

A novel method for computing the Hilbert transform with Haar multiresolution approximation[☆]

Chaoying Zhou^a, Lihua Yang^{a,*}, Yingjun Liu^a, Zhihua Yang^b

^a School of Mathematics and Computing Science, Sun Yat-sen University, Guangzhou city, 501275, PR China

^b Information Science and Technology School, Guangdong University of Business Studies, Guangzhou, 510320, PR China

Received 19 July 2007; received in revised form 31 January 2008

Abstract

In this paper, an algorithm for computing the Hilbert transform based on the Haar multiresolution approximation is proposed and the L^2 -error is estimated. Experimental results show that it outperforms the library function ‘hilbert’ in Matlab (The MathWorks, Inc. 1994–2007). Finally it is applied to compute the instantaneous phase of signals approximately and is compared with three existing methods.

© 2008 Elsevier B.V. All rights reserved.

Keywords: Hilbert transform; Haar function; Multiresolution approximation; Matlab

1. Introduction

The Hilbert transform of a function $f \in L^p(\mathbb{R})$ ($1 \leq p < \infty$) is defined as the Cauchy principle value integral [9]

$$Hf(x) := \frac{1}{\pi} \text{p.v.} \int_{\mathbb{R}} \frac{f(t)}{x-t} dt. \quad (1.1)$$

Obviously, the integral (1.1) exists for almost all x . In this paper, we consider primarily the case $f \in L^2(\mathbb{R})$, the set of square integrable functions, namely finite energy signals, defined on the real line.

The Hilbert transform plays an important role in many subjects of science and technology such as optics [2], waves in stratified fluids [1,8], signal processing and so on. As early as in 1940s, Gabor used Hilbert transform to define the instantaneous frequency of a non-stationary signal. Let $s(t)$ be a real valued signal and $(Hs)(t)$ be its Hilbert transform. The complex valued signal $z(t) = s(t) + i(Hs)(t) = \rho(t)e^{i\theta(t)}$ is called its analytic signal and the instantaneous frequency of $s(t)$ is defined as $\omega(t) = \frac{1}{2\pi} \frac{d\theta(t)}{dt}$, where $\theta(t) = \arctan\left(\frac{(Hs)(t)}{s(t)}\right)$ [3]. In 1970s, it was proved by Vakman that under some proper physical hypothesis, the Hilbert transform is necessary for the definition of the instantaneous frequency. Not long ago, Huang developed a new method, called Hilbert–Huang transform, for

[☆] This work is supported in part by NSFC (No.60475042,10631080).

* Corresponding author. Tel.: +86 20 84115508; fax: +86 20 84111696.
E-mail address: mcsylh@mail.sysu.edu.cn (L. Yang).

analyzing nonlinear and nonstationary data in [6]. With the help of the Hilbert transform, an energy-frequency-time distribution, called the Hilbert spectrum, is obtained.

However it is usually a very difficult task to compute the Hilbert transform of a given signal. According to the formula (1.1), the numerical computation of the Hilbert transform is unstable due to the singularity of the integrand and the infinity of the integral interval. It is still a challenging question to numerically compute the Hilbert transform effectively.

In the past, researchers made valuable contributions to the computation of the Hilbert transform. In [7], Kress and Martensen applied the rectangle rule to (1.1) to get the following numerical approximation:

$$Hf(x) \approx \frac{2}{\pi} \sum_{n=-\infty}^{+\infty} \frac{f(x - (2n + 1)h)}{2n + 1},$$

where h is the step size.

For $f \in L^2(\mathbb{R})$, it is well known that

$$Hf(k) = \mathcal{F}^{-1}\{-i\text{sgn}(k)\mathcal{F}\{f\}(k)\}$$

holds almost everywhere, where \mathcal{F} stands for the Fourier transform defined as

$$\mathcal{F}\{f\}(k) = \widehat{f}(k) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi kx} dx,$$

and \mathcal{F}^{-1} is the inverse Fourier transform defined by

$$\mathcal{F}^{-1}\{\widehat{f}\}(x) = f(x) = \int_{-\infty}^{\infty} \widehat{f}(k)e^{i2\pi kx} dk.$$

With the values of the function f at the equidistant knots in some interval $[-L, L]$ and utilizing the fast Fourier transform and the inverse Fourier transform, Henrici developed a method [5] to calculate the Hilbert transform approximately.

In 1995, based on the facts that functions

$$\rho_n(x) = \frac{(1 + ix)^n}{(1 - ix)^{n+1}}, \quad n \in \mathbb{Z},$$

are the eigenfunctions of the Hilbert transform and form a complete and orthogonal basis of $L^2(\mathbb{R})$, Weideman express the Hilbert transform of $f \in L^2(\mathbb{R})$ as follows (see [10] for details)

$$Hf(x) = \sum_{n=-\infty}^{+\infty} [-i\text{sgn}(n)]a_n \rho_n(x), \quad (1.2)$$

where

$$a_n = \int_{-\infty}^{+\infty} f(x)\overline{\rho_n(x)} dx, \quad n \in \mathbb{Z} \quad (1.3)$$

are the expansion coefficients. With (1.2), the Hilbert transform of a function f can be obtained immediately if all the coefficients a_n ($n \in \mathbb{Z}$) are known. However, the computation of these coefficients is very complicate and difficult usually. Weideman calculated them only for some special functions in [10].

In the engineering applications, we always like to directly use some existing software packages to compute the Hilbert transform. A built-in function named ‘hilbert’ in Matlab, an extensively used scientific and engineering software tool, is designed for the computation. For a vector v of finite data, the action of ‘hilbert’ on v produces a complex vector $\text{hilbert}(v)$ of the same dimension, whose imaginary is the Hilbert transform of v . The principle of computing the Hilbert transform of v by ‘hilbert’ in Matlab is to calculate the fast Fourier transform (FFT) Fv of v firstly, multiply Fv by the sign multiplier $-i\text{sgn}(\cdot)$, and apply the inverse fast Fourier transform (IFFT) to get the discrete Hilbert transform of v . In the sequel, we call the method for the computation of the Hilbert transform with ‘hilbert’ in Matlab the *Matlab method*, or MM for short. It is obviously seen that the principle of the Matlab method is

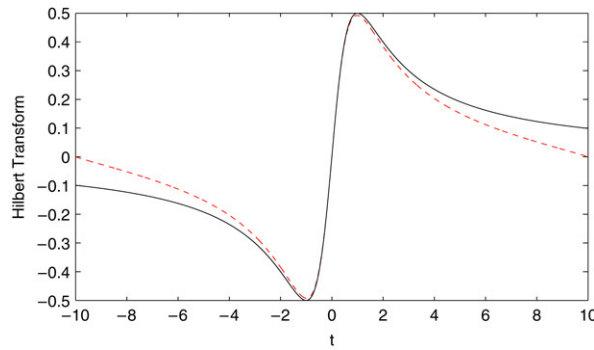


Fig. 1. The Hilbert transform of $f(t) = \frac{1}{1+t^2}$ on the interval $[-10, 10]$. The solid line: the exact result $Hf(t) = \frac{t}{1+t^2}$; the dashed line: the result computed with the Matlab method.

the same as that in [5]. For a given signal $f(t)$, a discrete sampling is needed before the Matlab method being applied and thus the sample error is caused. By virtue of the continuity of the Fourier transform, this error can be reduced by increasing the sampling rate if $f(t)$ is a continuous function. Another error is caused by the truncation of the signal f since only the information of $f(t)$ on some interval $[a, b]$ is used for the computation. Let us consider the Hilbert transform of the function $f(t) = \frac{1}{1+t^2}$. It can be solved analytically as $Hf(t) = \frac{t}{1+t^2}$, whose values on the interval $[-10, 10]$ are shown in Fig. 1 by a solid line. Meanwhile, we truncate the function on $[-10, 10]$ and discretize it by a uniform sample rate of 0.1 to compute its Hilbert transform with Matlab method. The result is shown in Fig. 1 by a dashed line. It is easy to see that the error by the Matlab method is too large to be reliable, particularly in the neighbourhoods of -10 and 10 .

The above example shows that the computation error of the Hilbert transform by the Matlab method cannot be ignored. In signal processing, when the instantaneous frequency of a given signal is computed through the Hilbert transform by Matlab method, the nonnegligible computation error may cause a completely wrong understanding of the physics of the signal. Therefore it is important to develop reliable methods to compute the Hilbert transform. In this paper a new algorithm based on the Haar multiresolution approximation is presented and the error estimation is given. Numerical simulations show encouraging efficacy.

The rest of the paper is organized as follows: In Section 2 a new scheme for the computation of the Hilbert transform is deduced based on the Haar multiresolution approximation. The corresponding error is estimated in Section 3. In Section 4 the proposed method is compared with the Matlab method by numerical simulation, which shows that it outperforms favourably over the Matlab method. Finally, in Section 5, our method is applied to estimate the instantaneous phase of signals and the experimental results are compared with those given in [4].

2. A new scheme for computation of the Hilbert transform based on Haar multiresolution approximation

Multiresolution analysis (MRA), proposed by S. Mallat and Y. Meyer in the late 1980s, essentially amounts to constructing a hierarchy of approximations to functions in various subspaces of a linear vector space. It is well known that the Haar functions $\varphi_{j,k}(x) := 2^{\frac{j}{2}}\varphi(2^jx - k)$ ($j, k \in \mathbb{Z}$), where

$$\varphi(x) = \begin{cases} 1 & 0 \leq x \leq 1, \\ 0 & \text{otherwise} \end{cases}$$

create an orthogonal multiresolution analysis

$$V_j = \overline{\text{span}\{\varphi_{j,k}(x) \mid k \in \mathbb{Z}\}} \quad (j \in \mathbb{Z})$$

of $L^2(\mathbb{R})$. Suppose that P_j is the orthogonal projector from $L^2(\mathbb{R})$ onto V_j . It is easy to see that

$$P_j f(x) = \sum_{k=-\infty}^{+\infty} c_{j,k}(f)\varphi_{j,k}(x) \tag{2.1}$$

for all $f \in L^2(\mathbb{R})$, where

$$c_{jk}(f) = \langle f, \varphi_{j,k} \rangle = \int_{\mathbb{R}} f(x)\varphi_{j,k}(x)dx, \tag{2.2}$$

and we can get

$$\lim_{j \rightarrow +\infty} \|f - P_j f\|_2 = 0, \quad \lim_{j \rightarrow -\infty} \|P_j f\|_2 = 0. \tag{2.3}$$

$P_j f$ is called the Haar multiresolution approximation of $f \in L^2(\mathbb{R})$ on $\{V_j\}_{j \in \mathbb{Z}}$ and (2.3) guarantees that f can be substituted by $P_j f$ for large enough j in practical applications.

It is calculated without difficulty that

$$(H\varphi_{j,k})(x) = (H\varphi)_{j,k}(x) = \frac{1}{\pi} 2^{\frac{j}{2}} \ln \left| \frac{2^j x - k}{2^j x - k - 1} \right| \tag{2.4}$$

for any $x \neq \frac{k}{2^j}$ and $x \neq \frac{k+1}{2^j}$, which together with (2.1) concludes that for any $f \in L^2(\mathbb{R})$,

$$H(P_j f)(x) = \sum_{k=-\infty}^{+\infty} c_{j,k}(f)(H\varphi_{j,k})(x) = \frac{1}{\pi} \sum_{k=-\infty}^{+\infty} c_{j,k}(f) 2^{\frac{j}{2}} \ln \left| \frac{2^j x - k}{2^j x - k - 1} \right| \quad (\forall 2^j x \notin \mathbb{Z}). \tag{2.5}$$

For a continuous function f , the coefficients $c_{j,k}(f)$ in (2.2) can be approximated as follows when j is large enough:

$$c_{j,k}(f) = \int_{\mathbb{R}} f(x)\varphi_{j,k}(x)dx = 2^{-\frac{j}{2}} \int_0^1 f\left(\frac{x+k}{2^j}\right) dx \approx 2^{-\frac{j}{2}} f\left(\frac{2k+1}{2^{j+1}}\right). \tag{2.6}$$

Since Hilbert transform is a unitary operator from $L^2(\mathbb{R})$ onto itself (see [9]), Hf can be approximated by $H(P_j f)$ for j large enough, which together with (2.5) and (2.6) yields the following approximation of Hf :

$$Hf(x) \approx \frac{1}{\pi} \sum_{k=-\infty}^{+\infty} f\left(\frac{2k+1}{2^{j+1}}\right) \ln \left| \frac{2^j x - k}{2^j x - k - 1} \right| \quad (\forall 2^j x \notin \mathbb{Z}). \tag{2.7}$$

In practice only finite samples of function f are given:

$$f_{j,k} = f\left(\frac{2k+1}{2^{j+1}}\right) \quad (k \in K_L := \{k_1, k_2, \dots, k_L\}),$$

where $k_j := k_1 + j - 1$ ($j = 1, 2, \dots, L$), then an extension is necessary to calculate approximately its Hilbert transform using (2.7). Because the values of function f at the points far from the point x are just slightly contributive to $Hf(x)$, the zero extension is reasonable, namely

$$f_{j,k} = f\left(\frac{2k+1}{2^{j+1}}\right) := 0 \quad (k \in \mathbb{Z} \setminus K_L).$$

Consequently (2.7) is rewritten as

$$Hf(x) \approx \frac{1}{\pi} \sum_{k \in K_L} f\left(\frac{2k+1}{2^{j+1}}\right) \ln \left| \frac{2^j x - k}{2^j x - k - 1} \right| \quad (\forall 2^j x \notin \mathbb{Z}). \tag{2.8}$$

For $x = \frac{2n+1}{2^{j+1}}$ ($n \in K_L$) we finally have

$$(Hf)\left(\frac{2n+1}{2^{j+1}}\right) \approx (Hf)_{j,n} := \frac{1}{\pi} \sum_{k \in K_L} f_{j,k}(h_{n-k} - h_{n-k-1}) \quad (n \in K_L), \tag{2.9}$$

where

$$h_k = \ln \left| k + \frac{1}{2} \right| \quad (k \in \mathbb{Z}).$$

In the sequel, we call the method given by (2.8) Haar scaling function method, or HM for short.

3. Error estimation

Below is the main result of this section, which gives the error of our proposed algorithm (2.8).

Theorem 1.

$$\left\| Hf - \frac{1}{\pi} \sum_{k \in K_L} f \left(\frac{2k+1}{2^{j+1}} \right) \ln \left| \frac{2^j \cdot -k}{2^j \cdot -k - 1} \right| \right\|_2 \leq \|f - f_J\|_2 + 3|J|^{1/2} \omega \left(f, \frac{1}{2^{j+1}} \right)_J, \tag{3.1}$$

where f_J is a truncation of f on the interval $J := [\frac{k_1}{2^j}, \frac{k_L+1}{2^j})$ defined by

$$f_J(x) = \begin{cases} f(x) & x \in J, \\ 0 & x \in \mathbb{R} \setminus J, \end{cases}$$

$|J| = \frac{k_L - k_1 + 1}{2^j}$ is the width of the interval J , and $\omega(f, h)_J$ is the continuity modulus of f over the interval J which is defined by

$$\omega(f, h)_J := \sup \{ |f(x) - f(y)| \mid x, y \in J, |x - y| \leq h \} \quad (\forall h > 0).$$

Proof. For any $f \in L^2(\mathbb{R})$, by (2.5) we have for $2^j x \notin \mathbb{Z}$,

$$H(P_j f_J)(x) = \frac{1}{\pi} \sum_{k=-\infty}^{+\infty} c_{j,k}(f_J) 2^{\frac{j}{2}} \ln \left| \frac{2^j x - k}{2^j x - k - 1} \right| \quad (f \in L^2(\mathbb{R})),$$

where $c_{j,k}(f_J)$, defined as (2.6), is easily shown to satisfy.

$$c_{j,k}(f_J) = \begin{cases} c_{j,k}(f) & k \in K_L, \\ 0 & k \notin K_L. \end{cases}$$

Therefore

$$\left| Hf(x) - \frac{1}{\pi} \sum_{k \in K_L} f \left(\frac{2k+1}{2^{j+1}} \right) \ln \left| \frac{2^j x - k}{2^j x - k - 1} \right| \right| \leq E_1(x) + E_2(x) + E_3(x) \tag{3.2}$$

where

$$E_1(x) := |Hf(x) - Hf_J(x)|,$$

$$E_2(x) := |Hf_J(x) - H(P_j f_J)(x)|,$$

$$E_3(x) := \frac{1}{\pi} \left| \sum_{k \in K_L} \left[c_{j,k}(f) 2^{j/2} - f \left(\frac{2k+1}{2^{j+1}} \right) \right] \ln \left| \frac{2^j x - k}{2^j x - k - 1} \right| \right|.$$

Since Hilbert transform is a unitary operator from $L^2(\mathbb{R})$ onto itself, it is clear that

$$\|E_1\|_2 = \|Hf - Hf_J\|_2 = \|f - f_J\|_2. \tag{3.3}$$

and

$$\|E_2\|_2 = \|Hf_J - H(P_j f_J)\|_2 = \|f_J - P_j f_J\|_2.$$

For $x \in \mathbb{R}$, let m be the unique integer satisfying $x \in [\frac{m}{2^j}, \frac{m+1}{2^j})$. Then

$$P_j f_J(x) = \sum_{k \in \mathbb{Z}} c_{j,k}(f_J) \phi_{j,k}(x) = c_{j,m}(f_J) 2^{j/2} = \int_0^1 f_J \left(\frac{t+m}{2^j} \right) dt.$$

If $x \notin J$, then

$$f_J(x) = 0, \quad P_j f_J(x) = \int_0^1 f_J \left(\frac{t+m}{2^j} \right) dt = 0,$$

or else, we have $x \in J$, which implies that $m \in K_L$ and $\frac{t+m}{2^j} \in J$ for any $t \in (0, 1)$. It is followed immediately by

$$\left| f_J \left(\frac{t+m}{2^j} \right) - f_J(x) \right| = \left| f \left(\frac{t+m}{2^j} \right) - f(x) \right| \leq \omega \left(f, \frac{1}{2^j} \right)_J.$$

Hence

$$|P_j f_J(x) - f_J(x)| \leq \omega \left(f, \frac{1}{2^j} \right)_J \leq 2\omega \left(f, \frac{1}{2^{j+1}} \right)_J$$

for any $x \in \mathbb{R}$. Hence

$$\|E_2\|_2 = \|f_J - P_j f_J\|_2 \leq |J|^{1/2} 2\omega \left(f, \frac{1}{2^{j+1}} \right)_J. \tag{3.4}$$

Finally, we turn to estimate $\|E_3\|_2$. According to (2.4), we have

$$E_3(x) = \left| \sum_{k \in K_L} \left[c_{j,k}(f) - 2^{-j/2} f \left(\frac{2k+1}{2^{j+1}} \right) \right] H\phi_{j,k}(x) \right| \quad (\forall 2^j x \notin \mathbb{Z}).$$

It is concluded that $\{H\phi_{j,k}\}_{k \in K_L}$ forms an orthonormal system of $L^2(\mathbb{R})$ since $\{\phi_{j,k}\}_{k \in K_L}$ is such an orthonormal system and H an unitary operator from $L^2(\mathbb{R})$ onto itself. Hence

$$\|E_3\|_2 = \left(\sum_{k \in K_L} \left| c_{j,k}(f) - 2^{-j/2} f \left(\frac{2k+1}{2^{j+1}} \right) \right|^2 \right)^{1/2}.$$

Using (2.6), we have

$$\|E_3\|_2 \leq L^{1/2} 2^{-j/2} \omega \left(f, \frac{1}{2^{j+1}} \right)_J = |J|^{1/2} \omega \left(f, \frac{1}{2^{j+1}} \right)_J. \tag{3.5}$$

Combining (3.2) (3.3) (3.4) and (3.5), the theorem is proved. ■

Estimation (3.1) shows that the L^2 -error of our proposed algorithm (2.8) to compute the Hilbert transform of a continuous function $f \in L^2(\mathbb{R})$ is bounded by two parts: one is

$$\|f - f_J\|_2 = \left(\int_{-a_J}^{a_J} |f(x)|^2 dx + \int_{b_J}^{\infty} |f(x)|^2 dx \right)^{1/2} \tag{3.6}$$

where $a_J := \frac{k_1}{2^j}$ and $b_J := \frac{k_L+1}{2^j}$ are respectively the left and right endpoints of interval J , which is produced by the truncation of f ; another is $3|J|^{1/2} \omega \left(f, \frac{1}{2^{j+1}} \right)_J$ caused by sampling. For a continuous function f , the continuity modulus $\omega(f, \frac{1}{2^{j+1}})_J$ tends to 0 as j tends to infinity. Particularly, if f is differentiable on J then

$$3|J|^{1/2} \omega \left(f, \frac{1}{2^{j+1}} \right)_J \leq 3|J|^{1/2} \frac{1}{2^{j+1}} \max_{x \in J} |f'(x)|. \tag{3.7}$$

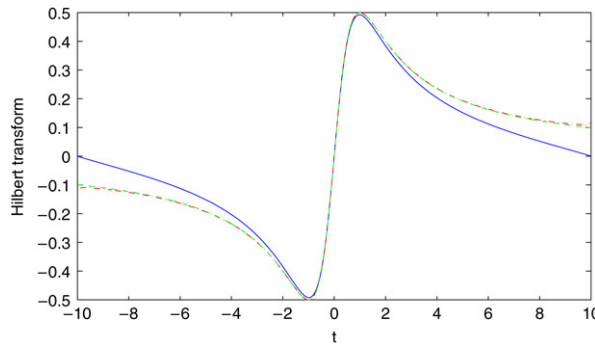


Fig. 2. The Hilbert transforms of $f(x) = \frac{1}{1+x^2}$ computed with our proposed method and MM. Dashed: $Hf(x) = \frac{x}{1+x^2}$; Solid: the result computed with MM; Dotted: our proposed algorithm.

Table 1
Five functions and their Hilbert transforms

$f(x)$	$Hf(x)$
$\frac{1}{1+x^2}$	$\frac{x}{1+x^2}$
$\frac{1}{1+x^4}$	$\frac{x(1+x^2)}{\sqrt{2}(1+x^4)}$
$\frac{\sin x}{1+x^2}$	$\frac{e^{-1}-\cos x}{1+x^2}$
e^{-x^2}	$-\frac{2}{\sqrt{\pi}}e^{-x^2} \int_0^x e^{t^2} dt$
$\frac{\sin x}{1+x^4}$	$\frac{1}{1+x^4} \left[e^{-\frac{1}{\sqrt{2}}} \cos\left(\frac{1}{\sqrt{2}}\right) + e^{-\frac{1}{\sqrt{2}}} \sin\left(\frac{1}{\sqrt{2}}\right) x^2 - \cos x \right]$

4. Experiments and comparison with Matlab’s method

In this section experiments for five functions listed in the left column of Table 1 are conducted. These functions are chosen for our experiments since their Hilbert transforms can be solved analytically, as shown in the right column of Table 1. We shall compute their Hilbert transforms numerically with Matlab method (MM) and our proposed algorithm (2.8) respectively and compare the results accordingly.

As the first example, let us consider function $f(t) = \frac{1}{1+t^2}$ and $J = [-10, 10)$. By (3.6) it is easily to show that the truncation error is

$$\|f - f_J\|_2 = \left(\frac{1}{3|a_J|^3} + \frac{1}{3|b_J|^3} \right)^{1/2} < 0.02582,$$

and according to (3.7) the error caused by sampling is bounded by

$$\begin{aligned} 3|J|^{1/2} \omega\left(f, \frac{1}{2^{j+1}}\right)_J &\leq 3|J|^{1/2} \frac{1}{2^{j+1}} \max_{x \in J} \frac{2|x|}{(1+x^2)^2} \\ &\leq 3|J|^{1/2} \frac{1}{2^{j+1}} \max_{x \in J} \frac{1}{1+x^2} = 3|J|^{1/2} \frac{1}{2^{j+1}} < \frac{7}{2^j}. \end{aligned}$$

Choose $j = 9$ we have $\frac{7}{2^j} < 0.01368$. Hence

$$\left\| Hf - \frac{1}{\pi} \sum_{k \in K_L} f\left(\frac{2k+1}{2^{j+1}}\right) \ln \left| \frac{2^j \cdot -k}{2^j \cdot -k - 1} \right| \right\|_2 < 0.02582 + 0.01368 < 0.0395.$$

The Hilbert transform computed with our proposed algorithm is graphically displayed with dotted line in Fig. 2. To be compared with Matlab’s method, the result computed with MM is shown in the same figure with solid line and

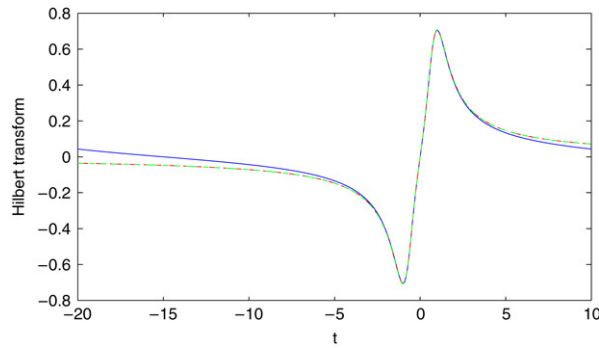


Fig. 3. The Hilbert transforms of $f(x) = \frac{1}{1+x^4}$ computed with our proposed method and MM. Dashed: $Hf(x) = \frac{x(1+x^2)}{\sqrt{2}(1+x^4)}$; Solid: the result computed with MM; Dotted: our proposed algorithm.

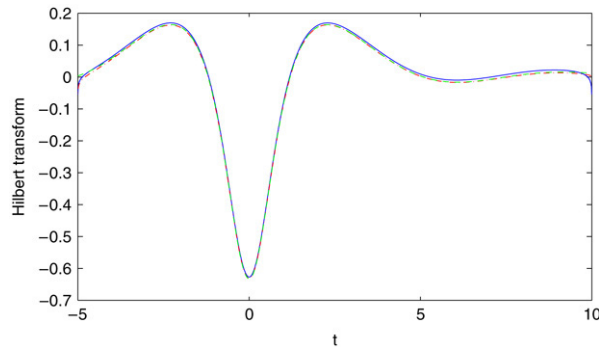


Fig. 4. The Hilbert transforms of $f(x) = \frac{\sin x}{1+x^2}$ computed with our proposed method and MM. Dashed: $Hf(x) = \frac{e^{-1-\cos x}}{1+x^2}$; Solid: the result computed with MM; Dotted: our proposed algorithm.

$Hf(x) = \frac{x}{1+x^2}$ shown with dashed line. One has no difficulty in seeing that the results computed using our method is much better than MM, especially in the neighbourhoods of the boundary of the interval J .

The second example is $f(x) = \frac{1}{1+x^4}$. Choosing $J = [-20, 10]$ and $j = 10$, we can get the following estimation similar to the last example:

$$\left\| Hf - \frac{1}{\pi} \sum_{k \in K_L} f \left(\frac{2k+1}{2^{j+1}} \right) \ln \left| \frac{2^j \cdot -k}{2^j \cdot -k - 1} \right| \right\|_2 < 0.00013 + 0.00977 < 0.01.$$

Fig. 3 show the Hilbert transform of $f(x) = \frac{1}{1+x^4}$ (dashed line) and the numerical results computed numerically by MM (solid line) and our proposed method (dotted line). It is obviously seen that our method provides much finer numerical computation.

The results computed with our proposed and Matlab’s method can be compared similarly. They are shown in Figs. 4–6 respectively corresponding to $f(x) = \frac{\sin x}{1+x^2}$, e^{-x^2} and $\frac{\sin x}{1+x^4}$. All these experiments show that our proposed algorithm gives much finer numerical computation.

Finally, let us consider function $f(x)$ defined by

$$f(x) = \begin{cases} |x+2| & -3 \leq x < -1, \\ |x| & -1 < x \leq 1, \\ |x-2| & 1 < x \leq 3, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

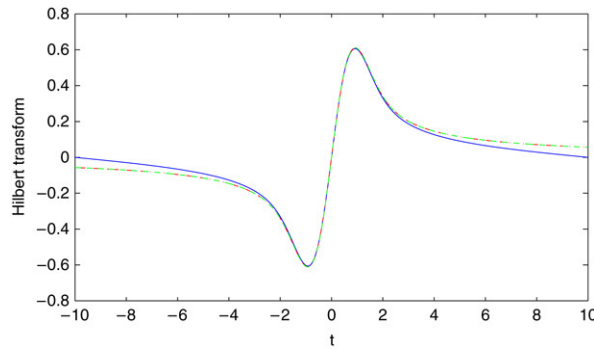


Fig. 5. The Hilbert transforms of $f(x) = e^{-x^2}$ computed with our proposed method and MM. Dashed: $Hf(x) = -\frac{2}{\sqrt{\pi}}e^{-x^2} \int_0^x e^{t^2} dt$; Solid: the result computed with MM; Dotted: our proposed algorithm.

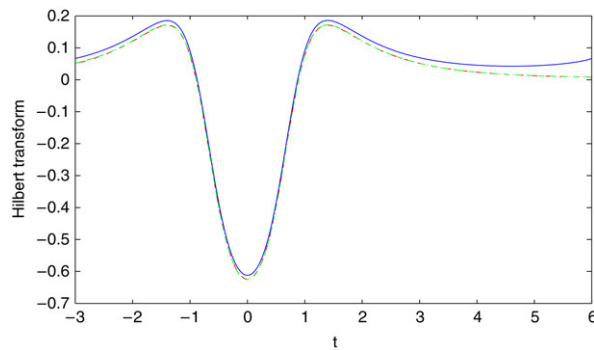


Fig. 6. The Hilbert transforms of $f(x) = \frac{\sin x}{1+x^4}$ computed with our proposed method and MM. Dashed: $Hf(x) = \frac{1}{1+x^4} \left[e^{-\frac{1}{\sqrt{2}}} \cos\left(\frac{1}{\sqrt{2}}\right) + e^{-\frac{1}{\sqrt{2}}} \sin\left(\frac{1}{\sqrt{2}}\right)x^2 - \cos x \right]$; Solid: the result computed with MM; Dotted: our proposed algorithm.

It is a function compactly supported on $[-3, 3]$ and discontinuous at $x = \pm 3$ as shown in Fig. 7. Choosing $J = [-3, 3)$, the truncation error is obviously

$$\|f - f_J\|_2 = 0$$

and the sampling error is

$$3|J|^{1/2}\omega\left(f, \frac{1}{2^{j+1}}\right)_J \leq 3|J|^{1/2}\frac{1}{2^{j+1}} = \frac{3\sqrt{6}}{2^{j+1}} < \frac{1}{2^{j-2}}.$$

Hence

$$\left\| Hf - \frac{1}{\pi} \sum_{k \in K_L} f\left(\frac{2k+1}{2^{j+1}}\right) \ln \left| \frac{2^j \cdot -k}{2^j \cdot -k - 1} \right| \right\|_2 < \frac{1}{2^{j-2}} \rightarrow 0 \quad (j \rightarrow \infty).$$

In this example the error is caused totally by sampling since the truncation error is 0. Fig. 8 shows that the results computed using our method (dotted line) and that by Matlab’s method (solid line) for $j = 10$. To compare them, the Hilbert transform of $f(x)$ is solved analytically as

$$Hf(x) = \frac{1}{\pi}x[2 \ln|x| - \ln|x+1| - \ln|x-1|] + \frac{1}{\pi}(x-2)[2 \ln|x-2| - \ln|x-3| - \ln|x-1|] + \frac{1}{\pi}(x+2)[2 \ln|x+2| - \ln|x+3| - \ln|x+1|] \tag{4.2}$$

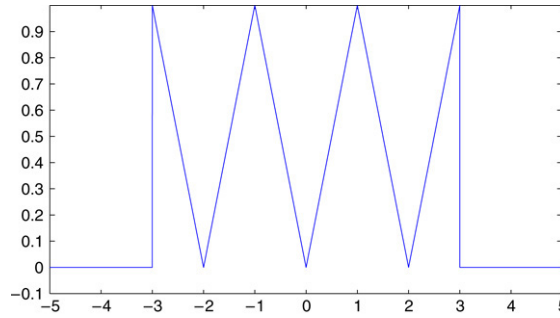


Fig. 7. The function $f(x)$ defined by (4.1).

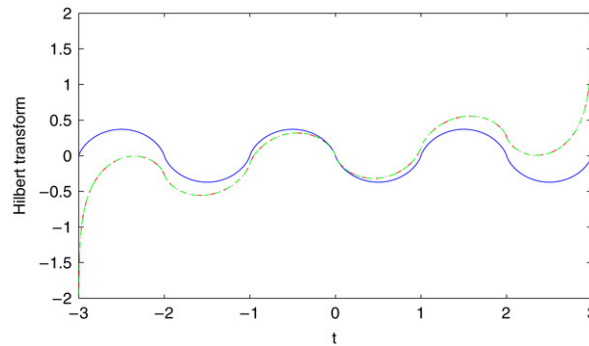


Fig. 8. The Hilbert transforms of $f(x)$ defined by (4.1). Dashed: $Hf(x)$ given by (4.2); Solid: the result computed with MM; Dotted: our proposed algorithm.

and shown with dashed line in Fig. 8. It is seen obviously that our proposed algorithm provides much better performance.

5. Computation of instantaneous phase: Experiments and comparison

As discussed in Section 1, the computation of the Hilbert transform is crucial in accurately estimating the instantaneous frequency of a signal. In [4], three methods: Hilbert transform method, Haar wavelet method, and generalized pencil of function methods, were employed to estimate signals' instantaneous frequency. The Hilbert transform method there is actually the Matlab method mentioned in Section 1 of this paper. With the proposed algorithm in this paper, the Hilbert transform of a signal can be computed numerically based on the Haar multiresolution approximation: consequently the instantaneous frequency can be estimated. In this section, this method is compared with those of [4].

We employ the same signal experimented with in [4] for comparison. Let

$$s(t) = \cos \left(20\pi t + 40\pi \int_0^t m(x)dx \right), \tag{5.1}$$

whose phase $\theta(t) = 20\pi t + 40\pi \int_0^t m(x)dx$ and frequency $\theta'(t)/2\pi = 10 + 20m(t)$ are shown in Fig. 9, where $m(x)$ is a modulating signal taking binary values.

Let us compute the instantaneous phase of the signal $s(t)$ with the four different methods: the Hilbert transform, Haar wavelet, GPOF, and our proposed method. The results are shown in Fig. 10, in which the left column are the phase $\tilde{\theta}(t)$ computed with the four methods and shown in solid lines. For convenience of comparison, the original phase is plotted there with dotted lines. It is easy to see that all the methods provide very good approximation and GPOF is the worst around the points of less smoothness. To show the error around the endpoint, an amplificatory display on [2.55, 2.625] is shown on the second column. It shows that our proposed method provides best approximation near the

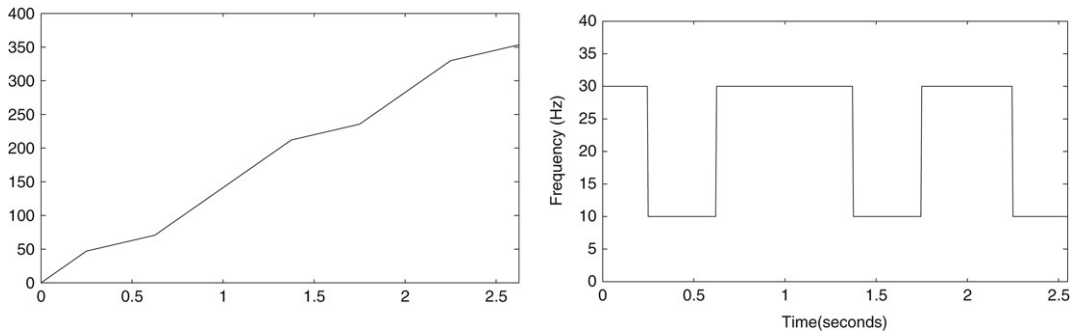


Fig. 9. The phase (left) and the frequency (right) of the signal defined by Eq. (5.1).

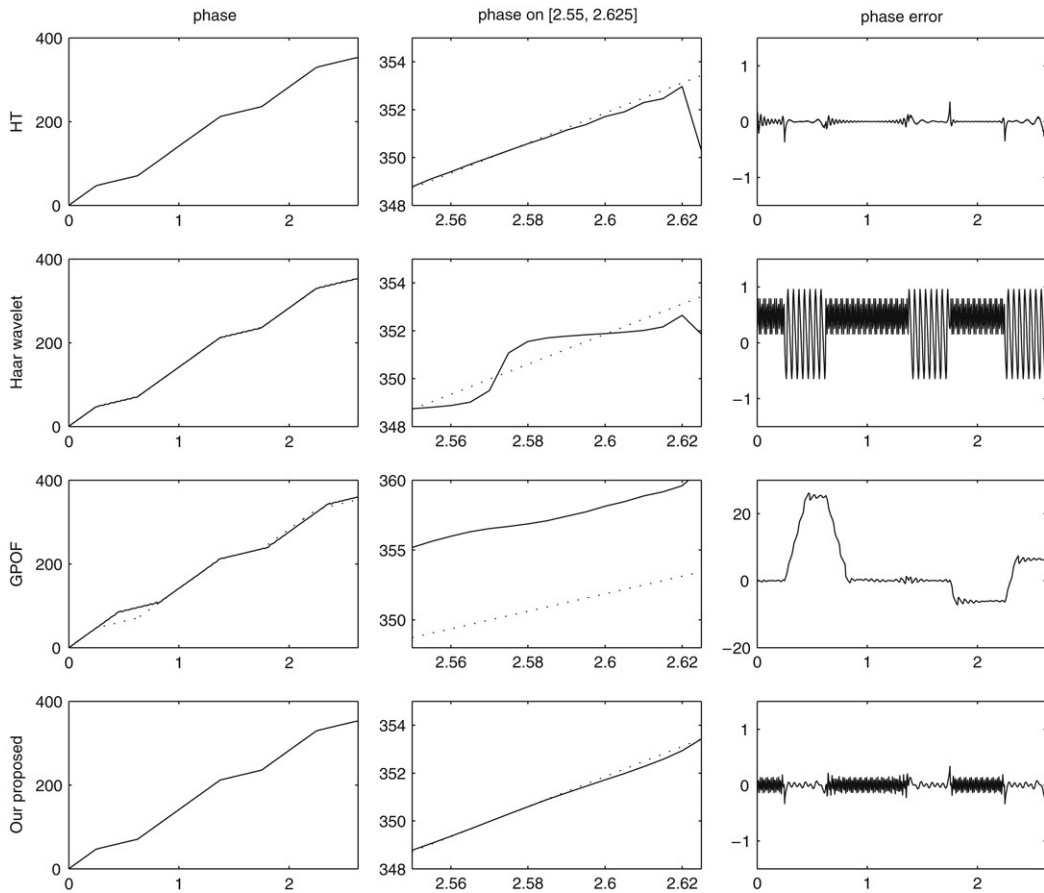


Fig. 10. The instantaneous phase of the signal $s(t)$ and the computation errors. The first column is the original phase $\theta(t)$ (dotted line) and the approximate values $\tilde{\theta}(t)$ (solid line) computed with Hilbert transform method (the first row), Haar wavelet (the second row), GPOF (the third row) and our proposed method (the fourth row) respectively. The second column is an amplificatory display of the first column on [2.55, 2.625]. The third column is the errors $\tilde{\theta} - \theta(t)$ corresponding to the Hilbert transform method, Haar wavelet, GPOF and our proposed method respectively from top to bottom.

right endpoint. The third column are the errors $\tilde{\theta} - \theta(t)$ corresponding to the four methods. It also shows that Hilbert transform and our proposed method outperform the other methods.

To compare the antinoise ability of these methods, we add a Gaussian noise with $SNR = 3.0$ into $s(t)$. The noisy signal is shown in Fig. 11. Then the instantaneous phases computed with the four methods are shown in Fig. 12. The experiment shows that GPOF and our proposed method are least sensitive to noise.

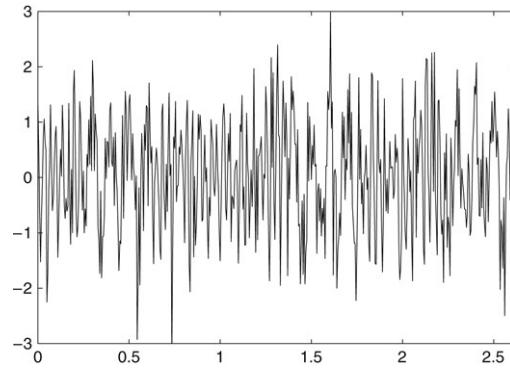


Fig. 11. The noisy signal with additive Gaussian noise of SNR = 3.0.

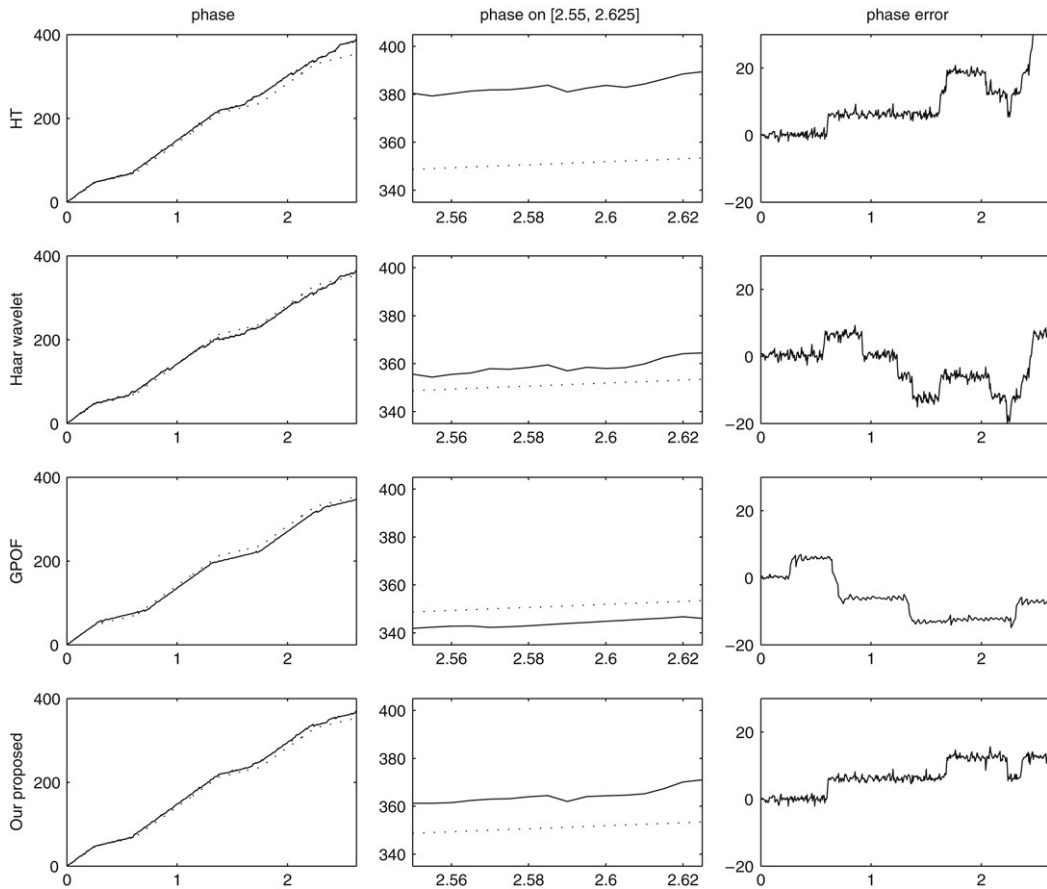


Fig. 12. The instantaneous phase of the noise signal, which is $s(t)$ plus an additive Gaussian noise with SNR = 3.0.

The instantaneous frequency can be obtained by computing the derivative of $\tilde{\theta}(t)$ numerically. However, such computation is usually a typical ill-posed problem. Therefore noisy data need to be smoothed before the computation. Considering the above computational results of the phase provides us with very clear comparison of the methods; we omit the computation of the frequency here.

6. Conclusion

In this paper, we present a novel method based on multiresolution approximation of Haar scaling function for computing the Hilbert transform of a function. The L^2 -error of the proposed method is estimated. Experiments show

that the performances of our algorithm are much better than that of the Matlab method. Finally it is applied to compute the instantaneous phase of signals approximately and is compared with the other three methods given in [4].

Acknowledgments

The authors would like to thank Drs Jaideva C. Goswami and Albert E. Hoefel for providing us with some experimental data and program codes.

References

- [1] T.B. Benjamin, Internal waves of permanent form in fluids of great depth, *J. Fluid Mech.* 25 (1967) 559–592.
- [2] C.F. Bohren, D.R. Huffman, *Absorption and Scattering of Light by Small Particles*, Wiley-Interscience, New York, 1983.
- [3] L. Cohen, *Time-Frequency Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [4] J.C. Goswami, A. Hoefel, Algorithms for estimating instantaneous frequency, *Signal Processing* 84 (8) (2004) 1423–1427.
- [5] P. Henrici, *Applied and Computational Complex Analysis*, Wiley-Interscience, New York, 1986.
- [6] N.E. Huang, Z. Shen, et al., The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis, *Proc. R. Soc. Lond. A* 454 (1998) 903–995.
- [7] R. Kress, E. Martensen, Anwendung der rechteckregel auf die reelle Hilbert transformation mit unendlichem intervall, *Z. Angew. Math. Mech.* 40 (1970) T61–T64.
- [8] H. Ono, Algebraic solitary waves in stratified fluids, *J. Phys. Soc. Japan* 39 (1975) 1082–1091.
- [9] E.M. Stein, G. Weiss, *Introduction to Fourier Analysis on Euclidean Spaces*, Princeton Univ. Press, Princeton, 1971.
- [10] J.A.C. Weideman, Computing the Hilbert transform on the real line, *Math. Comp.* 64 (210) (1995).