Li, Yuanlu, Liu, Fawang, Turner, Ian, & Li, Tao
(2018)
Time-fractional diffusion equation for signal smoothing.
Applied Mathematics and Computation, *326*, pp. 108-116.

https://doi.org/10.1016/j.amc.2018.01.007

Time Fractional Order Diffusion Equation for Signal Smoothing

Yuanlu Li[1,2], Fawang Liu[3], Ian W. Turner[3], Tiao Li[1,2]

1. B-DAT, School of Information and Control, Nanjing University of Information Science & Technology, Nanjing, China, 210044;

2. Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science & Technology, Nanjing, China, 210044

3.School of Mathematical Sciences, Queensland University of Technology, GPO Box 2434, Brisbane, Qld. 4001, Australia

Abstract: Time fractional order diffusion equation, as a generalization of classical diffusion equation, was proposed to peak-preserving smoothing of the spectra signal. Two implement methods: explicit difference scheme and implicit difference scheme was presented, respectively. Taking the under processing spectrum as a reference signal to design the diffusion function, so that the diffusion is weaker along the peak, thus, the peak-preserving smoothing can be achieved. The classical diffusion filtering is just a case of time fractional order diffusion filtering. Some simulated signals and an NMR spectrum has been used to verify the proposed method and compare the performance of classical smoothing methods (regularization method, Savitzky-Golay method and wavelet method). Results indicated that the time fractional order diffusion filtering is advantage over the classical diffusion filtering and their smoothing performance is better than that of classical smoothing methods.

Key words: Fractional order diffusion; smoothing; wavelet; regularization method

2010 Mathematics Subject Classification. **65N06, 65N12, 26A33.**

1. Introduction

Because of imperfections in the experimental apparatus, interfering physical or chemical processes, or any of other causes, experimental spectra often contain noise. So signal smoothing is often desired to help with subsequent analysis. For example, if derivatives of the signal are needed, the signal must be especially smooth otherwise the noise will be severely amplified in the derivative signal [1, 2].

At present, there are many methods which can be used to reduce noise in a spectrum [3-11]. Sliding average method is the simplest ones. The Savitzky–Golay method [8] is one of improvements to sliding average method, where a low-order polynomial is fitted to the data within a moving window rather than just taking their average. Thus an increase in the window size can improve the degree of smoothing, but, too broad of a window will reduce the effect of the resolution enhancement and distort the derivative spectra. In additional, a lower degree of the polynomial leads to more effective noise reduction, while a higher polynomial degree preserves more accurate the peak profiles. From Ref. [12], the parameters for the Savitzky–Golay method are selected usually by a trial-and-error method. Therefore, the Savitzky–Golay method has been a popular smoothing method. Similar method includes using a spline to smooth the spectra [7]. As an improved alternative to the Savitzky-Golay method, the regularization method was proposed based on penalized least squares [10, 11]. This smoother is very fast, gives continuous control over smoothness, interpolates automatically, and allows fast leave-one-out cross-validation.

Wavelet method [5, 6, 13-15] is an important method for spectra smoothing, in which the spectrum is decomposed by the discrete wavelet transform (DWT) and the high frequency components under threshold value are discarded then the inverse wavelet transform used to reconstruct the spectrum. However, the number of samples are demanded to be a power of two.

Since the nonlinear diffusion equation was firstly used to image denoising and edge detection by Perona and Malik [16], partial differential equations have been widely used in image processing [17-23].

In the last ten years, the development of fractional order systems theory has led to a new set of tools that began replacing classic procedures and implementations. For example, Bai and Feng [24] use the fractional-order anisotropic diffusion equations for image denoising. Their results indicated that the fractional-order anisotropic diffusion equations can yield good visual effects and better signal-to-noise ratio. The fractional-order level set model was proposed by replacing the integral-order gradient operator of level set function in the energy formulation into the fractional-order one [25]. The regularization operator is constructed by using fractional order derivatives, where the choice of the fractional order for each pixel in the image is driven by the texture map of the image. An adaptive strategy for the restoration of textured images was presented in [26].

Result indicated this is an efficient tool to preserve texture well in the texture regions while removing noise and staircase effects in the image. Three fractional-order TV-L2 models, which are based on the different numerical algorithms of fractional differential, are introduced for image denoising in [27]. The interplay between fractional calculus and signal processing will bring new challenges since the involved mathematical tools are more difficult to compute than the classic, but also richer allowing better models and performances.

In this paper, the time fractional order diffusion equation was proposed to smooth noisy spectra. The time fractional order diffusion equation is a generalization of the classical diffusion equation [16]. It can be obtained by replacing the first-order time derivative in the nonlinear diffusion equation into a fractional derivative of order $\alpha \in (0,1]$. Unlike the classical case, information about all the previous time layers is required when numerically approximating a time fractional diffusion equation on a certain time layer. Fortunately, there have been some methods for solving of the time fractional order diffusion. These methods mainly include explicit difference scheme [28], implicit difference scheme [29-33], and spectral method [34-36], etc.

2. Nonlinear Diffusion Filtering

Witkin [37] found that the solution of a heat diffusion equation with a signal as initial data is equivalent to the convolution of the signal with Gaussians function at each scale. That's to say, the diffusion process is equivalent to smoothing process with a Gaussian kernel. But smoothing process does not discriminate between local features and noise. To recognize invariant features at each scale, a variable diffusion processes was proposed by Perona and Malik [16]. They used the gradient of the actual image as a reference of diffusion function to control the diffusion process. It has been considered as an edge-preserving smoothing method. The process can be formulated as follows:

$$\frac{\partial U(x,t)}{\partial t} = div\left(g(x,t)\frac{\partial U(x,t)}{\partial x} \right), \tag{1}$$

with initial and boundary conditions:

$$U(x,0) = f(x), 0 \le x \le L \tag{2}$$

$$U(0,t)=U(L,t)=0, 0<t<T \tag{3}$$

where $g(x,t)$ is a diffusion function. In general, $g(x,t)$ is a smooth decreasing function with $g(0,t)=1$, $g(x,t)\geq 0$ and $g(x,t)$ tending to zero at infinity. The diffusion strength is controlled by $g(x,t)$.

Two different diffusion functions have been proposed [16]:

$$g_1(x,t)=\exp\left[-\left(\frac{|\nabla U(x,t)|}{\lambda}\right)^2\right] \tag{4}$$

$$g_2(x,t)=\frac{1}{1+\left(\frac{|\nabla U(x,t)|}{\lambda}\right)^2} \tag{5}$$

If the gradient $|\nabla U(x,t)|$ is large, $\mathrm{g}(x,t)$ is small. This means the diffusion is weak. Conversely, if the gradient $|\nabla U(x,t)|$ is small, $\mathrm{g}(x,t)$ is large, the diffusion is strong. The parameter λ is chosen by the noise level and the edge strength. A proper choice of the diffusion function can preserve the edges and even enhance them while being numerically stable [16].

However, we will take use of the spectrum to be smoothed as the reference signal of the diffusion function. So the diffusion functions (4) and (5) will be correspondingly replaced by

$$g_3(x,t)=\exp\left[-\left(\frac{|U(x,t)|}{\lambda}\right)^2\right] \tag{6}$$

$$g_4(x,t)=\frac{1}{1+\left(\frac{|U(x,t)|}{\lambda}\right)^2} \quad . \tag{7}$$

From $g_3(x,t)$ and $g_4(x,t)$, we can find the diffusion coefficient will become weaker along the peak of the signal. In this way, we will better preserve the peak shape of a spectrum.

3. Time Fractional Order Diffusion Filtering

In nonlinear diffusion, improvement of performance was achieved by design of diffusion function. Here, a time fractional diffusion equation was proposed to improve the smoothing performance. In fact, using the diffusion equation to smooth signal is to find a solution of Eq.(8), which meets the initial conditions. In other words, $u(x,0)$ is a noisy signal, and $u(x,T)$ is the smoothed signal. The diffusion process can be described by

$$\begin{cases} \frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = g(x,t)\frac{\partial^2 u(x,t)}{\partial x^2}, \quad \alpha\in(0,1), \quad 0<x<L, \quad 0<t\le T, \\ u(x,0) = f(x), \quad 0<x<L, \\ u(0,t)=u(L,t)=0, \quad 0<t<T. \end{cases} \tag{8}$$

where $g(x,t)$ is a known diffusion function.

3.1. Numerical methods

There are some existing numerical methods for the solution of the fractional order diffusion equation [28-33, 35, 38]. Since finite difference is easy to handle and a real digital signal already is discrete, the finite difference scheme was adopted in this paper.

Setting $\tau = T/N$ and $h = L/M$, where τ and h are time and space steps, respectively, we have $t_0 = 0$, $t_N = T$, $t_k = k\tau$ for $k = 1,2,\cdots,N-1$ and $x_0 = 0$, $x_M = L$, $x_i = ih$ for $i = 1,2,\cdots,M-1$. Let u_i^k be the numerical approximation to $u(x_i,t_k)$.

The Caputo fractional derivative of order α with respect to time is defined as

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)}\int_0^t \frac{\partial u(x,\xi)}{\partial \xi}\frac{1}{(t-\xi)^\alpha}d\xi, 0<\alpha<1. \tag{9}$$

For $0 < \alpha < 1$, the Caputo's fractional derivative is approximated by [39]:

$$\frac{\partial^\alpha u_i^k}{\partial t^\alpha} \cong \sigma_{\alpha,\tau} \sum_{j=1}^{k} \omega_j^{(\alpha)} \left(u_i^{k-j+1} - u_i^{k-j} \right) = \sigma_{\alpha,\tau} \left[u_i^k - \sum_{j=1}^{k-1} \left(\omega_j^{(\alpha)} - \omega_{j+1}^{(\alpha)} \right) u_i^{k-j} - \omega_k^{(\alpha)} u_i^0 \right], \quad (10)$$

where

$$\sigma_{\alpha,\tau} = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)}, \ \omega_j^{(\alpha)} = j^{1-\alpha} - (j-1)^{1-\alpha}, \quad \text{and } 1 = \omega_1^{(\alpha)} > \omega_2^{(\alpha)} > \cdots > \omega_k^{(\alpha)}. \quad (11)$$

3.1.1 Explicit difference scheme

Explicit difference scheme is very simple way to implement and, therefore, they are used almost exclusively. Now Let us consider the simplest discrete approximation of this process.

The explicit numerical method is given by

$$\sigma_{\alpha,\tau} \sum_{j=1}^{k} \omega_j^{(\alpha)} \left(u_i^{k-j+1} - u_i^{k-j} \right) = \frac{g_i^{k-1}}{h^2} \left(u_{i-1}^{k-1} - 2u_i^{k-1} + u_{i+1}^{k-1} \right) \quad (12)$$

Let $\beta_i^{k-1} = \dfrac{g_i^{k-1}}{\sigma_{\alpha,\tau} h^2}$, for k=1, we have

$$u_i^1 = \beta_i^0 u_{i-1}^0 + \left[1 - 2\beta_i^0 \right] u_i^0 + \beta_i^0 u_{i+1}^0, \quad (13)$$

For k>1, we have

$$u_i^k = \beta_i^{k-1} u_{i-1}^{k-1} + \left(1 - 2\beta_i^{k-1} \right) u_i^{k-1} + \beta_i^{k-1} u_{i+1}^{k-1} - \sum_{j=2}^{k} \omega_j^{(\alpha)} \left(u_i^{k-j+1} - u_i^{k-j} \right). \quad (14)$$

The explicit scheme is expressed by matrix-vector notation as following

$$\begin{aligned} U^1 &= \left[A^0 \right] U^0 \\ U^k &= \left[A^{k-1} \right] U^{k-1} - \sum_{j=2}^{k} \omega_j^{(\alpha)} \left(U^{k-j+1} - U^{k-j} \right), k \geq 2 \end{aligned} \quad (15)$$

$$\left[A^{k-1}\right]=\begin{bmatrix} 1-2\beta_1^{k-1} & \beta_1^{k-1} & & & \\ \beta_2^{k-1} & 1-2\beta_2^{k-1} & \beta_2^{k-1} & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{M-2}^{k-1} & 1-2\beta_{M-2}^{k-1} & \beta_{M-2}^{k-1} \\ & & & \beta_{M-1}^{k-1} & 1-2\beta_{M-1}^{k-1} \end{bmatrix},\ U^k=\begin{bmatrix} u_1^k \\ u_2^k \\ \vdots \\ u_{M-1}^k \end{bmatrix}.$$

One can see the explicit iteration scheme is computationally very cheap: It requires only to calculate the three nonvanishing matrix entries each row and to perform a matrix–vector multiplication. But, the time step size is restricted as a small number for stability reason.

3.1.2 Semi-implicit difference scheme

The implicit numerical method is given by

$$\sigma_{\alpha,\tau}\sum_{j=1}^{k}\omega_j^{(\alpha)}\left(u_i^{k-j+1}-u_i^{k-j}\right)=\frac{g_i^{k-1}}{h^2}\left(u_{i-1}^k-2u_i^k+u_{i+1}^k\right), \tag{16}$$

For k=1, we have

$$-\beta_i^0 u_{i-1}^1+\left[1+2\beta_i^0\right]u_i^1-\beta_i^0 u_{i+1}^1=u_i^0, \tag{17}$$

For k>1, we have

$$-\beta_i^{k-1}u_{i-1}^k+\left(1+2\beta_i^{k-1}\right)u_i^k-\beta_i^{k-1}u_{i+1}^k=u_i^{k-1}-\sum_{j=2}^{k}\omega_j^{(\alpha)}\left(u_i^{k-j+1}-u_i^{k-j}\right). \tag{18}$$

The implicit scheme can be expressed by matrix-vector notation as following

$$\begin{aligned} U^1 &= \left[B^0\right]^{-1}U^0 \\ U^k &= \left[B^{k-1}\right]^{-1}\left[U^{k-1}-\sum_{j=2}^{k}\omega_j^{(\alpha)}\left(U^{k-j+1}-U^{k-j}\right)\right], k\ge 2 \end{aligned} \tag{19}$$

$$\left[B^{k-1}\right]=\begin{bmatrix} 1+2\beta_1^{k-1} & -\beta_1^{k-1} & & & \\ -\beta_2^{k-1} & 1+2\beta_2^{k-1} & -\beta_2^{k-1} & & \\ & \ddots & \ddots & \ddots & \\ & & -\beta_{M-2}^{k-1} & 1+2\beta_{M-2}^{k-1} & -\beta_{M-2}^{k-1} \\ & & & -\beta_{M-1}^{k-1} & 1+2\beta_{M-1}^{k-1} \end{bmatrix},\ U^k=\begin{bmatrix} u_1^k \\ u_2^k \\ \vdots \\ u_{M-1}^k \end{bmatrix}.$$

One can see B^{k-1} is strictly diagonally dominant. So B^{k-1} are invertible.

3.2. Stability of implicit difference scheme

Assume $\tilde{u}_i^k$ ($i = 0,1,\cdots,M$; $k = 0,1,\cdots,N$) be the approximation of numerical solution, the error $\varepsilon_i^k = \tilde{u}_i^k - u_i^k$ satisfies

$$-\beta_i^0 \varepsilon_{i-1}^1 + \left(1+2\beta_i^0\right)\varepsilon_i^1 + -\beta_i^0 \varepsilon_{i+1}^1 = \varepsilon_i^0 , \tag{20}$$

$$-\beta_i^{k-1} \varepsilon_{i-1}^k + \left(1+2\beta_i^{k-1}\right)\varepsilon_i^k - \beta_i^{k-1} \varepsilon_{i+1}^k = \varepsilon_i^{k-1} - \sum_{j=2}^{k} \omega_j^{(\alpha)} \left(\varepsilon_i^{k-j+1} - \varepsilon_i^{k-j}\right). \tag{21}$$

The implicit difference scheme is unconditionally stable.

Proof. Let $\left\|E^1\right\|_\infty = \left|\varepsilon_l^1\right| = \max\limits_{1\le i\le M-1}\left|\varepsilon_i^1\right|$, we have

$$\begin{aligned}\left\|E^1\right\|_\infty &= \left|\varepsilon_l^1\right| \\ &\le -\beta_l^0 \left|\varepsilon_{l-1}^1\right| + \left(1+2\beta_l^0\right)\left|\varepsilon_l^1\right| - \beta_l^0 \left|\varepsilon_{l+1}^1\right| \\ &\le \left|-\beta_l^0 \varepsilon_{l-1}^1 + \left(1+2\beta_l^0\right)\varepsilon_l^1 - \beta_l^0 \varepsilon_{l+1}^1\right| \\ &= \left|\varepsilon_l^0\right| \le \left\|E^0\right\|_\infty\end{aligned} \tag{22}$$

Thus, $\left\|E^1\right\|_\infty \le \left\|E^0\right\|_\infty$. Assuming $\left\|E^n\right\|_\infty \le \left\|E^0\right\|_\infty$, $n = 2,3,\cdots,k-1$. $\left|\varepsilon_l^k\right| = \max\limits_{1\le i\le M-1}\left|\varepsilon_i^k\right|$, we have

$$\begin{aligned}\left\|E^k\right\|_\infty &= \left|\varepsilon_l^k\right| \\ &\le -\beta_l^{k-1}\left|\varepsilon_{l-1}^k\right| + \left(1+2\beta_l^{k-1}\right)\left|\varepsilon_l^k\right| - \beta_l^{k-1}\left|\varepsilon_{l+1}^k\right| \\ &\le \left|-\beta_l^{k-1}\varepsilon_{l-1}^k + \left(1+2\beta_l^{k-1}\right)\varepsilon_l^k - \beta_l^{k-1}\varepsilon_{l+1}^k\right| \\ &= \left|\varepsilon_l^{k-1} - \sum_{j=2}^{k}\omega_j^{(\alpha)}\left(\varepsilon_l^{k-j+1} - \varepsilon_l^{k-j}\right)\right| \\ &= \left|\varepsilon_l^{k-1} - \left[\omega_2^{(\alpha)}\varepsilon_l^{k-1} - \sum_{j=1}^{k-2}\left(\omega_{k-j}^{(\alpha)} - \omega_{k-j+1}^{(\alpha)}\right)\varepsilon_l^j - \omega_k^{(\alpha)}\varepsilon_l^0\right]\right| \\ &= \left|\left[1-\omega_2^{(\alpha)}\right]\varepsilon_l^{k-1} + \left[\sum_{j=1}^{k-2}\left(\omega_{k-j}^{(\alpha)} - \omega_{k-j+1}^{(\alpha)}\right)\varepsilon_l^j + \omega_k^{(\alpha)}\varepsilon_l^0\right]\right| \\ &\le \left[1-\omega_2^{(\alpha)}\right]\left\|E^0\right\|_\infty + \left[\left(\omega_2^{(\alpha)} - \omega_k^{(\alpha)}\right) + \omega_k^{(\alpha)}\right]\left\|E^0\right\|_\infty \\ &\le \left\|E^0\right\|_\infty\end{aligned} \tag{23}$$

i.e., $\left\|E^{k}\right\|_{\infty} \leq \left\|E^{0}\right\|_{\infty}$.

Unlike the explicit scheme, they can be fully adapted to the desired accuracy without the need to choose small time steps for stability reasons.

3.3. Convergence analysis of implicit difference scheme

Let $u(x_i,t_k)$, ($i=1,2,\cdots,M-1$; $k=1,2,\cdots,N$) be the exact solution of Eq.(8) at mesh point (x_i,t_k), and u_i^k be the numerical approximation to $u(x_i,t_k)$. Let $e_i^k = u(x_i,t_k)-u_i^k$, $i=1,2,\cdots,M-1$; $k=1,2,\cdots,N$ and $\xi^k = \begin{pmatrix} e_1^k & e_2^k & \cdots & e_{M-1}^k \end{pmatrix}^T$. Using $\xi^0 = 0$, substitution into Eq.(17) and (18) leads to

$$-\beta_i^0 e_{i-1}^1 + \left(1+2\beta_i^0\right)e_i^1 - \beta_i^0 e_{i+1}^1 = R_i^1, \tag{24}$$

$$-\beta_i^{k-1} e_{i-1}^k + \left(1+2\beta_i^{k-1}\right)e_i^k - \beta_i^{k-1} e_{i+1}^k = e_i^{k-1} - \sum_{j=2}^{k} \omega_j^{(\alpha)} \left(e_i^{k-j+1} - e_i^{k-j}\right) + R_i^k . \tag{25}$$

Since

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{\left[\left(u(x_i,t_k)-u(x_i,t_{k-1})\right) + \sum_{j=2}^{k}\left(b_j^{(\alpha)} - b_{j-1}^{(\alpha)}\right)\left(u(x_i,t_{k-j+1})-u(x_i,t_{k-j})\right)\right]}{\tau^\alpha \Gamma(2-\alpha)} + O\left(\tau\right)$$

and

$$g(x_i,t_{k-1})\frac{\partial^2 u(x_i,t_k)}{\partial x^2} = g(x_i,t_{k-1})\frac{u(x_{i-1},t_k)-2u(x_i,t_k)+u(x_{i+1},t_k)}{h^2} + O\left(h^2\right).$$

Hence,

$$R_i^k = \tau^\alpha \Gamma(2-\alpha)\left[\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} - g(x_i,t_{k-1})\frac{\partial^2 u(x_i,t_k)}{\partial x^2}\right] + C_1\left(\tau^{\alpha+1}\right) + C_2\left(\tau^\alpha h^2\right),$$

So,

$$\left|R_i^k\right| \leq \frac{\widehat{C}}{\omega_k^{(\alpha)}}\left(\tau^{\alpha+1} + \tau^\alpha h^2\right), i=1,2,\cdots,M-1; k=1,2,\cdots,N .$$

where $\widehat{C}$ is a non-negative constant.

For $k=1$, let $\left\|\xi^1\right\|_\infty = \left|e_l^1\right| = \max_{1\le i\le M-1}\left|e_i^1\right|$, we have

$$\begin{aligned}\left\|\xi^1\right\|_\infty &= \left|e_l^1\right| \le -\beta_l^0\left|e_{l-1}^1\right| + \left(1+2\beta_l^0\right)\left|e_l^1\right| - \beta_l^0\left|e_{l+1}^1\right| \\ &\le \left|-\beta_l^0 e_{l-1}^1 + \left(1+2\beta_l^0\right)e_l^1 - \beta_l^0 e_{l+1}^1\right| \\ &= \left|R_l^1\right| \le C\left(\tau^{1+\alpha}+\tau^\alpha h^2\right) \\ &= \frac{1}{\omega_1^{(\alpha)}}\widehat{C}\left(\tau^{1+\alpha}+\tau^\alpha h^2\right)\end{aligned} \tag{26}$$

Assuming $\left\|\xi^m\right\|_\infty \le \frac{1}{\omega_k^{(\alpha)}}\widehat{C}\left(\tau^{1+\alpha}+\tau^\alpha h^2\right)$, $m=2,3,\cdots,k-1$, and $\left\|\xi^k\right\|_\infty = \left|e_l^k\right| = \max_{1\le i\le M-1}\left|\varepsilon_i^k\right|$,

Then we have

$$\begin{aligned}\left\|\xi^k\right\|_\infty &= \left|e_l^k\right| \\ &\le -\beta_l^{k-1}\left|e_{l-1}^k\right| + \left(1+2\beta_l^{k-1}\right)\left|e_l^k\right| - \beta_l^{k-1}\left|e_{l+1}^k\right| \\ &\le \left|-\beta_l^{k-1}e_{l-1}^k + \left(1+2\beta_l^{k-1}\right)e_l^k - \beta_l^{k-1}e_{l+1}^k\right| \\ &= \left|e_l^{k-1} - \sum_{j=2}^{k}\omega_j^{(\alpha)}\left(e_l^{k-j+1}-e_l^{k-j}\right) + R_l^k\right| \\ &= \left|\left(1-\omega_2^{(\alpha)}\right)e_l^{k-1} + \left[\sum_{j=1}^{k-2}\left(\omega_{k-j}^{(\alpha)}-\omega_{k-j+1}^{(\alpha)}\right)e_l^j + R_l^k\right]\right| \\ &\le \left(1-\omega_2^{(\alpha)}\right)\left\|\xi^{k-1}\right\|_\infty + \sum_{j=1}^{k-2}\left(\omega_{k-j}^{(\alpha)}-\omega_{k-j+1}^{(\alpha)}\right)\left\|\xi^j\right\|_\infty + \left|R_l^k\right| \\ &\le \left(1-\omega_2^{(\alpha)}\right)\left\|\xi^{k-1}\right\|_\infty + \sum_{j=1}^{k-2}\left(\omega_{k-j}^{(\alpha)}-\omega_{k-j+1}^{(\alpha)}\right)\left\|\xi^j\right\|_\infty + \frac{\widehat{C}}{\omega_k^{(\alpha)}}\left(\tau^{1+\alpha}+\tau^\alpha h^2\right) \\ &\le \left[\left(1-\omega_2^{(\alpha)}\right) + \left(\omega_2^{(\alpha)}-\omega_k^{(\alpha)}\right) + \omega_k^{(\alpha)}\right]\frac{\widehat{C}}{\omega_k^{(\alpha)}}\left(\tau^{1+\alpha}+\tau^\alpha h^2\right) \\ &= \frac{1}{\omega_k^{(\alpha)}}\widehat{C}\left(\tau^{1+\alpha}+\tau^\alpha h^2\right)\end{aligned} \tag{27}$$

Since $\lim_{k\to\infty}\frac{1}{k^\alpha\omega_k^\alpha} = \frac{1}{1-\alpha}$, and $\tau k \le T$ is finite, hence there is a constant C, such as

$\left|\xi^k\right|_\infty \le \frac{\widehat{C}k^\alpha}{\omega_k^{(\alpha)}}\left(\tau^{\alpha+1}+\tau^\alpha h^2\right) \le C\left(\tau+h^2\right)$, so $\left|u(x_i,t_k)-u_i^k\right| \le C\left(\tau+h^2\right)$.

4. Data Acquisition

Under normal circumstances, the spectral peaks can be modelled by Gaussian peaks, Lorenz peaks or their combination.

As tested data, the Gaussian peaks is generated by

$$G_s(x) = \sum_{i=1}^{n} A_i \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2} \right], \tag{28}$$

and the Lorenzian peaks is generated by the following formula:

$$Lz(t) = \sum_{i=1}^{2} \frac{A_i}{1+\left(\dfrac{x-\mu_i}{\sigma_i}\right)^2} \tag{29}$$

where n is the number of peaks, A_i, μ_i, and σ_i the height, position and width of peak i, respectively. The white noise is generated by "awgn" function in matlab.

An NMR spectrum, which is part of the Wavelab toolbox (http://statweb.stanford.edu/~wavelab/), was used to compare the time-fractional order diffusion filtering to the other smoothing methods.

5. Results and Discussion

5.1 Assessment criteria

The signal to noise ratio (SNR) is use to assess the smoothing performance. SNR is computed by the following formula:

$$SNR = 10\lg \frac{\sum_n s^2(n)}{\sum_n |f(n) - s(n)|^2} , \tag{30}$$

where $s(n)$ is a genuine signal (the one with no noise added) and $f(n)$ is a noisy signal. The noisy signal is passed to the smoothing algorithm, and the smoothed signal compared to the genuine signal by calculating the SNR between the two signals.

5.2 Comparison between time-fractional order and classical diffusion filtering

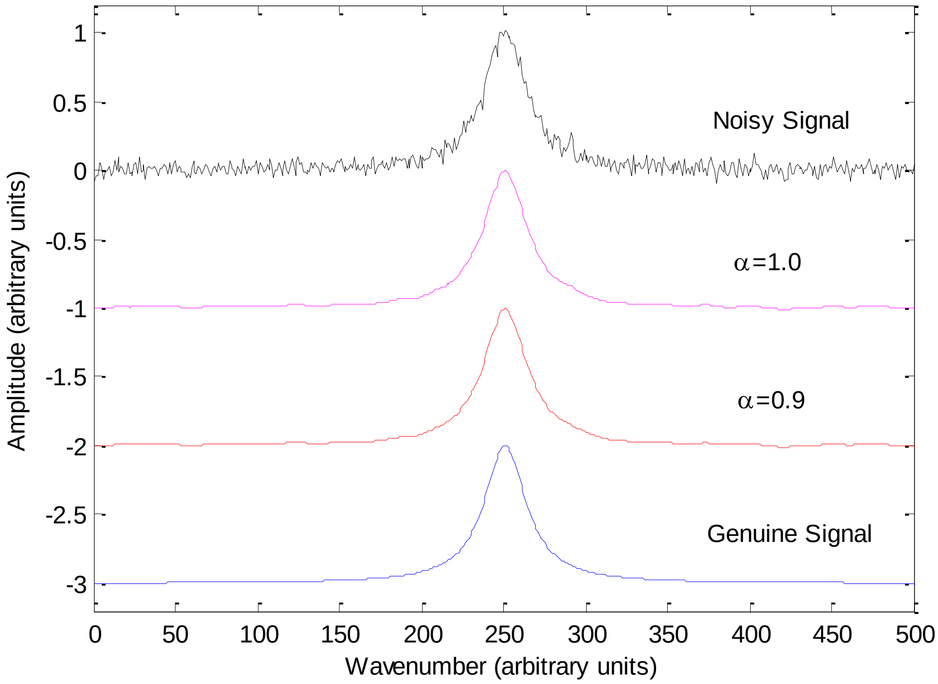
As an evaluation of time fractional order and classical diffusion filtering, four noisy signals are generated to compare the time fractional order diffusion filtering with the classical diffusion filtering ($\alpha = 1$) and their SNR are set to 15dB. One hand we compare their smoothing performance, we use the SNR to assess it. The other hand we compare their CPU time which is time-consumed in execution of the algorithm. The noisy signals and their smoothed signals are shown in figure 1. The CPU time and SNR are present in Table 1. In this experiment, the time fractional order derivative $\alpha = 0.9$, the iteration times are set to 80, time step size is 0.25 and parameter $\lambda = 0.25$. For classical diffusion filtering, the iteration times are set to 80, time step size is 0.25 and parameter $\lambda = 0.3$.

Table 1 Comparison of time fractional order diffusion filtering for $\alpha = 0.9$ and 1.0 in the case of 15dB

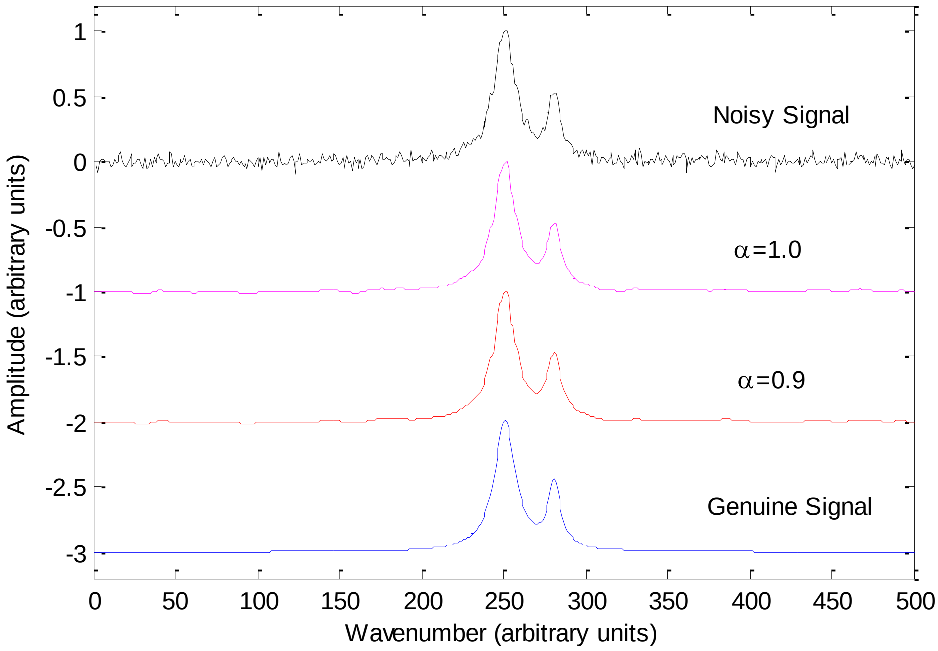
	Signal 1		Signal 2		Signal 3		Signal 4	
α	SNR	CPU time	SNR	CPU time	SNR	CPU time	SNR	CPU time

0.9	28.2339	3.388571	24.0287	3.696531	24.0666	3.410147	23.0530	3.482364
1.0	28.2176	3.206549	23.7776	3.391068	23.0910	3.199753	22.2486	3.380023

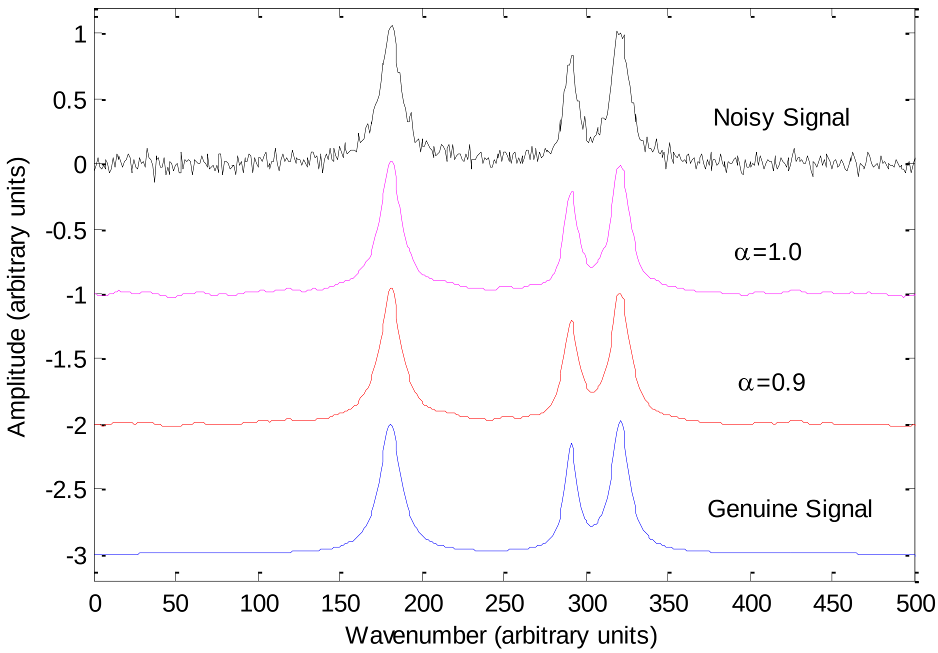
From Table 1, one can see that time-consuming of the time fractional order diffusion model is slightly more than that of the classical diffusion model. But smoothing performance of the time fractional order diffusion model is better than that of the classical diffusion model.



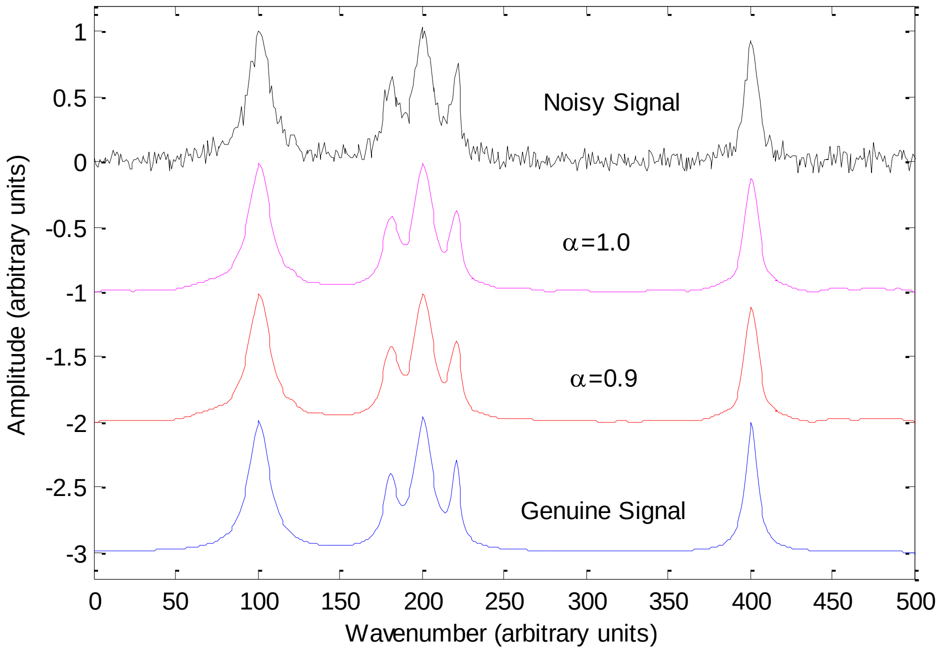
(a) Signal 1, the smoothed signals and the genuine signal have been shifted vertically



(b) Signal 2, the smoothed signals and the genuine signal have been shifted vertically



(c) Signal 3, the smoothed signals and the genuine signal have been shifted vertically



(d) Signal 4, the smoothed signals and the genuine signal have been shifted vertically

Figure 1 Comparison of time fractional diffusion model and the classical diffusion model, in the case of the SNR 15dB.

5.3 The optimal time fractional derivative order

In order to find the best time fractional derivative order α , the relationship between the SNR and the time fractional derivative order α is shown in figure 2. Usually, one can take 0.95 as the best time fractional diffusion derivative order to smooth the noisy signal.

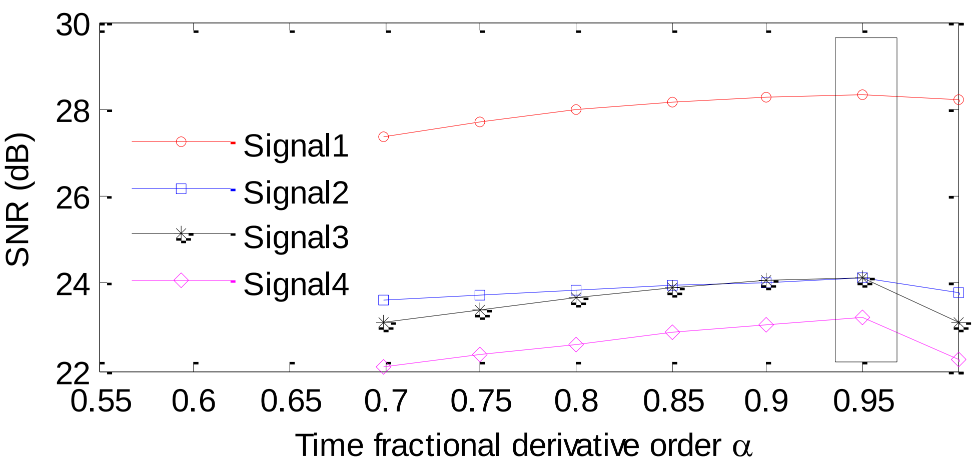


Figure 2 The relationship between the SNR and the time fractional derivative order α for 4 signals mentioned above.

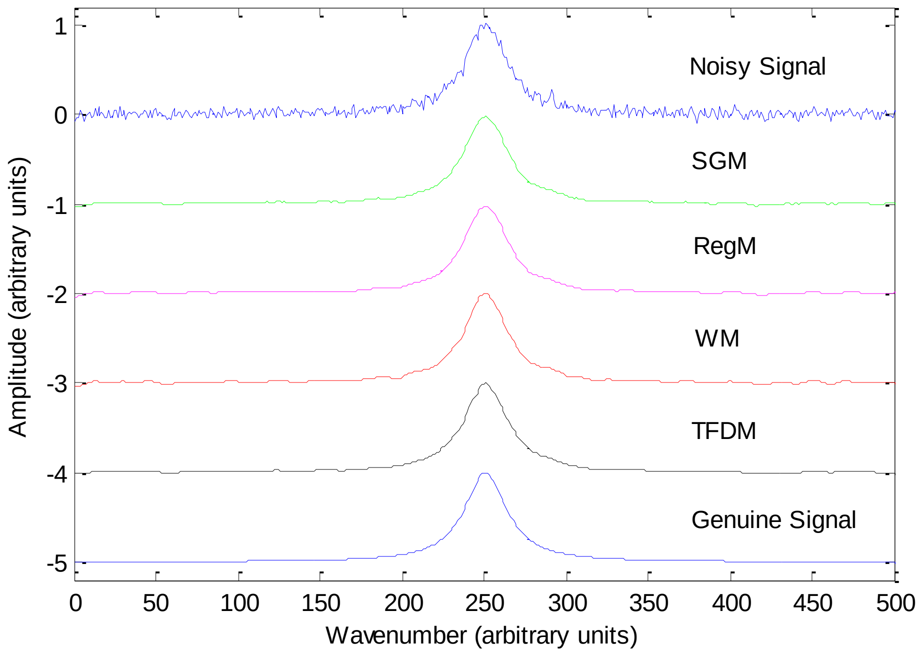
5.4 Smoothing performance comparison for different methods

Classical smoothing methods such as regularization method (RegM) [10, 11], Savitzky-Golay method (SGM) [8], Wavelet method (WM) [6, 15] and sliding average method (SAM) are performed as the comparisons. For the Savitzky-Golay method, runs are taken with sliding window sizes ranging from 3 to 31 points, and the window size that maximized the SNR is selected. In the case of the Savitzky–Golay method, degree of the polynomial is 2.

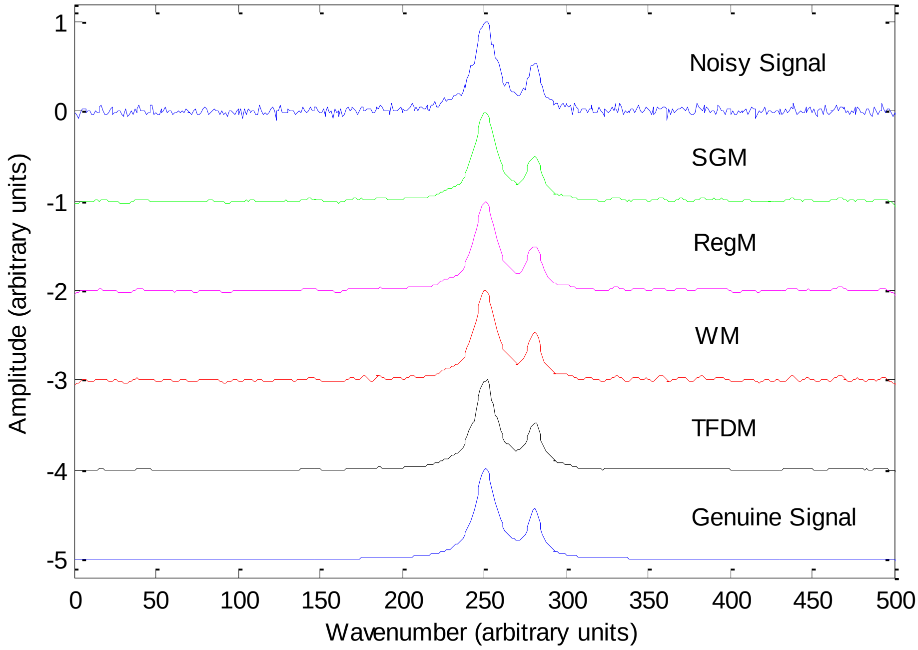
For wavelet method, sym14 wavelet is selected as the optimized wavelet. The decomposition level is also compared as an important factor to optimize smoothing efficiency of sym14 from level 1–5. The optimal level is selected.

For the regularization method, we directly use the codes given by ref. [11] to obtain the optimal smoothing result.

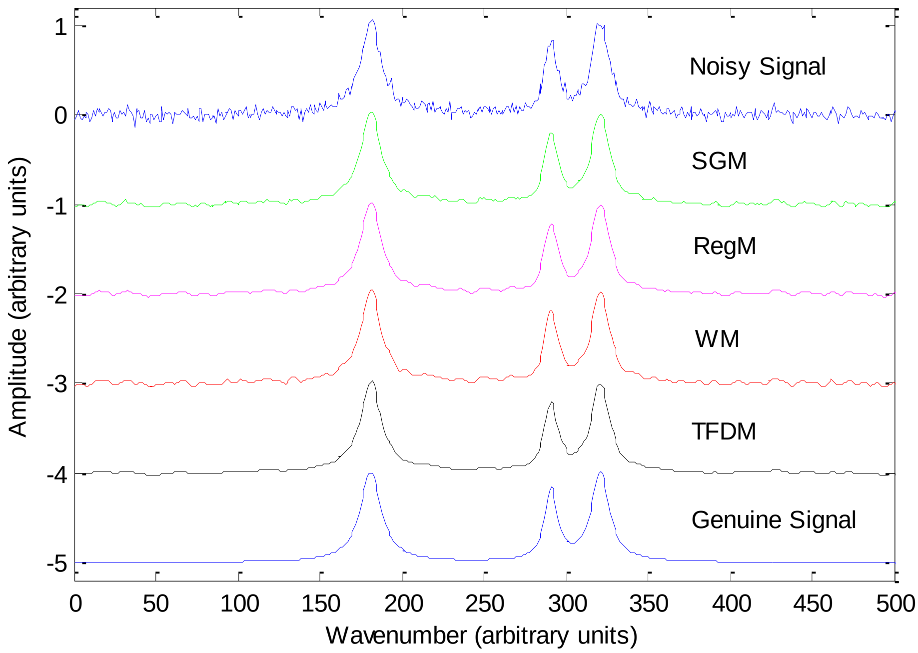
For the time fractional diffusion method, the time fractional order derivative $\alpha = 0.95$, the iteration times are set to 80, time step size is 0.25 and parameter $\lambda = 0.25$. The SNR of these smoothed signals are shown Table 2. Comparison for different smoothing methods is shown in figure 4.



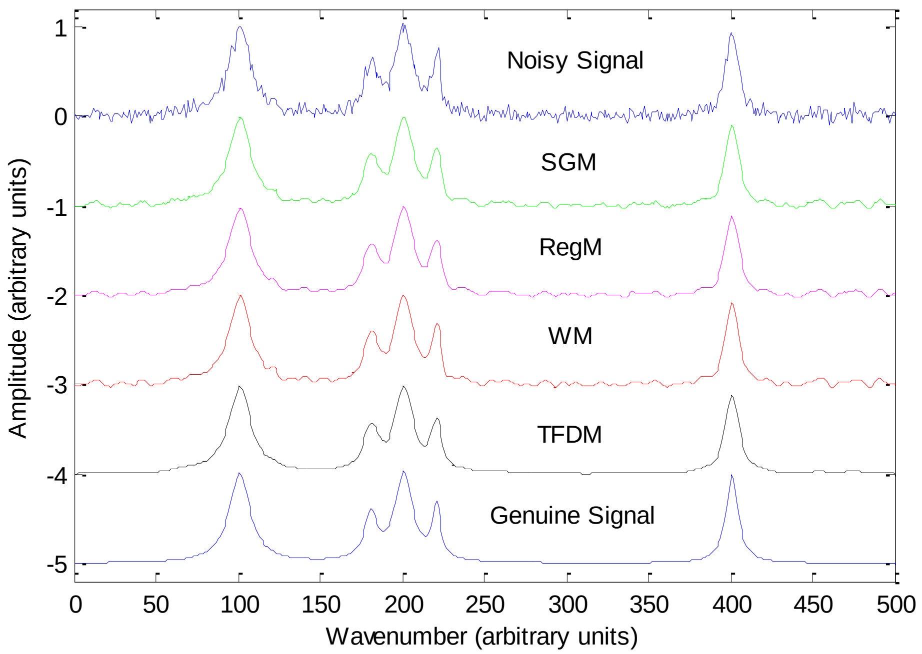
(a) Signal 1, the smoothed signals and the genuine signal have been shifted vertically



(b) Signal 2, the smoothed signals and the genuine signal have been shifted vertically



(c) Signal 3, the smoothed signals and the genuine signal have been shifted vertically



(d) Signal 4, the smoothed signals and the genuine signal have been shifted vertically

Figure 3 Comparison for different smoothing methods in the case of 15dB

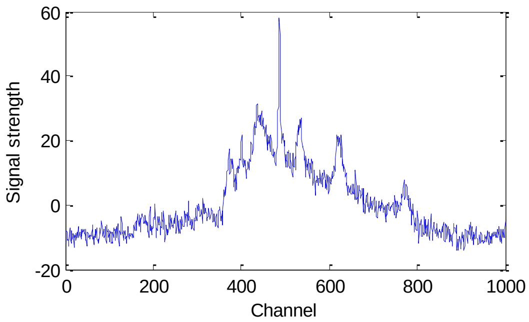
Table 2. SNR improvement with different smoothing method

Signal	TFDM	CDM	RegM	SGM	WM
1	28.2339	28.2476	27.0833	27.0100	25.4781
2	24.0287	23.7776	21.8826	21.8586	20.5150
3	24.0666	23.0910	22.1014	21.5116	20.8083
4	23.0530	22.2486	21.1560	20.9873	20.9227

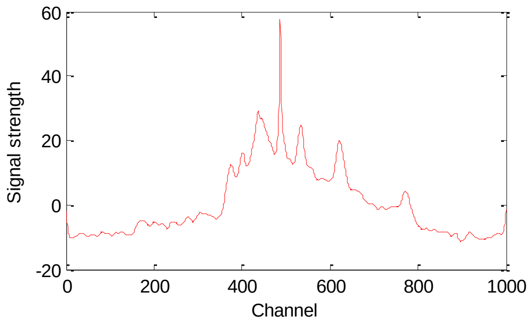
Form above results, one can the time fractional diffusion order model is an excellent method for signal smoothing. The smoothing performance of the time fractional order diffusion model is better than that of RegM, SGM, and WM.

5.5 Smoothing of a NMR spectrum

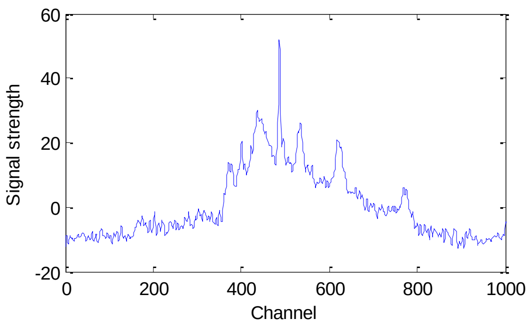
In the end we use the time fractional order diffusion filtering to smooth a NMR spectrum of a wood. Comparisons are also performed with different smoothing methods in common use, such as Savitzky–Golay method [8], regularization method [10], wavelet method [15]. The results are shown in Fig. 7. The number of points in the spectra was set to 1024. In regularization method, we directly use the codes given by ref. [11] to obtain the optimal smoothing result. In the Savitzky-Golay method, the width of the sliding window is 21 points and polynomial degree is 2. In wavelet method, the function "wden" in matlab toolbox was used to smooth the NMR, the wavelet is Sym10 and decomposed level is 5. In the time fractional order diffusion filtering ($\alpha = 0.95$), the iteration times are set to 30, time step size was 1/4 and parameter λ was set to 0.8. The Savitzky-Golay smooth is less effective at reducing noise, but more effective at retaining the shape of the original signal. The regularization method, as an improvement of the Savitzky-Golay smooth method, has the same advantage and disadvantage as the Savitzky-Golay smooth method. However, one can see that the time fractional order diffusion not only reduces the noise but also keeps the shape of original signal.



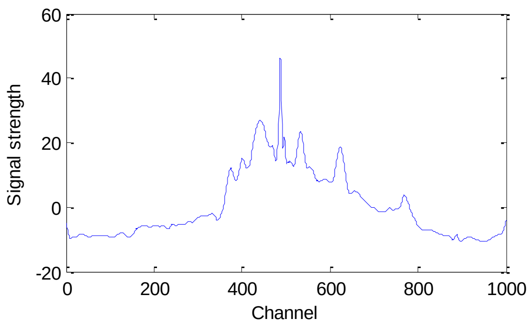
(a) Original NMR signal



(b) Time-fractional diffusion-wave mothod



(c) Regularization method



(d) Wavelet method

Fig.4. Comparison of different smoothing methods: an NMR spectrum of a wood being smoothed by the Savitzky–Golay method, regularization method, wavelet method and time fractional order diffusion filtering.

6. Conclusion

Based on time-fractional order diffusion equation, a novel smoothing method was proposed. Detail implement method include explicit difference scheme and implicit difference scheme were given. Effectiveness of time fractional order diffusion was verified by some simulated signals and an NMR spectrum. Results indicated that time fractional order diffusion filtering is an excellent smoothing method. It can not only reduce noise but also preserve the peak shape.

Acknowledgements

The work was partly supported by outstanding innovation team project of universities in jiangsu province and National Natural Science Foundation of China (Grant: 61271395).

Reference:

[1] Lu S, Pereverzev S. Numerical differentiation from a viewpoint of regularization theory. Mathematics of Computation. 2006;75:1853-70.

[2] Lubansky A, Yeow YL, Leong YK, Wickramasinghe SR, Han B. A general method of computing the derivative of experimental data. AIChE journal. 2006;52:323-32.

[3] Czarnecki MA. Resolution Enhancement in Second-Derivative Spectra. Applied spectroscopy. 2015;69:67-74.

[4] Kharintsev S, Kamalova D, Salakhov MK, Sevastianov A. Resolution enhancement of composite spectra using wavelet-based derivative spectrometry. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy. 2005;61:149-56.

[5] Leung AK-m, Chau F-t, Gao J-b. Wavelet transform: a method for derivative calculation in analytical chemistry. Analytical chemistry. 1998;70:5222-9.

[6] Liu BF, Sera Y, Matsubara N, Otsuka K, Terabe S. Signal denoising and baseline correction by discrete wavelet transform for microchip capillary electrophoresis. Electrophoresis. 2003;24:3260-5.

[7] Rowlands CJ, Elliott SR. Denoising of spectra with no user input: a spline-smoothing algorithm. Journal of Raman Spectroscopy. 2011;42:370-6.

[8] Savitzky A, Golay MJ. Smoothing and differentiation of data by simplified least squares procedures. Analytical chemistry. 1964;36:1627-39.

[9] Shao X, Ma C. A general approach to derivative calculation using wavelet transform. Chemometrics and Intelligent Laboratory Systems. 2003;69:157-65.
[10] Stickel JJ. Data smoothing and numerical differentiation by a regularization method. Computers & chemical engineering. 2010;34:467-75.
[11] Eilers PH. A perfect smoother. Analytical chemistry. 2003;75:3631-6.
[12] Zimmermann B, Kohler A. Optimizing Savitzky-Golay parameters for improving spectral resolution and quantification in infrared spectroscopy. Applied spectroscopy. 2013;67:892-902.
[13] Coombes KR, Tsavachidis S, Morris JS, Baggerly KA, Kuerer HM. Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform. Proteomics. 2005;5:4107-17.
[14] Lang M, Guo H, Odegard JE, Burrus CS, Wells Jr R. Noise reduction using an undecimated discrete wavelet transform. Signal Processing Letters, IEEE. 1996;3:10-2.
[15] Alsberg BK, Woodward AM, Winson MK, Rowland J, Kell DB. Wavelet denoising of infrared spectra. Analyst. 1997;122:645-52.
[16] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 1990;12:629-39.
[17] You Y-L, Kaveh M. Fourth-order partial differential equations for noise removal. Image Processing, IEEE Transactions on. 2000;9:1723-30.
[18] Catté F, Lions P-L, Morel J-M, Coll T. Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journal on Numerical analysis. 1992;29:182-93.
[19] Chan T, Marquina A, Mulet P. High-order total variation-based image restoration. SIAM Journal on Scientific Computing. 2000;22:503-16.
[20] Wei GW. Generalized Perona-Malik equation for image restoration. Signal Processing Letters, IEEE. 1999;6:165-7.
[21] Wang Y, Chen W, Zhou S, Yu T, Zhang Y. MTV: modified total variation model for image noise removal. Electronics Letters. 2011;47:592-4.
[22] Hajiaboli MR. An anisotropic fourth-order partial differential equation for noise removal. Scale space and variational methods in computer vision: Springer; 2009. p. 356-67.
[23] Barbu T. Robust Anisotropic Diffusion Scheme for Image Noise Removal. Procedia Computer Science. 2014;35:522-30.
[24] Bai J, Feng X-C. Fractional-order anisotropic diffusion for image denoising. Image Processing, IEEE Transactions on. 2007;16:2492-502.
[25] Tian D, Zhang X, Fan L. A Fractional-Order Level Set Model for Image Segmentation. International Journal of Digital Content Technology and its Applications. 2013;7:622.

[26] Chan R, Lanza A, Morigi S, Sgallari F. An adaptive strategy for the restoration of textured images using fractional order regularization. Numerical Mathematics: Theory, Methods and Applications. 2013;6:276-96.
[27] Chen D, Chen Y, Xue D. Three fractional-order TV-L2 models for image denoising. Journal of Computational Information Systems. 2013;9:4773-80.
[28] Shen S, Liu F, Anh V, Turner I. Detailed analysis of a conservative difference approximation for the time fractional diffusion equation. Journal of Applied Mathematics and Computing. 2006;22:1-19.
[29] Zhang Y-n, Sun Z-z, Liao H-l. Finite difference methods for the time fractional diffusion equation on non-uniform meshes. Journal of Computational Physics. 2014;265:195-210.
[30] Murio DA. Implicit finite difference approximation for time fractional diffusion equations. Computers & Mathematics with Applications. 2008;56:1138-45.
[31] Zhuang P, Liu F. Implicit difference approximation for the time fractional diffusion equation. Journal of Applied Mathematics and Computing. 2006;22:87-99.
[32] Yuste SB, Quintana-Murillo J. Fast, accurate and robust adaptive finite difference methods for fractional diffusion equations. Numerical Algorithms. 2015:1-22.
[33] Zhuang P, Liu F, Anh V, Turner I. New solution and analytical techniques of the implicit numerical method for the anomalous subdiffusion equation. SIAM Journal on Numerical analysis. 2008;46:1079-95.
[34] Esmaeili S, Garrappa R. A pseudo-spectral scheme for the approximate solution of a time-fractional diffusion equation. International Journal of Computer Mathematics. 2015;92:980-94.
[35] Lin Y, Xu C. Finite difference/spectral approximations for the time-fractional diffusion equation. Journal of Computational Physics. 2007;225:1533-52.
[36] Li X, Xu C. A space-time spectral method for the time fractional diffusion equation. SIAM Journal on Numerical analysis. 2009;47:2108-31.
[37] Witkin AP. Scale-space filtering: A new approach to multi-scale description. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84: IEEE; 1984. p. 150-3.
[38] Alikhanov AA. A new difference scheme for the time fractional diffusion equation. Journal of Computational Physics. 2015;280:424-38.
[39] Sweilam N, Khader M, Mahdy A. Crank-Nicolson finite difference method for solving time-fractional diffusion equation. Journal of Fractional Calculus and Applications. 2012;2:1-9.