

# Decomposition of one-dimensional waveform using iterative Gaussian diffusive filtering methods

BY YIH-NEN JENG<sup>1,\*</sup>, P. G. HUANG<sup>2</sup> AND YOU-CHI CHENG<sup>1</sup>

<sup>1</sup>*Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan 70101, Taiwan, Republic of China*

<sup>2</sup>*Department of Mechanical and Materials Engineering, Wright State University, Dayton, OH 45435, USA*

The Gaussian smoothing method is shown to have a wide transition zone around the cut-off frequency selected to filter a given dataset. We proposed two iterative Gaussian smoothing methods to tighten the transition zone: one being approximately diffusive and the other being strictly diffusive. The first version smoothes repeatedly the remaining high-frequency parts and the second version requires an additional step to further smooth the resulting smoothed response in each of the smoothing operation. Based on the choice of the criterion for accuracy, the smoothing factor and the number of iterations are derived for an infinite data length in both methods. By contrast, for a finite-length data string, results of the interior points (sufficiently away from the two endpoints) obtained by both methods can be shown to exhibit an approximate diffusive property. The upper bound of the distance affected by the error propagation inward due to the lack of data beyond the two ends is numerically estimated. Numerical experiments also show that results of employing the iterative Gaussian smoothing method are almost the same as those obtained by the strict diffusive version, except that the error propagation distance induced by the latter is slightly deeper than that of the former. The proposed method has been successfully applied to decompose the wave formation of a number of test cases including two sets of real experimental data.

**Keywords:** iterative Gaussian smoothing method; diffusive filter on time domain; time-series data

---

## 1. Introduction

The least-squares method is widely used in many mathematical, physical and engineering applications (Marr & Hildreth 1980; Lancaster & Salkauskas 1986; Jeng 2000; Fasshauer 2002; Liew *et al.* 2002). When the dataset to be analysed is complicated, a lower degree polynomial approximation is not sufficient to capture the profile trends and a high-degree polynomial is frequently suggested. However, a high-degree polynomial approximation is more complicated to implement and it can sometimes give rise to undesirable oscillations.

\* Author for correspondence (z6208016@email.ncku.edu.tw).

If the least-squares method is defined locally at every point, the method is called the moving least-squares method (Lancaster & Salkauskas 1986). The method can be applied in both discrete and continuous datasets, but in this work we shall consider only the former cases. Moreover, the moving least-squares methods can be used in conjunction with a weighting kernel of Gaussian exponential function to eliminate unnecessary spurious oscillation (Marr & Hildreth 1980; Mackworth & Mokhtarian 1984; Mokhtarian & Mackworth 1986; Lowe 1989; Weiss 1994; Jeng 2000; Fasshauer 2002; Liew *et al.* 2002; Jeng & Cheng 2004). If the approximated polynomial is of the zeroth degree, the method becomes the conventional Gaussian smoothing method (Mackworth & Mokhtarian 1984; Mokhtarian & Mackworth 1986; Lowe 1989). Among many available weighting kernels for the zeroth-order moving least-squares method, the Gaussian exponential function has the following two useful properties. First, by a careful selection of an appropriate smoothing factor,  $\sigma$ , as defined in the Gaussian function  $\exp[-x^2/(2\sigma^2)]$ , there exists a desired threshold frequency above which the waveforms can be effectively filtered out. Second, the relation between the original data and resulting smoothed data is akin to that observed between the initial temperature distribution and the exact solution at a corresponding instant of the equation for unsteady one-dimensional heat conduction (Carslaw & Jaeger 1957). In other words, the Gaussian smoothing method is closely related to the spontaneous diffusion process of nature and therefore obeys the second law of thermodynamics (Morse & Feshbach 1953).

The signal decomposition using the wavelet transform has been successfully applied to many fields (Hess-Nielsen & Wickerhauser 1996; Goswami & Chan 1999; Sonka *et al.* 1999; Gonzalez & Woods 2002; Misiti *et al.* 2007). The wavelet transform employs a weighting kernel function defined within a finite interval to resolve the local spectrum. Although many filters using the wavelet kernels perform very well, there is no evidence to show that they are non-diffusive. The other class of signal decomposition involves the Laplacian and Gaussian pyramids that are widely employed in the field of image processing (Burt & Adelson 1983; Sonka *et al.* 1999; Gonzalez & Woods 2002). While these methods aim to provide a significant data reduction for imaging procession, it is possible that the present approach can be used as a basic tool for the data reduction. In addition, there are a number of other methods proposed to decompose a time-series dataset consisting of many waves with different and variable wavelengths; for example, the matrix pencil method (Hua & Sarkar 1989; Ruscio 1997) and empirical mode decomposition method (Huang *et al.* 1998, 1999). However, except for the wavelet filters, a complete discussion of these topics is not within the scope of the present paper.

In §2*a*, the Gaussian smoothing method in terms of the zeroth-order moving least-squares method is illustrated. The diffusive property and transition zone of the resulting response are also discussed. In §2*b,c*, two iterative Gaussian smoothing methods are proposed, one with and one without strict diffusive properties, respectively. Their resulting smoothed part, which is defined as the difference between the original data and final high-frequency part, is shown to have a narrower transition zone than that obtained by employing the original Gaussian smoothing method. The applications of these iterative Gaussian smoothing methods for discrete finite-length data string are discussed in §2*d*.

Error estimations by applying the smoothing methods discussed in §2*b* to finite-length data string are also shown in §2*d*. A list of the iterative procedure is shown in §2*e*. Finally, numerical tests of iterative Gaussian methods for a number of cases are included in §3.

## 2. Theoretical development

### (a) Gaussian smoothing method for a discrete data string of infinite range

Consider a set of infinite data  $(x_i, y_i)$ ,  $x_i = i\Delta x$ ,  $-\infty < i < \infty$ , to be approximated by a polynomial. The error targeting function of the weighted moving least-squares method is defined at every data point  $x_j$ ,  $-\infty < j < \infty$  as

$$I_j = \sum_{i=-\infty}^{\infty} \exp\left[-\frac{(x_i - x_j)^2}{2\sigma^2}\right] \left[ y_i - \sum_{m=0}^M A_{m,j}(x_i - x_j)^m \right]^2. \quad (2.1)$$

The moving least-squares method requires that  $I_j$  is a minimum with respect to  $A_{0,j}, \dots, A_{M,j}$ . Thus, a set of simultaneous algebraic equations will have to be solved at every point  $x = x_j$  (Marr & Hildreth 1980; Jeng 2000; Fasshauer 2002; Liew *et al.* 2002). If the polynomial contains only the zero-degree term ( $M=0$ ), the result becomes the well-known Gaussian smoothing method (Mackworth & Mokhtarian 1984; Mokhtarian & Mackworth 1986; Lowe 1989), which takes the following form explicitly:

$$\left. \begin{aligned} \bar{y}_j &= \frac{1}{k} \sum_{i=-\infty}^{\infty} \exp\left[-\frac{(x_i - x_j)^2}{2\sigma^2}\right] y_i, \quad -\infty < j < \infty, \\ k &= \sum_{i=-\infty}^{\infty} \exp\left[-\frac{(x_i - x_j)^2}{2\sigma^2}\right]. \end{aligned} \right\} \quad (2.2)$$

As  $\Delta x \rightarrow 0$ ,  $k$  can be replaced by the factor of  $\sqrt{2\pi}\sigma/\Delta x$  and the above formula becomes the explicit form shown in Marr & Hildreth (1980), Mackworth & Mokhtarian (1984), Mokhtarian & Mackworth (1986), Lowe (1989), Weiss (1994), Jeng (2000), Fasshauer (2002) and Liew *et al.* (2002).

Assume  $y_j$  can be expressed in the following discrete form:

$$y_j = \sum_{l=0}^{\infty} \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \quad -\infty < j < \infty, \quad (2.3)$$

where  $\lambda_l$  is the wavelength of the  $l$ th mode. For a finite data string with a period  $T$ ,  $\lambda_l = T/l$ , the upper limit of the summation is a finite integer and equation (2.3) is just a discrete Fourier series expansion. It should be pointed out that although we use equation (2.3) as an example in the following discussion, it is not always possible to identify such a period,  $T$ , for a real data string because not all data are periodical by their nature. Therefore, filters defined on the spectral domain may introduce both amplitude attenuation and phase error because it may not be possible to filter a real dataset on the spectral domain precisely. By contrast, it

will be shown that the proposed filtering procedure, which is operated on the time domain, works very well without knowing the exact form of the spectrum, involving a prior knowledge of the values of  $c_i$ 's,  $d_i$ 's and  $\lambda_i$ 's.

After applying the Gaussian smoothing method, it can be shown that the response is

$$\bar{y}_j = \sum_{l=0}^{\infty} a(\sigma, \lambda_l) \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \quad (2.4)$$

where

$$a(\sigma, \lambda_i) = \frac{1}{k} \sum_{i=-\infty}^{\infty} \exp \left( -\frac{x_i^2}{2\sigma^2} \right) \cos \frac{2\pi x_i}{\lambda_i}. \quad (2.5)$$

By approximating the summation over all  $i$ 's of equation (2.5) with a proper integration formula and making use of the definition of  $k$ , equation (2.2), one can deduce the following relationship:

$$\begin{aligned} a(\sigma, \lambda_i) &\approx \frac{\left[ \int_{-\infty}^{\infty} \exp \left[ -\frac{x^2}{2\sigma^2} \right] \cos \frac{2\pi x}{\lambda_i} dx + O(\Delta x^2) \right]}{\left[ \int_{-\infty}^{\infty} \exp \left[ -\frac{x^2}{2\sigma^2} \right] dx + O(\Delta x^2) \right]} \\ &\approx \exp \left[ -2\pi^2 \sigma^2 / \lambda_i^2 \right] + O(\Delta x^2), \end{aligned} \quad (2.6)$$

which satisfies the following inequalities:

$$\pm \varepsilon \leq a(\sigma, \lambda_i) \leq 1, \quad (2.7)$$

where  $\varepsilon$  is a positive machine round-off error. The second inequality of equation (2.7) appears to be obvious because a cosine function is always less than 1. The first inequality of equation (2.7) can be verified only by extensive numerical tests. It reflects the possibility that, for modes having  $\sigma \gg \lambda_i$ , the exponential function of equation (2.6) is a positive value close to 0 but the error term,  $O(\Delta x^2)$ , may take a small negative value.

If the data are continuous, the diffusive property of the Gaussian smoothing method is

$$0 \leq a(\sigma, \lambda_i) \leq 1. \quad (2.8)$$

After applying the smoothing step to the low-frequency modes, the original  $y_i$  can be cast into a sum of the smooth part  $\bar{y}_i$  and the high-frequency part  $y'_i$ . This implies that the smoothing operation can be considered as a filter. An ideal filter is infinitely sharp such that, there exists a cut-off wavelength,  $\lambda_c$ , whereby  $\bar{y}_i$  does not contain any Fourier mode whose wavelength  $\lambda$  is less than  $\lambda_c$  and  $y'_i$  does not have any mode whose wavelength  $\lambda$  is greater than  $\lambda_c$ . Unfortunately, most time-domain filter operations are not ideal and therefore there exists an open interval,  $\lambda_1 < \lambda_c < \lambda_2$ , where one cannot separate  $\bar{y}_i$  and  $y'_i$  into two distinct regions. This open interval is referred to as the transition zone of the filter. Considering the Gaussian smoothing method as an example, one may identify a region,  $\varepsilon_a < a(\sigma, \lambda_j) < 1 - \varepsilon_a$ , being the corresponding transition zone, where  $\varepsilon_a$  is a small positive constant. Unfortunately, as shown in [figure 1a](#), the transition zone of the Gaussian smoothing can be too wide to be useful. A careful inspection

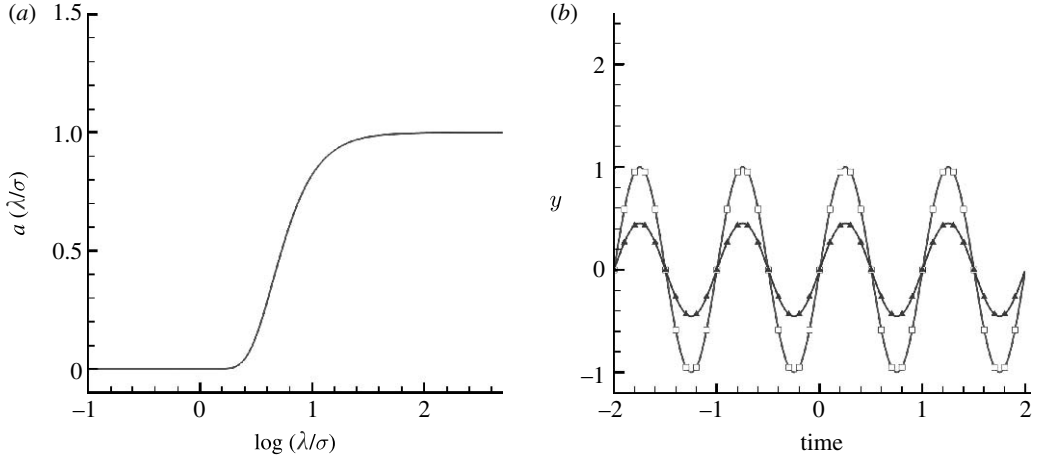


Figure 1. (a) The attenuation factor  $a(\sigma, \lambda_l)$  versus  $\lambda/\sigma$ ; and (b) the original (squares,  $\lambda=1$ ) and smoothed data (triangles,  $\sigma=0.2$ ).

of this figure reveals that

$$\left. \begin{aligned} a(\sigma, \lambda_j) &\rightarrow 0, & \text{if } \lambda_j < 0.6\sigma, \\ a(\sigma, \lambda_j) &\rightarrow 1, & \text{if } \lambda_j > 40\sigma, \\ \varepsilon_a < a < 1 - \varepsilon_a, & & \text{if } 0.6\sigma \leq \lambda_j \leq 40\sigma. \end{aligned} \right\} \quad (2.9)$$

This implies that, when  $\sigma > 1.7\lambda$ , the wave can be almost completely filtered; when  $\sigma < 0.025\lambda$ , the wave can be completely retained; when  $\sigma$  has a value in between  $1.7\lambda$  and  $0.025\lambda$ , it is a transition region where the wave can only be partially retained (or filtered). For example, figure 1b shows that the response of a single sine function with  $\lambda=1$  via the Gaussian smoothing method with  $\sigma=0.2$  has led to a smoothed profile showing  $a(0.2, 1) \approx 0.45404$ . Unfortunately, the transition zone of the Gaussian smoothing method is rather wide, and it is only useful to decompose two wave forms with a wavelength ratio larger than 40 : 0.6. This makes the method impractical for filtering or decomposing any real data. In order to derive a smoothing method having a narrower transition zone, the following iterative scheme is proposed.

(b) *Iterative Gaussian smoothing method for a discrete data string of infinite range*

By denoting  $S\{y_j\}$  as an operator for applying the Gaussian smoothing method to  $y_j$  and renaming  $\bar{y}_j$  on the l.h.s. of equation (2.2) as  $\bar{y}_{j,1}$ , where the subscript 1 denotes the smoothed part of the first cycle, one yields

$$\left. \begin{aligned} \bar{y}_{j,1} &= S\{y_j\} = \sum_{l=0}^{\infty} a(\sigma, \lambda_l) \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], & -\infty < j < \infty, \\ y'_{j,1} &= y_j - \bar{y}_j = \sum_{l=0}^{\infty} (1 - a(\sigma, \lambda_l)) \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \end{aligned} \right\} \quad (2.10)$$

where  $y'_{j,1}$  is the remaining high-frequency part of the first cycle. Likewise, the remaining high-frequency part is smoothed again to obtain yet again the smooth and high-frequency parts of the second cycle, say  $\bar{y}_{j,2}$  and  $y'_{j,2}$ , respectively. Repeating the procedure up to  $m$ th cycle, the corresponding smooth and high-frequency parts are, respectively,

$$\left. \begin{aligned} \bar{y}_{j,m} &= S\{y'_{j,m-1}\} = \sum_{l=0}^{\infty} a(\sigma, \lambda_l) (1 - a(\sigma, \lambda_l))^{m-1} \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \\ y'_{j,m} &= \sum_{l=0}^{\infty} (1 - a(\sigma, \lambda_l))^m \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \end{aligned} \right\} \quad (2.11)$$

The accumulated smooth parts,  $\bar{y}_j(m) = \bar{y}_{j,1} + \dots + \bar{y}_{j,m}$ , are related to the final high-frequency part through the following relation:

$$\bar{y}_j(m) = (y_j - y'_{j,1}) + (y'_{j,1} - y'_{j,2}) + (y'_{j,2} - y'_{j,3}) + \dots + (y'_{j,m-1} - y'_{j,m}) = y_j - y'_{j,m}. \quad (2.12)$$

For convenience, we rewrite the high-frequency part of equation (2.11) as

$$y'_{j,m} = \sum_{l=0}^{\infty} b(\sigma, \lambda_l, m) \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \quad (2.13)$$

where

$$b(\sigma, \lambda_l, m) = (1 - a(\sigma, \lambda_l))^m. \quad (2.14)$$

It can be shown that

$$0 \leq b(\sigma, \lambda_l, m) \leq 1 \pm m\varepsilon, \quad \forall \sigma > 0. \quad (2.15)$$

Figure 2 shows the variation of  $1 - b(\sigma, \lambda_l, m)$  with respect to the ratio  $\lambda_l/\sigma$  for a range of iteration steps,  $m$ . The line with  $m=1$  shows the attenuation factor of the low-frequency part generated by applying the Gaussian smoothing method once. As the iteration step,  $m$ , increases, the whole transition zone shifts to the left and its width becomes narrower and narrower. Since the lower bound of  $a(\sigma, \lambda_l)$  is  $\pm \varepsilon$ , as shown in equation (2.7), the upper bound of  $b(\sigma, \lambda_l, m)$  will have a value of  $1 \pm \varepsilon$  where  $-m\varepsilon \ll \varepsilon \ll m\varepsilon$ . This may cause some problems that the high-frequency wave may manifest itself during iterations and hence lead to a divergence of the filtering procedure. We shall discuss a fix to this problem in §2b.

Suppose we applied the iterative Gaussian smoothing method to a data string having two distinct wavelengths,  $\lambda_1$  and  $\lambda_2$ , where  $\lambda_1 > \lambda_2$ . If we would like to retain  $\lambda_1$ -wave while filter  $\lambda_2$ -wave, it is intuitive to require that

$$\left. \begin{aligned} [1 - a(\sigma, \lambda_1)]^m &\approx [1 - \exp \{-2\pi^2 \sigma^2 / \lambda_1^2\}]^m = 1 - \delta, \\ [1 - a(\sigma, \lambda_2)]^m &\approx [1 - \exp \{-2\pi^2 \sigma^2 / \lambda_2^2\}]^m = \delta, \end{aligned} \right\} \quad (2.16)$$

where the parameter  $\delta$  can take an arbitrarily small value. The solution of this set of simultaneous equations will give the value of the smoothing factor  $\sigma$  and the number of cycles,  $m$ , required to perform the decomposition of the two waves.

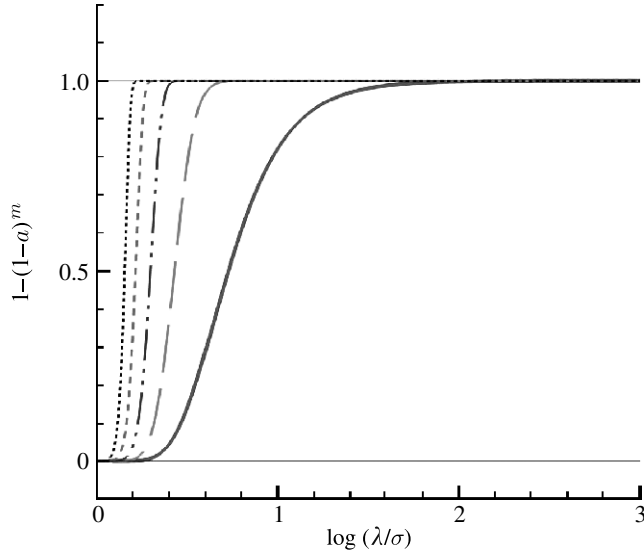


Figure 2. The attenuation factor of the iterative Gaussian smoothing method,  $1 - b = 1 - (1 - a)^m$  versus single smoothing method,  $\lambda/\sigma$  for several  $m$ 's in a cycle. Heavy solid line,  $m=1$ ; long-dashed line,  $m=10$ ; dash-dotted line,  $m=100$ ; dashed line,  $m=1000$ ; dotted line,  $m=10\,000$ .

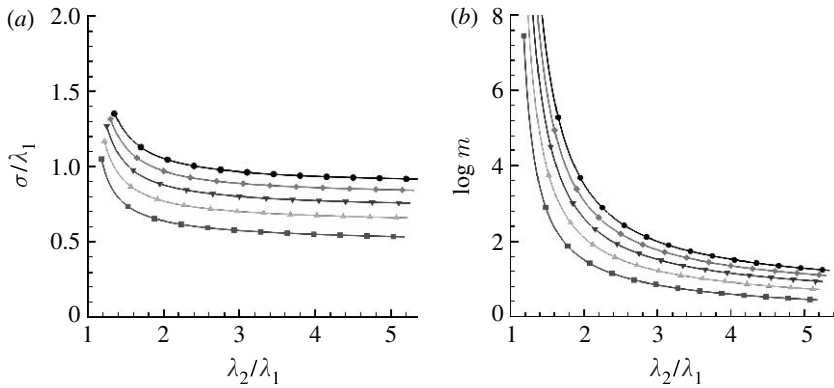


Figure 3. The parameters of iterative Gaussian smoothing method for several values of  $\delta$  of equation (2.16): (a)  $\sigma/\lambda_1$  versus  $\lambda_2/\lambda_1$  and (b)  $\log m$  versus  $\lambda_2/\lambda_1$  for both versions. Squares, 0.01–0.99; up triangles, 0.001–0.999; down triangles, 0.0001–0.9999; diamonds, 0.00001–0.99999; circles, 0.000001–0.999999.

A plot of these solutions is depicted in figure 3a,b, for the value of  $\sigma/\lambda_1$  and  $m$ , respectively. In §3 we will show how to apply the iterative Gaussian smoothing method to separate two distinct waves.

Note that, as the value of  $\lambda_2/\lambda_1$  becomes smaller than 2 (whose corresponding parameters are  $\delta=0.001$ ,  $\sigma/\lambda_1 \approx 0.7715$  and  $m \approx 127$ ), the required number of iteration cycles increases exponentially. For example, when  $\lambda_2/\lambda_1=1.5$ , 8134 iteration steps are needed to separate the two waves while when  $\lambda_2/\lambda_1=1.25$ ,  $m \approx 4\,615\,005$ . These numbers can also be affected by the choice of the value of  $\delta$ .

Table 1. Solutions of the required parameters of equation (2.16) for  $\lambda_2/\lambda_1=2$ .

	$\delta=0.01$	$\delta=0.001$	$\delta=0.0001$	$\delta=0.00001$	$\delta=0.000001$
$m^s$	33	127	410	1199	3306
$\sigma^s/\lambda_1^a$	0.639642	0.771546	0.878266	0.970770	1.053768
$m^d$	33	127	410	1199	3306
$\sigma^d/\lambda_1^a$	0.452295	0.545565	0.621028	0.686438	0.745127

<sup>a</sup>The superscript of ( )<sup>s</sup> denotes the iterative Gaussian smoothing method, while ( )<sup>d</sup> denotes the strict diffusive iterative Gaussian smoothing method.

As it can be seen from table 1, if the accuracy increases by an order of magnitude, the required value of  $\sigma/\lambda_1$  increases slightly while the iteration steps  $m$  increase approximately three to four orders of magnitude.

It is interesting to see that the underlying principle of the proposed iteration method is not very different from the classical iteration methods. For example, the filtering is operated on the smooth part in the current method while in the Ray & Ray iterative Gaussian scheme (Ray & Ray 1995) and the Gaussian pyramid (Burt & Adelson 1983; Sonka et al. 1999; Gonzalez & Woods 2002; Misiti et al. 2007), it is the high-frequency part that is repeatedly smoothed. Owing to the fact that the result of applying the Gaussian smoothing method can be looked upon as the exact solution of the one-dimensional time-dependent heat conduction problem, the result of one smoothing step can be interpreted as the corresponding solution of the one-dimensional time-dependent heat conduction problem at the same time instant. The method is therefore akin to a time-marching scheme for solving a partial differential equation by repeatedly reducing the residue to zero in a sequential manner.

(c) *An iterative Gaussian smoothing method with a strict diffusive property*

As shown in equation (2.15), the upper bound of  $b(\sigma, \lambda_l, m)$  has a value of  $1 \pm m\epsilon$  which may cause some problems. In order to make the iterative procedure satisfy a strict diffusive property such that  $0 \leq b(\sigma, \lambda_l, m) \leq 1$ , the Gaussian smoothing method is employed to smooth  $\bar{y}_j$  of equation (2.2) once more and denotes the resulting smoothed part as  $\bar{y}_{j,1}$ ,

$$\left. \begin{aligned} \bar{y}_{j,1} &= S^2\{y_j\} = \sum_{l=0}^{\infty} a^2(\sigma, \lambda_l) \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \quad -\infty < j < \infty, \\ y'_{j,1} &= y_j - \bar{y}_{j,1} = \sum_{l=0}^{\infty} (1 - a^2(\sigma, \lambda_l)) \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \end{aligned} \right\} \quad (2.17)$$

where  $S^2\{y_j\}$  denotes the application of the Gaussian smoothing method, firstly to  $y_j$  and then to  $\bar{y}_j$  once again. The  $m$ th cycle's results are

$$\bar{y}_{j,m} = S^2\{y'_{j,m-1}\} = \sum_{l=0}^{\infty} a^2(\sigma, \lambda_l) (1 - a^2(\sigma, \lambda_l))^{m-1} \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right],$$



$$\begin{aligned}
 y'_{j,m} &= \sum_{l=0}^{\infty} (1 - a^2(\sigma, \lambda_l))^m \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right] \\
 &= \sum_{l=0}^{\infty} b(\sigma, \lambda_l, m) \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \quad -\infty < j < \infty, \quad m = 1, 2, \dots
 \end{aligned}
 \tag{2.18}$$

Owing to the strict non-negative property of  $a^2$ , the factor  $b(\sigma, \lambda_l, m)$  satisfies

$$0 \leq b(\sigma, \lambda_l, m) = [1 - a^2(\sigma, \lambda_l)]^m \leq 1.
 \tag{2.19}$$

In other words, this iterative procedure is a strict diffusive smoothing method, and the corresponding variation of this new attenuation factor,  $1 - b(\sigma, \lambda_l, m)$ , with respect to  $\lambda/\sigma$  is exactly the same as the one shown in figure 2, provided that  $\sigma$  is replaced by  $\sigma/\sqrt{2}$ .

The estimated smoothing factor  $\sigma_d$  and iteration  $m_d$  (the subscript ‘d’ stands for the double smoothing that leads to the strict diffusive method) are solved from equation (2.16) with the factor  $a(\sigma, \lambda_l)$  replaced by  $a^2(\sigma, \lambda_l)$  and the factor 2 of the exponential function replaced by 4, respectively. The corresponding parameters  $\sigma_d/\sqrt{2}$  and  $m_d$  for different values of  $\delta$  are exactly the same as shown in figure 3a,b, respectively. It is noted that the required computing time of employing the strict diffusive iterative Gaussian smoothing method is twice that for the iterative Gaussian smoothing method.

(d) *Filtering a discrete data string with finite length*

In practice, data are not periodic and therefore they cannot be interpreted as infinite data strings. The extension of the two proposed iterative Gaussian smoothing methods to a data string of finite length is discussed in this section.

For a finite-range data string, say  $(x_j, y_j)$ ,  $j=0, 1, 2, \dots, n$ , the corresponding discrete Fourier expansion is

$$y_j = \sum_{l=0}^{n-1} \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right], \quad j = 0, 1, 2, \dots, n-1.
 \tag{2.20}$$

The application of the Gaussian smoothing method, which is the zeroth-order moving least-squares method for a finite data length, will give

$$\left. \begin{aligned}
 \bar{y}_j &= \frac{1}{\bar{k}_j} \sum_{i=0}^{n-1} \exp(-(x_i - x_j)^2/(2\sigma^2)) \sum_{l=0}^{n-1} \left[ c_l \cos \frac{2\pi x_i}{\lambda_l} + d_l \sin \frac{2\pi x_i}{\lambda_l} \right], \\
 \bar{k}_j &= \sum_{i=0}^{n-1} \exp(-(x_i - x_j)^2/(2\sigma^2)).
 \end{aligned} \right\}
 \tag{2.21}$$

Unlike  $k$  of equation (2.2),  $\bar{k}_j$  is not a constant but changes with respect to  $x_j$  now. Following the similar transform to obtain equation (2.4), the following relations

are obtained:

$$\begin{aligned}
 \bar{y}_j &= \frac{1}{\bar{k}_j} \sum_{l=0}^{n-1} \left\{ \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right] \sum_{i=0}^{n-1} \exp \left( -(x_i - x_j)^2 / (2\sigma^2) \right) \cos \frac{2\pi(x_i - x_j)}{\lambda_l} \right. \\
 &\quad \left. + \left[ d_l \cos \frac{2\pi x_j}{\lambda_l} - c_l \sin \frac{2\pi x_j}{\lambda_l} \right] \sum_{i=0}^{n-1} \exp \left( -(x_i - x_j)^2 / (2\sigma^2) \right) \sin \frac{2\pi(x_i - x_j)}{\lambda_l} \right\} \\
 &= \frac{1}{\bar{k}_j} \sum_{l=0}^{n-1} \left\{ \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right] \sum_{i=-j}^{n-1-j} \exp \left( -x_i^2 / (2\sigma^2) \right) \cos \frac{2\pi x_i}{\lambda_l} \right. \\
 &\quad \left. + \left[ d_l \cos \frac{2\pi x_j}{\lambda_l} - c_l \sin \frac{2\pi x_j}{\lambda_l} \right] \sum_{i=-j}^{n-1-j} \exp \left( -x_i^2 / (2\sigma^2) \right) \sin \frac{2\pi x_i}{\lambda_l} \right\}. \quad (2.22)
 \end{aligned}$$

If  $x_m > 5\sigma$  where  $m = \min(j, n-1-j)$ , the magnitude of the Gaussian functions,  $\exp[-x_m^2/(2\sigma^2)]$ , is of the order of  $O(-6)$  and is of no significance and hence equation (2.22) can be considered to be unaffected by the finite-length assumption of the data

$$\begin{aligned}
 \frac{1}{\bar{k}_j} \sum_{i=-j}^{n-1-j} \exp \left( -x_i^2 / (2\sigma^2) \right) \cos \frac{2\pi x_i}{\lambda_l} &\approx \frac{1}{k} \sum_{i=-\infty}^{\infty} \exp \left( -x_i^2 / (2\sigma^2) \right) \cos \frac{2\pi x_i}{\lambda_l} = a(\sigma, \lambda_l), \\
 \sum_{i=-j}^{n-1-j} \exp \left( -x_i^2 / (2\sigma^2) \right) \sin \frac{2\pi x_i}{\lambda_l} &\approx \sum_{i=-\infty}^{\infty} \exp \left( -x_i^2 / (2\sigma^2) \right) \sin \frac{2\pi x_i}{\lambda_l} = 0. \quad (2.23)
 \end{aligned}$$

As a result, equation (2.22) can be written as

$$\bar{y}_j \approx \sum_{l=0}^{\infty} a(\sigma, \lambda_l) \left[ c_l \cos \frac{2\pi x_j}{\lambda_l} + d_l \sin \frac{2\pi x_j}{\lambda_l} \right]. \quad (2.24)$$

On the other hand when data points are close to the boundary, the upper bound of the difference between smoothing for finite data points and infinite data points,  $\bar{y}_j$  and  $\bar{y}_j^\infty$ , respectively, can be estimated as

$$\begin{aligned}
 |\bar{y}_j - \bar{y}_j^\infty| &= \left| \frac{1}{\bar{k}_j} \sum_{i=0}^n \exp \left[ -\frac{(i-j)^2 (\Delta x)^2}{2\sigma^2} \right] y_i - \frac{1}{j} \sum_{i=-\infty}^{\infty} \exp \left[ -\frac{(i-j)^2 (\Delta x)^2}{2\sigma^2} \right] y_i \right| \\
 &\leq \frac{y_{\max}}{\bar{k}_j} \Phi(-j\Delta x) + \left| \frac{1}{\bar{k}_j} - \frac{1}{k} \right| k \bar{y}_j^\infty \approx \frac{2\Phi(-j\Delta x)}{1 - \Phi(-j\Delta x)} y_{\max,1}, \quad (2.25)
 \end{aligned}$$

where  $\Phi$  is the error function and  $y_{\max,1} = \max_{j \approx 0} [\bar{y}_{\max}, \bar{y}_j^\infty]$  in which  $\bar{y}_{\max} = \max_{i \approx 0} [\bar{y}_{i < 0}]$  and the index,  $i \approx 0$ , implies the points in the vicinity of region where  $\exp[-(i-j)^2(\Delta x)^2/2\sigma^2] \approx 1$ .

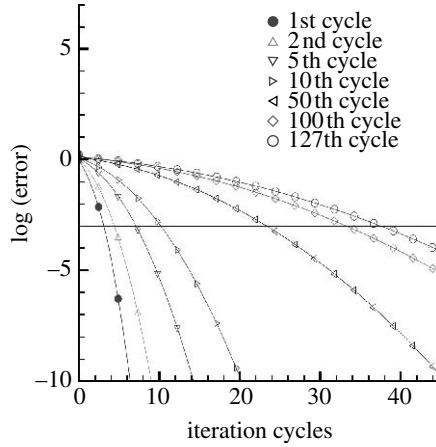


Figure 4. The upper bound estimations of error with respect to iteration cycles.

Table 2. Estimated contaminated distances with respect to different accuracy level of equation (2.16).

	$m$	$\delta=0.01$	$\delta=0.001$	$\delta=0.00001$	$\delta=0.000001$
$x_{fade}^s/\sigma$	1	2.35	3.11	4.28	5.22
$x_{face}^d/\sigma$	1	3.39	4.41	6.05	7.37
$x_{fade}^s/\sigma$	2	3.39	4.41	6.05	7.37
$x_{face}^d/\sigma$	2	4.91	6.33	8.60	10.43
$x_{fade}^s/\sigma$	10	7.99	10.20	13.73	16.60
$x_{face}^d/\sigma$	10	11.49	14.61	19.57	23.61
$x_{fade}^s/\sigma$	50	18.45	23.37	31.20	37.57
$x_{face}^d/\sigma$	50	26.31	33.26	44.33	53.32
$x_{fade}^s/\sigma$	126	29.71	37.55	50.02	60.15
$x_{face}^d/\sigma$	126	42.25	53.32	70.95	85.28

When the iterative Gaussian smoothing method is employed, the error estimation at the  $m$ -step can be approximated by the following recurrent equation:

$$e_{j,m} = \frac{2\Phi(-j\Delta x)}{1 - \Phi(-j\Delta x)} y_{\max,m} + \frac{1}{k_j} \sum_{i=0}^n e_{j,m-1} \exp \left[ -\frac{(i-j)^2(\Delta x)^2}{2\sigma^2} \right], \quad (2.26)$$

where  $e_{j,m-1}$  is the estimated error generated in the previous iteration cycle and  $e_{j,0}=0$  initially. The resulting upper bound distributions with respect to iteration cycles are shown in figure 4. For the diffusive iterative Gaussian smoothing method, the error bound estimation is exactly the same provided that the iteration number  $m$  in equation (2.26) is interpreted as the accumulated number of smoothing. The penetration distances for different error levels estimated for both the approximate and strict diffusive methods are depicted in figure 4 and table 2. It can be seen that the penetration distance increases with the increasing

number of iterations in an approximately exponential manner. It is noted that figure 4 and table 2 give the maximum possible errors and therefore the estimated distance of influence can be much larger than real numerical tests. This exercise only serves to give an indication of how the error propagates with iterations.

(e) *The algorithm for the iterative smoothing procedure*

The algorithm for the proposed iterative procedure can be summarized as follows.

- (i) Select the scheme of the iterative Gaussian smoothing method—be it a diffusive or a strict diffusive procedure.
- (ii) Determine the transition zone, say  $\lambda_1$  and  $\lambda_2$ . It is recommended to allow the transition zone to satisfy  $\lambda_2/\lambda_1 \geq 2$  as discussed in §2b.
- (iii) Determine values of  $\sigma$  and  $m$  by solving equation (2.16). If the strict diffusive version is employed, the resulting  $\sigma$  is replaced by  $\sigma/\sqrt{2}$ .
- (iv) Repeatedly apply the corresponding iterative method with the parameter  $\sigma$  for  $m$  iteration cycles.
- (v) The resulting high-frequency part is the desired high-frequency part and the difference between the original data and the high-frequency part becomes the smooth part.

### 3. Results and discussions

In order to examine the proposed iteration procedure, the following composite waveform is made up of two independent waves having wavelengths 0.5 and 0.25, respectively:

$$y(x) = \sin(4\pi x) + \sin(8\pi x). \quad (3.1)$$

If  $\delta=0.001$  and the iterative Gaussian smoothing method are chosen, table 1 suggests that the critical  $\sigma$  and  $m$  are 0.193 and 127, respectively. In order to simulate the infinite-domain limit, the calculation is taken in the range of  $0 \leq x_j \leq 10$  and  $-10 \leq x_i \leq 20$  with  $\Delta x=0.005$  where  $x_i$  and  $x_j$  are defined in equations (2.1) and (2.2), respectively. After all the data at  $x_j$ 's are evaluated in every cycle, they are periodically mapped to those points  $x_i$  in the range of  $-10 < x_i < 0$  and  $10 < x_i < 20$ , respectively. These conditions ensure that  $\exp[-(x_i - x_j)^2/(2\sigma^2)] < 10^{-50}$  for all  $|x_i - x_j| > 10$  and  $0 \leq x_j \leq 10$  simultaneously. Figure 5 shows the resulting accumulated smooth part of the 1st, 5th, 10th, 20th and 50th iteration cycles. It is clear that the smooth part gradually approaches the original long-wave mode. The converging history is shown as the dotted line in figure 6a, which shows an exponential decay of the  $\ell_2$  error and confirms the estimation of the attenuation factor  $a(\sigma, \lambda_i)$  in equation (2.6). The final  $\ell_2$  error is 0.0009794 while the  $\ell_\infty$  error of 0.001726 shows that the maximum error will be slightly larger than  $\delta=0.001$ . Figure 6a also shows the convergent histories for several different  $\delta$  values, all of which show similar exponential convergent behaviour. In figure 6b, there is a solid line that shows a non-convergent behaviour with  $\Delta x=1/7$  and all the other parameters are the same as that of the dotted line (fine-grid solution). Numerical experiments show that if  $\Delta x < 1/8$ , the convergent history

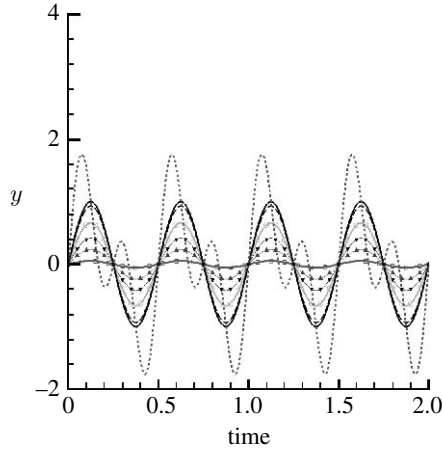


Figure 5. Original data (dashed line) and results of applying the iterative Gaussian smoothing method of 1 (squares,  $y-y'$ ), 5 (up triangles,  $y-y'$ ), 10 (down triangles,  $y-y'$ ), 20 (circles,  $y-y'$ ) and 50 (dashed lines,  $y-y'$ ) cycles, a single smoothing per iteration cycle is employed and other parameters are:  $\delta=0.001$ ,  $\sigma=0.193$ ,  $m=127$ . Original smooth part, solid line.

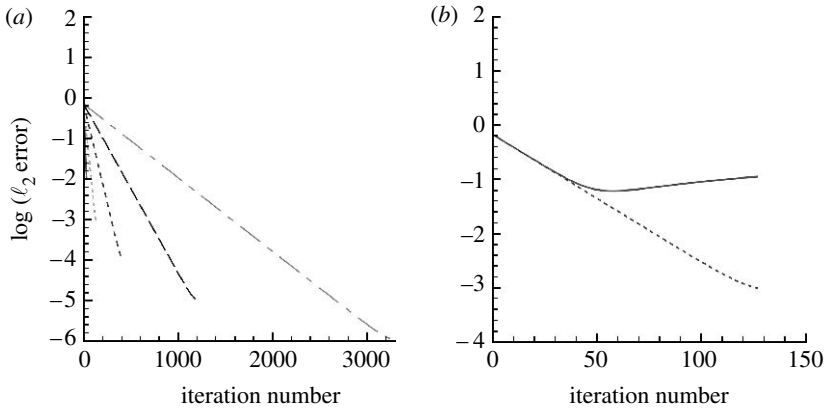


Figure 6. Convergent histories for infinite-domain data (log (square error) versus iteration number of applying single Gaussian smoothing in a cycle) for (a) several criteria  $\delta$  (solid line, 0.01–0.99; dotted line, 0.001–0.999; dashed line, 0.0001–0.9999; long dashed line, 0.00001–0.99999; dot-dashed line, 0.000001–0.999999) and (b)  $\delta=0.001$  with different grid sizes (dotted line, 200–4000 points for  $0 < x < 20$ ; solid line, 140 points for  $0 < x < 20$ ).

is almost the same as that of the dotted line of figure 6b. It seems that a coarse resolution of the original data will affect the accuracy of the wave decomposition procedure.

If the strict diffusive iterative Gaussian smoothing method is chosen, keeping all the rest parameters except  $\sigma=0.13647$ , the corresponding result coincides with that of figure 5 and the converging history is almost equal to the dotted line shown in figure 6a. The final  $\ell_2$  error is 0.0009795 and the  $\ell_\infty$  error is 0.001726, which are almost equal to that of the single smoothing per cycle. Table 3 shows the comparison of both versions for different values of  $\delta$ , provided that the grid size is fine enough. It seems that both the  $\ell_2$  and  $\ell_\infty$

Table 3. Comparison between the iterative Gaussian smoothing method and strict diffusive iterative Gaussian smoothing method for  $\lambda_2/\lambda_1=2$ .

	$\delta=0.01$	$\delta=0.001$	$\delta=0.0001$	$\delta=0.00001$	$\delta=0.000001$
$\Delta x^s$	0.01	0.005	0.005	0.0025	0.00125
$\ell_2^s$ error	0.00929	0.0009795	0.00009879	0.000009954	0.000001001
$\ell_\infty^s$ error	0.01640	0.001726	0.0001742	0.00001753	0.000001767
$\Delta x^d$	0.01	0.005	0.005	0.0025	0.00125
$\ell_2^d$ error	0.00929	0.0009794	0.00009879	0.000009954	0.000001002
$\ell_\infty^d$ error	0.01640	0.001726	0.0001741	0.00001753	0.000001770

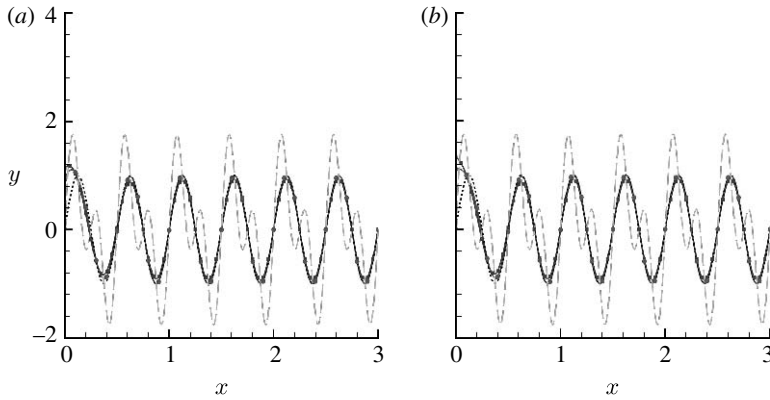


Figure 7. Original data and results of applying the iterative smoothing methods to a finite data string and detailed plots around the left end of  $x \approx 0$  are shown: (a) the iterative single Gaussian smoothing method ( $\sigma=0.193$ ;  $m=127$ ) and (b) the strict diffusive iterative Gaussian smoothing method ( $\sigma=0.13647$ ;  $m=127$ ). Long-dashed line, original data; solid line, original smooth part; squares, 20th cycle  $y-y'$ ; circles, 50th cycle  $y-y'$ .

errors are the same for most  $\delta$ 's, except a small difference for the case with  $\delta=0.000001$ . Besides, all the  $\ell_2$  and  $\ell_\infty$  errors are also approximately equal to the corresponding values of  $\delta$ .

Now the finite-domain wave decomposition for the waveform of equation (3.1) is examined. The data range of  $0 \leq x_i, x_j \leq 20$  is employed where data in the ranges of  $x < 0$  and  $x > 20$  are not considered. Figure 7a,b shows detailed results of 20th and 50th cycles for both versions of employing the iterative Gaussian smoothing method ( $\sigma=0.193$ ) and diffusive version ( $\sigma=0.13746$ ), respectively, where other parameters use values of  $\Delta x=0.005$   $\delta=0.001$ ,  $m=127$ , etc. Obviously, except at the region around the endpoint where  $x=0$ , both solutions coincide with each other. The corresponding final error plots of two versions are plotted in figure 8a,b, respectively. Again, their differences are insignificantly small in most regions.

Table 4 shows the region affected by the endpoints for different values of  $\delta$ . It is clear that the diffusive iterative Gaussian smoothing method results in a slightly deeper influence region (for cases with  $\delta=0.01-0.00001$  with moderate

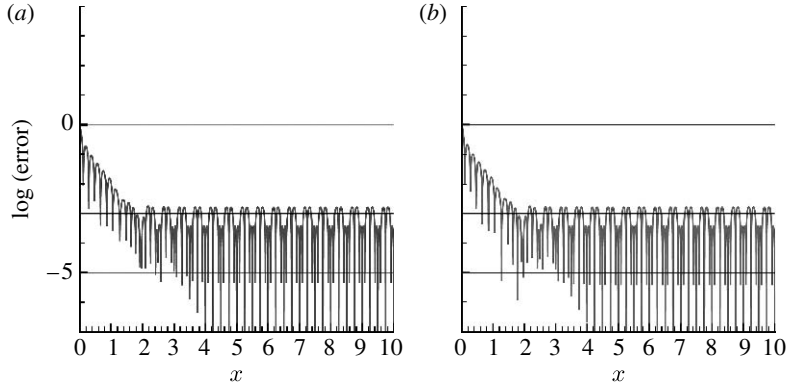


Figure 8. Final log-error plots of applying the iterative smoothing methods to a finite data string and detailed plots around the left end of  $x \approx 0$  are shown: (a) the iterative Gaussian smoothing method ( $\sigma=0.193$ ;  $m=127$ , with and without random number. The 127th cycle error,  $0.001+0.999$  criteria;  $xt=20$ , 4000 points, single smoothing in a cycle) and (b) the strict diffusive iterative Gaussian smoothing method ( $\sigma=0.13647$ ;  $m=127$ , with and without random number. The 127th cycle error,  $0.001+0.999$  criteria;  $xt=20$ , 4000 points, double smoothing in a cycle).

Table 4. Comparison of the ranges affected by the boundary at the two ends calculated by the iterative Gaussian smoothing method and the strict diffusive iterative Gaussian smoothing method for  $\lambda_2/\lambda_1=2$ .

	$\delta=0.01$	$\delta=0.001$	$\delta=0.0001$	$\delta=0.00001$	$\delta=0.000001$
$x_{\text{fade}}^s/\sigma \approx$	1.2	2.3	4.0	5.0	6.6
$x_{\text{fade}}^d/\sigma \approx$	1.2	2.5	4.2	5.4	8

accuracy) than that using the iterative Gaussian smoothing method. For the highly accurate one with  $\delta=0.000001$ , the diffusive version is approximately 20% deeper than the approximate diffusive version.

The second test case adds random number to the function of equation (3.1) so that it becomes

$$y(x) = \sin(4\pi x) + \sin(8\pi x) + 0.5(r'(x) - 0.5), \tag{3.2}$$

where the function  $r'(x)$  is the high-frequency part of  $r(x)$  and  $r(x)$  is the random number generated by the random number function of the Microsoft F-77 software. At first, a long enough data of  $r(x)$  are filtered by the iterative Gaussian smoothing method with  $\sigma=0.0175428$  and  $m \approx 17$  to drop those modes whose  $\lambda > 0.025$ . That  $r'(x)$  employed in equation (3.2) is taken from the central part of the resulting high-frequency part. A typical result shown in figure 9 is corresponding to that of figure 5. All the other data are almost the same as those shown in figures 6 and 7 and tables 3 and 4. Since the spectrum of the random number is non-zero for all high-frequency modes, the consistent results between that of equations (3.1) and (3.2) verify that the proposed filter works.

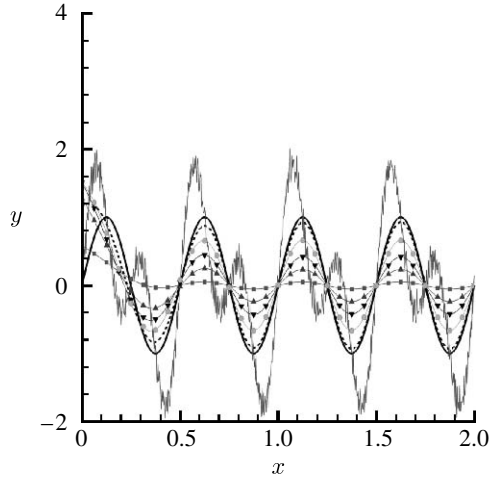


Figure 9. Original data with random number and results of applying the iterative Gaussian smoothing method 1 (squares,  $y-y'$ ), 5 (up triangles,  $y-y'$ ), 10 (down triangles,  $y-y'$ ), 20 (circles,  $y-y'$ ), 50 (dashed lines,  $y-y'$ ) cycles, and other parameters are:  $\delta=0.001$ ,  $\sigma=0.193$ ,  $m=127$ . Thin solid line, original data; thick solid line, original smooth part.

The third test case uses the following composite waveform that involves a non-sinusoidal part.

$$\begin{aligned}
 y(t) = & 1 + 0.2x + 0.005x^2 + 0.3 \exp[-0.005x^2] \sin(0.6\pi x) + 0.4 \sin(10\pi x) \\
 & + 0.2 \sin(5.4\pi x) + 0.3 \exp[-0.0005x^2](1 + 0.2x + 0.01x^2) \sin(3.2\pi x).
 \end{aligned}
 \tag{3.3}$$

In figure 10*a,b*, the original data are shown as a thin solid line, the original low-frequency and non-sinusoidal parts (the first line of equation (3.3)) are shown as a thick solid line, and the estimated low-frequency and non-sinusoidal parts are shown as a dashed line. The error around the two ends is again obvious. The corresponding log-error plots are shown in figure 11*a,b*, respectively. The error around the two ends attenuates in an exponential manner as points move towards the interior region such that the diffusive version has a slightly longer penetration length (approx. 1.8 for both versions) than the iterative Gaussian smoothing method.

The above comparisons show that, in the interior point far from the two ends, the difference between the results of employing the iterative and diffusive iterative Gaussian smoothing methods is insignificant for both the finite- and infinite-domain data. Note that the latter version should remove a longer segment around the two ends than that of the former version. Moreover, the required computing time of employing the strict diffusive iterative Gaussian smoothing method is about twice that obtained using the iterative Gaussian smoothing method. The authors have examined many other test cases with iteration step  $m \leq 10\,000$  and similar conclusions for the original and diffusive smoothing versions were obtained. Therefore, although the iterative Gaussian smoothing method cannot be proven to be a strict diffusive filter, it



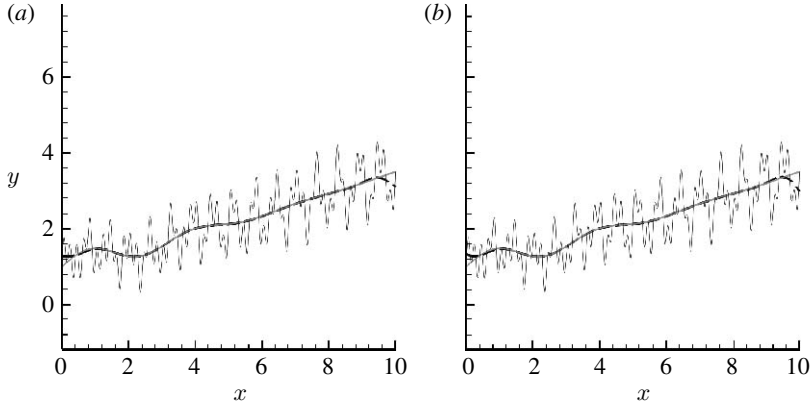


Figure 10. Results of filtering a composite waveform involve a non-sinusoidal and low-frequency part,  $y(x) = y_s + 0.4 \sin(10\pi x) + 0.2 \sin(5.4\pi x) + 0.3 \exp(-0.0005x^2)(1 + 0.2x + 0.01x^2) \sin(3.2\pi x)$ ;  $y_s = 1 + 0.2x + 0.005x^2 + 0.3 \exp(-0.005x^2) \sin(0.6\pi x)$ , using: (a) the iterative Gaussian smoothing method with  $\delta = 0.001$ ,  $\sigma = 0.411$ ,  $m = 5$  and (b) the strict diffusive iterative Gaussian smoothing method with  $\delta = 0.001$ ,  $\sigma = 0.2906$ ,  $m = 5$ ; double smoothing in a cycle. Thin solid line, original data; thick solid line, original smooth part; dashed line, estimated smooth part.

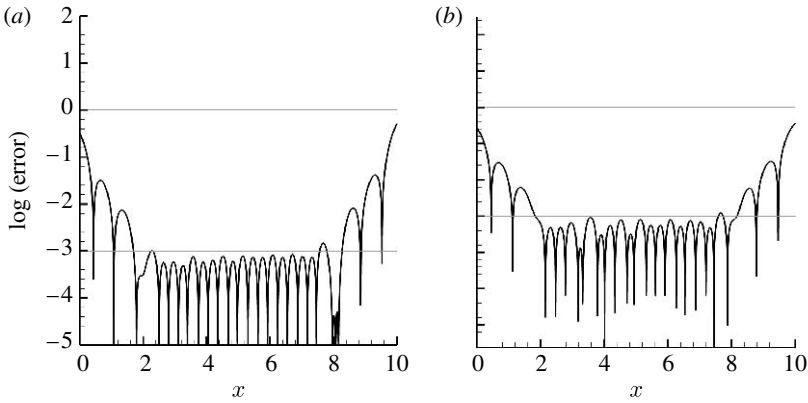


Figure 11. Final log-error plots of applying the iterative smoothing methods to a finite data string of figure 10. (a) Error of employing the iterative Gaussian smoothing method ( $\sigma = 0.4110$ ;  $m = 5$ ; single smoothing in a cycle) and (b) error of employing the strict diffusive iterative Gaussian smoothing method ( $\sigma = 0.2906$ ;  $m = 5$ ; double smoothing in a cycle).

can be applied to the cases with  $m < 10\,000$  for both finite- and infinite-domain data so as to save the computing resource. Based on this argument, the following practical applications merely employ the iterative Gaussian smoothing method.

According to the above-mentioned tests and many other cases not shown here, it seems that the penetration distance of employing the iterative Gaussian smoothing method confirms the estimation given in table 2. As the number of iterations increases, the distance  $x_{\text{fade}}^s / \sigma$  may be scaled down by dividing by an empirical factor of  $1 + 0.8 \log(m)$ . Since the resulting  $\bar{y}$  and  $y'$  in regions of  $(x_0, x_{\text{fade}} + x_0)$  and  $(x_N - x_{\text{fade}}, x_N)$  may involve large errors, it is recommended to discard them if the data within these regions are not important and if the data length is long enough.

Table 5. Comparison of different filtering techniques.

filters	parameters	$\max_{4 \leq x \leq 6}  \text{error} ^a$
original Gaussian smoothing	$\sigma=0.4$	$10^{-1.46}$
iterative Gaussian smoothing	$\delta=10^{-3}$ , $m=5$ , $\sigma=0.411$	$10^{-3.09}$
iterative Gaussian smoothing	$\delta=10^{-4}$ , $m=9$ , $\sigma=0.476$	$10^{-4.01}$
iterative Gaussian smoothing	$\delta=10^{-4.5}$ , $m=11$ , $\sigma=0.403$	$10^{-4.28}$
Dajbecnkies wavelet (dbN)	$N=9$	$10^{-3.00}$
Symlet wavelet (symN)	$N=11$	$10^{-3.49}$
Coiflet wavelet (coifN)	$N=5$	$10^{-3.14}$
biorthogonal wavelet pair	Nr=6, Nd=8	$10^{-2.68}$
reverse biorthogonal wavelet pair (rbioNr.Nd)	Nr=6, Nd=8	$10^{-2.90}$

<sup>a</sup> $|\text{error}| = |y_s - y_s^e|$ , where  $y_s$  and  $y_s^e$  are the original and estimated smooth part of figure 10.

A comparison between the present smoothing method and five well-known wavelet filters has been performed. Since all these filters are subject to the errors caused by the end effects induced by the finite data range, only the interior points ( $4 < x < 6$ ) are considered. Table 5 shows maximum absolute errors deduced from the estimated smooth parts as shown in figure 10. The schemes selected for comparison are the Gaussian smoothing method, the iterative Gaussian smoothing method and the following five well-known wavelet filters: Dajbecnkies family wavelet (dbN); Symlet wavelet (symN); Coiflet wavelet (coifN); biorthogonal wavelet pair (biorNr.Nd); and reverse biorthogonal wavelet pair (rbioNr.Nd). In deducing the data using the five wavelet filters, we employed MATLAB (Misiti et al. 2007) toolbox by applying the optimal tuning of the control parameters to suppress error. As shown in table 5, even under the optimal conditions, the performance of these five wavelet filters is not better than the current iterative Gaussian smoothing method and not surprisingly the conventional Gaussian smoothing method is worse than the other approaches. If one opts for a smaller value of  $\delta$  under a moderate data length, the proposed iterative Gaussian smoothing method may turn out to be better than the wavelet filters.

The next case involves the decomposition of a real electrocardiogram (ECG) dataset downloadable from [www.physionet.org](http://www.physionet.org) website. The data record represents an ECG of a healthy adult—data record number edb/e0103. The total number of points is 16 500 and the duration of measurement is 60 s. Since this dataset is a real-time measurement, it contains both the high-frequency noise and the low-frequency baseline wander. The inter-beat (RR) interval of a healthy human at rest is of the order of 1 s. Therefore, we selected  $\sigma=1.4$  s to filter the low-frequency baseline wander and  $\sigma=0.05$  s to filter the high-frequency noise. The critical iterations for both smoothings were fixed at 127 implying that the cut-off transition zone for the low-frequency baseline wander is between wavelengths of 1.82 and 3.64 s, and the cut-off transition zone for the high-frequency noise is between wavelengths of 0.065 and 0.13 s. These choices were selected by trial and error aiming at eliminating both the baseline wander and the high-frequency noise while preserving the primary wave signatures.

Figure 12 shows the comparison of the raw and the smoothed ECG data in which both the low-frequency baseline wander and the high-frequency noise are filtered. It can be seen that the current smoothed data clearly display signatures

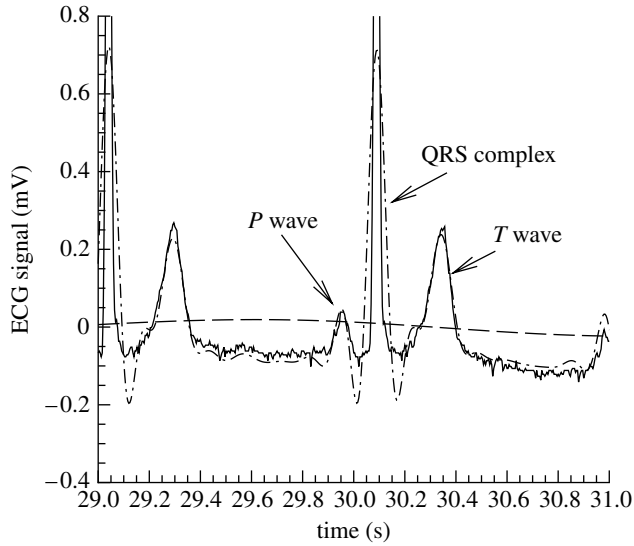


Figure 12. The high/low-pass filtering of an ECG signal. Solid line, raw data; dot-dashed line, smoothed data; dashed line, baseline wander.

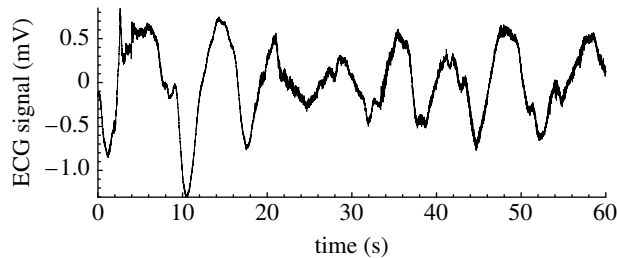


Figure 13. The baseline wander of an ECG signal.

of a healthy ECG pattern—it starts with a P wave, followed by the QRS complex and the T wave. The baseline wander of the raw data of the ECG data examined is fairly gradual. To create a challenge, we have altered the ECG data by adding a measured baseline wander, which is also downloadable from the same website. As shown in figure 13, the magnitude of this baseline wander is of the same order as that of the ECG signals and it is also accompanied by additional random noise. After the iterative Gaussian smoothing method is applied to the dataset to filter the low-frequency baseline wander and the high-frequency noise, the results are shown in figure 14. As shown in the figure, the current method provides excellent smoothing of the raw data, and the repeating pattern of a healthy ECG is clearly observed in the smoothed data.

The next case involves some experimental data from NASA Langley Research Center Workshop on Computational Fluid Dynamics (CFD) Validation of Synthetic Jets and Turbulent Separation Control ([cfdval2004.larc.nasa.gov](http://cfdval2004.larc.nasa.gov)). One of the cases in the workshop involves a zero-mass-flux, or synthetic, jet issuing from a circular orifice. The zero-mass jet is driven by the rapid motion of a piston-driven deformable wall inside a cavity. The piston is driven

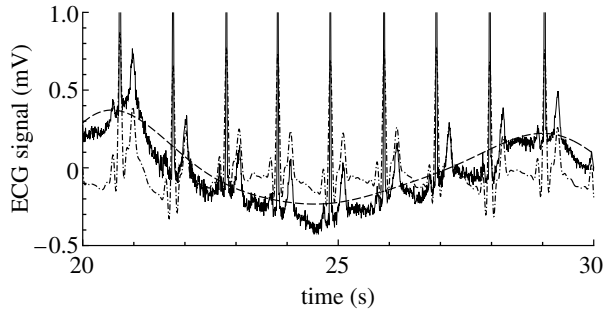


Figure 14. High/low-pass filtering of an ECG signal perturbed by a strong baseline wander (solid line, raw data; dot-dashed line, smoothed data; dashed line, predicted baseline wander).

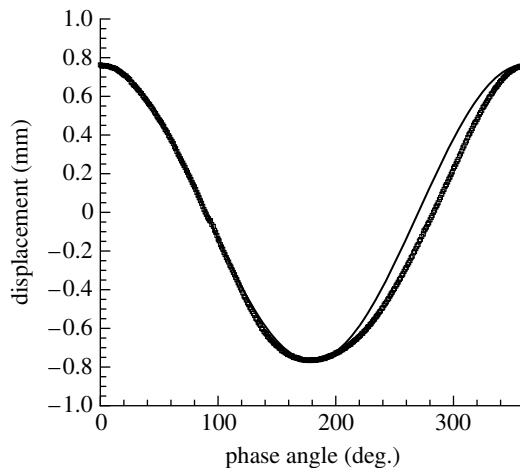


Figure 15. The comparison of the piston displacement and the data best fit using a sine wave (squares, experimental data; solid line, best-fit single wave).

electromechanically using a sinusoidal voltage input with a prescribed frequency. The phase-averaged data for the actuator's displacement about its neutral position are provided and the derivative of these data must be taken to obtain the velocity boundary condition of the diaphragm for later CFD calculations. Figure 15 shows the phase-averaged data for the piston displacement. Although the profile seems smooth, the best fit of a single sine wave reveals that the data profile is not exactly a sine wave and the data may consist of a combination of several different waves. Since these data are phase averaged with the sinusoidal voltage frequency, the data would contain only wave combinations of the voltage frequency and its subharmonics. Indeed, a Fourier power spectrum analysis has revealed that only waveforms with wavelengths of  $360^\circ$ ,  $180^\circ$ ,  $120^\circ$  and  $90^\circ$  of the sinusoidal voltage frequency are visible in the data. The wavelength ratios of two adjacent waveforms are therefore 2, 1.5 and 1.3333. The solutions of equation (2.16) provide choices of  $m$  and  $\sigma$  to divide these waveforms: a critical pair of  $\sigma=139^\circ$  and  $m=127$  is needed to divide the  $360^\circ$  and  $180^\circ$  waves; a critical pair of  $\sigma=108^\circ$  and  $m=8134$  is needed to divide  $180^\circ$  and  $120^\circ$  waves, and a critical pair of  $\sigma=91^\circ$  and  $m=596\ 134$  is needed to divide the  $120^\circ$  and  $90^\circ$  waves.

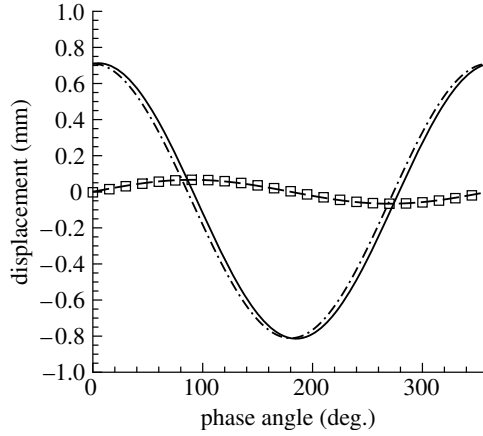


Figure 16. The smoothed displacement profile of the piston's  $360^\circ$  wave (solid line); dot-dashed line,  $0.76 \cos(2\pi t/T) - 0.051$ ; squares, combined wave minus cosine wave equation; dashed line,  $0.066 \sin(2\pi t/T)$ .

The wave decomposition is performed using the iterative Gaussian method discussed with the periodic assumption such that the computing range is extended to the point  $t$ , where  $\exp[-(t - t_e)^2/2\sigma^2] < 10^{-7}$ ,  $t_e = 0$  or  $2\pi$ . The predominant waveform is the  $360^\circ$  one shown as the solid line in figure 16. The  $360^\circ$  waveform consists of two waves. The primary wave can be approximated by fitting a cosine function as  $0.076 \cos(2\pi/T) - 0.051$ , where  $T$  is the time period of the piston cycle. The secondary wave can be obtained by subtracting the primary wave from the smoothed  $360^\circ$  wave and the difference is shown as symbols in the figure. We can further fit the secondary wave with a sine function as  $0.066 \sin(2\pi/T)$ , as shown in figure 16.

The decomposed  $180^\circ$ ,  $120^\circ$  and  $90^\circ$  waves are depicted in figure 17. These waves can be further approximated by three cosine functions with variations in phase shifts and magnitudes:  $0.43 \cos(4\pi(t/T - 12.45/360))$ ,  $0.0177 \cos(6\pi(t/T + 24.53/360))$  and  $0.074 \cos(8\pi(t/T + 0.44/360))$  for the  $180^\circ$ ,  $120^\circ$  and  $90^\circ$  waveforms, respectively.

Finally, the velocity of the piston movement can be obtained analytically by differentiating the sum of five piston displacement equations with respect to time. The analytical form of the velocity is depicted as a solid line in figure 18. Also, a dashed line is presented showing the velocity derived from the best-fit single sine wave to the piston displacement data. These results are then compared with experimental velocity data, which are approximated by the slopes of the displacement data at every two adjacent points. It can be seen that the comparison of the experimental data and the current composite profile is excellent.

#### 4. Conclusions

Two iterative algorithms based on the Gaussian smoothing method to filter and decompose time-series data are developed: one is approximately diffusive and the other is strictly diffusive. While these algorithms are general, they are used here in conjunction with the moving least-squares method to handle the finite data length.

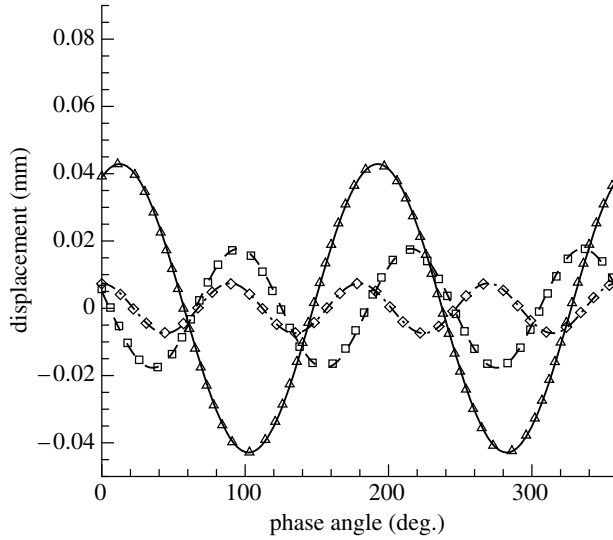


Figure 17. The decomposed  $180^\circ$  (triangles),  $120^\circ$  (squares) and  $90^\circ$  (diamonds) waveforms of the piston displacement. Solid line,  $0.043 \cos [4\pi(t/T - 12.45/360)]$ ; dashed line,  $0.0177 \cos [6\pi(t/T + 24.53/360)]$ ; dot-dashed line,  $0.0074 \cos [8\pi(t/T + 0.44/360)]$ .

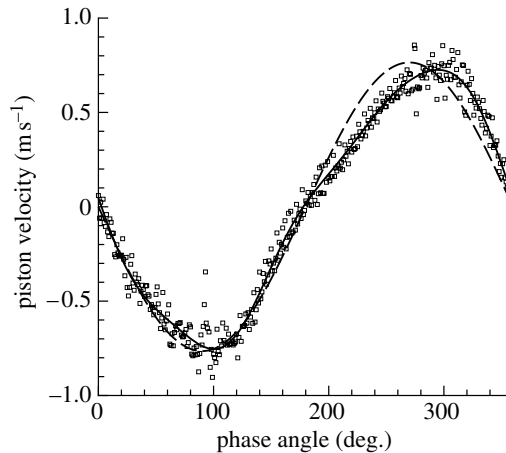


Figure 18. Comparison of the final piston velocities. Squares, experimental data; dashed line, best-fit single wave; solid line, composite waves.

The connection between the frequency domain and the physical space can be quantified by the selection of three parameters: the accuracy criterion; the Gaussian smoothing factor; and the number of iteration performed. The distance of the error penetration from the two data ends into the interior region is studied. It is proven that, in the interior point remote from the two ends, the result is the same as that of the infinite data length. The method has been successfully tested in a number of problems.

The first author is supported by the National Science Council of Taiwan under the grant no. NSC-94-2212-E006-084.

## References

- Burt, P. J. & Adelson, E. H. 1983 The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.* **31**, 532–540. (doi:10.1109/TCOM.1983.1095851)
- Carslaw, H. S. & Jaeger, J. C. 1957 *Conduction of heat in solids*. New York, NY: Oxford University Press.
- Fasshauer, G. 2002 Approximate moving least-squares approximations: a fast and accurate multivariate approximation method. *Curve Surf. Fitting* **1**, 1–10.
- Gonzalez, R. C. & Woods, R. E. 2002 *Digital image processing*, 2nd edn. Upper Saddle River, NJ: Pearson Education, Inc.
- Goswami, J. C. & Chan, A. K. 1999 *Fundamentals of wavelets, theory, algorithms, and applications*. New York, NY: Wiley.
- Hess-Nielsen, N. & Wickerhauser, M. V. 1996 Wavelets and time-frequency analysis. *Proc. IEEE* **84**, 523–540. (doi:10.1109/5.488698)
- Hua, R. & Sarkar, T. K. 1989 Generalized pencil-of-function method for extracting poles of an EM system from its transient response. *IEEE Trans. Anten. Propag.* **37**, 229–234. (doi:10.1109/8.18710)
- Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C. & Liu, H. H. 1998 The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. A* **454**, 903–995. (doi:10.1098/rspa.1998.0193)
- Huang, N. E., Shen, Z. & Long, S. R. 1999 A new view of nonlinear water waves: the Hilbert spectrum. *Annu. Rev. Fluid Mech.* **51**, 417–457. (doi:10.1146/annurev.fluid.31.1.417)
- Jeng, Y. N. 2000 The moving least-squares and least  $p$ -power methods for random data. In *Proc. 7th National Computational Fluid Dynamics Conference, Kenting, Taiwan, August*, pp. 9–14. The Netherlands: Kluwer Academic Publishers.
- Jeng, Y. N. & Cheng, Y. C. 2004 A simple strategy to evaluate the frequency spectrum of a time series data with non-uniform intervals. *Trans. Aeronaut. Astronaut. Soc. (Republic of China)* **36**, 207–214.
- Lancaster, P. & Salkauskas, K. 1986 *Curve and surface fitting: an introduction*, ch. 2, 10. New York, NY, Academic Press.
- Liew, K. M., Huang, Y. Q. & Reddy, J. N. 2002 A hybrid moving least-squares and differential quadrature (MLSDQ) meshfree method. *Int. J. Comput. Eng. Sci.* **3**, 1–12. (doi:10.1142/S1465876302000526)
- Lowe, D. G. 1989 Organization of smooth image curves at multiple scales. *Int. J. Comput. Vis.* **3**, 119–130. (doi:10.1007/BF00126428)
- Mackworth, A. K. & Mokhtarian, F. 1984 Scale-based description of planar curves. In *Proc. 5th Canadian Society for Computational Studies of Intelligence, London, Ontario, May*, pp. 114–119.
- Marr, D. & Hildreth, E. 1980 Theory of edge detection. *Proc. R. Soc. B* **207**, 187–217. (doi:10.1098/rspb.1980.0020)
- Misiti, M., Misiti, Y., Oppenheim, G. & Poggi, J. M. 2007 *Wavelet toolbox 4 user's guide*. Natick, MA: The MathWorks, Inc.
- Mokhtarian, F. & Mackworth, A. 1986 Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 34–43.
- Morse, P. M. & Feshbach, H. 1953 *Methods of theoretical physics*, vol. 2. New York, NY: McGraw-Hill.
- Ray, B. K. & Ray, K. 1995 Corner detection using iterative Gaussian smoothing with constant window size. *Pattern Recogn.* **28**, 1765–1781. (doi:10.1016/0031-3203(95)00046-3)
- Ruscio, D. D. 1997 On subspace identification of extended observability matrix. In *Proc. 36th Conf. on Decision and Control, San Diego, CA, USA, December*, paper no. TA05, pp. 1841–1847. San Diego, CA: IEEE.
- Sonka, M., Hlavac, V. & Boyle, R. 1999 *Image processing analysis, and machine vision*. London, UK: Brooks/Cole Publishing Company.
- Weiss, I. 1994 High-order differentiation filters that work. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 734–739. (doi:10.1109/34.297955)