg

Abstract

The initialization of wavelet transforms and the inner product computations of wavelets with their derivatives are very important in many applications. In this correspondence, the interpolatory subdivision scheme (ISS) is proposed to solve these problems efficiently. We introduce a general procedure to compute the exact values of derivatives of the interpolatory fundamental function and then derive a fast recursive algorithm for the realization of the initialization and inner product evaluations. Error analysis of the algorithm and its comparison with other approaches are discussed. Numerical experiments demonstrate high performance of the algorithm.

Keywords

Wavelet transform, Interpolatory subdivision scheme, Wavelet-Galerkin algorithm.

I. Introduction

Wavelet transform has become a powerful tool for signal analysis. Efficient implementation of wavelet transforms is an important step towards further applications. The initialization of wavelet transform arises from the pyramid algorithm for discrete wavelet transform or wavelet packet transform. One usually needs to compute the following coefficients exactly before using the recursive pyramid algorithm:

$$c_J^0(k) = 2^{-\frac{J}{2}} \int f(t)\psi(2^{-J}t - k)dt. \tag{1}$$

There are several approaches to solve this problem (cf. [1], [2], [3]). Usually this problem is treated by approximating both the signal and the wavelet (or scaling function) using certain basis function:

$$f(t) = \sum_k f(k)\chi(t-k), \tag{2}$$

$$\psi(t) = \sum_k h(k)\phi(t-k). \tag{3}$$

In [1], [2] the approximation basis is chosen to be the standard *sinc* function as in the Shannon sampling theorem [4]. In [3] the signal f is assumed to be piecewise constant in dyadic intervals. Equivalently, 0th-order B-splines were used to approximate the signal. Moreover, Chui's interpolatory graphical display algorithm was used, which was in fact due to the refinable property of splines [5], [6]. Therefore, most of these approaches can be attributed to the curve fitting problem using certain types of subdivision schemes. In this correspondence, we aim to use the interpolatory subdivision scheme (ISS) [12], [13] to deal with this problem. In particular, we propose the problem in a more general setting. For the numerical solution of PDEs using wavelets,

the computations of the inner products of the signal (wavelets) with the derivatives of scaling functions or wavelets $\psi^{(m)}$ are usually needed. In wavelet-Garlerkin method, one usually needs to evaluate the following integral for all $m, J \in Z_+$ and $k \in Z$:

$$c_J^{(m)}(k) = 2^{-\frac{J}{2}} \int f(t)\psi^{(m)}(2^{-J}t - k)dt. \tag{4}$$

As an alternative, our method also provides a simple and efficient way for the computation of the inner products of wavelets and their derivatives discussed in [7], [8].

II. An interpolatory subdivision scheme

A. The definitions

A uniform subdivision scheme which is also called the *binary subdivision algorithm or stationary subdivision*, is defined as follows. Suppose that the initial *control points* in $\mathbf{R}^3$ (or in $\mathbf{R}^d,\ d \geq 1$) are denoted by $P_i^0,\ i \in Z$, then, the refined control points $\{P_i^{k+1}, i \in Z\}$ are obtained from $\{P_i^k\}$ recursively by the following refinement equations

$$P_i^{k+1} = \sum_{j \in Z} a_{i-2j} P_j^k, \quad i \in Z, \quad k \geq 0. \tag{5}$$

Obviously, a typical example of a binary subdivision scheme is the scheme generating uniform B-splines of order n. In this case the mask is given by the binomial coefficients. The scheme (5) is a stepwise interpolatory scheme if and only if the mask $\{a_j, j \in Z\}$ satisfies $a_{2i} = \delta_i, \quad \forall i \in Z$. Two frequently used interpolatory subdivision schemes are the '4–point interpolatory scheme' [10], [11] and the '6–point interpolatory scheme' [12]. A more general symmetric interpolatory subdivision algorithm for curves was given by [12], [13]:

$$\begin{cases} P_{2i}^{k+1} &= P_i^k, \\ P_{2i+1}^{k+1} &= \sum\limits_{j=0}^{n} L_{n,j}(P_{i-j}^k + P_{i+j+1}^k), \end{cases} \tag{6}$$

where n is called the degree of the scheme and $\{L_{n,j}\}$ are given by

$$L_{n,j} = \frac{[(2n+1)!!]^2}{2 \cdot 4^n \cdot (2n+1)!} \cdot \frac{(-1)^j}{2j+1} \cdot \begin{pmatrix} 2n+1 \\ n-j \end{pmatrix}, \quad j = 0, 1, \cdots, n, \tag{7}$$

where $\begin{pmatrix} 2n+1 \\ n-j \end{pmatrix}$ denotes the binomial coefficient.

The scheme defined by (6) produces $C^{\frac{n}{2}}$ curves for any initial data. Furthermore, for such choice of coefficients the scheme reproduces all parametric polynomial curves of degree less than or equal to $2n+1$ [13].

Let $\phi_n(t)$ be the limit curve generated from the cardinal data $\{P_i = (i, \delta_0)^T\}$. Then $\phi_n(i) = \delta_i, i \in Z$ and $\phi_n(t)$ satisfies the following two-scale relation

$$\phi_n(t) = \sum_{j=-(2n+1)}^{2n+1} g_k^n \phi_n(2t-k), \quad t \in R, \tag{8}$$

where $g^n = \{L_{n,n}, 0, L_{n,n-1}, \cdots, 0, L_{n,0}, 1, L_{n,0}, 0, \cdots,$

$L_{n,n-1}, 0, L_{n,n}\}$. Usually $\phi_n(t)$ is called the ***interpolatory fundamental function or basis***.

B. Computing procedure for the derivative $\phi_n^{(m)}$

In this section, we present the principle for the computation of the exact values of derivatives of the interpolatory function ϕ_n. As an example, we only consider the case of $n = 2$. More details can be found in [14], [15] where the exact values of the interpolatory function ϕ_n at integers for $n \geq 2$ are presented.

The techniques for the computation of $\phi_n^{(m)}$ are the local subdivision method and the divided difference approximation. The derivatives are just the limits of their corresponding divided difference approximations. From the construction of the algorithm, it can be found that the local iteration matrix C when $n = 2$ is given by

$$\begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{256} & -\frac{25}{256} & \frac{150}{256} & \frac{150}{256} & -\frac{25}{256} & \frac{3}{256} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3}{256} & -\frac{25}{256} & \frac{150}{256} & \frac{150}{256} & -\frac{25}{256} & \frac{3}{256} & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{3}{256} & -\frac{25}{256} & \frac{150}{256} & \frac{150}{256} & -\frac{25}{256} & \frac{3}{256} & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{3}{256} & -\frac{25}{256} & \frac{150}{256} & \frac{150}{256} & -\frac{25}{256} & \frac{3}{256} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}.$$

In another word, C is the matrix form of the local subdivision scheme for $n = 2$. The entries of C are determined by the scheme (6) or (8) and the size of C is just large enough to determine the local property of $\phi_2(t)$ at a single point. It can be shown from the reproduction of quintic

polynomials that the first six eigenvalues and their corresponding eigenvectors of C are given by

$$\lambda_k = 2^{-k},$$

$$\xi_k = ((-4)^k, (-3)^k, (-2)^k, (-1)^k, 0, 1, 2^k, 3^k, 4^k)^T, \ k = 0, \cdots, 5$$

and the other eigenvalues are:

$$1, \ \frac{1}{2}, \ \frac{1}{4}, \ \frac{9}{64}, \ \frac{1}{8}, \ -\frac{9}{128}, \ \frac{1}{16}, \ -\frac{1}{16}, \ \frac{1}{32}.$$

The exact values of all the corresponding normalized left and right eigenvectors can be obtained by *Maple*. For example, the first three pairs are given by the following:

$$\begin{aligned}
\xi_0 &:= (1,1,1,1,1,1,1,1,1)^T, \\
\eta_0 &:= (0,0,0,0,1,0,0,0,0)^T, \\
\xi_1 &:= (-4,-3,-2,-1,0,1,2,3,4)^T, \\
\eta_1 &:= (-3,-128,1272,-6528,0,6528,-1272,128,3)^T/8760, \\
\xi_2 &:= (16,9,4,1,0,1,4,9,16)^T, \\
\eta_2 &:= (9,192,-1472,5696,-8850,5696,-1427,192,9)^T/3360.
\end{aligned}$$

In order to evaluate the divided difference easily, the following preliminary result is needed [14].

Lemma 1: Given a square matrix $\mathbf{A}$ of order l, let the normalized left and right (generalized) eigenvectors of $\mathbf{A}$ be denoted by $\{\eta_i, \zeta_i\}$. Then, for any vector $f \in R^l$, we have the following "Fourier" expansion

$$f = \sum_{i=1}^{l} (f^T \eta_i) \zeta_i. \tag{9}$$

From the above properties of C, Lemma 1 and the divided difference approximation, the following result is obtained (cf. [14]).

Theorem 1: The fundamental solution ϕ_2 is twice continuously differentiable and supported on $(-5, 5)$, and its first and second order derivatives at integers are given by

$$\begin{array}{llll}
\phi_2'(0) & = 0, & \phi_2''(0) & = -\frac{295}{56}, \\
\phi_2'(\pm 1) & = \mp\frac{272}{365}, & \phi_2''(\pm 1) & = \frac{356}{105}, \\
\phi_2'(\pm 2) & = \pm\frac{53}{365}, & \phi_2''(\pm 2) & = -\frac{1427}{1680}, \\
\phi_2'(\pm 3) & = \mp\frac{16}{1095}, & \phi_2''(\pm 3) & = \frac{4}{35}, \\
\phi_2'(\pm 4) & = \mp\frac{1}{2920}, & \phi_2''(\pm 4) & = \frac{3}{560}.
\end{array} \tag{10}$$

The graphs of $\phi_2(t)$ and its derivatives $\phi_2'(t)$ and $\phi_2''(t)$ are shown in Figure 1.

It is easy to check that the derivatives of $\phi_n(t)$ also satisfy a two-scale relation similar to (8):

$$\phi_n^{(m)}(t) = 2^m[\phi_n^{(m)}(2t) + \sum_{j=0}^{n} L_{n,j}\phi_n^{(m)}(2t \pm (2j+1))]. \tag{11}$$

Therefore, from the above values of the derivatives of $\phi_n(t)$ we can compute all the derivative values of $\phi_n(t)$ at dyadic points easily. For simplicity, (11) can be rewritten as

$$\phi_n^{(m)}(t) = [M * \phi_n^{(m)}](2t), \tag{12}$$

where the mask or transfer function M is given by the following sequence:

$$2^m\{L_{n,n}, 0, L_{n,n-1}, \cdots, 0, L_{n,0}, 1, L_{n,0}, 0, \cdots, L_{n,n-1}, 0, L_{n,n}\}. \tag{13}$$

III. Initialization and inner product computations using ISS

Suppose N is the number of samples of the signal and the sampling rate is 1 without loss of generality. Our strategy is to use the interpolatory basis ϕ_n to approximate both the signal and wavelets in (2) and (3). We now formulate (1) and (4) in the following general case

$$c_J^{(m)}(k) = 2^{-\frac{J}{2}} \int f(t)\psi^{(m)}(2^{-J}t - k)dt, \quad m = 0, 1, \cdots, \tag{14}$$

where m denotes the m-th order derivative of the wavelet function. If we represent both the signal f and $\psi^{(m)}$ using the interpolatory basis ϕ_n in (2) and (3) we will arrive at the following

$$c_J^{(m)}(k) =$$

$$\begin{aligned} &2^{-\frac{J}{2}} \int (\sum_{i\in Z} f(i)\phi_n(t-i))(\sum_{l\in Z} \psi(l)\phi_n^{(m)}(2^{-J}t - k - l))dt \\ &= 2^{-\frac{J}{2}} \sum_i \sum_l f(i)\psi(l) \int \phi_n(t-i)\phi_n^{(m)}(2^{-J}t - k - l))dt. \end{aligned} \tag{15}$$

By defining $\Phi_J^{(m)}(x) = \int \phi_n(t)\phi_n^{(m)}(2^{-J}t - x)dt$, we can conclude that $\Phi_J^{(m)}$ is also refinable and it is easy to verify that the values of $\Phi_J^{(m)}$ at integers can be computed recursively from $\Phi_0^{(m)}$ through the two-scale relation (12):

$$\Phi_J^{(m)}(k) = (\Phi_J^{(m)} * M)(2k) = \cdots$$

$$= (\Phi_0^{(m)} * M * M_{\uparrow 2} * \cdots * M_{\uparrow 2^{J-1}})(2^J k),$$

where $M_{\uparrow J}$ denotes the up-sampling operation which is obtained by inserting $J-1$ zeros between every two adjacent elements of the mask M given in (13). Thus, we only need to evaluate $\Phi_0^{(m)}$ at integers

$$\begin{aligned}\Phi_0^{(m)}(k) &= \int \phi_n(t)\phi_n^{(m)}(t-k)dt = (-1)^m \int \phi_n(t)\phi_n^{(m)}(k-t)dt \\ &= (-1)^m(\phi_n * \phi_n^{(m)})(k) = (-1)^m \phi_n^{(m)}(k).\end{aligned}$$

Hence this formula is exact. At last, we can express (15) as follows:

$$c_J^{(m)}(k) = 2^{-\frac{J}{2}} \sum_i \sum_l f(i)\psi(l)(\Phi_0^{(m)} * M * M_{\uparrow 2} * \cdots * M_{\uparrow 2^{J-1}})$$

$$(2^J(k+l)-i) = 2^{-\frac{J}{2}} \sum_l \psi(l)(f * \Phi_0^{(m)} * M * M_{\uparrow 2} * \cdots * M_{\uparrow 2^{J-1}})(2^J(k+l)).$$

In the above formula, the initialization and inner products are computed recursively and thus the computational complexity is $\mathcal{O}(N)$.

IV. Discussions

A. Approximation errors

In the previous section, we project both the signal f and the wavelets $\psi_n^{(m)}$ into the hierarchical sampling space. Error estimates of the approximation to the limit curves by the piecewise linear interpolants $P^k(t)$ are given by the following (cf. [13], [14]).

Theorem 2: Suppose $F(t),\ t \in R$, is a regular and C^{2n+2} curve in R^m, $m \geq 2$. Let $P(t),\ t \in R$, be the limit curve generated by scheme (6) from the initial data $P_i := F(ih),\ i \in Z$, $0 < h < 1$. Then, on any finite interval $[a, b]$, we have the following estimate

$$\|F(ht) - P(t)\|_\infty \leq \frac{M_{2n+2}(F)}{(2n+2)!} h^{2n+2} = \mathcal{O}(h^{2n+2}),$$

where the number $M_{2n+2}(F)$ depends only on the derivatives of $F(t)$ and n. Similarly, we have

$$\|h^m F^{(m)}(ht) - P^{(m)}(t)\|_\infty = \mathcal{O}(h^{2n+2-m}), \quad m = 0, 1, \cdots \frac{n}{2}.$$

Therefore, we can conclude by the triangle inequality

$$\begin{aligned}\|c_{2^J} f - \tilde{c}_{2^J} f\|_\infty &= \|f * \psi_{2^l}^{(m)} - \tilde{f} * \tilde{\psi}_{2^l}^{(m)}\|_\infty \\ &\leq \|f - \tilde{f}\|_\infty \|\psi^{(m)}\|_\infty + \|\psi_{2^l}^{(m)} - \tilde{\psi}_{2^l}^{(m)}\|_\infty \|\tilde{f}\|_\infty\end{aligned}$$

that the approximation order of our algorithm is $\mathcal{O}(h^{2n+2-m})$.

In Figure 2 we show the interpolation results of the function $f(t) = e^t sin(t), t \in [0, 10]$ using the ISS approach and the *sinc* bases. Such a function is not band-limited. It can be shown that the ISS method provides a better result than the Shannon bases.

B. Comparison with other approaches

The asymptotic behavior of the coefficients (7) when the order n tends to infinity is (cf. [9]): $\lim_{n\to\infty} L_{n,j} = \frac{1}{\pi}\frac{(-1)^j}{2j+1}$. Therefore, the impulse responses $\{g_k^\infty\}$ in the two-scale relation (8) become

$$g_k^\infty = \begin{cases} \delta_j & if \quad k = 2j, \\ \frac{1}{\pi}\frac{(-1)^j}{2j+1} & if \quad k = 2j+1 \end{cases} = \frac{sin\frac{k\pi}{2}}{\frac{k\pi}{2}}$$

and the interpolatory function ϕ_n approaches the *sinc* function $\phi_\infty(t) = \frac{sin\pi t}{\pi t}$ when n goes to infinity. Therefore, for band-limited signal, one can increase the order of ISS to recover the signal from its finite samples.

In [1], [2] the signal $f(t)$ is supposed to be approximated by the standard *sinc* bases and the initial values are obtained by projecting the signal into the multiresolution space V_J. Then the computation of these initial values becomes

$$c_J^0(k) = \sum_n f(n)\alpha(k - 2^{-J}n) \tag{16}$$

and

$$\alpha(m) =< sinc, \varphi_{Jm} >= \frac{1}{2\pi}\int \Xi(\omega)\sqrt{2^J}\Phi(2^J\omega)exp(i2^J\omega m)d\omega \tag{17}$$

where $\Xi(\omega) = 1$, for $|\omega| \leq \pi$ and 0 elsewhere. It is assumed in [1] and [2] that for large $J \geq 0$ the behavior of $\Phi(2^J\omega)$, for $|\omega| \geq \pi$ can be neglected. Then $sinc(t)$ can be assimilated to a Dirac function. Hence (16) reduces to

$$c_J^0(k) = \sqrt{2^{-J}}\sum_n f(n)\varphi(2^{-J}k - n).$$

The quality of such an approximation lies in the regularity of the low-pass filter. The higher degree of the regularity of φ, the better of the approximation so that most of the energy lies within $|\omega| \leq \pi$. Therefore, such approach would be useless in the case of scaling function whose support is $-1 \leq t \leq 1$ (e.g., the Haar wavelet). In Figure 3 we compare the results using ISS with this approach. For illustration the signal is supposed to be Dirac function $f(t) = \delta(t - t_0)$. Then the initialization coefficients are just the sampling points of $\sqrt{2^{-J}}\varphi(2^{-J}t_0 - k)$. Good results can be obtained by using ISS as shown in Figure 3.

Similarly, for the case $J < 0$ one can roughly take $\Phi(2^J\omega) = 1$ for $|\omega| \leq \pi$ as discussed in [2]. Then the computation of the initial coefficients reduces to

$$c_J^0(k) = \sqrt{2^J}\sum_n f(n)sinc(2^Jk - n),$$

which uses the Shannon functions as interpolating bases. This approach can only work well when $\Phi(2^J\omega) = 1$ for $|\omega| \leq \pi$ which is not always the case in practice.

We now compare our method with the spline approach used in [3]. If B-splines of order greater than 0 are used, the size of the interpolating mask is infinite [5], which has to be truncated or using other method. For ISS the size of mask is finite and therefore computationally efficient. Another advantage of ISS is that such an approach is local. One can choose the interpolation points non-uniformly by assigning more interpolation points around irregular locations. The disadvantage of ISS is that the generated curves are not polynomial splines.

V. Conclusions

In this correspondence, we propose the ISS approach for the initialization of wavelet transform which is very important for further applications. Moreover, the proposed method also provides an efficient evaluation of inner products of wavelets and their derivatives which is essential in the wavelet-Galerkin method. We formulate a general procedure for the computation of exact derivative values of the interpolatory fundamental function at dyadic points. The error analysis shows that the method is exact for certain polynomials. The numerical experiments have shown the high accuracy and efficiency of the method.

References

[1] P. Abry and P. Flandrin, On the initialization of the discrete wavelet transform algorithm, *IEEE Signal Processing Letters*, vol. 1, no. 2, pp. 32-34, 1994.

[2] X. P. Zhang, L. S. Tan and Y.-N. Peng, From the wavelet series to the discrete wavelet transform - the initialization, *IEEE Trans. Signal Processing*, vol. 44, no. 1, pp. 129-133, 1996.

[3] C. J. Zarowski, An approach to initializing the wavelet packet algorithm, *IEEE Signal Processing Letters*, vol. 4, no. 5, pp. 132-134, 1997.

[4] P. L. Butzer, W. Engels, S. Ries, R. L. Stens, The Shannon sampling series and the reconstruction of signals in terms of linear, quadrature and cubic splines, *SIAM J. Appl. Math.*, vol. 46, no. 2, pp. 299-323, 1986.

[5] M. Unser, A. Aldroubi and M. Eden, Fast B-spline algorithms for continuous image representation and interpolation, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 277-285, 1991.

[6] Yu-Ping Wang, S. L. Lee, Scale-space derived from B-splines, to appear in *IEEE Trans. Pattern Anal. Machine Intell.*

[7] W. Dahmen and C. A. Michelli, Using the refinement equation for evaluating integrals of wavelets, *SIAM J. Numer. Anal.*, vol. 30, pp. 507-537, 1993.

[8] J. M. Restrepo and G. K. Leaf, Inner product computations using periodized Daubechies wavelets, *Int J. Numer. Method in engineering*, vol. 40, pp. 3557-3578, 1997.

[9] Yuping Wang and Ruibin Qu, Fast implementation of scale-space filtering using interpolatory subdivision scheme, preprint, Wavelets Strategic Research Programme, National University of Singapore, 1997.

[10] S. Dubuc, Interpolation through an iterative scheme, *J. Math. Anal. and Appl.*, vol. 144, pp. 185–204, 1986.

[11] N. Dyn, J. A. Gregory and D. Levin, A 4-point interpolatory subdivision scheme for curve design, *Computer Aided Geometric Design*, vol. 4, pp. 257–268, 1987.

[12] Ruibin Qu, Curve and surface interpolation by subdivision, *Computer Aided Drafting, Design and Manufacturing*, vol. 4, no. 2, pp. 28-39, 1994.

[13] Ruibin Qu and R. P. Agarwal, A cross difference approach to the analysis of subdivision algorithms, *Neural, Paralell & Scientific Computations*, vol. 3, no. 3, pp. 393-416, 1995.

[14] Ruibin Qu and R. P. Agarwal, Solving two point boundary value problems by interpolatory subdivision algorithms, *Intern. J. Computer Math.*, vol. 60, pp. 279-294, 1996.

[15] Ruibin Qu, A new approach to numerical differentiation and integration, *Mathl. Comput. Modeling*, vol. 24, no. 10, pp. 55-68, 1996.

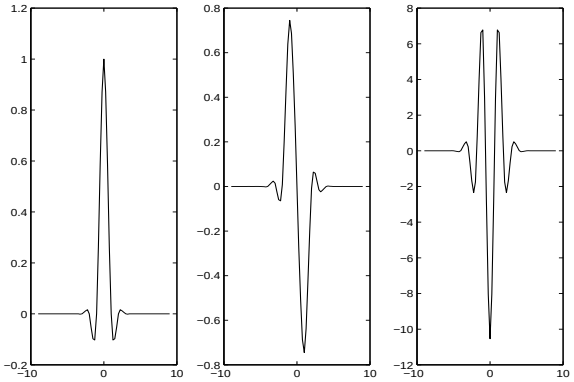

Fig. 1. Interpolatory basis function $\phi_2(t)$ and its first and second derivatives $\phi_2'(t)$ and $\phi_2''(t)$.

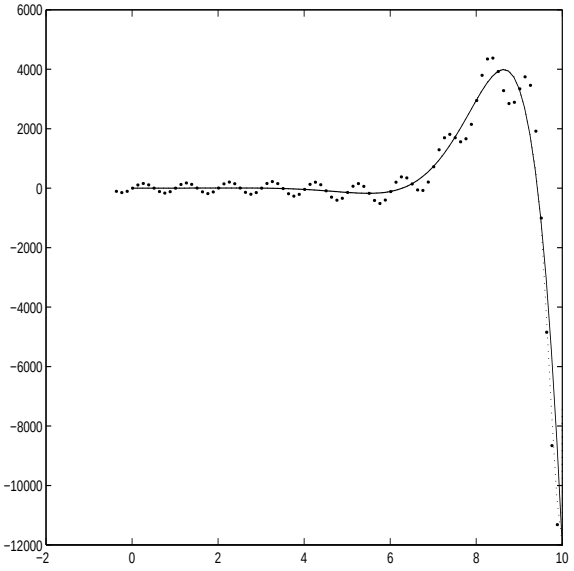

Fig. 2. Interpolation of the function $f(t) = e^t sin(t), t = 0, 0.125, 0.25, ..., 10$ (solid line) using the interpolatory subdivision scheme (dotted line) and the Shannon bases $sinc(t)$ (shown using points). The interpolating points are taken at $t = 0, 0.5, 1, ..., 10$.

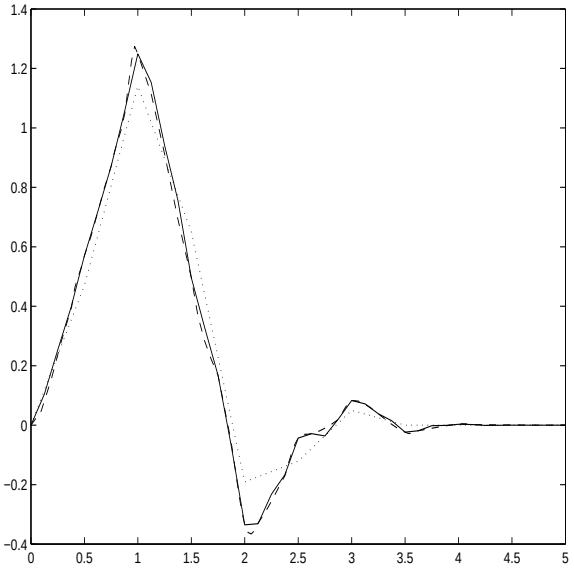

Fig. 3. Comparisons of the initialization results between different methods. The scaling function is chosen as the Daubechies scaling function of order 3. The ideal initialization coefficients are drawn in solid line. The result using ISS is shown by the dashed line. The result using algorithms in [1], [2] is shown by the dotted line.