

## A VARIANT OF THE EMD METHOD FOR MULTI-SCALE DATA

THOMAS Y. HOU\* and MIKE P. YAN†

*Applied and Computational Mathematics  
California Institute of Technology Pasadena, CA 91125, USA*  
\*hou@acm.caltech.edu  
†yan@acm.caltech.edu

ZHAOHUA WU

*Center for Ocean-Atmospheric Prediction Studies  
and Department of Meteorology  
Florida State University, Tallahassee, FL 32306-2840, USA*  
zwu@fsu.edu

In this paper, we propose a variant of the Empirical Mode Decomposition method to decompose multiscale data into their intrinsic mode functions. Under the assumption that the multiscale data satisfy certain scale separation property, we show that the proposed method can extract the intrinsic mode functions accurately and uniquely.

*Keywords:* Empirical Mode Decomposition (EMD); adaptive data analysis; sparse representation.

### 1. Introduction

The Empirical Mode Decomposition method (EMD for short), which was introduced by Dr. Norden Huang, is an effective tool for adaptive local time–frequency decomposition. Since its introduction in the mid-1990s, it has found successful applications in a number of scientific and engineering fields. The detailed description of the EMD method can be found in the two pioneering papers of Huang *et al.*<sup>10,11</sup> Unlike other traditional data analysis methods, such as the Fourier Transform and various wavelet decomposition methods, the EMD method does not use *a priori* determined basis functions. Moreover, it allows the local time frequency (the so-called instantaneous frequency) and its envelope to vary in time. Thus it is capable of capturing some intrinsic physical features hidden in a nonlinear, nonstationary signal. An important recent development was the introduction of the Ensemble Empirical Mode Decomposition method (EEMD) by Wu and Huang.<sup>16,18</sup> The effects of the decomposition using the EEMD are that the added white noise series cancel each other, and the mean intrinsic mode functions (IMFs) stay within the natural dyadic filter windows as discussed in Flandrin *et al.*<sup>7</sup> and Wu and

Huang,<sup>15,17</sup> significantly reducing the chance of mode mixing and preserving the dyadic property. We refer to the recent review paper<sup>13</sup> for detailed discussions on their latest developments and applications.

Despite of its considerable success, EMD still has some limitations. One of the main difficulties is the lack of a solid mathematical foundation. In some cases, this could lead to ambiguity in choosing the optimal parameters in EMD and the loss of physical uniqueness of the decomposed IMFs. The lack of physical uniqueness makes it difficult to interpret the physical significance of the decomposed IMFs.

One of the main purposes of this paper is to provide some theoretical understanding of EMD for a class of multiscale data. One of the advantages of considering multiscale data is that there is already a well-established theory called homogenization,<sup>1</sup> which provides a guidance for us in decomposing a multiscale signal into its IMFs. We would like to understand under what conditions on the multiscale data that we can decompose them accurately and uniquely into their IMFs. It turns out that the near orthogonality of the IMFs is required to ensure physical uniqueness of the decomposed IMFs. For multiscale data, this near orthogonality property of IMFs is related to the scale separation of the multiscale data among different IMFs. Another main finding of this paper is that if the multiscale data have a sparse representation in certain bases, then one can exploit this sparsity property of the multiscale data to obtain highly accurate and efficient recovery of the IMFs.

Let us first recall the basic steps in EMD. For a given signal,  $f(x)$ , we would like to decompose it as the sum of a local median  $m_{n-1}(x)$ , and an IMF, which is expressed as  $a_n(x) \cos(\theta_n(x))$ :

$$f(x) = m_{n-1}(x) + a_n(x) \cos(\theta_n(x)). \quad (1)$$

The EMD method provides an approximation to the local median via a sifting procedure. Specifically, the EMD method uses a cubic spline polynomial to interpolate all the local maxima to obtain an upper envelope, and a cubic spline to interpolate all the local minima to obtain a lower envelope, then average the upper and lower envelopes to obtain an approximate median for  $m_{n-1}(x)$ . One then decides whether or not to accept the obtained  $m_{n-1}(x)$  as our local median depending on whether  $f(x) - m_{n-1}(x)$  gives an acceptable IMF. The conditions on the IMFs are (roughly) the following: (1) the number of zeros and the number of extrema of an IMF must be equal or differ at most by one, (2) an IMF must be “symmetric” with respect to zero (the local median is zero). If  $f(x) - m_{n-1}(x)$  does not satisfy these conditions, one can treat  $f(x) - m_{n-1}(x)$  as a new signal and construct a new candidate for IMF for  $f(x) - m_{n-1}(x)$  using the same procedure described above. This sifting procedure continues until we obtain a satisfactory IMF, which we denote as  $f_n(x)$ . Now we can treat  $f(x) - f_n(x)$  as a new signal, and apply the same procedure to generate the second IMF,  $f_{n-1}(x)$ . This procedure can continue until  $f_0(x)$  no longer contains any local extrema.

After the individual IMFs are found by EMD, their instantaneous frequencies are typically obtained via the Hilbert transform. However, this technique can be justified only under certain conditions that are seldom satisfied by components of practical data. Some empirical methods have been proposed to circumvent this difficulty with some success, but pose their own problems.

In this paper, we propose several alternative ways of determining the instantaneous frequency without using the Hilbert Transform. We consider two types of data. The first type of data is a periodic data set consisting of a smooth median function  $a_0(x)$ , a smooth envelope function,  $a_1(x)$ , and a smooth phase function,  $\theta_1(x)$  of the form:

$$f(x) = a_0(x) + a_1(x) \cos \theta_1(x). \quad (2)$$

We assume that  $a_0$ ,  $a_1$  and  $\theta_1'$  have a sparse representation in the Fourier bases in the sense that the number of nonzero Fourier coefficients is small compared with the size of the data. It is important to point out that the data we consider are not sparse in the Fourier bases although we assume that  $a_0$ ,  $a_1$  and  $\theta_1'$  are sparse in the Fourier bases. Further, we assume that the data satisfy certain scale separation property, which will be made precise in the next section. For this class of periodic data, we introduce a variant of EMD which we call *the Newton–Raphson-based EMD method*. By exploiting the sparsity of data, we design a Newton–Raphson iterative method to recover the sparse Fourier coefficients of  $a_0$ ,  $a_1$ , and  $\theta_1'$  accurately and efficiently. We remark that the Newton–Raphson-based EMD method can be applied to nonperiodic data as long as  $a_0$ ,  $a_1$  and  $\theta_1'$  have a sparse representation with respect to certain bases which could be cubic spline bases, wavelet bases, or curvelet bases for example.

The Newton–Raphson-based EMD method can be considered as a nonlinear version of the sifting procedure in the original EMD method since the Newton–Raphson iterations provide a nonlinear correction mechanism to correct the errors we make in the sifting algorithm used in the EMD method. This method can also be considered as a nonlinear version of compressed sensing<sup>2–5</sup> in the sense that the bases in which the signal is sparse are unknown *a priori*. These bases are constructed adaptively to fit a given physical signal in a sparsest possible way so that it preserves some important physical features of the signal. This is similar in spirit to the Multiscale Finite Element Methods (see e.g. Refs. 6 and 8), in which multiscale finite element bases are constructed adaptively to incorporate the key physical multiscale features of the underlying solution into the bases so that we obtain an uniformly accurate coarse grid approximation (or sparse representation) of the multiscale solution via the local multiscale bases.

The convergence of the Newton–Raphson method depends critically on our ability in producing an accurate initial guess. To this end, we propose a second order method, via either EMD or least square spline fit (LSSF), to approximate  $a_0$  and  $a_1$ . Once we obtain an accurate approximation to  $a_0$  and  $a_1$ , we can produce an

approximation to  $\theta_1$  by using the relationship (2). However, this involves the division operation which could be unstable or lead to loss of accuracy when  $a_1$  has small amplitude in some grid points. To overcome this difficulty, we introduce a second-derivative method which gives an accurate approximation to the instantaneous frequency without the need of first obtaining a good approximation of  $a_1$ . This method works well for data whose envelope functions have small amplitude. Our numerical experiments show that the Newton–Raphson-based EMD method gives a very accurate recovery of the IMFs when the data satisfy our scale-separation condition. The recovery can be made as accurate as the machine precision when the amplitude is not too small.

The method that we use in generating the initial guess for the Newton–Raphson iterations is robust and it is interesting by itself. With some extra effort, we generalize it to study the second type of data, the multiscale data (nonperiodic), which are of the form:

$$\begin{aligned} f(x) &= f_0(x) + f_1(x, x/\epsilon_1(x)) + f_2(x/\epsilon_1(x), x/\epsilon_2(x)) \\ &= a_0(x) + a_1(x) \cos(x/\epsilon_1(x)) + a_2(x/\epsilon_1(x)) \cos(x/\epsilon_2(x)), \end{aligned} \quad (3)$$

where  $a_i(x)$  and  $\epsilon_i(x)$  ( $i = 0, 1, 2$ ) are assumed to be smooth, and  $0 < \epsilon_2(x) \ll \epsilon_1(x) \ll 1$ . The method for determining  $\epsilon_i(x)$  ( $i = 1, 2$ ) now involves solving a first order nonlinear ordinary differential equation. For this reason, we call this method *the ODE-based EMD method*. For multiscale data that satisfy a similar scale separation property, we demonstrate that the ODE-based EMD method can recover the exact IMFs accurately and uniquely, even for multiscale data with small amplitude in their envelopes or nearly singular instantaneous frequencies. It is worth emphasizing that the ODE-based EMD method extracts the envelope of an IMF, and its instantaneous frequency independently. Thus they can be computed independently and in parallel.

We also compare the method that uses the local median with the method that uses the local mean in the EMD decomposition. Under the assumption that the envelope function and the instantaneous frequency of an IMF are smoother than the IMF itself, both approaches give accurate approximations to the envelope and the instantaneous frequency. However, when this assumption is violated, the method that uses the local median gives a much more accurate EMD decomposition than the method that uses the local mean. We provide some preliminary analysis and numerical evidence to support this conclusion.

We also study how to deal with noisy data using our approach. As we mentioned earlier, EEMD provides an effective way to deal with noisy data by adding additional white noise to the noisy signal and decomposing the noise as an IMF. While EEMD is robust, it could be expensive since it needs to apply EMD to a large number of random samples with the added Gaussian noise. In this paper, we propose an alternative denoising method based on the Least Square Spline Fitting method (LSSF) to extract the noise from a noisy signal. Our preliminary numerical experiments show that the LSSF denoising method gives results

comparable with those obtained by EEMD but with considerable computational saving.

We remark that the methods we introduce in this paper still have some limitations. They work only for special synthetic data that satisfy our scale separation property. More effort is required to generalize these methods for more complicated data that arise from realistic applications. This important issue will be addressed in an upcoming paper using a different approach.<sup>9</sup>

The rest of this paper is organized as follows. In Sec. 2, we introduce the Newton–Raphson-based EMD method for periodic data. We also describe an accurate and robust method to generate the initial guess for the Newton–Raphson-based EMD method. In Sec. 3, we introduce the ODE-based EMD method for multiscale data. In Sec. 4, we introduce our LSSF denoising method for noisy data and compare its performance with that of EEMD. Some concluding remarks are given in Sec. 5.

## 2. Sparse Periodic Data

In this section, we consider periodic data which satisfy certain scale separation property. Specifically, we consider data that consist of  $n$  number of IMFs, each of them has the form:

$$f_j(x) = a_j(x) \cos \theta_j(x), \quad j = 1, \dots, n, \quad a \leq x \leq b, \quad (4)$$

where  $a_j(x)$  is a positive envelope function, and  $\theta_j(x)$  the phase function, and its instantaneous frequency is defined as  $d\theta_j(x)/dx$ . We assume that the data satisfy the following the scale separation property:

**Assumption 1 (Scale-Separation).** The IMFs  $\{f_j\}$  ( $j = 1, \dots, n$ ) are said to satisfy the scale-separation property if (1)  $f_i(x)$ ,  $f_j(x)$  with  $i \neq j$  are nearly orthogonal; (2) for each pair of IMFs  $(f_{j-1}, f_j)$  ( $1 \leq j \leq n$ ),  $f_{j-1}$ ,  $a_j$ , and  $d\theta_j(x)/dx$  are smoother than  $\cos(\theta_j(x))$ .

Throughout this paper, we say that  $g_1(x)$  is smoother than  $g_2(x)$  if the amplitude of the first order derivative of  $g_1$  is much smaller than that of  $g_2$ , that is

$$\left| \frac{g_1'(x)}{g_2'(x)} \right| \ll 1.$$

The above assumption implies that  $f_{j-1}$  is smoother than  $f_j$ . This is to ensure that we can decompose a given signal by a finite number of IMFs. The near orthogonality condition between two different IMFs is to ensure the physical uniqueness of the EMD decomposition. We can measure the degree of orthogonality between  $f_i$  and  $f_j$  ( $j \neq i$ ) by using the following quantity:

$$\mu(f_i, f_j) = \frac{1}{\|f_i\|_{L^2[a,b]} \|f_j\|_{L^2[a,b]}} \int_a^b f_i(x) f_j(x) dx. \quad (5)$$

This quantity is similar to the mutual coherence of two columns in a matrix which has been used in the Compressed Sensing literature, see e.g. page 40 of the recent review paper.<sup>2</sup> Near orthogonality between  $f_i$  and  $f_j$  requires that the mutual coherence between  $f_i$  and  $f_j$  be small. Without imposing the near orthogonality condition on  $f_i(x)$  and  $f_j(x)$ , there may be more than one way to decompose the data into their IMFs, thus violating the physical uniqueness property. How small the mutual coherence needs to be in order to guarantee the physical uniqueness of the EMD decomposition is a nontrivial question which requires further analysis. This will be investigated in an upcoming paper.<sup>9</sup>

Next, we will discuss a variant of the EMD method to separate the median,  $f_0(x)$ , from its original signal  $f(x)$ .

### 2.1. Approximation of the local median

In this section, we discuss how to approximate the local median,  $f_0$ , from a given signal,  $f$ . Again, we express the signal  $f$  in the form:

$$f(x) = f_0(x) + a_1(x) \cos \theta_1(x). \quad (6)$$

Under the assumption stated above, one can show that

$$\frac{df(x)}{dx} \approx a_1(x) \frac{d \cos \theta_1(x)}{dx}.$$

By  $f \approx g$ , we mean that  $|f - g| \ll 1$ . The above statement can be verified directly by our assumption that  $a_0$  and  $a_1$  are smoother than  $\cos(\theta_1(x))$ . Thus, the local maximum of  $f(x)$  is achieved approximately at the points in which  $\cos \theta_1(x) = 1$ . Denote  $x_{\max}$  the points of local maxima of the discretized data and  $x_{\min}$  the minima. Then we have  $\cos \theta_1(x_{\max}) \approx 1$  and  $\cos \theta_1(x_{\min}) \approx -1$ .

#### 2.1.1. EMD

By applying a cubic spline interpolation, we can get an approximation of the upper envelope function which connects all the local maxima. The upper envelope, denoted as  $\bar{f}(x)$ , is given approximately as

$$\bar{f}(x) \approx f_0(x) + a_1(x). \quad (7)$$

Similarly, the lower envelope function, denoted as  $\underline{f}(x)$ , is given approximately by

$$\underline{f}(x) \approx f_0(x) - a_1(x). \quad (8)$$

Note that the accuracy of the upper and lower envelope functions depends on both the local wavelength of the given signal and the grid resolution in which we sample the signal. The accuracy of the spline interpolation depends on the local wavelength. On the other hand, the accuracy of the location of the local extrema,  $x_{\max}$  or  $x_{\min}$ , depends on the grid resolution.

We define the *local median* of a signal to be the average of the two envelope functions given in (7) and (8), respectively. This local median gives a good approximation to the low frequency IMF,  $f_0(x)$ ,

$$f_0(x) \approx \frac{1}{2}(\bar{f}(x) + \underline{f}(x)).$$

After we obtain an approximation to  $f_0$ , we can extract the high frequency IMF by defining  $f_1(x) = f(x) - f_0(x)$ .

2.1.2. *Least square spline fit (LSSF)*

In this section, we introduce the Least Square Spline Fit method (LSSF), which we will use extensively throughout this paper. First, we divide the domain into finite intervals, i.e.

$$a = x_0 < x_1 < \dots < x_m = b.$$

By our assumption, the *median* function  $f_0(x)$  and the envelope  $a_1(x)$  are smoother than the original signal. We will approximate  $f_0$  and  $a_1$  by piecewise polynomials  $\tilde{f}_0(x)$  and  $\tilde{a}_1(x)$ , respectively, with the following properties:

- (1)  $\tilde{f}_0(x), \tilde{a}_1(x) \in C^2(a, b)$ , i.e. they are second order continuously differentiable in the domain  $[a, b]$ .
- (2)  $\tilde{f}_0(x)$  and  $\tilde{a}_1(x)$  are third order polynomials in each interval  $[x_{k-1}, x_k]$  for  $k = 1, \dots, m$ .

It is not difficult to see that the basis functions that generate this class of polynomials are given by

$$1, x, x^2, \left( \begin{cases} (x - x_k)^3, & \text{for } x > x_k; \\ 0 & \text{for } x < x_k, \end{cases} \right)_{k=0,1,\dots,m-1}. \tag{9}$$

In order to find the least-square fit of  $\tilde{f}_0(x)$  and  $\tilde{a}_1(x)$  in terms of the bases given in Eq. (9), we formulate it as a least square problem by imposing the following condition using the data from the local extrema

$$\begin{aligned} f(x_{\max}) &= \tilde{f}_0(x_{\max}) + \tilde{a}_1(x_{\max}), \\ f(x_{\min}) &= \tilde{f}_0(x_{\min}) - \tilde{a}_1(x_{\min}). \end{aligned}$$

In general, LSSF requires that the number of extrema be greater than  $m + 3$ , and the system is solved by the least square method.

2.2. *Local median versus local mean*

Another possible approach to approximate  $f_0$  is to use the local mean of the signal instead of its local median. The local mean can be obtained by averaging the signal over one local period, which is defined as the interval between two local maxima

or minima. Under Assumption 1,  $f_0(x)$  and  $a_1(x)$  can be considered approximately as constant in one local period of the signal. Let  $x_1$  and  $x_2$  be the two neighboring grid points at which the signal achieves its local maxima or minima. Using Eq. (6), we approximate the local mean at the midpoint between  $x_1$  and  $x_2$  as follows:

$$\tilde{f}(x_0) \equiv \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} f(x) dx \approx f_0(x_0) + a_1(x_0) \int_{x_1}^{x_2} \cos \theta_1(y) dy,$$

where  $x_0 = 1/2(x_1 + x_2)$ . If we further have  $\int_{x_1}^{x_2} \cos \theta_1(y) dy = 0$ , then the local mean  $\tilde{f}(x_0)$  gives a good approximation to  $f_0(x_0)$ . However, in many cases, this is not true. If we make an additional assumption that  $\theta'_1(x)$  is smoother than  $\cos \theta_1(x)$ , then one can show easily that  $\int_{x_1}^{x_2} \cos \theta_1(y) dy \approx 0$ . Then we have  $\tilde{f}(x_0) \approx f_0(x_0)$ .

Our numerical results show that it is in general more accurate to approximate  $f_0$  using its local median than using its local mean. For example, if we choose  $f_0 = 0$ ,  $a_1 = 1$ , and  $\theta_1 = 2\pi x + 1 + \cos(2\pi x)$ , then we have  $\theta'_1 = 2\pi(1 - \sin(2\pi x))$ , which is as oscillatory as  $\cos(\theta_1)$  itself. In this case,  $\int_{x_1}^{x_2} \cos \theta_1(y) dy = 0$  is not valid and the local mean of  $f(x)$  is not zero, see Fig. 1. If we approximate  $f_0$  by its local mean, then  $a_1$  will be highly oscillatory with frequency of the same order as that of  $\cos(\theta_1)$ . This would violate Assumption 1. This is not surprising since this signal does not satisfy the scale-separation condition stated in Assumption 1.

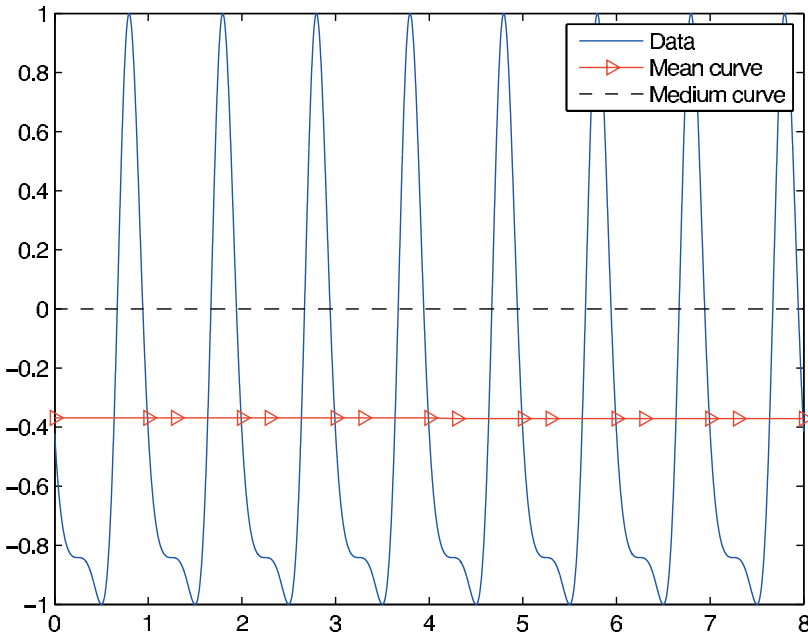


Fig. 1.  $f(x) = \cos(2\pi x + 1 + \cos 2\pi x)$ . The median curve is obtained by averaging the upper and lower envelopes. The mean curve is obtained by averaging the upper mean curve, which interpolates the local mean between two neighboring maxima, and the lower mean curve, which interpolates the local mean between two neighboring minima.



### 2.3. EMD decomposition via the Newton–Raphson method

In this section, we introduce an iterative method based on the Newton–Raphson iterations to improve the accuracy of the IMFs. The success of this iterative EMD method depends on two factors. The first one is that the signal has a sparse representation in some appropriate bases. The second one is that we can generate a good initial guess for the Newton–Raphson iterations. In this paper, we will only focus on the simplest case of periodic data. We assume that  $a_0$ ,  $a_1$  and  $\theta'_1$  have a sparse representation in the Fourier bases. More precisely, we assume that  $a_0(x)$ ,  $a_1(x)$  and  $\theta_1(x)$  have the following form:

$$a_0(x) = b_0^{(0)} + \sum_{k=1}^{K_0} b_{2k-1}^{(0)} \cos(2\pi kx) + \sum_{k=1}^{K_0} b_{2k}^{(0)} \sin(2\pi kx); \tag{10}$$

$$a_1(x) = b_0^{(1)} + \sum_{k=1}^{K_1} b_{2k-1}^{(1)} \cos(2\pi kx) + \sum_{k=1}^{K_1} b_{2k}^{(1)} \sin(2\pi kx); \tag{11}$$

$$\theta_1(x) = 2T\pi x + b_0^{(2)} + \sum_{k=1}^{K_2} b_{2k-1}^{(2)} \cos(2\pi kx) + \sum_{k=1}^{K_2} b_{2k}^{(2)} \sin(2\pi kx). \tag{12}$$

The sparsity means that  $K_0 + K_1 + K_2$  is relatively small compared with the size of the discrete data set  $f(x)$ . In this case, the EMD decomposition is equivalent to determining the Fourier coefficients of  $a_0$ ,  $a_1$  and  $\theta_1$ . Obviously, this idea can be generalized to data which have a sparse representation in other bases, which are not necessarily Fourier bases.

In our implementation, we will choose an upper bound for  $K_0$ ,  $K_1$ , and  $K_2$ . We rearrange all the Fourier coefficients of  $a_0(x)$ ,  $a_1(x)$  and  $\theta_1(x)$  into a new vector  $\mathbf{b}$ . Then the EMD decomposition is reduced to solve a nonlinear system with an unknown vector  $\mathbf{b}$ , i.e.

$$F(x_j; \mathbf{b}) = f(x_j), \quad j = 1, 2, \dots, N,$$

where  $x_j$  ( $j = 1, 2, \dots, N$ ) are the discrete sampling grid points. We will use the Newton–Raphson Iterative Method to solve the above nonlinear system. Specifically, if we already obtain the  $n$ th iteration,  $\mathbf{b}^{(n)}$ , then we update the  $(n + 1)$ th iteration by

$$\mathbf{b}^{(n+1)} = \mathbf{b}^{(n)} - B(\mathbf{b}^{(n)})f(\mathbf{b}^{(n)}), \tag{13}$$

where  $B$  is an approximate inverse to the Jacobian matrix

$$D(\mathbf{b}) = \partial f / \partial \mathbf{b}.$$

Under the sparsity assumption, we have  $N \gg (K_0 + K_1 + K_2)$ . This means that the Jacobian matrix  $D$  has many more rows than columns. Therefore, we cannot invert the Jacobian matrix directly. Instead, we approximate the inverse of the

Jacobian matrix using the least square method. More precisely, if we denote  $y^{(n)} = B(\mathbf{b}^{(n)})f(\mathbf{b}^{(n)})$ , we approximate  $y^{(n)}$  by solving the following least square problem:

$$(D^T(\mathbf{b}^{(n)})D(\mathbf{b}^{(n)}))y^{(n)} = D^T(\mathbf{b}^{(n)})f(\mathbf{b}^{(n)}). \tag{14}$$

The method we described above involves solving a nonlinear system using the Newton–Raphson method. We will call this method *the Newton–Raphson-based EMD method*.

For the convergence of the Newton–Raphson iterative method, it is essential to generate a good initial guess for  $\mathbf{b}$ . Next we discuss how to generate a good initial guess for the coefficients of  $a_0(x)$ ,  $a_1(x)$  and  $\theta_1(x)$ .

**2.4. Initial guess of  $a_0(x)$  and  $a_1(x)$**

The initial guess for  $a_0$  and  $a_1$  can be generated by using the upper and lower envelopes defined in Sec. 2.1. In order to improve the accuracy in approximating the upper and lower envelopes, we use a second order accurate interpolation to generate a more accurate approximation to the local extrema. Specifically, if  $f(x_j)$  is a discrete extremum point of the data, we generate an updated extremum at  $x_j + \delta x$  with second order accuracy,

$$\delta x = \begin{cases} \frac{f(x_{j-1}) - f(x_{j+1})}{2(f(x_{j-1}) + f(x_{j+1}) - 2f(x_j))}, & \text{for } f(x_{j-1}) + f(x_{j+1}) \neq 2f(x_j); \\ 0, & \text{for } f(x_{j-1}) + f(x_{j+1}) = 2f(x_j). \end{cases}$$

The function value at the updated extremum is updated to

$$f(x_j) - \frac{\delta x^2}{2}(f(x_{j-1}) - 2f(x_j) + f(x_{j+1})).$$

With the updated extrema, we can use either a finite Fourier interpolation of the updated extrema with some low pass filtering (Fourier smoothing) or LSSF to generate an approximation to the upper and lower envelope functions,  $\bar{f}(x)$  and  $\underline{f}(x)$ , respectively:

$$\begin{aligned} \bar{f}(x) &\approx a_0(x) + a_1(x), \\ \underline{f}(x) &\approx a_0(x) - a_1(x), \end{aligned}$$

from which we obtain

$$\begin{aligned} a_0(x) &\approx \frac{1}{2}(\bar{f}(x) + \underline{f}(x)), \\ a_1(x) &\approx \frac{1}{2}(\bar{f}(x) - \underline{f}(x)). \end{aligned}$$

Once we obtain  $a_0$  and  $a_1$ , we can extract their finite Fourier coefficients by using the least square method. This produces a good initial guess for the Fourier coefficients of  $a_0$  and  $a_1$ .

**2.5. Initial guess of instantaneous frequency  $\theta'_1(x)$**

There are several ways to generate a good initial guess for  $\theta_1$ . A natural approach is to use the relation:

$$\cos \theta_1(x) = \frac{f_1(x)}{a_1(x)}, \tag{15}$$

where  $f_1(x) = f(x) - a_0(x)$ . First, we would like to point out that an interesting method has been recently proposed in Ref. 14 without using the Hilbert transform. This method uses the following equation:

$$\theta'_1(x) \approx \left| \frac{\cos \theta_1(x_{j+1}) - \cos \theta_1(x_{j-1}))}{2\Delta x \sqrt{1 - \cos \theta_1(x_j)}} \right|. \tag{16}$$

Note that the above formula is not well defined at those points in which  $\cos \theta_1(x_j) = 1$ . Thus, one cannot use this formula around those points. Instead one may interpolate the instantaneous frequency based on those points in which  $\cos \theta_1(x_j) \neq 1$ , e.g.  $\cos \theta_1(x_j) < 0.9$ . Another difficulty is that due to numerical errors in approximating  $a_0$  and  $a_1$ , the amplitude of the approximated cosine function (15) at some grid points may be strictly greater than one. In this case, one needs to normalize the right hand side of (15) so that its amplitude is less than or equal to one for all grid points. While such procedure works well for many cases, it suffers from some numerical instability in the region where the amplitude of  $a_1$  is small.

Next, we introduce a different method which uses the discrete second-derivative of the signal to determine the instantaneous frequency  $\theta'_1(x)$ . This method is second order accurate and requires that the envelope function be smooth and satisfy  $a''_1(x)/a_1(x) \approx 0$ . This requirement can be easily satisfied by normalizing the IMF with the approximated envelope given by EMD or least-square-spline method. In this case, we would have  $a_1(x) \approx 1$ .

Our method explores the smoothness of  $a_1$  and  $\theta_1$ . By using a Taylor expansion of  $f_1(x_{j+1})$  and  $f_1(x_{j-1})$  around  $x_j$ , we obtain

$$\begin{aligned} f_1(x_{j+1}) &= \left( a_1(x_j) + \Delta x a'_1(x_j) + \frac{1}{2} \Delta x^2 a''_1 \right) \cos \left( \theta_1(x_j) + \Delta x \theta'_1(x_j) + \frac{1}{2} \Delta x^2 \theta''_1 \right) \\ &\quad + o(\Delta x^2), \\ f_1(x_{j-1}) &= \left( a_1(x_j) - \Delta x a'_1(x_j) + \frac{1}{2} \Delta x^2 a''_1 \right) \cos \left( \theta_1(x_j) - \Delta x \theta'_1(x_j) + \frac{1}{2} \Delta x^2 \theta''_1 \right) \\ &\quad + o(\Delta x^2). \end{aligned}$$

Thus, we have

$$\frac{f_1(x_{j+1}) + f_1(x_{j-1})}{2f_1(x_j)} = \cos \left[ \Delta x \theta'_1 \sqrt{1 - \tan^2 \theta_1(x_j) \left( \frac{\theta''_1}{(\theta'_1)^2} + \frac{2a'_1}{a_1 \theta'_1} \right) + \frac{a''_1}{a_1 (\theta'_1)^2}} \right] + o(\Delta x^2).$$

We apply the above formula at the local extrema of the IMF such that  $\tan \theta_1(x_j) = 0$ . By using the assumption  $a''_1/a_1 \approx 0$ , we can further simplify the above equation to

$$\frac{f_1(x_{j+1}) + f_1(x_{j-1})}{2f_1(x_j)} = \cos[\Delta x \theta'_1(x_j)] + o(\Delta x^2).$$

Note that since we apply our formula only at the local extrema, the left-hand side of the above equation is less than or equal to one. Then the values of instantaneous frequency at those local extrema of an IMF are approximated by

$$\theta'_1(x_j) = \frac{1}{\Delta x} \arccos \frac{f_1(x_{j+1}) + f_1(x_{j-1})}{2f_1(x_j)}. \tag{17}$$

Then, we can interpolate the instantaneous frequency using EMD or LSSF. The scale-separating property implies that the instantaneous frequency can be well approximated by its values at the local extrema.

One advantage of this method is that it can be applied directly to the signal of the form  $f_1(x) = a_1(x) \cos \theta_1(x)$  which satisfies  $a''_1(x)/a_1(x) \approx 0$ . As we will see later, this method works well even when the approximation of the envelope  $a_1(x)$  is not very accurate, see e.g. our simulations of signals (24) and (25).

In addition, we can exploit the sparsity of the instantaneous frequency with respect to certain bases. If the instantaneous frequency is sparse in the Fourier space, we can extract its Fourier coefficients from those local extrema as we did for the upper and lower envelopes. This gives a very accurate recovery of the instantaneous frequency.

**2.6. Numerical results using the Newton–Raphson-based EMD method**

In this section, we present some numerical results to demonstrate how well the Newton–Raphson-based EMD method works. Without loss of generality, we assume that all signals have period 1. We test four different signals, which all have sparse Fourier representations in  $a_0(x)$ ,  $a_1(x)$  and  $\theta'_1(x)$ . In the first two examples, the amplitude of  $a_1$  is not small. In this case, it is relatively easier to obtain a perfect recovery. In the last two examples, we test a more difficult case in which the amplitude of  $a_1$  is small in some grid points. We will show that the Newton–Raphson iterations still converge and give a reasonably accurate recovery of its IMFs and instantaneous frequency.

### 2.6.1. Regular periodic data

We first test the signal for which  $a_1$  is not small. We choose the signal of the form

$$f(x) = a_0(x) + a_1(x) \cos \theta_1(x), \tag{18}$$

and sample 256 equally spaced points in the domain  $[0, 1)$ . In our example,  $a_0(x)$ ,  $a_1(x)$ , and the instantaneous function,  $\theta_1(x)$ , are sparse in the Fourier bases and are given as follows:

$$\begin{aligned} a_0(x) &= b_0^{(0)} + \sum_{k=1}^3 b_{2k-1}^{(0)} \cos(2\pi kx) + \sum_{k=1}^3 b_{2k}^{(0)} \sin(2\pi kx); \\ a_1(x) &= b_0^{(1)} + \sum_{k=1}^3 b_{2k-1}^{(1)} \cos(2\pi kx) + \sum_{k=1}^3 b_{2k}^{(1)} \sin(2\pi kx); \\ \theta_1(x) &= 2T\pi x + b_0^{(2)} + \sum_{k=1}^3 b_{2k-1}^{(2)} \cos(2\pi kx) + \sum_{k=1}^3 b_{2k}^{(2)} \sin(2\pi kx), \end{aligned}$$

where

$$\{b_i^{(0)}\}_{i=0,1,\dots,6} = [0, 0, 1, 0.5, 0, 0.2, 0]; \tag{19}$$

$$\{b_i^{(1)}\}_{i=0,1,\dots,6} = [2, 1, 0, 0.5, 0, 0, 0.3]; \tag{20}$$

$$\{b_i^{(2)}\}_{i=0,1,\dots,6} = [2, 0, 1, 0.4, 0, 0, 0]. \tag{21}$$

In order to satisfy the scale-separation assumption (Assumption 1) between  $f_0(x)$  and  $f_1(x)$ , we choose the coefficient,  $T$ , which appears in the linear term of  $\theta_1(x)$ , to satisfy the following condition:

$$T \geq K_0 + K_1. \tag{22}$$

This is not required in general, but it is a sufficient condition to satisfy the scale separation condition. The inequality (22) implies that  $T \geq 6$ . In practice, we observe that even if we choose  $T$  as small as 5, we can still obtain a perfect recovery. If we choose a larger  $T$ , we can accommodate more Fourier modes in  $a_0$  and  $a_1$ . For a fixed number of Fourier modes in  $a_0$  and  $a_1$ , larger  $T$  gives a more accurate initial guess for  $a_0$ ,  $a_1$  and  $\theta_1$ . In Fig. 2, we illustrate the convergence of the Newton–Raphson iteration for two cases with  $T = 5$  and  $T = 10$ , respectively. In both cases, we can see that we recover the  $f$ ,  $a_0$ ,  $a_1$  and  $\theta_1$  up to the machine precision. The iterations converge exponentially fast, reaching the machine precision within just a few iterations. Moreover, the larger the value of  $T$  is, the faster is the convergence.

### 2.6.2. Periodic data with small amplitude envelope

In the previous two examples, we recover  $a_0(x)$ ,  $a_1(x)$  and  $\theta_1(x)$  almost perfectly up to the machine precision. In the next two examples, we consider a more difficult case when the envelope  $a_1(x)$  has small amplitude in some grid points. We keep the

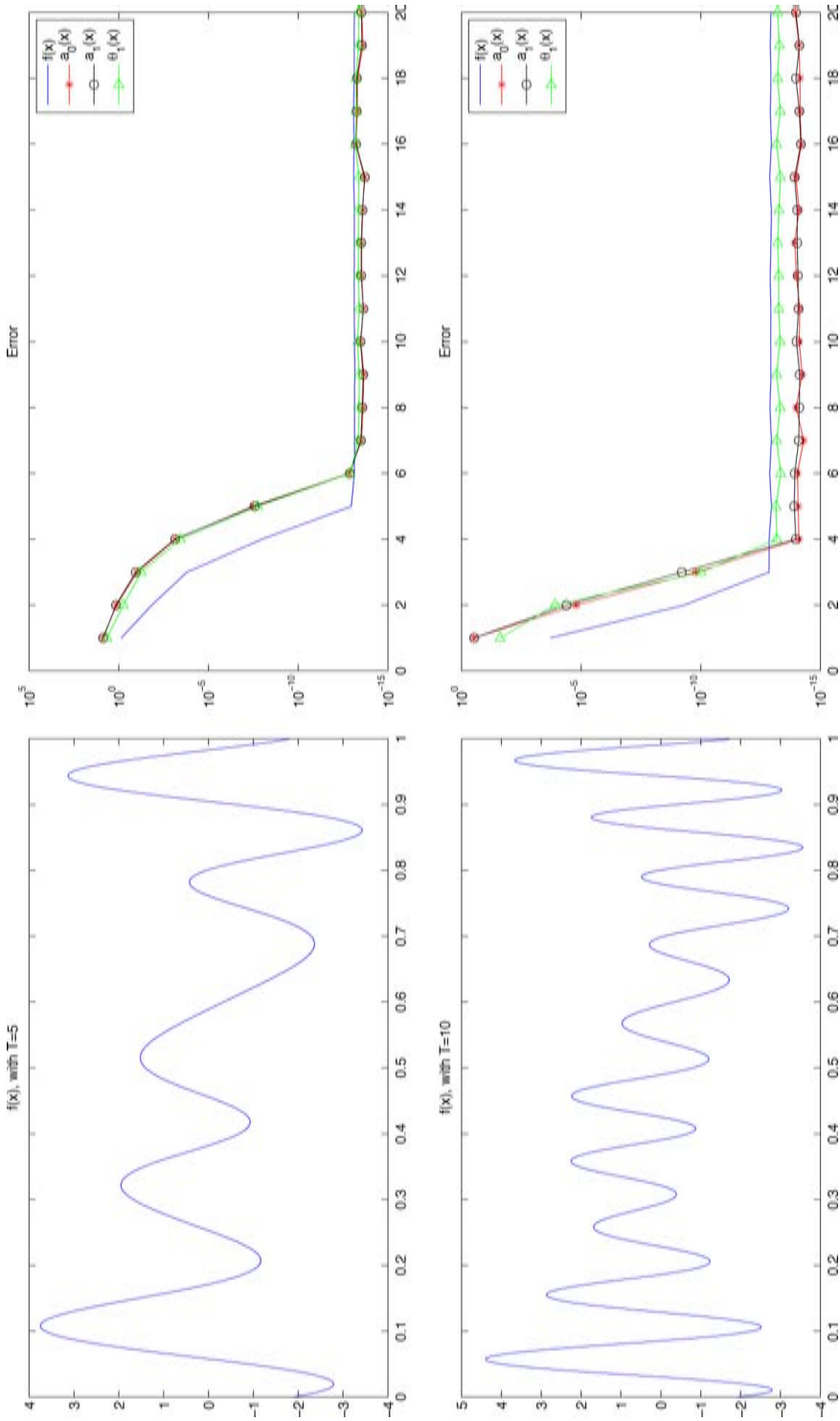


Fig. 2. The periodic signals with  $T = 5$  (upper) and  $T = 10$  (lower). The original signals are displayed in the left columns and the error figures are in the right. The horizontal axis of the error figures denotes the number of Newton-Raphson iterations. The vertical axis is the  $l^2$ -norm of the errors. Four curves (solid line, dash-star, dash-circle, and dash-triangle) in the right column represent the errors of the recovered  $f(x)$ ,  $a_0(x)$ ,  $a_1(x)$ , and  $\theta_1(x)$ , respectively.

coefficients of  $a_0(x)$  in (19) and  $\theta_1(x)$  in (21) the same and replace those of  $a_1(x)$  in (20) by

$$\{b_i^{(1)}\}_{i=0,1,\dots,2K_2} = [1.01, 1, 0, 0.5, 0, 0, 0.3]. \tag{23}$$

The envelope  $a_1(x)$  has two local minimum points at  $x_1 = 0.6051$  and  $x_2 = 0.2848$ , respectively, see the upper-left corner of Fig. 3. The minimum values of  $a_1$  at these two local minima are

$$a_1(0.6051) = 0.0689, \quad a_1(0.2848) = 0.1024.$$

One of the main difficulties in recovering accurately the IMFs is due the ill-conditioning of the resulting linear system when  $a_1$  has small amplitude at some grid points. Our numerical results show that as long as the values of  $\cos \theta_1(x)$  are not too small at the points at which  $a_1$  achieves its minimum, then we can still expect a perfect recovery of  $a_0$ ,  $a_1$  and  $\theta_1$ . For example, if we choose  $T = 6$ , we have

$$\cos \theta_1(0.6051) = 0.6704, \quad \cos \theta_1(0.2848) = 0.7089.$$

The Newton–Raphson iteration can still recover the Fourier coefficients of  $a_0$ ,  $a_1$  and  $\theta_1$  up to the machine precision (see the lower figures of Fig. 3). On the other hand, if we choose  $T = 10$ , then we have

$$\cos \theta_1(0.6051) = -0.2319, \quad \cos \theta_1(0.2848) = -0.0857.$$

The small amplitude of  $\cos \theta_1(0.2848)$  makes the resulting linear system ill-conditioned. Although the signal of  $T = 10$  has a better scale-separation property than that of  $T = 6$ , the Newton–Raphson iteration does not give a perfect recovery of the signal in the case of  $T = 10$ , see the upper-right figure of Fig. 3. The errors are the largest at the points in which  $a_1(x)$  achieves its minimum. Even in this very difficult test, we still retain a reasonably good recovery of its IMFs.

### 2.7. Computing instantaneous frequencies for nearly singular data

In this section, we will apply the method that we introduced earlier in Sec. 2.5 to compute instantaneous frequencies for nearly singular data. This method has the advantage of determining the instantaneous frequency only requiring a smoother second derivative of the envelope. Thus, it works very effectively for data with very small amplitude. We also compare the performance of our method with the EMD method that uses the Hilbert–Huang transform and renormalizes the amplitude of  $a_1$ .<sup>13,14</sup> For the purpose of comparison, we call our method the second-derivative method and the normalized EMD method the first-derivative method.

#### 2.7.1. Envelopes with small amplitude

We first test two examples with different envelope functions  $a_1$ . In both cases, we set the median to be zero. In the first example, we choose

$$f_1(x) = (0.1 + (x - 4)^2) \cos\{20\pi x + 0.5 \sin(2\pi x)\}. \tag{24}$$

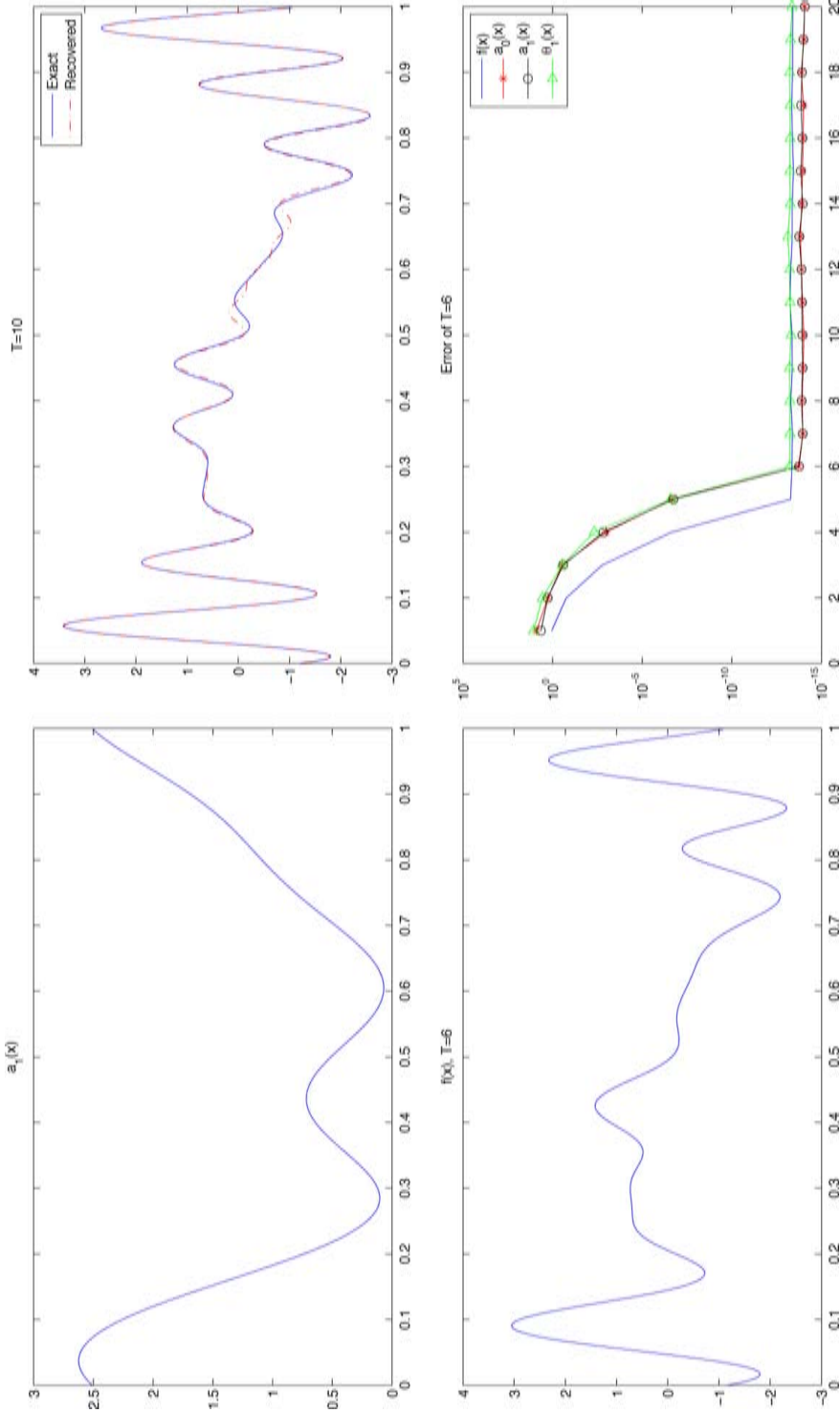


Fig. 3. The periodic data sets with  $T = 10$  (upper-right) and  $T = 6$  (lower). The envelope  $a_1(x)$  (upper-left) shows two small local minimum points. The recovery with  $T = 10$  (upper-right) has a larger error. The signal with  $T = 6$  is displayed (lower-left). The horizontal axis of the error figure (lower-right) denotes the number of iterations and the vertical axis is the  $l^2$ -norm of the errors of the recovered functions.



In the second example, we choose

$$f_1(x) = (0.2 + |(x - 4)(x - 1.5)(x - 6.5)|) \cos\{20\pi x + 0.5 \cos(2\pi x)\}. \quad (25)$$

Note that the absolute value sign in the second example (25) is to ensure that  $a_1$  is positive.

In the first signal, we observe that the error obtained by the second-derivative method is significantly smaller than that obtained by the first-derivative method. In the second example, the signal (25) has three nearly singular points in the envelope  $a_1(x)$ . When we use the first-derivative method, the error of the instantaneous frequency is magnified near the points at which  $a_1$  reaches its minimum, while the second-derivative method is relatively insensitive to the smallness of the envelope (see the right-lower corner of Fig. 4).

### 2.7.2. Instantaneous frequency with jump

Next, we study an interesting signal with a frequency jump in the time domain. For simplicity, we ignore the effect of the envelope function and the low-frequency IMF by setting

$$a_0(x) = 0, \quad a_1(x) = 1.$$

Then the signal becomes  $f(x) = \cos \theta(x)$ . We let  $\theta(0) = 0$  and choose the instantaneous frequency as

$$\theta'(x) = \begin{cases} 20\pi + \pi \cos 2\pi x, & \text{for } x \in (0, 4]; \\ 30\pi + \pi \cos 2\pi x, & \text{for } x \in (4, 8]. \end{cases}$$

The signal we choose has a frequency-jump at  $x = 4$  (see the lower-left of Fig. 5). We compare the relative error of the recovered instantaneous frequency to the exact one, using these two methods. As we can see from Fig. 5, the error curve of the second-derivative method (thick line) is an order of two magnitude smaller than that of the normalized first-derivative method throughout the domain, except at the location of the frequency jump (because of under-sampling). The improvement of the new method over the original EMD method (first-derivative method) is quite significant.

## 3. Multiscale Data Set

In this section, we generalize the method that we developed in the previous section to multiscale data which have the form of

$$f(x) = f_0(x) + f_1(x, x/\epsilon_1(x)) + f_2(x/\epsilon_1(x), x/\epsilon_2(x)), \quad (26)$$

where  $\epsilon_1(x)$  and  $\epsilon_2(x)$  satisfy:

$$0 < \epsilon_2(x) \ll \epsilon_1(x) \ll 1.$$

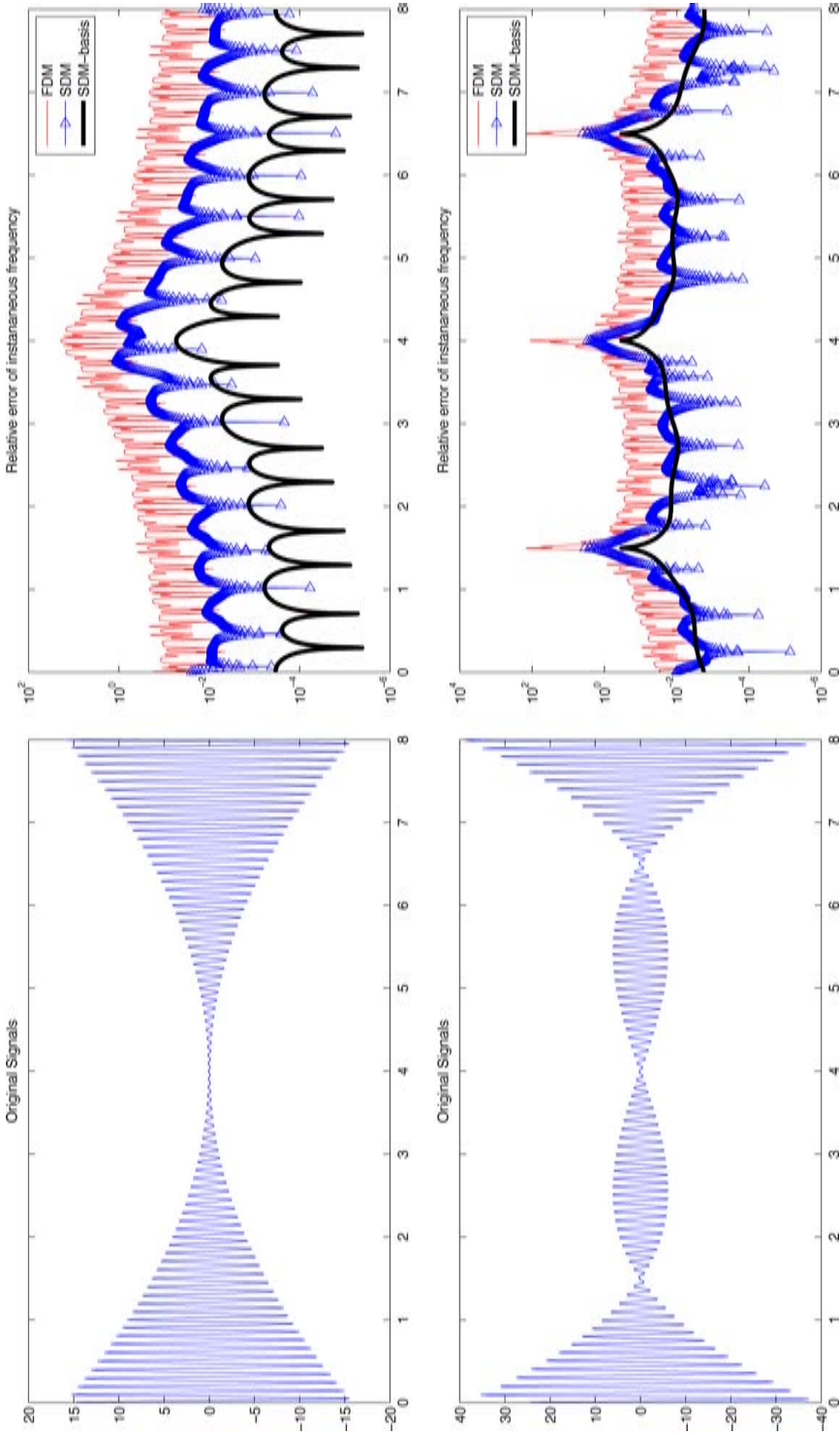


Fig. 4. Figures on the left are original signals of first example (24) (above) and second example (25) (below). The right column figures are relative errors of the first-derivative method (solid line, FDM, Eq. (16)), the second-derivative method (dash-triangle, SDM), and the second-derivative method with finite Fourier bases fit (solid wide line, SDM-basis), respectively.

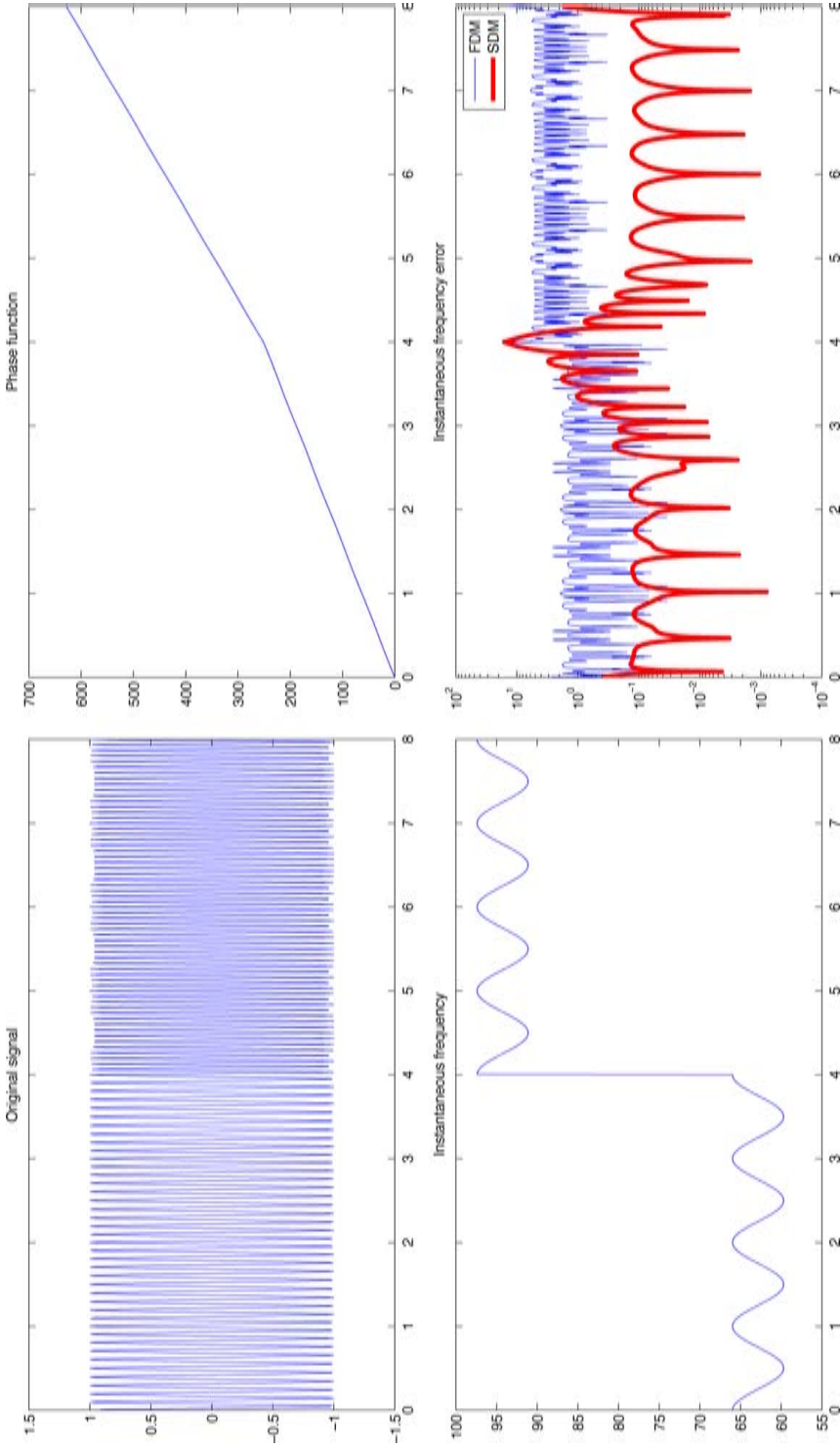


Fig. 5. Signal (upper-left corner) with a jump frequency at  $x = 4$ . The phase function  $\theta_1(x)$  and the instantaneous frequency  $\theta'_1(x)$  are shown at the upper-right and the lower-left corner. The error figure (lower-right) gives a comparison of the results obtained using the second-derivative method (solid wide line, SDM) with those obtained using the first-derivative method (solid line, FDM, Eq. (16)).

We further assume that  $f_1(x, y)$  and  $f_2(x, y)$  are periodic in  $y$  with period 1. For this class of multiscale data, we express its high frequency IMF as

$$f_i(x, x/\epsilon_1) = a_i(x) \cos\{2\pi x/\epsilon_i(x)\}, \quad i = 1, 2. \tag{27}$$

We assume that the multiscale data satisfy the same scale-separation property as stated in Assumption 1 for periodic signals. However, we do not assume that the multiscale data are periodic with respect to  $x$ .

We will use the same second order method which was introduced in Sec. 2.4 to approximate its local median, denoted as  $r_{n-1}(x)$ , and its envelope function  $a_n(x)$ . Based on the second order approximation of  $a_n$ , we can extract the Fourier coefficients of the envelope by using a finite number of Fourier modes and the least square method.

After we obtain an approximation to  $r_{n-1}(x)$ , we can approximate  $f_n$  by using  $f_n(x) = f(x) - r_{n-1}(x)$ . By our assumption,  $a_n(x)$  and  $\epsilon_n(x)$  are smoother than  $\cos(2\pi x/\epsilon_n)$ . We denote the phase function

$$\theta_n(x) = \frac{2\pi x}{\epsilon_n(x)}.$$

The instantaneous frequency  $\theta'_n(x)$  can be generated using the methods we described in Sec. 2.5. On the other hand, the instantaneous frequency is given by

$$\theta'_n(x) = \frac{2\pi}{\epsilon_n(x)} - \frac{2\pi x \epsilon'_n(x)}{\epsilon_n^2(x)}. \tag{28}$$

Once we have the approximated  $\theta'_1(x)$ , we have an ordinary differential equation

$$\frac{d\epsilon_n(x)}{dx} = \frac{-\theta'_n(x)\epsilon_n^2(x) + 2\pi\epsilon_n(x)}{2\pi x}. \tag{29}$$

The additional work here is that we need to solve an ordinary differential equation to obtain an approximation for  $\epsilon_n(x)$ . Notice that when  $x = 0$  the IMF (27) reaches its local maximum point. Without loss of generality, we denote local maximum point  $x_j = 0$  as an initial time and apply a high order ODE solver such as the classical fourth order Runge–Kutta method (RK4) to solve the initial value problem.

By our assumption, the length scale function  $\epsilon_n(x)$  is a smooth function which has a well-defined first order derivative. This implies that the numerator in the quotient of (29) must be zero at  $x = 0$ , i.e.

$$\epsilon_n(0) = 2\pi/\theta'_n(0). \tag{30}$$

In addition, the right-hand side of the ODE (29) has weak singularity at  $x = 0$ . In order to remove it, we apply the L'Hospital rule to the quotient of Eq. (29) to derive

$$\lim_{x \rightarrow 0} (\theta'_n(x)\epsilon_n^2(x))' = 0.$$

Hence,  $\theta'_n(x)\epsilon_n^2(x)$  is approximately a constant near  $x = 0$ . Denote  $x_1$  as the nearest point to  $x = 0$ . Then, we have

$$\theta'_n(x_1)\epsilon_n^2(x_1) \approx \theta'_n(0)\epsilon_n^2(0) = \frac{4\pi^2}{\theta'_n(0)}.$$

Thus, we obtain a new initial condition associated with  $x = x_1$ :

$$\epsilon_n(x_1) = 2\pi\sqrt{\frac{1}{\theta'_n(0)\theta'_n(x_1)}}.$$

We will choose  $\epsilon_n(x)$  to be positive. In our implementation, we solve the ODE, in both directions, for  $\epsilon_n(x)$  over an interval between a discrete maximum and minimum point. This is to prevent the error propagation from one local period to another. Within each local interval, the initial point  $x_{j+1}$  of ODE (29) is chosen to be the closest point to the discrete maximum.

The  $\epsilon_n(x)$  function we obtain from solving the ODE over each interval is a piecewise smooth curve defined on each interval. Because of the periodicity of the cosine function, we need to generate its global approximation in a proper way. Specifically, once the discrete point crosses one local maximum, the phase function,  $(2\pi x)/\epsilon_n(x)$ , has to increase by  $2\pi$ . After we obtain the global approximation of  $\epsilon_n(x)$ , we will apply a low pass filtering to smooth out the high frequency error by using the least square method and imposing the sparse representation of  $\epsilon_n(x)$  in its Fourier bases, as we did for the envelope functions.

For a general multi-scale signal consisting of multiple IMFs, we can apply the above procedure recursively to find all its IMFs. As in the original EMD method, we will first use the second order method to find its upper and lower envelopes. From these two envelopes, we can determine the envelope function,  $a_n(x)$ . The length-scale function,  $\epsilon_n(x)$ , will be determined by solving the first order ODE that we described above. This gives the first and the highest frequency IMF,

$$f_n(x) = a_n(x) \cos(2\pi x/\epsilon_n(x)).$$

Now we can apply the same procedure to  $f(x) - f_n(x)$  by treating  $f(x) - f_n(x)$  as a new signal. This will generate the next envelope function  $a_{n-1}(x)$  and the new length-scale function  $\epsilon_{n-1}(x)$ . This procedure can continue until  $f_0(x)$  is a monotone function. Since the above method involves solving an ODE to determine the length scale function, we call this method *the ODE-based EMD method*.

### 3.1. Numerical results for the ODE-based EMD method

In this section, we will present some numerical results to demonstrate the robustness of our ODE-based EMD method. The implementation of the ODE-based EMD method consists of two steps. The first step is to use the second order method to approximate the upper and lower envelopes of a given signal. Once we obtain these two envelopes, we can extract the local median of the signal,  $r_{n-1}(x)$ , which

contains the lower frequency information and its envelope function,  $a_n(x)$ . In the second step, we define the approximate IMF,  $\tilde{f}_n(x) = f(x) - r_{n-1}(x)$ , and solve for the ODE based on this approximate IMF to obtain the length-scale function,  $\epsilon_n(x)$ . After we solve for  $\epsilon_n(x)$ , we obtain the first and the highest frequency IMF,  $f_n(x)$ , as

$$f_n(x) = a_n(x) \cos(2\pi x / \epsilon_n(x)).$$

Then we repeat the same procedure to the lower frequency signal,  $f(x) - f_n(x)$ , to extract the remaining IMFs,  $f_{n-1}, \dots, f_0(x)$ .

We test a multiscale signal consisting of three IMFs. We also compare the performance of the method that uses the local median with the method that uses the local mean in our approximation of the envelope function. Our numerical results show that the method that uses the local median gives a superior performance than the method that uses the local mean.

The multiscale signal that we consider is defined on the interval between 0.1 and 2.1. The grid resolution is given by  $\Delta x = 1/2560$ . The signal is of the form:

$$\begin{aligned} f(x) &= f_0(x) + f_1(x) + f_2(x) \\ &= a_0(x) + a_1(x) \cos \frac{2\pi x}{\epsilon_1(x)} + a_2(x) \cos \frac{2\pi x}{\epsilon_2(x)}, \end{aligned} \quad (31)$$

where the envelopes and length-scale functions are given by

$$a_0(x) = \sin(2\pi x), \quad (32)$$

$$a_1(x) = 0.2(1 + 0.2 \cos(2\pi x)), \quad (33)$$

$$\epsilon_1(x) = \frac{1}{16}(1 - 0.05 \cos(2\pi x)), \quad (34)$$

$$a_2(x) = 0.04(1 + 0.2 \cos(2\pi x))^2, \quad (35)$$

$$\epsilon_2(x) = \frac{1}{256}(1 - 0.05 \cos(2\pi x))^2. \quad (36)$$

In Fig. 6, we plot the three components,  $f_0, f_1, f_2$ , and the signal  $f(x)$ . In Fig. 7, we plot the relative errors of the recovered IMFs for the multiscale signal defined above. We observe that the relative errors for the envelope function of the highest frequency IMF,  $f_2$ , are below 1.4%. The relative errors for the length-scale function are even smaller, below 0.004%. The relative errors for the second IMF are larger than the corresponding errors for the first IMF, but are still reasonably small. This behavior is to be expected since the major source of errors depend on the length-scale  $\epsilon_i$ . This can be traced back to the approximation errors in the two steps of our method. In the first step, the grid space between two local extrema is roughly  $\epsilon_i$ . Thus the interpolation step in approximating the envelope functions produces errors that depend on  $\epsilon_i$  directly. This interpolation errors will propagate into the second step of our method since we use  $f_n(x) = f(x) - r_{n-1}(x)$  as the starting point in deriving our ODE for the length-scale function,  $\epsilon_n(x)$ .

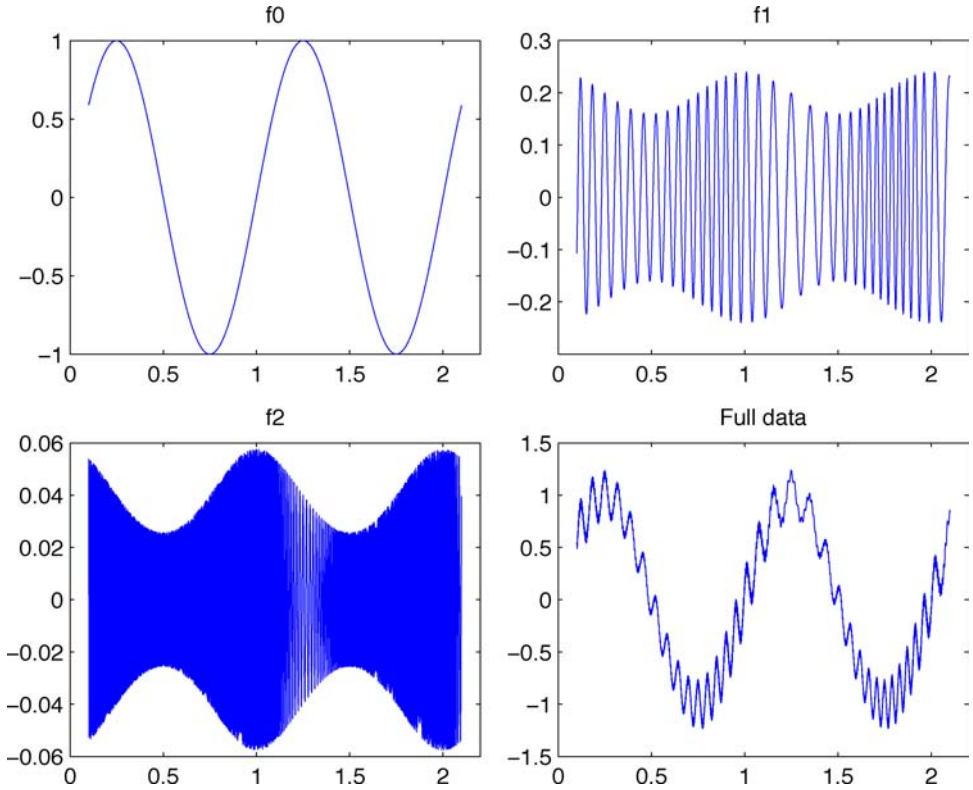


Fig. 6. The multi-scale data (lower right figure) is the sum of  $f_0$ ,  $f_1$  and  $f_2$ .

In the previous example, we observe that the method that uses the local median and the method that uses the local mean give a comparable performance. It is mainly due to the smoothness property of the instantaneous frequency in this example. In order to show how the regularity of the instantaneous frequency may affect the performance of these two methods, we test another multiscale signal by doubling the amplitude of  $\epsilon_2(x)$ .

$$\epsilon_2(x) = \frac{1}{128}(1 - 0.05 \cos(2\pi x))^2, \tag{37}$$

by keeping all other parameters the same as in Eqs. (32)–(35). In this case, the distance between two closest extrema increases. It is not hard to show that in order for the local mean method to give a good approximation to the envelope function, the function,  $\epsilon_2^2/(\epsilon_2 - x\epsilon_2')$ , must be close to a constant over one discrete period between two extrema. By increasing the period between two extrema, this quantity has a larger variation within one period. As a result, the method that uses the local mean gives a worse approximation to the envelope function. This error will propagate to the length-scale function, resulting in a larger error in our approximation of the IMF. In Fig. 8, we plot the errors for both the envelope

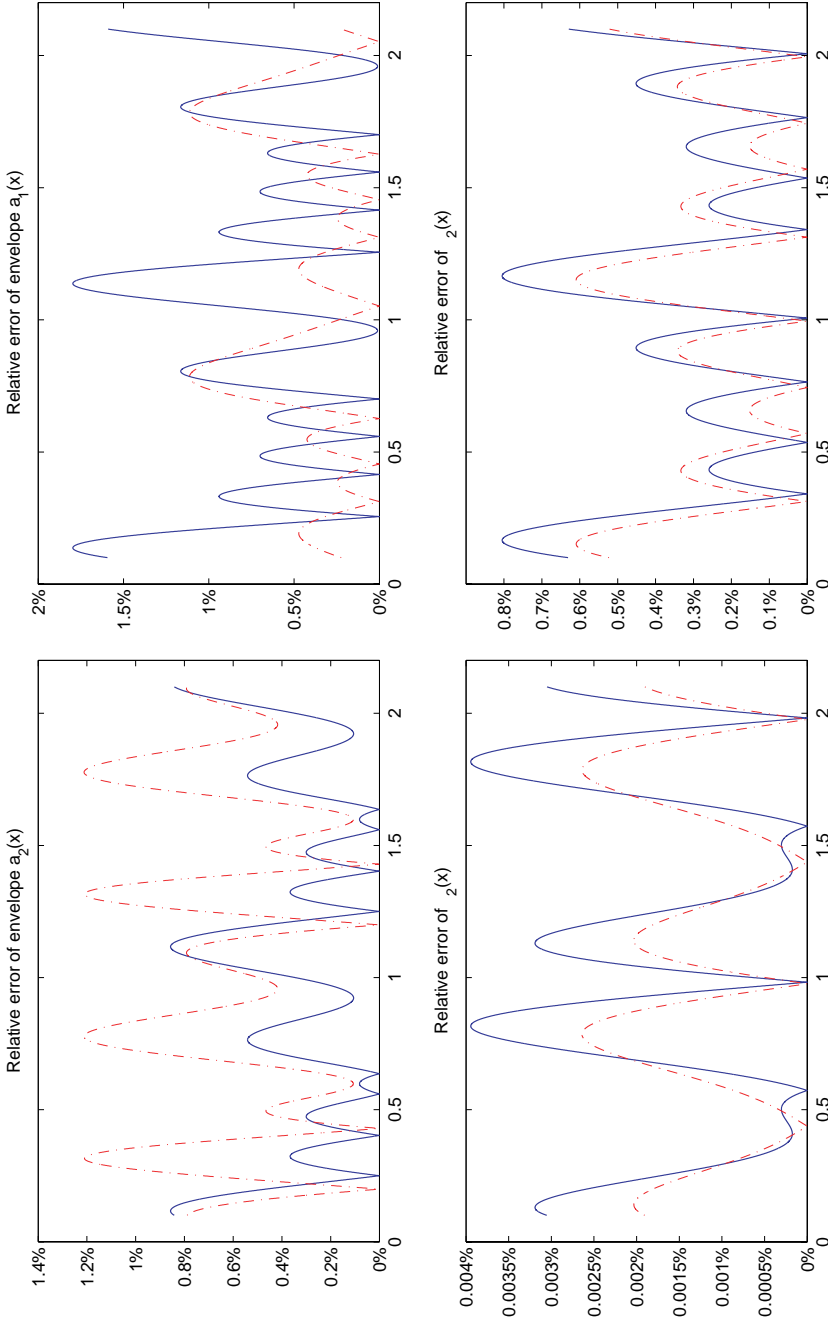


Fig. 7. Relative error figures of the recovered functions. The upper two subplots are the relative errors of the envelope functions,  $a(x)$ , corresponding to different IMFs. The lower two subplots are the relative errors of the length-scale functions,  $\epsilon(x)$ , corresponding to different IMFs. The solid curves correspond to the computations that use the local median to determine the envelope functions, while the dashed curves correspond to the computations that use the local mean. The computations are done for the first example with  $\epsilon_2$  defined by (36).



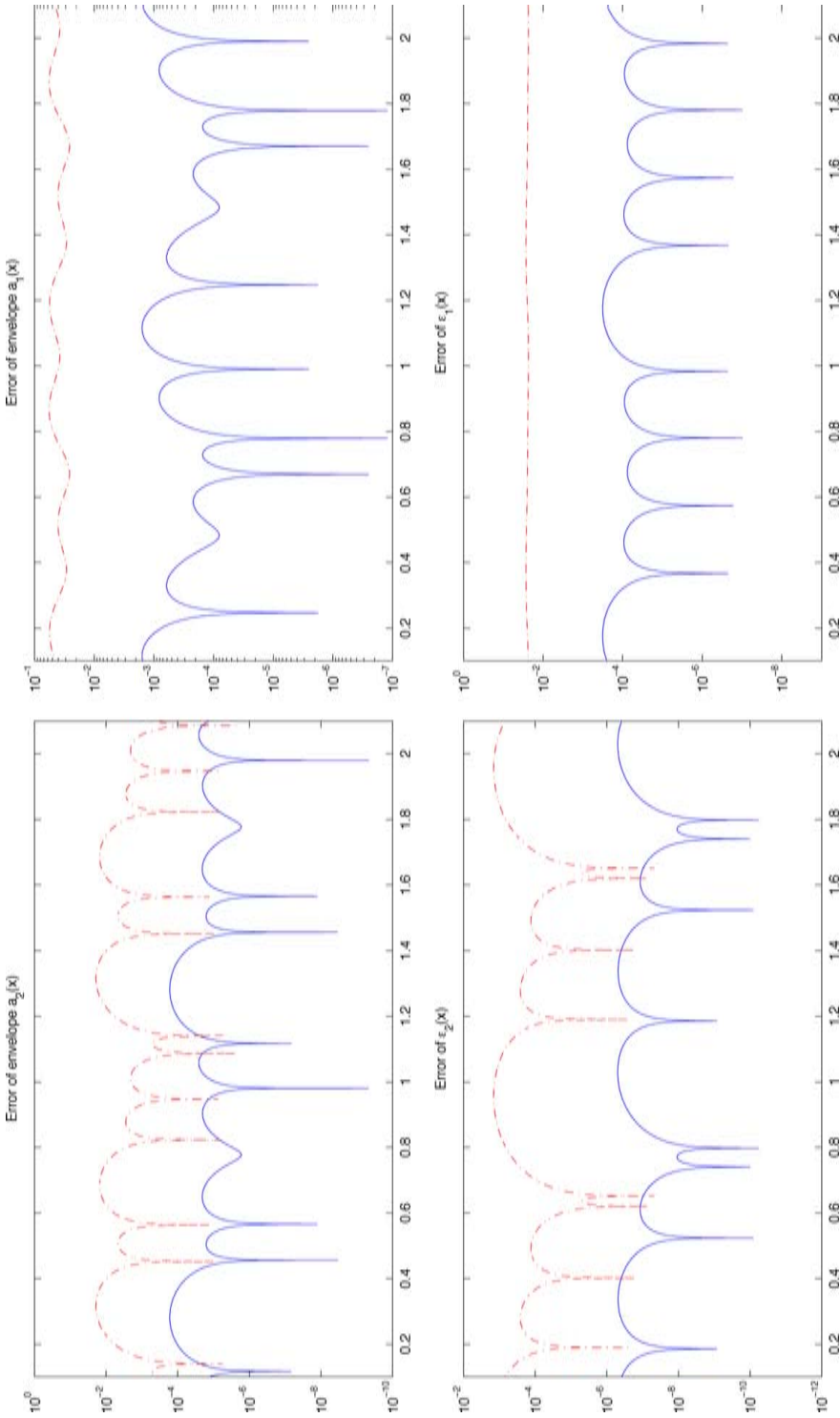


Fig. 8. Error curves of the envelope functions  $a(x)$  (upper row) and the length-scale functions  $\epsilon(x)$  (lower row). The left two subplots are the error curves of the high-frequency IMF  $f_2(x)$ , and the right two subplots are the error curves of the low-frequency IMF  $f_1(x)$ . We compare the performance of the method that uses the local mean (dashed) with the method that uses the local median (solid). The computations are done for the second example with  $\epsilon_2$  defined by (37).

functions and the length-scale functions using these two approaches. As we can see, the errors produced by the method that uses the local median are two orders of magnitude smaller than those produced by the method that uses the local mean. This is another strong evidence that the local median method gives a superior performance than the local mean method.

#### 4. LSSF Denoising and Decomposition

In this section, we introduce a least-square-spline-fit (LSSF) method to denoise or decompose a noisy signal. We compare the performance of the method with EEMD (or EMD for clean signal) for the data that we have tested in the previous sections. The LSSF denoising method is a variant of the methods that we described earlier, with only small modifications near the boundaries. Below we will describe the LSSF denoising method in some detail.

##### 4.1. LSSF denoising

To illustrate the main idea of the LSSF denoising method, we assume that a clean signal  $f(x)$  is polluted by a Gaussian white noise with mean zero and variance  $\sigma^2$ , i.e.

$$\tilde{f} = f + N(0, \sigma^2). \quad (38)$$

Since the Gaussian noise has very high frequency oscillations, we would like to first denoise the signal. The main idea of the LSSF denoising is similar to that of finding the local median in Sec. 2.1.2. We treat  $f(x)$  as the median function of a noisy signal,  $\tilde{f}(x)$ , and apply the LSSF method to denoise the signal.

In comparison, the EEMD method first adds a white noise to a signal, applies EMD to each realization, and average over all ensembles. Specifically, EEMD generate  $M$  samples of noisy signals by adding a Gaussian noise  $N(0, \tilde{\sigma}^2)$  into the original signal with a given noise level  $\tilde{\sigma}$ . Then one applies EMD to each sample, and average these  $M$  sets of generated IMFs. The median generated by such procedure is treated as the denoised signal  $f(x)$ .

In the LSSF denoising method, we divide the domain  $[a, b]$  into a finite number of sub-intervals, i.e.

$$a = x_0 < x_1 < \cdots < x_m = b.$$

The size of these sub-intervals is determined by the smoothness of  $f(x)$ . Furthermore, we denote  $\mathbb{K}$  as a class of functions, whose elements  $g(x)$  satisfy the following conditions:

- (1)  $g(x) \in C^2(a, b)$ , i.e. it is second order continuously differentiable in the domain  $[a, b]$ .
- (2)  $g(x)$  is a third order polynomial in each interval  $[x_{k-1}, x_k]$ , for  $k = 1, \dots, m$ .

Then the LSSF method is to approximate  $f(x)$  by  $\bar{f}(x)$ :

$$\bar{f}(x) = \{g(x) \in \mathbb{K} \mid \min \|\bar{f}(x) - g(x)\|_2\}. \quad (39)$$

Note that the dimension of the class  $\mathbb{K}$  is  $m+3$  and one possible basis decomposition of  $\mathbb{K}$  is to choose the bases as

$$1, x, x^2, \left( \begin{cases} (x - x_k)^3, & \text{for } x > x_k; \\ 0, & \text{for } x < x_k, \end{cases} \right)_{k=0,1,\dots,m-1}. \quad (40)$$

Then we use all data points to approximate the basis coefficients of  $\bar{f}_0(x)$  by minimizing the energy norm (39).

#### 4.2. The end-point effect of the LSSF decomposition

In this section, we will address an important issue, i.e. the end-point effect in our LSSF decomposition. Since the frequency of a Gaussian noise is quite high, thus the distance between the boundary of a signal and its nearest extremum is very small. Hence the chance of an over-shoot in our piecewise cubic polynomial approximation near boundaries is small. However, in the LSSF decomposition of IMFs that have median to low frequencies, the chance of over-shoot is large due to the large increasing distance between the boundary and its nearest extremum.

In order to alleviate this difficulty, we set up some reasonable criteria in our polynomial approximation of the IMFs near the end-point of a signal. Here we adopt the idea of linear extensions which has been introduced in Ref. 10. Denote  $x_1$  and  $x_2$  the first and second maximum points, and  $x_0$  the left boundary point. Then the boundary value of the upper envelope function  $\bar{f}(x_0)$  is the linear extension of the values of  $x_1$  and  $x_2$ , i.e.

$$\bar{f}(x_0) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x_0 - x_1).$$

The same applies to the lower envelope function  $\underline{f}(x_0)$ , which comes from the extension of the first and second minima. The way we treat the right end-point of  $\bar{f}$  and  $\underline{f}$  are analogous. Then the spline coefficients of both  $a_0(x)$  and  $a_1(x)$  are specified by the least square method using the identities

$$\begin{aligned} \bar{f}(x) &= a_0(x) + a_1(x); \\ \underline{f}(x) &= a_0(x) - a_1(x). \end{aligned}$$

#### 4.3. Denoising simulations

In the simulations via EEMD, we choose  $m = 800$  samples of random realizations. The numerical output of EEMD is a matrix of several columns. The first column is the original signal, the last one is the median and the rest are different IMFs. As the noise mainly stays in the high frequency IMFs, we compare the sum of the low frequency IMFs (i.e. EEMD denoised signal) to the actually one,  $f(x)$ .

However, it is not clear which column is the best cut-off column that separates the low frequency IMFs from the high frequency IMFs. In our simulations, we test different cut-off columns and choose the one whose low-frequency sum is closest to the true median,  $f(x)$ .

In the LSSF method, we need to specify the spline sub-intervals. In general, the sub-intervals are determined by the smoothness of the original noise free signal,  $f(x)$ . In the algorithm, we define  $s$  periods of noisy signals as a new spline interval. Here one period is defined as the distance from one maximum (or minimum) to the next maximum (or minimum). Typically, one needs to use several periods to average out the Gaussian noise. The specific number of periods would depend on the noise level. In our simulations, we determine the optimal number of periods empirically by testing different values of  $s$  and choosing the one that gives the most stable median  $f(x)$ .

In addition, we use the Newton–Raphson-based method to decompose different IMFs and compare the recovered functions with those obtained by EMD.

We choose the periodic signal we used before in Eqs. (19)–(21)

$$\begin{aligned}\theta(x) &= \sin(2\pi x) + 0.4 \cos(4\pi x) + 20\pi x + 2; \\ a_0(x) &= \sin(2\pi x) + 0.5 \cos(4\pi x) + 0.2 \cos(6\pi x); \\ a_1(x) &= 2 + \cos(2\pi x) + 0.5 \cos(4\pi x) + 0.3 \sin(6\pi x); \\ \tilde{f}(x) &= a_0(x) + a_1(x) \cos \theta(x) + N(0, \sigma^2),\end{aligned}\tag{41}$$

over the domain  $[0, 1)$  with grid size  $h = 1/1024$ . In our denoising via LSSF, we choose the best period number  $s = 4$  for noise-level 0.1 and  $s = 5$  for noise-level 0.2. For the comparison, we choose the sum of fourth, fifth and sixth columns of EEMD matrix as the IMF  $f_1(x) = a_1 \cos \theta_1(x)$  for noise-level  $\sigma = 0.1$ , and the sum of fifth and sixth columns as the IMF for  $\sigma = 0.2$ . The sum of the columns after the sixth one is treated as the median function  $a_0(x)$ .

Note that although the signal we used is periodic, we do not use this special property when we denoise it via LSSF. They are treated as a general time signal.

From the simulations, we can see the errors of the denoised signals, via either EEMD or LSSF, are at the same level. One advantage of LSSF over EEMD is the computational cost. In our simulations, EEMD takes a long time to generate 800 samples of white noise in the averaging step with 1024 discrete points. If one increases the size of the time sequence or increases the number of samples, the computational cost would increase linearly for EEMD. In comparison, the LSSF denoising method runs quite fast and effectively without generating any additional noise.

#### 4.4. *Simulations of LSSF decomposition*

In this section, we compare the multiscale-signal decompositions of two methods, LSSF and EMD. The data we use in the simulations are the same as those given in Eqs. (32)–(36).

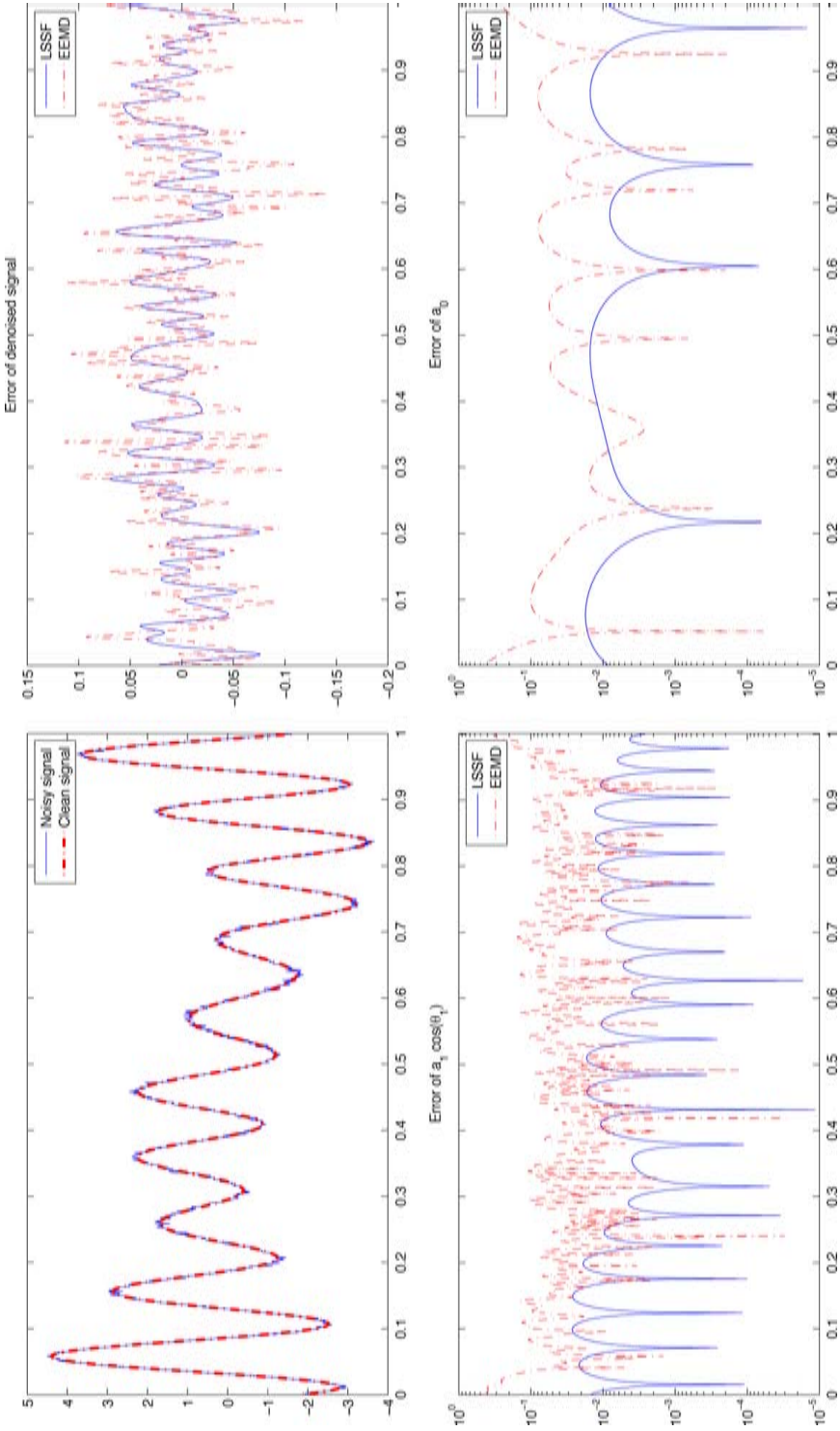


Fig. 9. Upper-left, clean signal (dashed) and noisy signal of noise-level  $\sigma = 0.1$  (solid curve). Upper-right, absolute errors of denoised signal via the LSSF (solid curve) and EEMD (dashed). Lower-left, absolute errors of recovered IMF  $a_1 \cos \theta_1(x)$  via LSSF-Newton-Raphson-based method (solid curve) and EEMD (dashed). Lower-right, absolute errors of the recovered median function  $a_0(x)$  via the LSSF-Newton-Raphson-based method (solid curve) and EEMD (dashed).

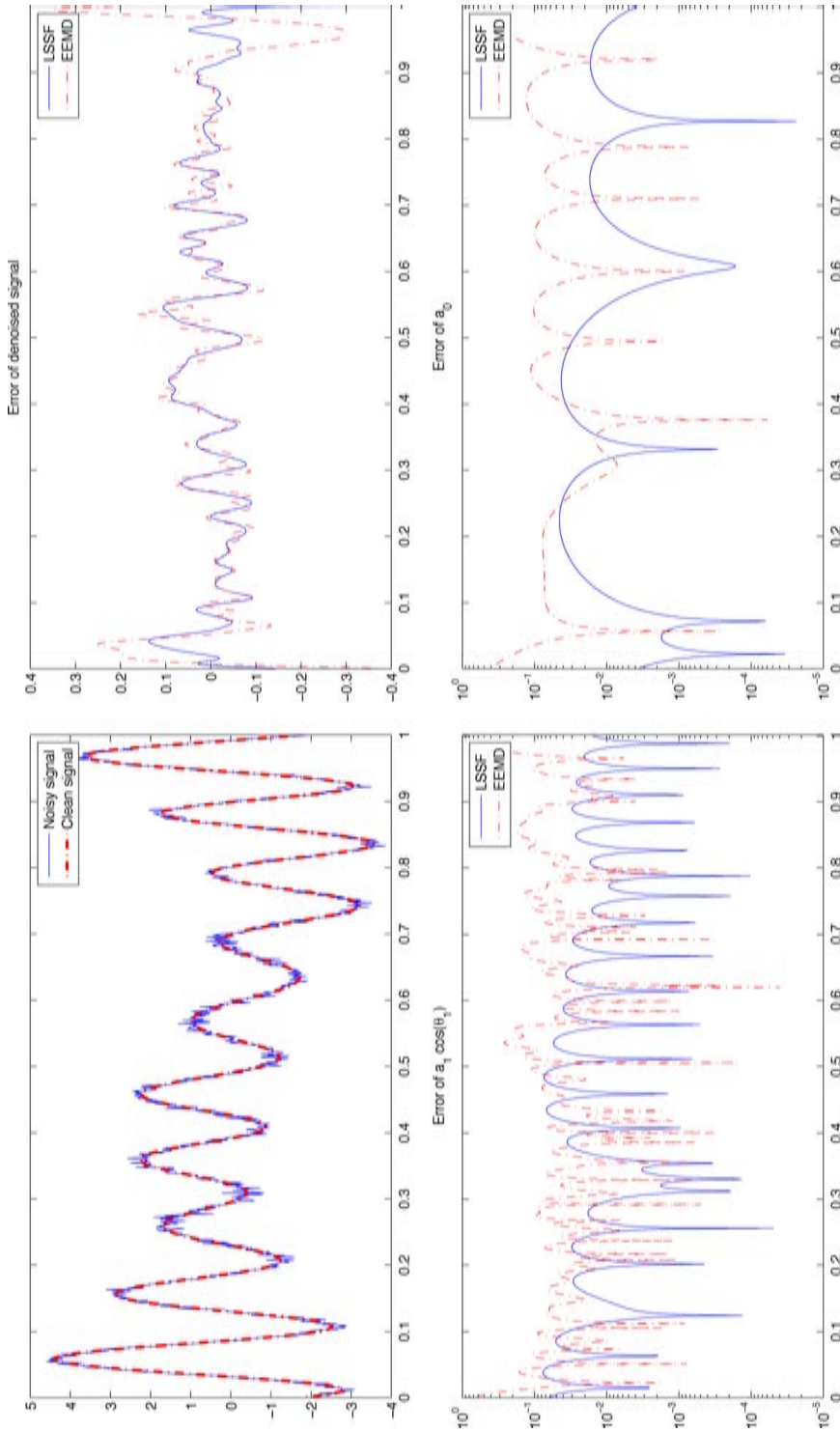


Fig. 10. Upper-left, clean signal (dashed) and noisy signal of noise-level  $\sigma = 0.2$  (solid curve). Upper-right, absolute errors of denoised signal via LSSF (solid curve) and EEMD (dashed). Lower-left, absolute errors of recovered IMF  $a_1 \cos \theta_1(x)$  via the LSSF-Newton-Raphson-based method (solid curve) and EEMD (dashed). Lower-right, absolute errors of the recovered mean function  $a_0(x)$  via the LSSF-Newton-Raphson-based method (solid curve) and EEMD (dashed).

The key of LSSF is to derive the spline coefficients of the local median  $a_0(x)$  and high-frequency envelope  $a_1(x)$  by the information of local extrema. Because of the scale-separating property, the envelope function  $a_1(x)$  is smoother than the data before decomposition, and hence we define a larger period number  $s = 5$  for the spline intervals. Meanwhile, the local median  $a_0(x)$  may still consist of several IMFs and the median function. In order to preserve the high frequency information in  $a_0(x)$ , we choose the period number to be  $s = 1$ . The errors of the recovered IMFs and median function are illustrated in Fig. 11, in which we compare LSSF with EMD. As we can see, the errors produced by the two methods are comparable with the error curves of EMD stay slightly below those of the LSSF method.

## 5. Conclusion

In this paper, we have introduced a variant of the EMD method to decompose a sparse periodic signal and a multiscale signal into its IMFs. In principle, one can separate the high frequency IMF by using either the local median method or the local mean method. One of the main observations of this paper is that the local median method gives a more superior performance than the local mean method. The difference in their performance is especially pronounced when the instantaneous frequency is not smoother than the signal itself. We provided some preliminary analytical and numerical evidence to support this.

Another important observation of this paper is that the IMFs of the given data need to satisfy certain scale-separation property. This scale-separation property is related to the local orthogonality of the IMFs. This condition is required for the sake of both accuracy and the physical uniqueness. Violation of such scale-separation property would require additional selection criteria to define a unique and accurate EMD decomposition. This is an important question, and is beyond the scope of this paper. This issue will be addressed in a systematic way in an upcoming paper<sup>9</sup> by using a Total-Variation-Diminishing (TVD) based optimization approach and borrowing some ideas from compressed sensing.<sup>2-5</sup>

For the class of periodic signals with a finite number of Fourier modes in both the envelopes and the instantaneous frequencies, we introduced a Newton–Raphson-based EMD method. A less accurate but more robust method was proposed to produce a good initial guess for the Newton–Raphson iterations. The Newton–Raphson-based EMD method has the advantage of converging rapidly and producing a highly accurate recovery of the IMFs (often up to the machine precision) for data satisfying our scale-separation assumption. We also introduced a method that uses the second-derivative of the signal (Sec. 2.5) to determine the instantaneous frequencies in the EMD decomposition instead of using the Hilbert transform. This method has the advantage of not requiring a good approximation of the envelope function. This method is especially useful when it is difficult to approximate the envelope function accurately (e.g. it is nearly singular or has small amplitude).

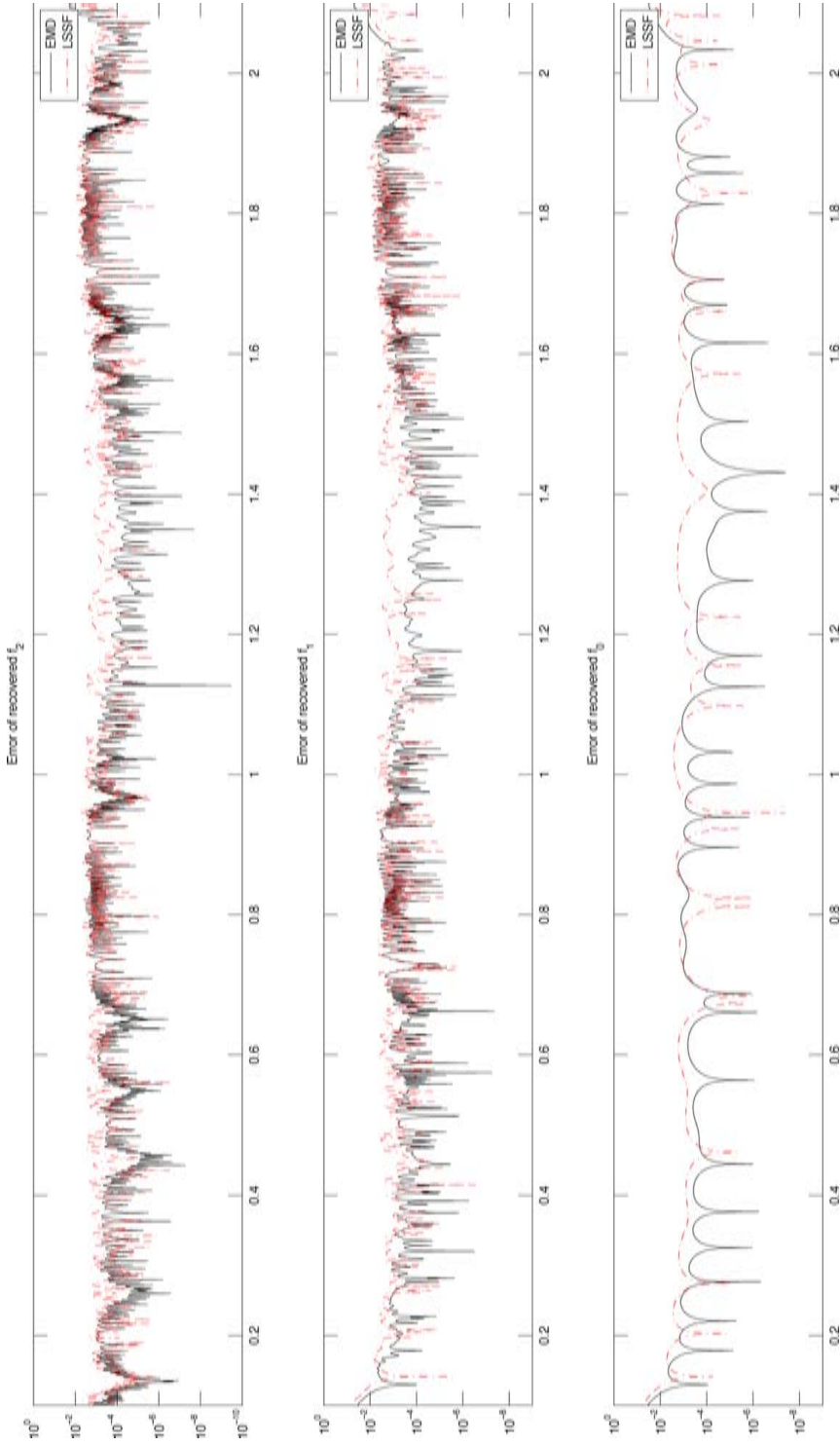


Fig. 11. Decomposition of multiscale signal (32–36) via EMD (solid curve) and LSSF (dashed). Relative error figures of recovered small-scale IMFs  $f_1$ ,  $f_2$  and mean  $f_0$  (31) are given from top to bottom.



For multiscale data, we developed an ODE-based EMD method to determine the length-scale function in the EMD decomposition without the prior knowledge of the envelope. As for the periodic data, this ODE-based EMD method works very effectively when the envelope is nearly singular or has small amplitude. Numerical simulations confirm the superior performance of the proposed method.

For noisy signals, we proposed an LSSF-based denoising algorithm. The computational cost of the LSSF denoising algorithm is significantly smaller than that of EEMD, especially when the data size is huge. Our preliminary numerical experiments indicate that for the special class of multiscale data considered here, the two methods give a comparable performance.

There remain some limitations in the methods proposed in this paper. Right now, we use a Fourier interpolation or a cubic spline to interpolate the envelope function. In the future, we will explore other alternative approaches to better interpolate the envelope in the whole domain. Another open question is the recovery of the instantaneous frequency for data with small amplitude in their high frequency oscillations. The second-derivative method or the ODE-based method we introduced here can alleviate this difficulty to some extent. However, this problem remains when the scale-separation property is not satisfied. We still need to develop a robust randomized sampling technique to extract data with noise or incomplete data. This is a topic of our future research.

## Acknowledgments

This work was in part supported by the NSF Grant DMS-0713670. This research was initiated during Prof. Hou's visit to the National Central University as a Visiting Chair Professor in December of 2008. Prof. Hou would like to thank the National Central University for their support and hospitality during his visit. We would also like to thank Dr. Zuoqiang Shi for many helpful discussions on the research presented in this paper.

## References

1. A. Bensoussan, J. L. Lions and G. Papanicolaou, Asymptotic analysis for periodic structures, Volume 5 of *Studies in Mathematics and Its Applications* (North-Holland Publ., New York, 1978).
2. A. M. Bruckstein, D. L. Donoho and M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, *SIAM Rev.* **51** (2009) 34–81.
3. E. Candes, J. Romberg and T. Tao, Robust uncertainty principles: Exact signal recovery from highly incomplete frequency information, *IEEE Trans. Inform. Theory* **52** (2006) 489–509.
4. E. Candes, J. Romberg and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, *Comm. Pure Appl. Math.* **59** (2006) 1207–1223.
5. D. L. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory* **52** (2006) 1289–1306.
6. Y. Efendiev and T. Y. Hou, *Multiscale Finite Element Methods: Theory and Applications* (Springer, New York, 2009).

7. P. Flandrin, G. Rilling and P. Goncalves, Empirical mode decomposition as a filter-bank, *IEEE Signal Process. Lett.* **11** (2004) 112–114.
8. T. Y. Hou and X. H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, *J. Comput. Phys.* **134** (1997) 169–189.
9. T. Y. Hou and Z. Shi, *An Adaptive TVD-Based Optimization Approach for the Analysis of Multiscale Nonlinear Signals*, in preparation (2009).
10. N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N. Yen, C. Tung and H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proc. Roy. Soc. London A* **454** (1998) 903–993.
11. N. Huang, Z. Shen and S. Long, A new view of nonlinear water waves: The Hilbert spectrum, *Ann. Rev. Fluid Mech.* **31** (1999) 417–457.
12. N. Huang, M. Wu, S. Long, S. Shen, W. Qu, P. Gloersen and K. Fan, A confidence limit for the empirical mode decomposition and Hilbert spectral analysis, *Proc. Roy. Soc. London A* **459** (2003) 2317–2345.
13. N. Huang and Z. Wu, A review on Hilbert–Huang transform: Method and its applications to geophysical studies, *Rev. Geophys.* **46** (RG2006) doi:10.1029/2007RG000228.
14. N. Huang, Z. Wu, S. Long, K. Arnold, X. Chen and K. Blank, On instantaneous frequency, *Adv. Adapt. Data Anal.* **1** (2009) 177–229.
15. Z. Wu and N. E. Huang, A study of the characteristics of white noise using the empirical mode decomposition method, *Proc. Roy. Soc. London* **460** (2004) 1597–1611.
16. Z. Wu and N. E. Huang, Ensemble Empirical Mode Decomposition: A noise-assisted data analysis method, *COLA Technical Report* **193** (2005) [ftp://grads.iges.org/pub/ctr/ctr\\_193.pdf](ftp://grads.iges.org/pub/ctr/ctr_193.pdf).
17. Z. Wu and N. E. Huang, Statistical significant test of intrinsic mode functions, in *Hilbert–Huang Transform: Introduction and Applications*, eds. N. E. Huang and S. S. P. Shen (World Scientific, Singapore, 2005), pp. 125–148.
18. Z. Wu and N. Huang, Ensemble Empirical Mode Decomposition: A noise-assisted data analysis method, *Adv. Adapt. Data Anal.* **1** (2009) 1–41.