RAS Techniques and Instruments

RASTAI **2**, 307–323 (2023) https://doi.org/10.1093/rasti/rzad020
Advance Access publication 2023 June 13

Comparisons between fast algorithms for the continuous wavelet transform and applications in cosmology: the 1D case

Yun Wang[1] and Ping He[1,2★]

[1]*College of Physics, Jilin University, Changchun 130012, P. R. China*
[2]*Center for High Energy Physics, Peking University, Beijing 100871, P. R. China*

Accepted 2023 June 6. Received 2023 May 15; in original form 2023 February 9

ABSTRACT

The continuous wavelet transform (CWT) is very useful for processing signals with intricate and irregular structures in astrophysics and cosmology. It is crucial to propose precise and fast algorithms for the CWT. In this work, we review and compare four different fast CWT algorithms for the 1D signals, including the FFTCWT, the V97CWT, the M02CWT, and the A19CWT. The FFTCWT algorithm implements the CWT using the Fast Fourier Transform (FFT) with a computational complexity of $\mathcal{O}(N\log_2 N)$ per scale. The rest algorithms achieve the complexity of $\mathcal{O}(N)$ per scale by simplifying the CWT into some smaller convolutions. We illustrate explicitly how to set the parameters as well as the boundary conditions for them. To examine the actual performance of these algorithms, we use them to perform the CWT of signals with different wavelets. From the aspect of accuracy, we find that the FFTCWT is the most accurate algorithm, though its accuracy degrades a lot when processing the non-periodic signal with zero boundaries. The accuracy of $\mathcal{O}(N)$ algorithms is robust to signals with different boundaries, and the M02CWT is more accurate than the V97CWT and A19CWT. From the aspect of speed, the $\mathcal{O}(N)$ algorithms do not show an overall speed superiority over the FFTCWT at sampling numbers of $N \lesssim 10^6$, which is due to their large leading constants. Only the speed of the V97CWT with real wavelets is comparable to that of the FFTCWT. However, both the FFTCWT and V97CWT are substantially less efficient in processing the non-periodic signal because of zero padding. Finally, we conduct wavelet analysis of the 1D density fields, which demonstrate the convenience and power of techniques based on the CWT. We publicly release our CWT codes as resources for the community.

Key words: Numerical methods – Algorithms – Wavelet analysis – Cosmology.

1 INTRODUCTION

Wavelets are wave-like functions that are localized in both the real and Fourier domains. Hence, by convolving a signal under investigation with the dilated or contracted wavelets, the local features at various scales will be extracted, and this process is called wavelet transform (WT; e.g. Daubechies 1992; Kaiser & Hudgins 1994; Addison 2017). There are two basic types of WT: discrete WT (DWT) and continuous WT (CWT). The DWT, using orthogonal wavelets, operates over coarse dyadic scales and positions. In contrast to the DWT, the CWT offers a highly redundant representation of the signal, which ensures that intricate structures or textures can be resolved quite well (Addison 2018). However, the redundancy also makes the direct computation of the CWT terribly inefficient, which requires a time complexity of $\mathcal{O}(N^2)$ per scale, where N is the number of data points. One more efficient way to implement CWT is to use the Fast Fourier Transform (FFT) with a complexity $\mathcal{O}(N\log_2 N)$, since the convolution in the real domain is equivalent to multiplication in the frequency domain, i.e. convolution theorem (Torrence & Compo 1998; Pérez-Rendón & Robles 2004; Press et al. 2007; Arts et al. 2022).

Consequently, the CWT is becoming increasingly popular in many areas of science and engineering. In the context of astrophysics and cosmology, the CWT has been used for various studies including but not limited to, identifying structures and substructures from the galaxy catalogue (e.g. Slezak et al. 1990, 1993; Escalera & Mazure 1992; Escalera et al. 1994; Flin & Krywult 2006; Schwinn et al. 2018), analysing the fractal properties of the galaxy distribution (e.g. Martínez et al. 1993; Rozgacheva et al. 2012), analysing galactic images (e.g. Frick et al. 2001, 2016; Tabatabaei et al. 2013; Robitaille et al. 2014; Arshakian & Ossenkopf 2016), detecting baryon acoustic oscillation features (e.g. Tian et al. 2011; Arnalte-Mur et al. 2012; Labatie et al. 2012), investigating the turbulence in the intracluster medium (e.g. Shi et al. 2018; Roh et al. 2019) and characterizing the cosmic density fields at low redshifts (e.g. Wang & He 2022; Wang et al. 2022).

Since the COBE detection of the CMB anisotropy in 1992, cosmology has emerged as a precision, data-driven science (Turner 2022). The observational experiments such as the *Euclid* space mission (*Euclid*; Laureijs et al. 2011), the Dark Energy Spectroscopic Instrument (DESI; Levi et al. 2013), and the Square Kilometre Array (SKA; Bacon et al. 2020), and state-of-the art cosmological

★ E-mail: hep@jlu.edu.cn

simulations such as the IllustrisTNG (Pillepich et al. 2018), the SIMBA (Davé et al. 2019), and the MillenniumTNG (Hernández-Aguayo et al. 2022), are producing increasingly growing amounts of data, which need to be analysed by high-performance algorithms and methods. Therefore, the fast CWT algorithms with $\mathcal{O}(N)$ complexity are obviously more attractive than the FFT-based implementation of the CWT (FFTCWT) with $\mathcal{O}(N\log_2 N)$ complexity. Fortunately, a great effort has been made to develop fast CWT algorithms without using the FFT (e.g. Unser et al. 1994; Berkner & Wells 1997; Vrhel et al. 1997; Muñoz et al. 2002; Omachi & Omachi 2007; Arizumi & Aksenova 2019), which achieve the time complexity of $\mathcal{O}(N)$ per scale. However, some $\mathcal{O}(N)$ algorithms are only applicable to particular cases. For example, the algorithm of Unser et al. (1994) is restricted to integer scales, the algorithm of Berkner & Wells (1997) is only available for wavelets which are derivatives of the Gaussian function, and the algorithm of Omachi & Omachi (2007) is only applicable for polynomial wavelets.

What we need are fast CWT algorithms with no restrictions on the wavelet, and with arbitrarily fine scale resolution. Therefore, in this study, we will consider the $\mathcal{O}(N)$ algorithms proposed by Vrhel et al. (1997), by Muñoz et al. (2002), and by Arizumi & Aksenova (2019). For convenience, we denote these three algorithms as the V97CWT, the M02CWT, and the A19CWT, respectively. The V97CWT is a fast recursive algorithm based on the finite impulse response (FIR) and infinite impulse response (IIR) filtering techniques with filter coefficients determined by two compactly supported auxiliary functions. The M02CWT reaches the linear complexity by decomposing both the wavelet and the signal into B-splines, and the A19CWT approximates the wavelet as piecewise polynomials and reduces the number of operations using integration by parts.

Motivated by the facts that (1) all these powerful algorithms are 1D, and (2) there is no any publicly available source code for the V97CWT, M02CWT, and A19CWT algorithms, we must conduct a systematic comparison study of them to benchmark their actual performance, which is the basis for developing high-dimensional fast CWT algorithms to analyse high-dimensional data, e.g. the 2D weak-lensing maps and 3D spatial distribution of matter. For some simple 1D functions, such as sine, cosine, and Gaussian functions, their CWTs can be evaluated by analytical calculations. So the accuracy of their numerical CWTs can be verified by the corresponding analytical results. Finally, it should be noted that the CWT for 1D signals is not trivial in astrophysics and cosmology, as it is also applicable to a wide range of scenarios, such as analysing the light curves of astronomical sources (e.g. Tarnopolski et al. 2020; Ren et al. 2023), subtracting the foreground emission from the 21 cm signal (e.g. Gu et al. 2013; Li et al. 2019), measuring the small-scale structure in the Lyman-α forest (e.g. Lidz et al. 2010; Garzilli et al. 2012; Wolfson et al. 2021), investigating the time-frequency properties of the gravitational waves (e.g. Tary et al. 2018), characterizing the 1D density fields (e.g. da Cunha et al. 2018; Wang & He 2021; Wang et al. 2022), and so on. We publicly release the Fortran 95 implementations[1] and their Python WRAPPERS[2] of the fast CWT algorithms described in this paper, in the hope that the community will use them to perform wavelet analysis of 1D signals.

[1]The Fortran 95 codes are available at https://github.com/WangYun1995/FortranCWT

[2]The Python WRAPPERS are available at https://github.com/WangYun1995/pyFortranCWT

Table 1. The acronyms frequently used in the paper, with their meanings explained.

Acronym	Meaning
CWT	Continuous wavelet transform
ICWT	Inverse continuous wavelet transform
CBSW	Cubic B-spline wavelet
GDW	Gaussian-derived wavelet
CW-GDW	Cosine-weighted Gaussian-derived wavelet
MW	Morlet wavelet
FT	Fourier transform
FFT	Fast Fourier transform
FFTCWT	The fast CWT algorithm based on the FFT
V97CWT	The fast CWT algorithm of Vrhel et al. (1997)
M02CWT	The fast CWT algorithm of Muñoz et al. (2002)
A19CWT	The fast CWT algorithm of Arizumi & Aksenova (2019)

The paper is organized as follows. We briefly introduce the mathematical formalism of the CWT in Section 2. We review the fast CWT algorithms in Section 3, and compare the performance between them in Section 4. We present simple applications of the 1D CWT in cosmology in Section 5. Finally, in Section 6, we summarize our main findings and present the conclusions.

For convenience of the readers, in Table 1, we list the acronyms frequently used in our paper, with their meanings explained.

2 THE FORMALISM OF THE CONTINUOUS WAVELET TRANSFORM

The CWT $W_f(w, x)$ of a 1D real signal $f(x)$ is defined as the convolution of $f(x)$ with a scaled wavelet, i.e.

$$W_f(w,x)=\int_{-\infty}^{+\infty} f(u)\psi(w,x-u)\mathrm{d}u, \tag{1}$$

where w is the scale parameter with dimension of $[x]^{-1}$, and

$$\psi(w,x)=\sqrt{w}\psi(wx) \tag{2}$$

is the scaled version of the mother wavelet

$$\psi(x)=\psi(1,x). \tag{3}$$

There are many different choices for the mother wavelet. In this study, we consider four kinds of wavelets: the cubic B-spline wavelet (CBSW; Muñoz et al. 2002), the Gaussian-derived wavelet (GDW; Wang & He 2021), the cosine-weighted Gaussian-derived wavelet (CW-GDW; Wang & He 2022), and the Morlet wavelet (MW; Addison 2017). Table 2 shows their formulas and properties, and Fig. 1 gives a graphical representation.

As well known, the classical inverse CWT (ICWT) formula is a double integral over scale and space (see e.g. Addison 2017). In fact, there are simpler inverse ways. If the complex wavelet satisfies $\hat{\psi}(k)=0$ for $k<0$ and $0<|\mathcal{K}_\psi|<\infty$, where $\mathcal{K}_\psi=\int_0^{+\infty}\frac{\hat{\psi}(k)}{k}\mathrm{d}k$, then the original signal can be reconstructed by the known Morlet formula (see e.g. Shensa 1993; Daubechies et al. 2011) as follows

$$f(x)=\bar{f}+2\mathrm{Re}\left\{\frac{1}{\mathcal{K}_\psi}\int_0^{+\infty}\frac{W_f(w,x)}{\sqrt{w}}\mathrm{d}w\right\}, \tag{4}$$

where Re{…} denotes the real part, and $\bar{f}=\lim_{L\to\infty}\frac{1}{L}\int_{-L/2}^{L/2}f(x)\mathrm{d}x$ is the average value of $f(x)$ over all space. If the real wavelet satisfies $\psi(x)=\psi(-x)$ and $0<|\mathcal{K}_\psi|<\infty$, then a single integral ICWT

Table 2. Four mother wavelet functions and their properties. $\hat{\psi}(k)$ is the FT of $\psi(x)$, C_N is the normalization constant that makes $\int_{-\infty}^{+\infty}|\psi(x)|^2\mathrm{d}x = 1$, $\mathcal{K}_\psi = \int_0^{+\infty}\frac{\hat{\psi}(k)}{k}\mathrm{d}k$ is a constant that ensures the existence of the single integral ICWT formula, χ is the half width of the wavelet's support $[-\chi, \chi]$, and $c_w = w/k_\mathrm{pseu}$ is the ratio between the wavelet scale and the corresponding pseudo Fourier frequency (see Wang et al. (2022) for the definition of c_w). Note the GDW, CW-GDW, and MW are not compactly supported, but decay exponentially. For these three wavelets, we set the values of χ to confirm $\int_{-\infty}^{+\infty}|\psi(x)|^2\mathrm{d}x \approx 1$ and $\psi(x) \approx 10^{-14}$.

	$\psi(x)$	$\hat{\psi}(k)$	C_N	$\mathcal{K}_\psi$	χ	c_w
CBSW	$C_\mathrm{N}(2\beta^3(x) - \beta^3(x+1) - \beta^3(x-1))$ [a]	$64C_\mathrm{N}\sin^6(k/2)/k^4$	$\sqrt{30/31}$	$\frac{1}{2}\sqrt{\frac{15}{62}}(27\ln 3 - 32\ln 2)$	3	0.46609 [b]
GDW	$C_\mathrm{N}(2-x^2)e^{-\frac{x^2}{4}}$	$8\sqrt{\pi}C_\mathrm{N}k^2e^{-k^2}$	$1/(18\pi)^{1/4}$	$2(8\pi/9)^{1/4}$	12	$2/\sqrt{5}$
CW-GDW	$C_\mathrm{N}\left((1-x^2)\cos x - x\sin x\right)e^{\frac{1-x^2}{2}}$	$\sqrt{2\pi}C_\mathrm{N}k(k\cosh k - \sinh k)e^{-\frac{k^2}{2}}$	$\sqrt{\frac{8}{1+5e}}/\pi^{1/4}$	$4\pi^{1/4}/\sqrt{1+5e}$	8	0.42822 [c]
MW	$C_\mathrm{N}(e^{-4\mathrm{i}x} - e^{-8})e^{-\frac{x^2}{2}}$	$\sqrt{2\pi}C_\mathrm{N}e^{-8}(e^{4k}-1)e^{-\frac{k^2}{2}}$	$\frac{e^8}{\sqrt{1-2e^4+e^{16}}}/\pi^{1/4}$	1.27484 [d]	7.5	0.24264 [e]

[a] see Appendix B for the definition of B-splines.
[b] 0.466094761079290.
[c] 0.428218886729052.
[d] 1.274837568937901.
[e] 0.242640671273266.

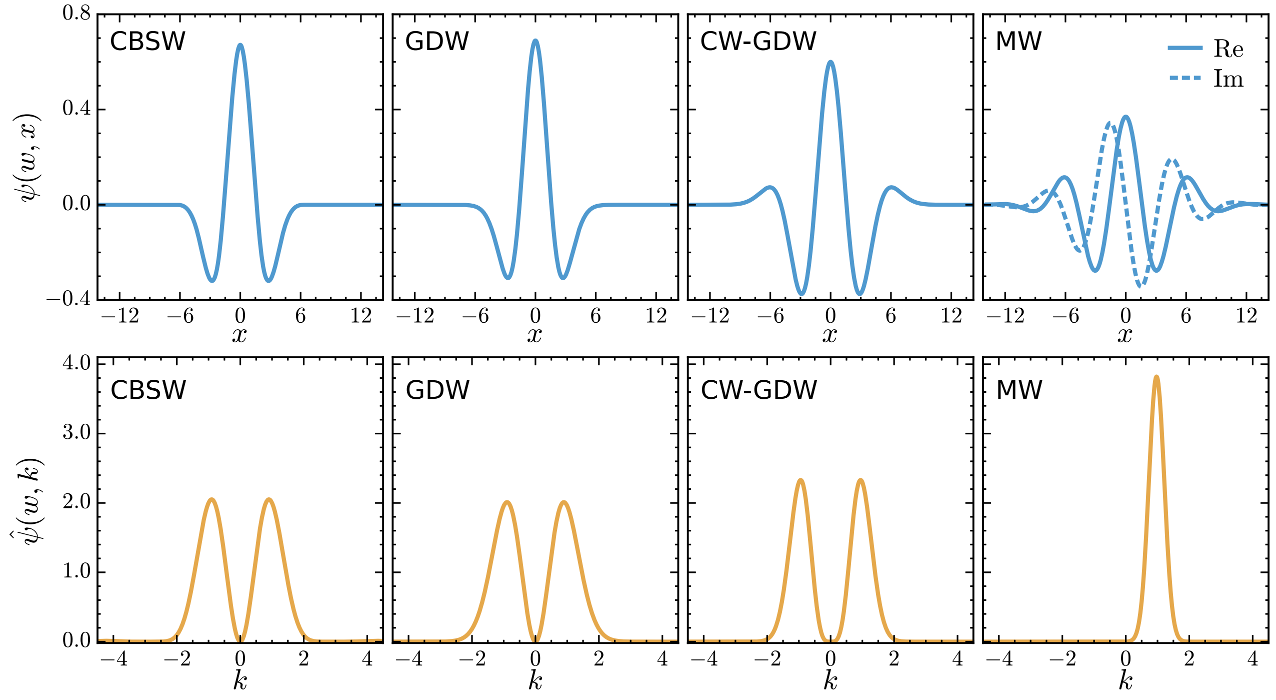


Figure 1. Top row: the CBSW, GDW, CW-GDW, and MW in the real domain, at scale $w = c_w$, i.e. $k_\mathrm{pseu} = 1$. Bottom row: the respective Fourier transforms of the wavelets in the top row.

formula also exists, which is

$$f(x) = \bar{f} + \frac{1}{\mathcal{K}_\psi}\int_0^{+\infty}\frac{W_f(w,x)}{\sqrt{w}}\mathrm{d}w. \tag{5}$$

Note that equation (5) is the generalization of the inverse formula in Wang & He (2021) and Wang et al. (2022), which holds for wavelets derived from the smoothing window function. We refer to Appendix A for the derivation of equation (5).

3 FAST ALGORITHMS FOR THE 1D CWT

For a discrete signal $f(n) \equiv f(n\Delta x)$ with sampling interval Δx, the CWT will be discretized in the following form:

$$W_f(\tilde{w}, n) = \sqrt{\tilde{w}\Delta x}\sum_m f(m)\psi[\tilde{w}(n-m)], \tag{6}$$

where $\tilde{w} = w\Delta x$ is the dimensionless scale parameter. It is clear that the computation of equation (6) requires N^2 multiplications and additions per scale, where N is the number of sampling points. Therefore, the high computational complexity makes this algorithm impractical for use. Next we will review four CWT algorithms with better performance, namely the FFTCWT, the V97CWT (Vrhel et al. 1997), the M02CWT (Muñoz et al. 2002), and the A19CWT (Arizumi & Aksenova 2019).

3.1 FFTCWT

If the discrete signal $f(n)$ is periodic with period $L = N\Delta x$, then it can be decomposed into a Fourier series as follows

$$f(n) = \frac{1}{L}\sum_m \hat{f}(m)e^{-2\pi imn/N}, \tag{7}$$

where the Fourier transform (FT) $\hat{f}(m)$ is defined as

$$\hat{f}(m) = \frac{L}{N}\sum_n f(n)e^{2\pi imn/N}. \tag{8}$$

By substituting equation (7) into equation (6), we get

$$W_f(\tilde{w}, n) = \frac{1}{L}\sum_m \hat{W}_f(\tilde{w}, m)e^{-2\pi imn/N}, \tag{9}$$

where $\hat{W}_f(\tilde{w}, m) = \sqrt{\frac{L}{N\tilde{w}}}\hat{f}(m)\hat{\psi}(\frac{2\pi m}{N\tilde{w}})$, and $\hat{\psi}(k)$ is the FT of the wavelet $\psi(x)$. Clearly, as the inverse FT of the product $\hat{W}_f(\tilde{w}, m)$, the CWT $W_f(\tilde{w}, n)$ can be computed efficiently by a standard FFT routine, like the `FFTW`[3] we used (Frigo & Johnson 2005).

Note that it is necessary to choose a set of discrete scales to use in equation (9). For the CWT, the choice of scales is arbitrary. It is convenient to discretize the scales evenly on a logarithmic scale:

$$\tilde{w} = \tilde{w}_{\min}2^{i+j/N_{\text{subs}}}, \tag{10}$$

where $\tilde{w}_{\min} = c_w\pi/N$ is the largest scale. Here, the scales are first divided into N_{levs} levels, numbered by i; then each level is divided into N_{subs} sub-levels, numbered by j. Thus, there is a total of $N_{\text{scales}} = N_{\text{levs}}N_{\text{subs}}$ scales. The number of scale levels is determined by $N_{\text{levs}} = \text{Nint}(\log_2 \frac{c_w k_{\text{Nyq}}}{\tilde{w}_{\min}/\Delta x})$, where k_{Nyq} is the Nyquist frequency and Nint(...) denotes the nearest integer, while the number of sub-levels is determined by the user to allow adjustment of the scale resolution.

For clarity, the sequence of the FFTCWT algorithm is shown as a flowchart in Fig. 2.

3.2 V97CWT

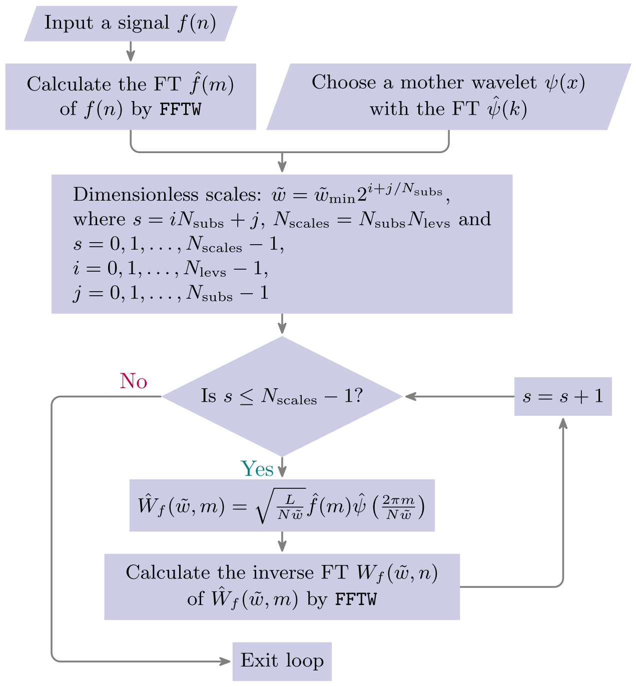
The fundamental idea of the V97CWT algorithm is to approximate the wavelet using two scaling functions, e.g. the zero order B-spline $\beta^0(x)$ and the cubic B-spline function $\beta^3(x)$ (see Appendix B for the definitions and properties of the B-splines). By using $\beta^3(x)$, the wavelet can be approximated as

$$\begin{aligned}\psi_j(\tilde{x}) = \psi\left(\frac{\tilde{w}_{\max}\tilde{x}}{2^{j/N_{\text{subs}}}}\right) &\approx (p_j * \beta^3)(\tilde{x}) \\ &= \sum_n p_j(n)\beta^3(\tilde{x} - n),\end{aligned} \tag{11}$$

where $\tilde{x} = x/\Delta x$ is the dimensionless coordinate, and $\tilde{w}_{\max}$ is the smallest scale. By convolving the above equation with $\beta^0(x)$, we get

$$p_j(n) = (q_j * q_{12})(n), \tag{12}$$

[3] https://www.fftw.org/



Input a signal $f(n)$
Calculate the FT $\hat{f}(m)$ of $f(n)$ by FFTW
Choose a mother wavelet $\psi(x)$ with the FT $\hat{\psi}(k)$
Dimensionless scales: $\tilde{w} = \tilde{w}_{\min}2^{i+j/N_{\text{subs}}}$, where $s = iN_{\text{subs}} + j$, $N_{\text{scales}} = N_{\text{subs}}N_{\text{levs}}$ and $s = 0, 1, \ldots, N_{\text{scales}} - 1$, $i = 0, 1, \ldots, N_{\text{levs}} - 1$, $j = 0, 1, \ldots, N_{\text{subs}} - 1$
No
Is $s \leq N_{\text{scales}} - 1$?
$s = s + 1$
Yes
$\hat{W}_f(\tilde{w}, m) = \sqrt{\frac{L}{N\tilde{w}}}\hat{f}(m)\hat{\psi}\left(\frac{2\pi m}{N\tilde{w}}\right)$
Calculate the inverse FT $W_f(\tilde{w}, n)$ of $\hat{W}_f(\tilde{w}, m)$ by FFTW
Exit loop

Figure 2. Schematic representation of the FFTCWT algorithm. We define the loop variable $s = iN_{\text{subs}} + j$ to merge the two nested loops (for i and j) into one single loop.

where the FIR filter $q_j(n)$ is

$$\begin{aligned}q_j(n) &= (\psi_j * \beta^0)(n) \\ &= \int_{n-1/2}^{n+1/2} \psi_j(\tilde{x})\mathrm{d}\tilde{x}, \quad n = -N_q, \ldots, -1, 0, 1, \ldots, N_q,\end{aligned} \tag{13}$$

and the IIR filter $q_{12}(n) = (\beta^4)^{-1}(n)$ is the convolution inverse of $\beta^4(n)$, i.e.

$$(q_{12} * \beta^4)(n) = \delta^K(n), \tag{14}$$

where $\delta^K(n)$ is the Kronecker delta function.

Substituting equations (11) and (12) into equation (6), we have the following equations

$$f_0(n) = (f * \beta^3)(n), \tag{15}$$

$$F_0(n) = (f_0 * q_{12})(n), \tag{16}$$

$$W_f\left(\frac{\tilde{w}_{\max}}{2^{j/N_{\text{subs}}}}, n\right) = \sqrt{\frac{\tilde{w}_{\max}\Delta x}{2^{j/N_{\text{subs}}}}}(F_0 * q_j)(n). \tag{17}$$

Exploiting the two-scale relation of the B-splines, we can obtain the CWT at scales of $\tilde{w}_{\max}/2^{i+j/N_{\text{subs}}}$ as follows

$$f_i(n) = (f_{i-1} * [h]_{\uparrow 2^{i-1}})(n), \tag{18}$$

$$F_i(n) = (f_i * [q_{12}]_{\uparrow 2^i})(n), \tag{19}$$

$$W_f\left(\frac{\tilde{w}_{\max}}{2^{i+j/N_{\text{subs}}}}, n\right) = \sqrt{\frac{\tilde{w}_{\max}\Delta x}{2^{i+j/N_{\text{subs}}}}}(F_i * [q_j]_{\uparrow 2^i})(n), \tag{20}$$

where '$[\ldots]_{\uparrow 2^i}$' denotes the insertion of $2^i - 1$ zeros between each point, and $h(n)$ is given by equation (B5). Equations (15), (17), (18), and (20) perform the FIR filtering. Equations (16) and (19) perform the IIR filtering, please refer to Appendix C for

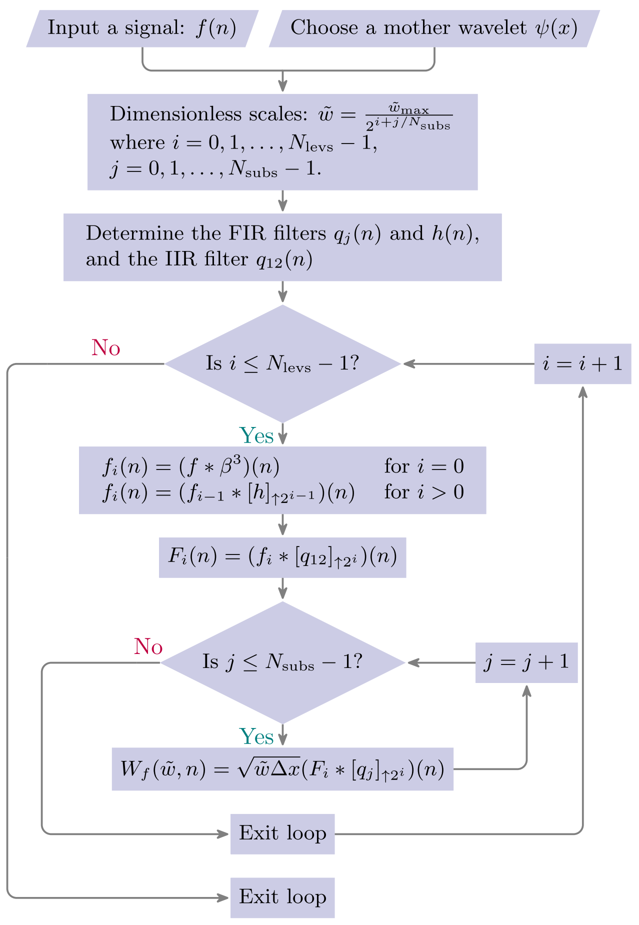

Input a signal: $f(n)$
Choose a mother wavelet $\psi(x)$
Dimensionless scales: $\tilde{w} = \frac{\tilde{w}_{\max}}{2^{i+j/N_{\text{subs}}}}$ where $i = 0, 1, \ldots, N_{\text{levs}} - 1$, $j = 0, 1, \ldots, N_{\text{subs}} - 1$.
Determine the FIR filters $q_j(n)$ and $h(n)$, and the IIR filter $q_{12}(n)$
Is $i \le N_{\text{levs}} - 1$?
No
Yes
$i = i + 1$
$f_i(n) = (f * \beta^3)(n)$ for $i = 0$
$f_i(n) = (f_{i-1} * [h]_{\uparrow 2^{i-1}})(n)$ for $i > 0$
$F_i(n) = (f_i * [q_{12}]_{\uparrow 2^i})(n)$
Is $j \le N_{\text{subs}} - 1$?
No
Yes
$j = j + 1$
$W_f(\tilde{w}, n) = \sqrt{\tilde{w}\Delta x}(F_i * [q_j]_{\uparrow 2^i})(n)$
Exit loop
Exit loop

Figure 3. Schematic representation of the V97CWT algorithm.

its details. The maximum scale level $N_{\text{levs}} - 1$ is determined by $N_{\text{levs}} = \text{Nint}(\log_2 \frac{\tilde{w}_{\max}}{c_w \pi/N})$.

The sequence of the V97CWT algorithm is shown as a flowchart in Fig. 3.

3.3 M02CWT

The M02CWT algorithm represents both the wavelet and the input signal as B-splines. The wavelet function is expressed as

$$\psi(\tilde{w}\tilde{x}) \approx \sum_{n=-N_d}^{N_d} d(n)\beta^3\left(\frac{\tilde{w}\tilde{x}}{h} - n\right), \tag{21}$$

where the parameter h is used to regulate the accuracy of the B-spline approximation. If the support of the wavelet $\psi(x)$ is $[-\chi, \chi]$, then the relation between h and N_d is $h = \chi/(N_d + 2)$. Likewise, the continuous signal $f_c(\tilde{x})$ is represented by its cubic B-spline interpolant:

$$\begin{aligned} f_c(\tilde{x}) &= (c * \beta^3)(\tilde{x}) \\ &= \sum_n c(n)\beta^3(\tilde{x} - n). \end{aligned} \tag{22}$$

For spline wavelets, e.g. the CBSW, the coefficient sequence $d(n)$ can be easily obtained from its analytic form. For general wavelet functions, the sequence $d(n)$ is calculated in the same method as $c(n)$, which are calculated by (see Appendix C)

$$c(n) = \left(f * (\beta^3)^{-1}\right)(n), \tag{23}$$

where $f(n)$ is the discrete input signal.

Substituting equations (21) and (B6) into equation (1), we get

$$W_f(\tilde{w}, \tilde{x}) = \sqrt{\tilde{w}\Delta x}\left(\frac{\tilde{w}}{h}\right)^3 \sum_{n=-N_d}^{N_d+4} b(n)v\left(\tilde{x} - \frac{nh}{\tilde{w}}\right),$$

where $b(n) = \sum_{n'=0}^{4} d(n - n')a(n')$, and $v(\tilde{x}) = D^{-4} f_c(\tilde{x} + 2h/\tilde{w})$. Then by using equations (22) and (B11), and considering that we are typically interested in the integer values of $\tilde{x}$, the above equation becomes

$$\begin{aligned} W_f(\tilde{w}, n) = \sqrt{\tilde{w}\Delta x}\left(\frac{\tilde{w}}{h}\right)^3 \sum_{n'=-N_d}^{N_d+4} \sum_{l=l_0}^{l_0+7} \\ \times b(n')g(l)\beta^7\left(n - \frac{(n'-2)h}{\tilde{w}} - l - 2\right), \end{aligned} \tag{24}$$

where l_0 is the ceiling integer of $n - (n' - 2)h/\tilde{w} - 6$, and

$$g(l) = (\Delta^{-4} * c)(l). \tag{25}$$

Notice that the computation of $g(l)$ needs to calculate cumulative sum of the sequence $c(l)$ four times, which indicates that in the case of a large amount of data, $g(l)$ becomes increasingly inaccurate as l increases due to the limited precision of floating points. The way to alleviate this issue is to divide the sequence $c(l)$ into many small segments and then compute $g(l)$ locally on each segment. To do this, we set scales as follows

$$\tilde{w} = \tilde{w}_0 2^{i+j/N_{\text{subs}}}, \tag{26}$$

where $\tilde{w}_0 = 2\chi/N$, and the scale level i takes the range of $I_{\min} = \text{Nint}(\log_2 \frac{c_w\pi/N}{\tilde{w}_0})$ to $I_{\max} = \text{Nint}(\log_2 \frac{c_w\pi/2}{\tilde{w}_0})$. In the case of $i < 1$, we do not split $c(l)$; whereas in the case of $i \ge 1$, we split $c(l)$ into 2^i parts, as shown by the flowchart in Fig. 4.

3.4 A19CWT

Since the mother wavelet is zero outside the support interval $[-\chi, \chi]$, the CWT at integer positions can be written as

$$W_f(\tilde{w}, n) = \sqrt{\frac{\Delta x}{\tilde{w}}} \int_{-\chi}^{\chi} f_c(n - \tilde{u}/\tilde{w})\psi(\tilde{u})\mathrm{d}\tilde{u}. \tag{27}$$

By partitioning the support duration $[-\chi, \chi]$ evenly into $2N_\chi$ intervals, i.e.

$$-\chi = \chi_{-N_\chi} < \ldots < \chi_{-1} < \chi_0 < \chi_1 < \ldots < \chi_{N_\chi} = \chi,$$

we approximate $\psi(\tilde{x})$ with cubic piecewise polynomials as shown below

$$\psi(\tilde{x}) \approx \begin{cases} \psi_{-N_\chi}(\tilde{x}), & \chi_{-N_\chi} \le \tilde{x} < \chi_{1-N_\chi}, \\ \vdots & \vdots \\ \psi_{n'}(\tilde{x}), & \chi_{n'} \le \tilde{x} < \chi_{n'+1}, \\ \vdots & \vdots \\ \psi_{N_\chi-1}(\tilde{x}), & \chi_{N_\chi-1} \le \tilde{x} < \chi_{N_\chi}, \end{cases} \tag{28}$$

where $\psi_{n'}(\tilde{x}) = \sum_{i=0}^{3} \alpha_{n',i}(\tilde{x} - \chi_{n'})^i$ for $\chi_{n'} \le \tilde{x} < \chi_{n'+1}$.

Substituting equation (28) into equation (27), we have

$$W_f(\tilde{w}, n) = \sqrt{\frac{\Delta x}{\tilde{w}}} \sum_{n'=-N_\chi}^{N_\chi-1} \int_{\chi_{n'}}^{\chi_{n'+1}} f_c\left(n - \frac{\tilde{u}}{\tilde{w}}\right)\psi_{n'}(\tilde{u})\mathrm{d}\tilde{u}. \tag{29}$$

Then we apply integration by parts to equation (29) and arrive at

$$
\begin{aligned}
W_f\left(\tilde{w}, n\right) &= \tilde{w}^3\sqrt{\frac{\Delta x}{\tilde{w}}}\sum_{n'=-N_\chi}^{N_\chi-1} 6\alpha_{n',3}\int_{\chi_{n'}}^{\chi_{n'+1}} F_c^{(3)}\left(n-\frac{\tilde{u}}{\tilde{w}}\right)\mathrm{d}\tilde{u} \\
&= \tilde{w}^4\sqrt{\frac{\Delta x}{\tilde{w}}}\sum_{n'=-N_\chi}^{N_\chi-1} 6\alpha_{n',3}\left(F_c^{(4)}\left(n-\frac{\chi_{n'}}{\tilde{w}}\right)\right. \\
&\quad \left. - F_c^{(4)}\left(n-\frac{\chi_{n'+1}}{\tilde{w}}\right)\right) \\
&= \tilde{w}^3\sqrt{\tilde{w}\Delta x}\sum_{n'=-N_\chi}^{N_\chi} B\left(n'\right) F_c^{(4)}\left(n-\frac{\chi_{n'}}{\tilde{w}}\right), \qquad (30)
\end{aligned}
$$

where $F_c^{(4)}(\tilde{x})$ is the fourth antiderivative of $f_c(\tilde{x})$, and

$$
\begin{aligned}
B(-N_\chi) &= 6\alpha_{-N_\chi,3}, \\
B(n) &= 6(\alpha_{n,3} - \alpha_{n-1,3}), \quad \text{for} \quad 1 - N_\chi \le n \le N_\chi - 1, \\
B(N_\chi) &= -6\alpha_{N_\chi-1}. \qquad (31)
\end{aligned}
$$

In fact, equation (30) assumes that the third derivative of the wavelet, i.e. ψ''' is constant on the interval $[\chi_{n'}, \chi_{n'+1})$, which can be approximated as

$$
\psi'''(\tilde{x}) \approx \frac{\psi''(\chi_{n'+1}) - \psi''(\chi_{n'})}{\chi_{n'+1} - \chi_{n'}}, \qquad (32)
$$

where ψ'' is the second derivative of the wavelet, which can be obtained analytically. Hence the coefficient $\alpha_{n',3}$ is given by

$$
\alpha_{n',3} = \frac{\psi''(\chi_{n'+1}) - \psi''(\chi_{n'})}{6(\chi_{n'+1} - \chi_{n'})}. \qquad (33)
$$

By using equations (22), (B7), and (B11), $F_c^{(4)}(\tilde{x})$ can be calculated as

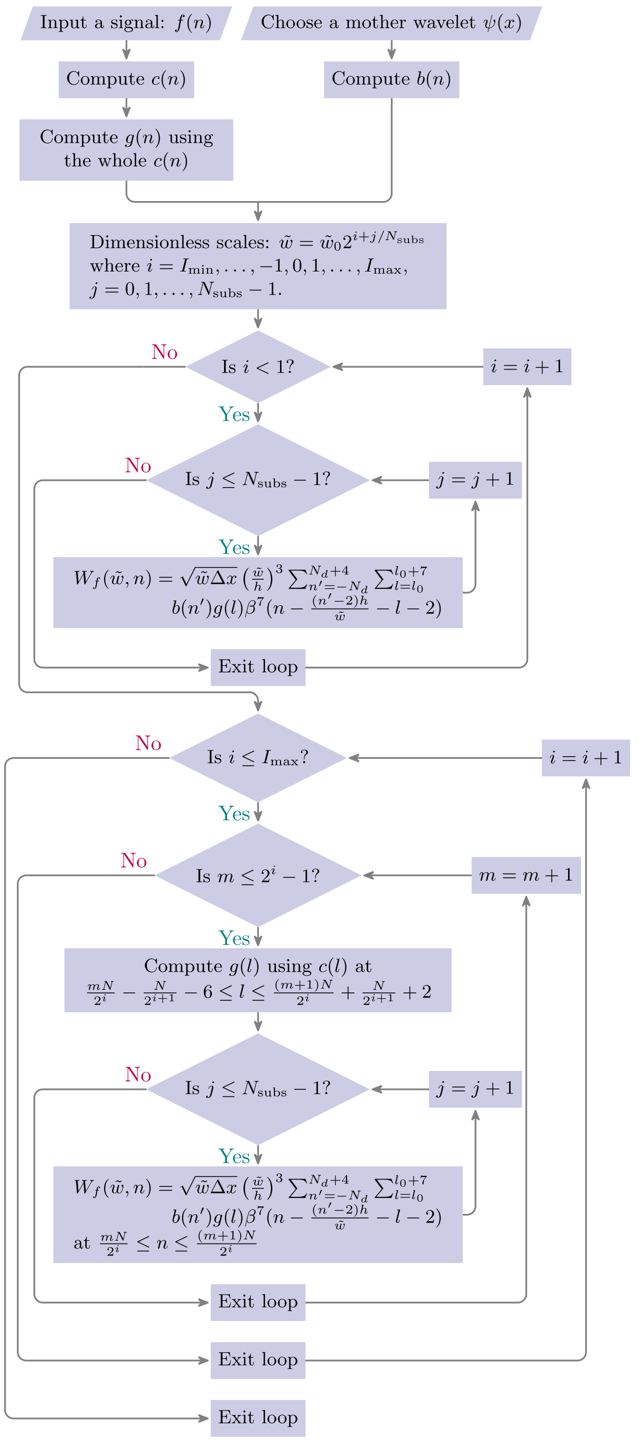
$$
F_c^{(4)}(\tilde{x}) = \sum_l g(l)\beta^7(\tilde{x} - l - 2). \qquad (34)
$$

Therefore equation (30) can be expressed as

$$
W_f(\tilde{w}, n) = \tilde{w}^3\sqrt{\tilde{w}\Delta x}\sum_{n'=-N_\chi}^{N_\chi}\sum_{l=l_1}^{l_1+7} B(n')g(l)\beta^7\left(n-\frac{\chi_{n'}}{\tilde{w}}-l-2\right), \qquad (35)
$$

where l_1 is the ceiling integer of $n - \chi_{n'}/\tilde{w} - 6$, and the coefficient sequence $g(l)$ is computed by equation (25). By comparing equations (24) and (35), we find that the M02CWT and A19CWT are very similar, but the theoretical derivation of the A19CWT is much simpler. To solve the accuracy issue of $g(l)$, we adopt the same scheme as the M02CWT algorithm.

The sequence of the A19CWT algorithm is shown as a flowchart in Fig. 5.



Input a signal: $f(n)$
Choose a mother wavelet $\psi(x)$
Compute $c(n)$
Compute $b(n)$
Compute $g(n)$ using the whole $c(n)$
Dimensionless scales: $\tilde{w} = \tilde{w}_0 2^{i+j/N_{\text{subs}}}$ where $i = I_{\min}, \ldots, -1, 0, 1, \ldots, I_{\max}$, $j = 0, 1, \ldots, N_{\text{subs}} - 1$.
No
Is $i < 1$?
$i = i + 1$
Yes
No
Is $j \le N_{\text{subs}} - 1$?
$j = j + 1$
Yes
$W_f(\tilde{w}, n) = \sqrt{\tilde{w}\Delta x}\left(\frac{\tilde{w}}{h}\right)^3 \sum_{n'=-N_d}^{N_d+4}\sum_{l=l_0}^{l_0+7} b(n')g(l)\beta^7(n - \frac{(n'-2)h}{\tilde{w}} - l - 2)$
Exit loop
No
Is $i \le I_{\max}$?
$i = i + 1$
Yes
No
Is $m \le 2^i - 1$?
$m = m + 1$
Yes
Compute $g(l)$ using $c(l)$ at $\frac{mN}{2^i} - \frac{N}{2^{i+1}} - 6 \le l \le \frac{(m+1)N}{2^i} + \frac{N}{2^{i+1}} + 2$
No
Is $j \le N_{\text{subs}} - 1$?
$j = j + 1$
Yes
$W_f(\tilde{w}, n) = \sqrt{\tilde{w}\Delta x}\left(\frac{\tilde{w}}{h}\right)^3 \sum_{n'=-N_d}^{N_d+4}\sum_{l=l_0}^{l_0+7} b(n')g(l)\beta^7(n - \frac{(n'-2)h}{\tilde{w}} - l - 2)$ at $\frac{mN}{2^i} \le n \le \frac{(m+1)N}{2^i}$
Exit loop
Exit loop
Exit loop

Figure 4. Schematic representation of the M02CWT algorithm.

3.5 Boundary conditions

In the definition of the CWT (see equation 1), the signal is assumed to be extended to infinity. Nevertheless, in reality, the length of the analysed signal is finite. Hence, we must make assumptions about the data outside its finite extent. Periodic boundary conditions are the most common choice. On the one hand, many cosmic fields are considered to be periodic. On the other hand, periodic boundary conditions are easy to implement. The FFTCWT inherits the attribute that the signal is assumed to be periodic in the FFT. The V97CWT imposes periodic boundary conditions on the signal by the IIR filtering (see Appendix C). For the M02CWT and A19CWT, the fact that the signal is periodic only imply that the coefficient $c(l)$ is periodic. To ensure that $g(l)$ is periodic, we take the following operations:

In addition, the M02CWT and A19CWT can also easily handle signals with zero boundaries without padding zeros at scale levels of $i < 1$. After calculating $c(l)$ by equations (C9), (C10), and (C13)–(C16), the coefficient $g(l)$ of the signal with zero boundaries can be obtained by the following operations:

```
if $i < 1$ then                                          ▷ Scale levels less than 1
    $g(0 : N-1) \leftarrow c(0 : N-1)$
    for $j = 1$ to 4 do
        $g \leftarrow \Delta^{-1} * g$
        $g \leftarrow g - \texttt{Mean}(g)$
    end for
else if $i \geq 1$ then                  ▷ Scale levels greater than or equal to 1
    Periodically padding $\frac{N}{4}+6$ values at start and $\frac{N}{4}+3$ values at end of $c$
    for $m = 0$ to $2^i - 1$ do
        $l_{t1} \leftarrow \frac{mN}{2^i} - \frac{N}{2^{i+1}} - 6, l_{t2} \leftarrow \frac{(m+1)N}{2^i} + \frac{N}{2^{i+1}} + 2$
        $g(l_{t1} : l_{t2}) \leftarrow (\Delta^{-4} * c)(l_{t1} : l_{t2})$
    end for
end if
```

```
if $i < 1$ then                                          ▷ Scale levels less than 1
    $g(-6 : N+1) \leftarrow c(-6 : N+1)$
    for $i = 1$ to 4 do
        $g \leftarrow \Delta^{-1} * g$
        $C_i \leftarrow g(N+1)$
    end for
    for $l = l_0$ to $l_0 + 7$ do          ▷ Replace $l_0$ with $l_1$ in the A19CWT
        if $l < -6$ then
            $g(l) \leftarrow 0$
        else if $l > N+1$ then
            $l' \leftarrow l - (N+1)$
            $g(l) \leftarrow C_4 + C_3 l' + \frac{1}{2} C_2 l'(l'+1) + \frac{1}{6} C_1 l'(l'+1)(l'+2)$
        end if
    end for
else if $i \geq 1$ then                  ▷ Scale levels greater than or equal to 1
    Padding $\frac{N}{4}$ zeros at start and $\frac{N}{4}+1$ zeros at end of $c$
    for $m = 0$ to $2^i - 1$ do
        $l_{t1} \leftarrow \frac{mN}{2^i} - \frac{N}{2^{i+1}} - 6, l_{t2} \leftarrow \frac{(m+1)N}{2^i} + \frac{N}{2^{i+1}} + 2$
        $g(l_{t1} : l_{t2}) \leftarrow (\Delta^{-4} * c)(l_{t1} : l_{t2})$
    end for
end if
```

3.6 Parameter settings

There are some unspecified parameters in the above algorithms, which are $\tilde{w}_{\max}$, N_q, h, N_d, and N_χ. In this subsection, we will discuss how to tune these parameters to make the algorithms sufficiently precise and efficient.

For the V97CWT algorithm, we define the approximation error as below

$$\mathrm{AE_V}(\tilde{w}_{\max}) = \frac{\sum_n \left|\psi(\tilde{w}_{\max}\tilde{x}_n) - \sum_m p(m)\beta^3(\tilde{x}_n - m)\right|}{\sum_n |\psi(\tilde{w}_{\max}\tilde{x}_n)|}, \tag{36}$$

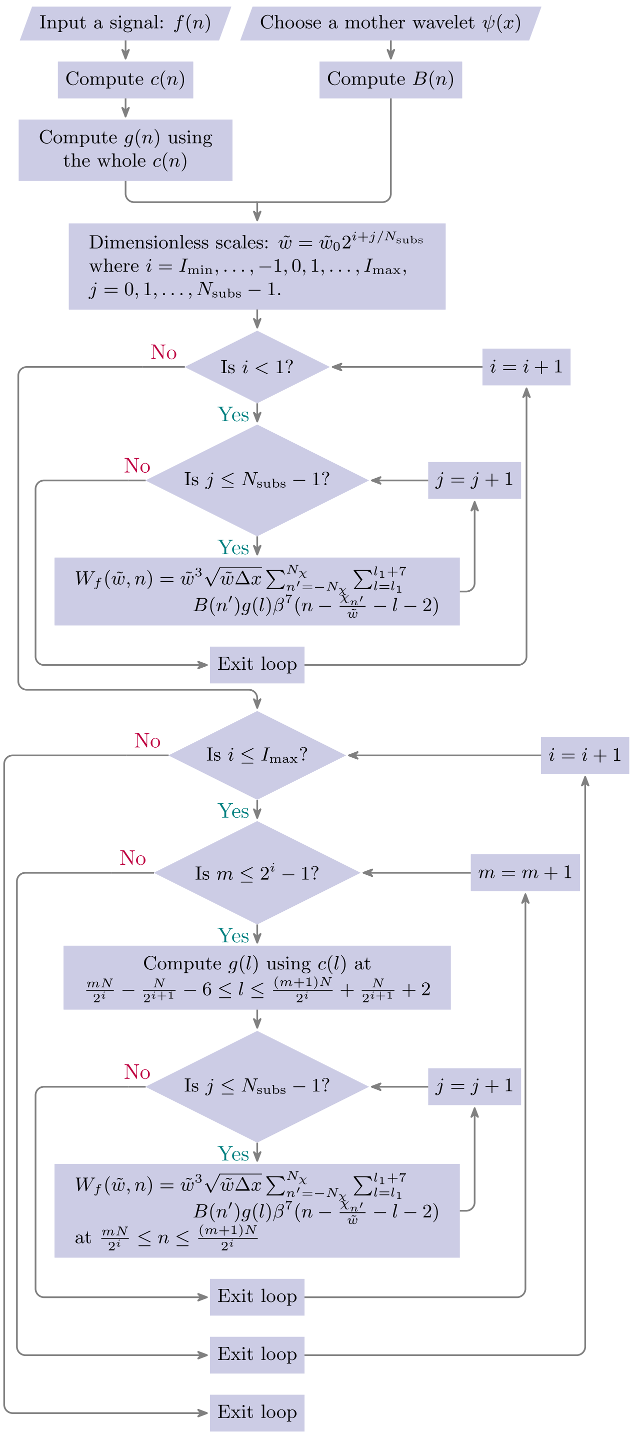
the result of which is shown in Fig. 6. We find that the error $\mathrm{AE_V}(\tilde{w}_{\max})$ increases with increasing $\tilde{w}_{\max}$. To ensure a high precision as well as a sufficiently large scale range, we set $\tilde{w}_{\max} = 1.34c_w$, which satisfy $\mathrm{AE_V}(\tilde{w}_{\max}/2) = 0.1$. According to equation (13), $q_j(N_q + 1) = 0$ yields the relationship $N_q(j) = 2^{j/N_{\mathrm{subs}}}\chi/\tilde{w}_{\max} - 1/2$. For simplicity, we use the same value of $N_q(j)$ at each j level, i.e.

$$N_q = \frac{2\chi}{\tilde{w}_{\max}} - \frac{1}{2}, \tag{37}$$

which is the upper limit of $N_q(j)$.

For the M02CWT algorithm, we define the approximation error as below

$$\mathrm{AE_M}(h) = \frac{\sum_n \left|\psi(\tilde{x}_n) - \sum_m d(m)\beta^3(\tilde{x}_n/h - m)\right|}{\sum_n |\psi(\tilde{x}_n)|}, \tag{38}$$



Input a signal: $f(n)$
Choose a mother wavelet $\psi(x)$
Compute $c(n)$
Compute $B(n)$
Compute $g(n)$ using the whole $c(n)$
Dimensionless scales: $\tilde{w} = \tilde{w}_0 2^{i+j/N_{\mathrm{subs}}}$ where $i = I_{\min}, \ldots, -1, 0, 1, \ldots, I_{\max}$, $j = 0, 1, \ldots, N_{\mathrm{subs}} - 1$.
No
Is $i < 1$?
$i = i + 1$
Yes
No
Is $j \leq N_{\mathrm{subs}} - 1$?
$j = j + 1$
Yes
$W_f(\tilde{w}, n) = \tilde{w}^3\sqrt{\tilde{w}\Delta x}\sum_{n'=-N_\chi}^{N_\chi}\sum_{l=l_1}^{l_1+7} B(n')g(l)\beta^7(n - \frac{\tilde{x}_{n'}}{\tilde{w}} - l - 2)$
Exit loop
No
Is $i \leq I_{\max}$?
$i = i + 1$
Yes
No
Is $m \leq 2^i - 1$?
$m = m + 1$
Yes
Compute $g(l)$ using $c(l)$ at $\frac{mN}{2^i} - \frac{N}{2^{i+1}} - 6 \leq l \leq \frac{(m+1)N}{2^i} + \frac{N}{2^{i+1}} + 2$
No
Is $j \leq N_{\mathrm{subs}} - 1$?
$j = j + 1$
Yes
$W_f(\tilde{w}, n) = \tilde{w}^3\sqrt{\tilde{w}\Delta x}\sum_{n'=-N_\chi}^{N_\chi}\sum_{l=l_1}^{l_1+7} B(n')g(l)\beta^7(n - \frac{\tilde{x}_{n'}}{\tilde{w}} - l - 2)$ at $\frac{mN}{2^i} \leq n \leq \frac{(m+1)N}{2^i}$
Exit loop
Exit loop
Exit loop

Figure 5. Schematic representation of the A19CWT algorithm.

the result of which is shown in Fig. 7. Since the cubic spline decomposition is perfectly exact to represent the CBSW with $h = 1$ (see Table 2), we only consider the approximation error for the GDW, CW-GDW, and MW. We see that the smaller the h, the smaller the error. However, considering the efficiency of the algorithm, the value of h cannot be chosen too small. Hence, we set the value of h such that the error $\mathrm{AE_M}(h)$ equals 5×10^{-4}, and then N_d can be determined by $h = \chi/(N_d + 2)$.

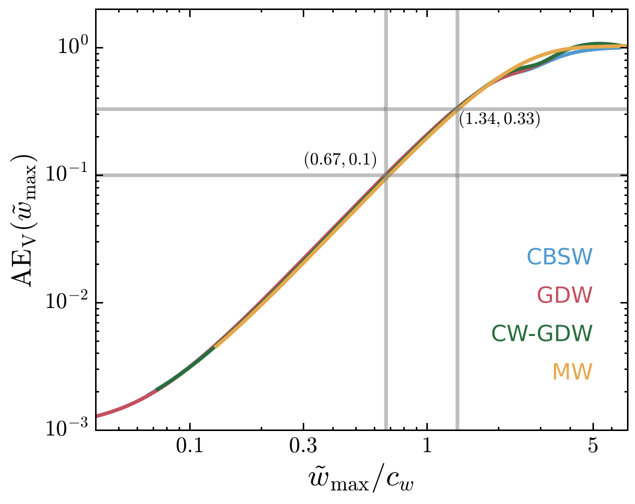


Figure 6. The Approximation error defined by equation (36) for different wavelets as labelled. At $\tilde{w}_{\max} = 0.67c_w$, the error reaches 0.1. When $\tilde{w}_{\max} = 1.34c_w$, i.e. twice as large as $0.67c_w$, the error is roughly 0.33.

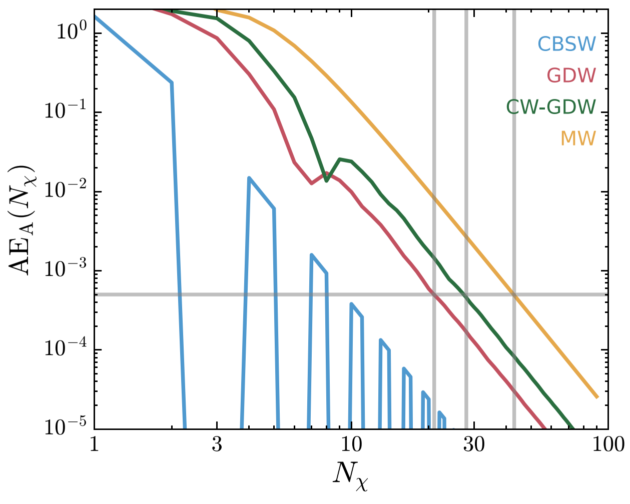


Figure 8. The Approximation error defined by equation (39) for different wavelets as labelled. For the CBSW, at N_χ equal to the integer multiples of 3, the error is very tiny and nearly 10^{-16}. So $N_\chi = 3$ is the best choice for it. The grey horizontal line shows the error level of 5×10^{-4}, and the grey vertical lines denote N_χ's values where the error reaches 5×10^{-4} for the GDW, CW-GDW, and MW from left to right.

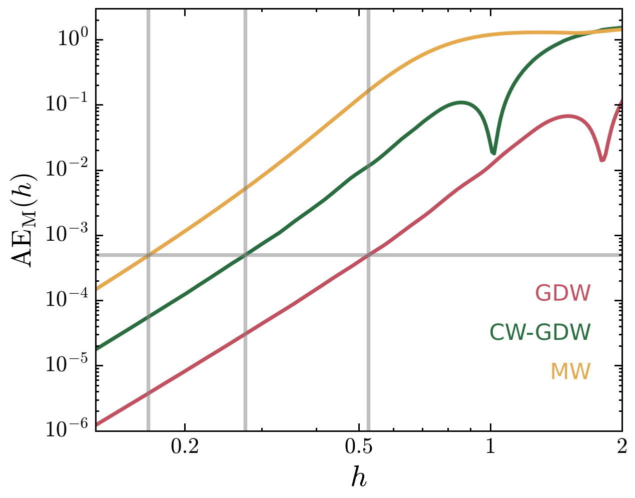


Figure 7. The Approximation error defined by equation (38) for different wavelets as labelled. The grey horizontal line shows the error level of 5×10^{-4}, and the grey vertical lines denote h's values where the error reaches 5×10^{-4} for the MW, CW-GDW, and GDW from left to right.

For the A19CWT algorithm, we define the approximation error as below

$$\mathrm{AE_A}(N_\chi) = \frac{\sum_n \left|\psi(\tilde{x}_n) - \psi_{\mathrm{pp}}(\tilde{x}_n)\right|}{\sum_n |\psi(\tilde{x}_n)|}, \tag{39}$$

where $\psi_{\mathrm{pp}}(\tilde{x}_n)$ is the piecewise polynomial function given by equation (28). Because the CBSW is actually a cubic piecewise polynomial function with compact support width $2\chi = 6$ and segment width $\Delta\chi = 1$, the approximation error $\mathrm{AE_A}(N_\chi)$ should be very small at the integer multiples of 3, which is illustrated in Fig. 8. Hence for the CBSW, $N_\chi = 3$ is the best choice. For the GDW, CW-GDW, and MW, we set the value of N_χ such that the error $\mathrm{AE_A}(N_\chi)$ roughly equals 5×10^{-4}, which is in accordance with the parameter settings of the M02CWT.

For clarity and convenience, we list the parameters and their values in Table 3.

Table 3. Parameter settings for the V97CWT, M02CWT, and A19CWT algorithms.

	$\tilde{w}_{\max}$	N_q	h	N_d	N_χ
CBSW	0.62457 [a]	9	1	1	3
GDW	1.19853 [b]	20	0.526	21	21
CW-GDW	0.57381 [c]	27	0.275	27	28
MW	0.32514 [d]	46	0.165	43	43

[a] 0.6245669798462486.
[b] 1.1985324359398872.
[c] 0.5738133082169298.
[d] 0.3251384995061764.

4 PERFORMANCE COMPARISON BETWEEN ALGORITHMS

In the 1D case, it is easy to find functions whose CWTs can be calculated analytically by using equation (1). Therefore, we can use their analytical results to examine the accuracy of the corresponding numerical outcomes. For instance, we here use the periodic function $f_1(x)$ with period 2π and the Gaussian function $f_2(x)$ as test signals, which are given below

$$f_1(x) = 2\cos(x) + \frac{1}{2}\cos(8x) + \frac{1}{4}\sin(32x), \tag{40}$$

$$f_2(x) = e^{-x^2/2}. \tag{41}$$

The two signals and their analytical CWTs are shown in Fig. 9.

For the following numerical tests, we write the double precision codes in Fortran 95 language, and compile them by gfortran 6.3.1 with the `-O3` flag under the Intel Xeon CPU E5-2678 v3 @ 2.50 GHz processor with 250 GB RAM running Linux (Fedora release 24).

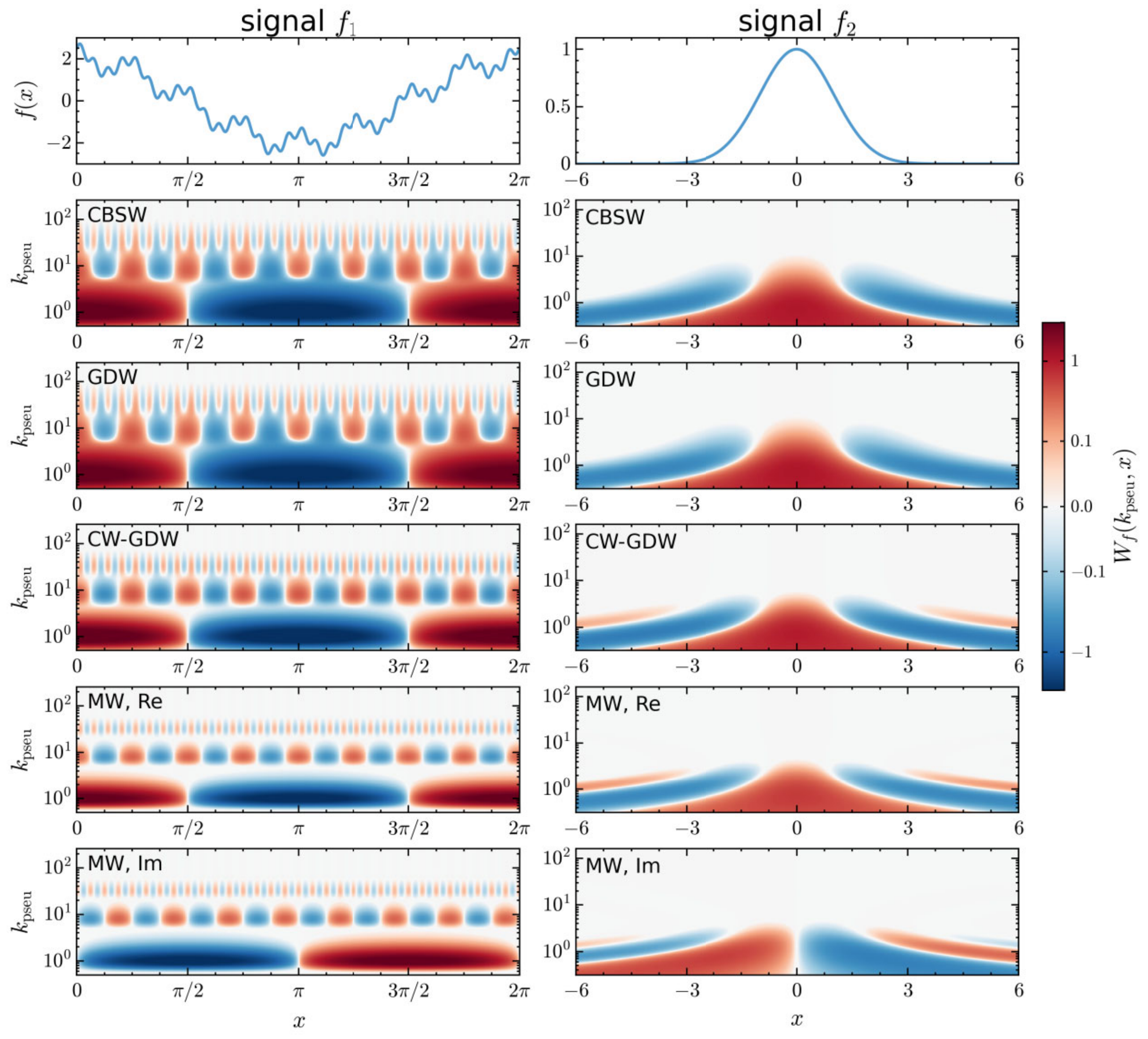


Figure 9. Left-hand column: the periodic signal f_1 and its CWTs which are calculated analytically for different wavelets. Right-hand column: the same as the left-hand column but for the non-periodic signal f_2. For comparison between different wavelets, we replace the wavelet scale w by the pseudo wavenumber $k_{\rm pseu}$ (see Table 2), and keep this convention throughout the subsequent plots.

4.1 Accuracy comparison

To check the accuracy of these algorithms, we define the error spectrum as follows

$$\mathrm{ES}(w) = \frac{\sum_n |W^{\rm n}_f(w, x_n) - W^{\rm a}_f(w, x_n)|}{\sum_n |W^{\rm a}_f(w, x_n)|} \times 100 \text{ per cent}, \tag{42}$$

where $W^{\rm a}_f(w, x)$ is the analytical CWT, and $W^{\rm n}_f(w, x)$ is the numerical CWT.

The computation of the numerical CWT $W^{\rm n}_f(w, x)$ requires the sampling of the signal. For the signals $f_1(x)$ and $f_2(x)$, we take $N = 512$ evenly spaced sample points on the intervals $[0, 2\pi)$ and $[-6, 6)$, respectively, which is sufficient to avoid the aliasing effect. The periodic boundary condition is used for $f_1(x)$, and the zero boundary condition for $f_2(x)$. Since the FFTCWT and V97CWT always assume the signals are periodic, we should pad zeros at both ends of the signal before execute the CWT of $f_2(x)$. Let $N_{\rm zeros}$ denote the number of padded zeros at each end, then it can be determined by χ and $\tilde{w}_{\rm min}$:

$$\begin{aligned} N_{\rm zeros} &= \mathrm{Nint}(\chi/\tilde{w}_{\rm min}), \\ &= \mathrm{Nint}\left(\frac{\chi}{c_w \pi}\right) N. \end{aligned} \tag{43}$$

However, padding zeros takes up more computational resources and reduce the efficiency of the algorithm, which we will see in the next subsection.

In Fig. 10, we show the error spectra of the periodic signal $f_1(x)$. We see that the FFTCWT algorithm yields the highest accuracy, the error of which is less than 10^{-10} per cent. The error of the V97CWT algorithm is between 0.01 per cent and 1 per cent. The errors of these two algorithms do not show any significant dependence on the kinds of wavelets. We observe that for the CBSW, the errors of both M02CWT and A19CWT are approximately between 3×10^{-9} per cent and 0.003 per cent. But for other wavelets, the errors are clearly higher than that for the CBSW, which are ranging from 0.003 per cent to 1 per cent in the case of the M02CWT,

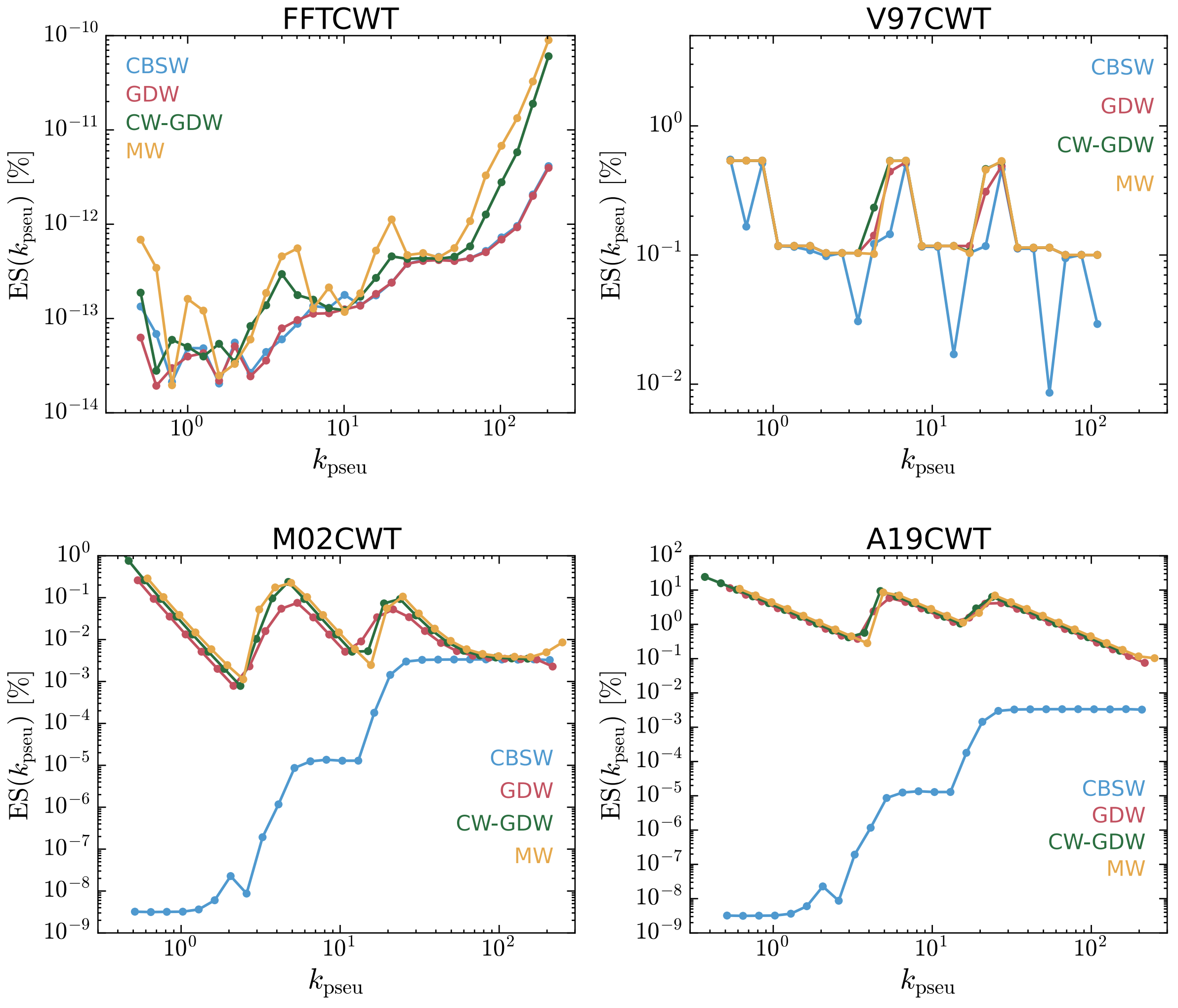


Figure 10. The error spectra of the periodic signal $f_1(x)$ for different wavelets, computed by the FFTCWT, V97CWT, M02CWT, and A19CWT algorithms, respectively.

and from 0.1 per cent to 10 per cent in the case of the A19CWT. The larger error of the A19CWT may be due to the too coarse approximation in equation (32). If N_χ is larger, this approximation will be more accurate, but the A19CWT will be less efficient.

In Fig. 11, we show the error spectra of the non-periodic signal $f_2(x)$. We observe that the error magnitudes of the FFTCWT (10^{-8} per cent – 0.1 per cent) in handling the non-periodic signal $f_2(x)$ are much higher than that (10^{-14} per cent – 10^{-10} per cent) in handling the periodic signal $f_1(x)$, whereas the error magnitudes of the other algorithms do not change much. Even so, the FFTCWT still provides the best accuracy among all algorithms for all wavelets. Only for the CBSW, the accuracy of the M02CWT and A19CWT can rival the accuracy of the FFTCWT.

In summary, the V97CWT, M02CWT, and A19CWT algorithms, which perform CWT calculations in real space, are not as precise as the FFTCWT. The reason for this is mainly that the former three make approximations to the wavelet function to trade off the efficiency, but yet the latter does not. For the general wavelets, A19CWT yields the largest error, which is due to twice approximations, namely equations (28) and (32), as stated above. However, for the special wavelet CBSW, the M02CWT and A19CWT algorithms provide a quite high accuracy owing to the fact that equations (21), (28), and (32) describe the CBSW exactly.

4.2 Speed comparison

To check the actual efficiency of the algorithms, we measured the variation of their CPU time with the number of sampling points per scale.

Fig. 12 shows the measurements of the periodic signal $f_1(x)$. We see that the V97CWT, M02CWT, and A19CWT algorithms without using the FFT are indeed very fast and they all have the complexity of $\mathcal{O}(N)$ but with different leading constants. However, they do not show a huge speed advantage over the FFTCWT at the sampling number of $N \lesssim 10^6$, which can be due to two reasons. On the one hand, the FFT library we used, `FFTW`, is very well optimized, and is the fastest free library available for computing the FFT. On the other hand, the leading constants of the V97CWT, M02CWT, and A19CWT are too large. The V97CWT performances better than the M02CWT and A19CWT, due to its recursive nature. Its CPU time are comparable to that of the FFTCWT for the real wavelets.

Fig. 13 shows the measurements of the non-periodic signal $f_2(x)$. It is clearly seen that the FFTCWT and V97CWT consume much

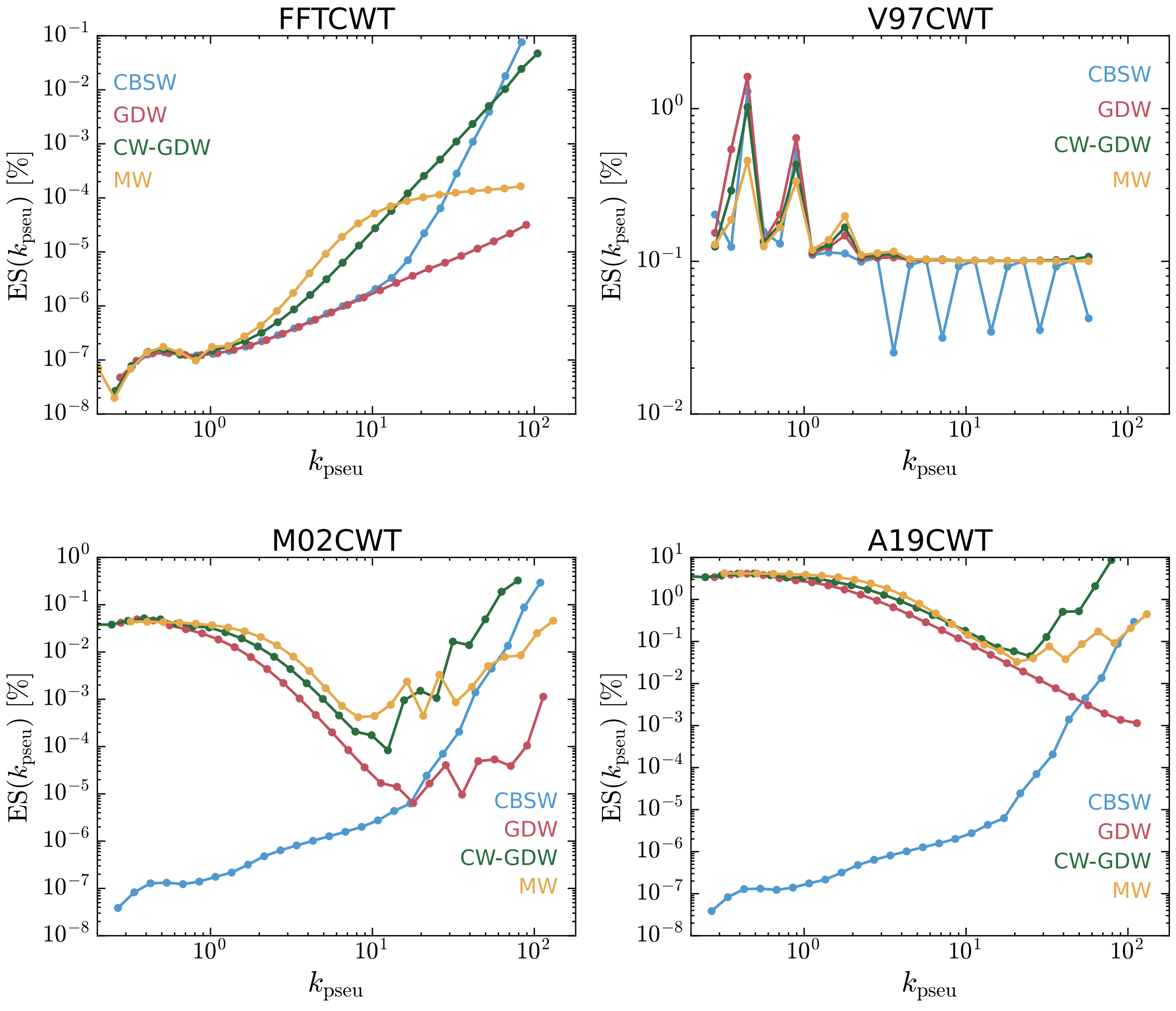


Figure 11. Same as Fig. 10, but for the results of the non-periodic signal $f_2(x)$.

more computational time than they do in processing the periodic signal, since we pad many zeros to the signal $f_2(x)$ before executing the CWT. Only for the complex wavelet, MW, the FFTCWT still maintains the speed advantage over the other algorithms. For other wavelets, the FFTCWT has no distinct speed advantage. The CPU time consumed by the M02CWT and A19CWT do not differ by whether the signal is periodic or non-periodic.

5 APPLICATIONS IN COSMOLOGY

There are many 1D signals in the astrophysics and cosmology, such as the light curves of astronomical sources, the Lyman-α forest, the 21 cm signal, the gravitational waves, and the cosmic fields obtained by solving 1D perturbative equations. The CWT can map them into the 2D time-frequency or space-scale domains, which reveals totally the complex and irregular structures at various positions and scales. Furthermore, we can construct some statistics based on the CWT to characterize the signals more quantitatively, for example, the wavelet power spectrum, the wavelet cross-correlation, the wavelet bicoherence, the wavelet modulus maxima and so on (e.g. Muzy et al. 1991; Hudgins et al. 1993; van Milligen et al. 1995a, 1995b).

As illustration, we perform the wavelet analysis of the density fields obtained by the 1D Zel'dovich approximation, which provides the exact non-linear solution for the perturbative equations of collisionless matter up to the first appearance of orbit-crossing singularities. The non-linear density field is given by

$$\delta(x,\theta)+1=\frac{1}{1-\theta\delta_0(x)}, \tag{44}$$

where θ is the growth factor and used as the time variable, and $\delta_0(x) = \delta(x, \theta = 1)$ is the initial Gaussian density field satisfying periodic boundary conditions, which is generated by the power-law spectrum $P(k) = Ak^{-2}$ with $A = 2.5 \times 10^{-6}$. For more details about the 1D Zel'dovich approximation, we refer the reader to Wang et al. (2022).

From the results in Section 4.1, it is clear that the FFTCWT algorithm is optimal for the periodic signal. In addition, as can be seen from Fig. 1, the CW-GDW achieves a better balance between spatial resolution and scale resolution compared to other wavelets. Therefore, we compute the CWTs of density fields by the FFTCWT algorithm with the CW-GDW, the results of which are illustrated in Fig. 14. By visual inspection, the CWT of the initial density field is dominated by large-scale components with a relatively random spatial distribution. As a consequence of the non-linear gravitational effect, the CWT of the density field at $\theta = 100$ shows a non-random structure with many small-scale components, which do not exist at the initial time.

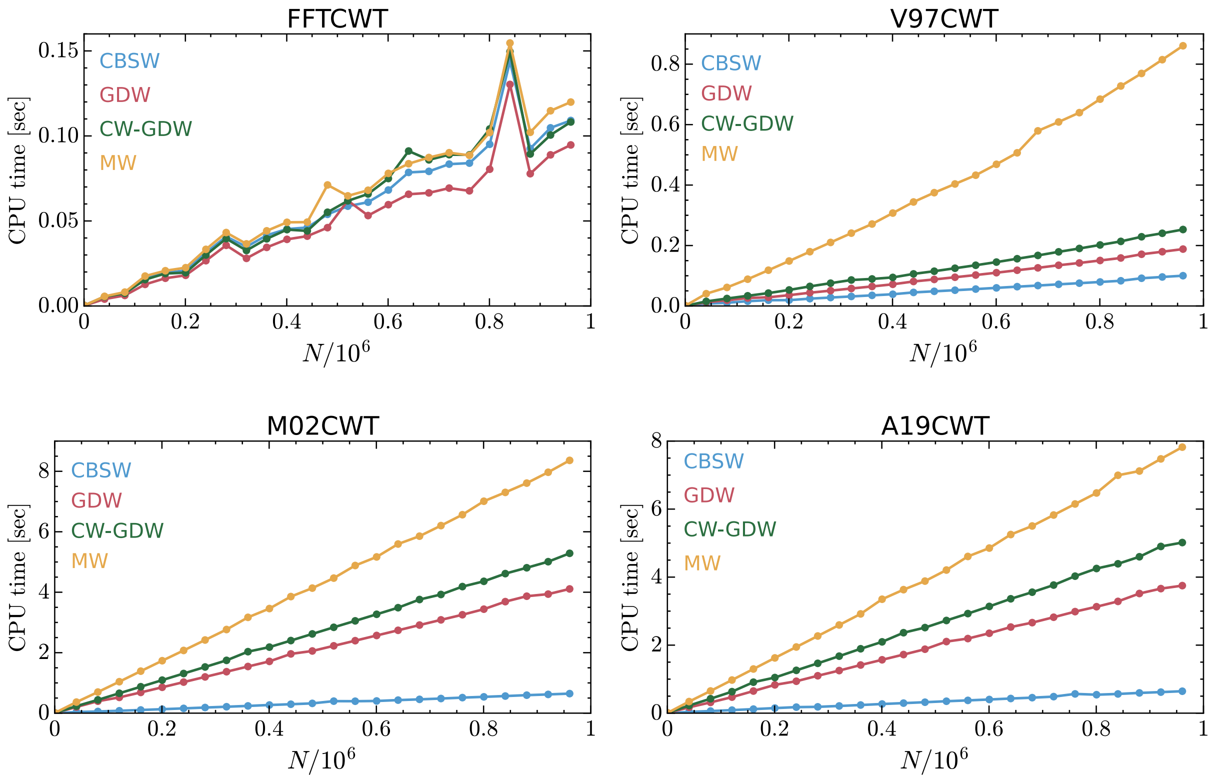


Figure 12. The CPU time per scale of the different algorithms with different wavelets to compute the numerical CWT of the periodic signal $f_1(x)$.

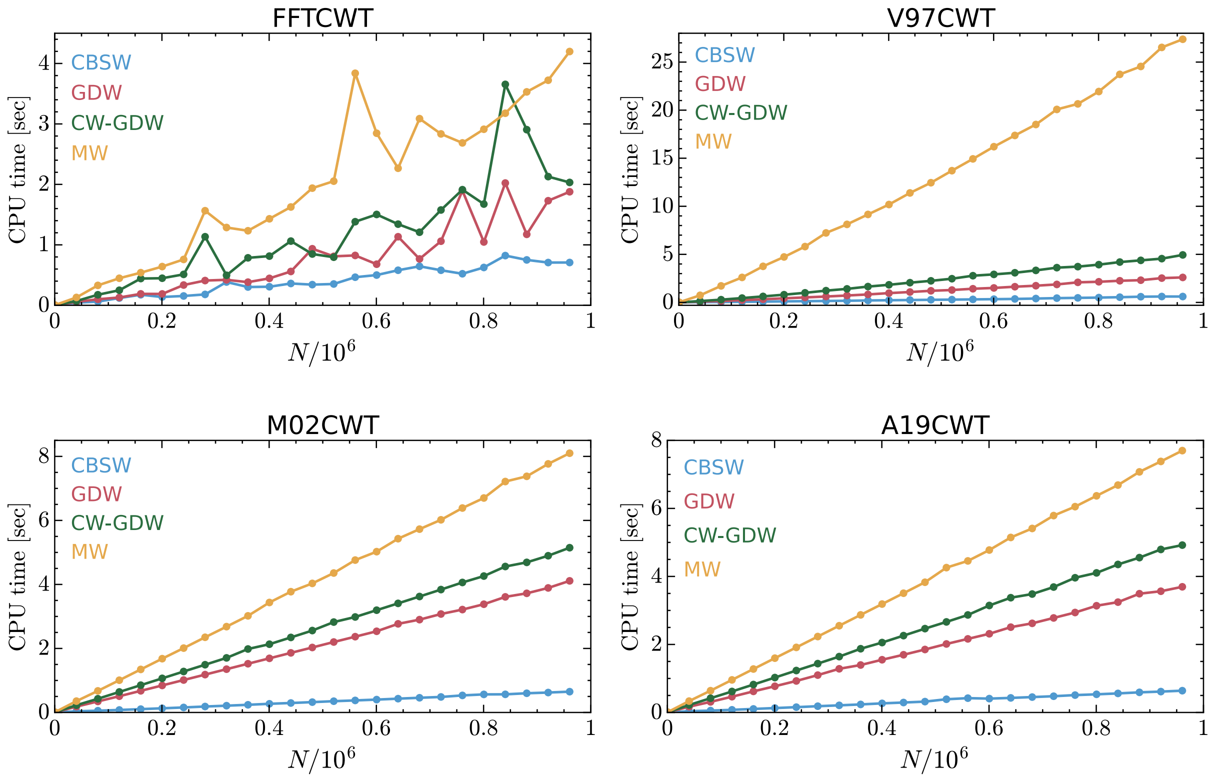


Figure 13. Same as Fig. 12, but for the measurements of the non-periodic signal $f_2(x)$.

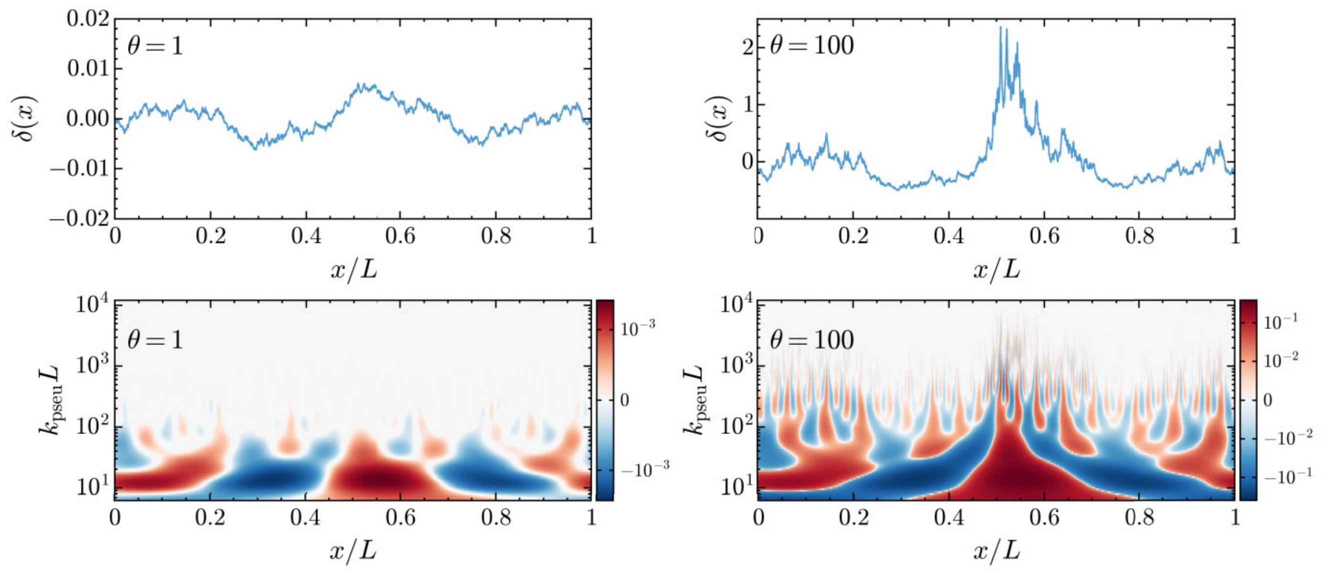


Figure 14. Top row: the initial density field (left-hand panel) and the non-linear density field at $\theta = 100$ (right-hand panel). Bottom row: the corresponding CWTs of the density fields, which are computed by using the FFTCWT algorithm with the CW-GDW. In these plots, the coordinates x and scales $k_{\rm pseu}$ are made dimensionless by dividing and multiplying the length size L of the density field, respectively.

In our previous work (Wang & He 2022), we proposed the environment-dependent wavelet power spectrum (env-WPS) to measure the dependence of matter clustering on both the scale and environment, which is given by

$$P(w, \delta') = \langle |W_\delta(w, x)|^2 \rangle_{\delta(x)=\delta'}, \tag{45}$$

where $W_\delta(w, x)$ is the CWT of $\delta(x)$, and '$\langle \ldots \rangle_{\delta(x)=\delta'}$' denotes the statistical average of the wavelet coefficients at each scale with the same local density, i.e. $\delta(x) = \delta'$. If we average over all the possible densities, then the env-WPS will degenerate to the global WPS as bellow

$$P(w) = \langle |W_\delta(w, x)|^2 \rangle_{\text{all } \delta'}. \tag{46}$$

Thus the relation between the global WPS $P(w)$ and the env-WPS $P(w, \delta')$ is

$$P(w) = \sum_{\delta'} f_{\delta'} P(w, \delta'), \tag{47}$$

where $f_{\delta'} = N_{\delta'}/N$ is the fraction of the env-WPS relative to the global WPS, and $N_{\delta'}$ is the number of grids at $\delta(x) = \delta'$. In fact, the env-WPS can be generalized to other kinds of signals, just by replacing δ' with the corresponding attribute.

For simplicity, we here split the densities into: (i) $\delta > 0$, i.e. the overdense environments and (ii) $\delta < 0$, i.e. the underdense environments, and then compute the env-WPSs, the results of which are shown in Fig. 15. For the initial density field, we can see that its env-WPSs have the same amplitudes with the global WPS. However, for the fully evolved density field at $\theta = 100$, the env-WPSs exhibit an obvious environment dependence. Specifically, the env-WPS with $\delta > 0$ is larger than the global WPS, while that with $\delta < 0$ is less than the global WPS.

As can be seen, the env-WPS provides more information about matter clustering than the traditional two-point statistics, e.g. FT-

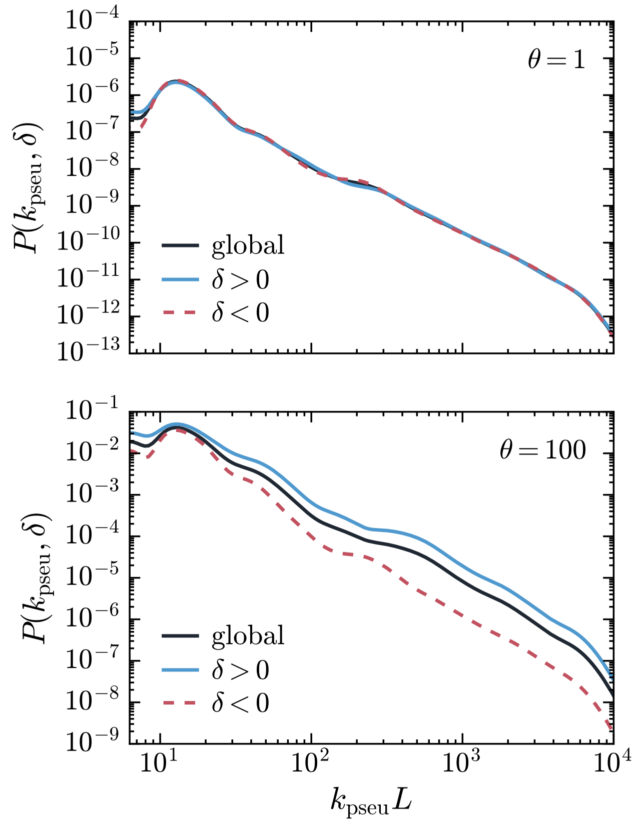


Figure 15. Comparison of the env-WPS of the initial density field and that of the late time density field. Top panel: the env-WPS of the initial density field with power-law power spectrum $P(k) \propto k^{-2}$. Bottom panel: the env-WPS of the non-linear density field at $\theta = 100$. In each panel, the global WPS is denoted by the black line.

based power spectrum, which completely lost the characteristics of the matter spatial distribution.[4]

6 SUMMARY AND CONCLUSIONS

In this paper, we review the fast algorithms for the CWT, including the FFTCWT with complexity of $\mathcal{O}(N\log_2 N)$ per scale, and other three algorithms with complexity of $\mathcal{O}(N)$ per scale, i.e. the V97CWT proposed by Vrhel et al. (1997), the M02CWT proposed by Muñoz et al. (2002), and the A19CWT proposed by Arizumi & Aksenova (2019).

By the convolution theorem, the FFTCWT converts convolution calculations in the real domain into multiplications in the Fourier domain, and then returns the final result by the inverse FT (see Section 3.1 and Fig. 2). The V97CWT use the daughter wavelets with scales of $\tilde{w}_{\max}/2^{j/N_{\rm subs}}$ and approximate them with two scaling functions, i.e. the zero order and the cubic B-splines. Using the two-scale relation of the B-splines, then the CWT can be calculated recursively (see Section 3.2 and Fig. 3). The M02CWT approximate the daughter wavelet with the rescaled cubic B-spline, and interpolating the discrete input signal with the cubic B-spline. Therefore, the large wavelet convolution kernel is translated into smaller B-spline kernel (see Section 3.3 and Fig. 4). The A19CWT achieves the same purpose as the M02CWT by approximating the mother wavelet as cubic piecewise polynomials and applying integration by parts (see Section 3.4 and Fig. 5). In fact, the precision of algorithms originally mentioned in Muñoz et al. (2002) and Arizumi & Aksenova (2019) is terrible on small scales, and we remedy this issue in our M02CWT and A19CWT algorithms (see Appendix D).

We compare the accuracy and speed between these fast CWT algorithms in Figs 10–13 by using two specific signals. Our main findings are summarized as follows:

(i) Even though for the non-periodic signal with zero boundaries, the accuracy of the FFTCWT is much lower compared to that for the periodic signal, it is still more accurate than other algorithms.

(ii) When the $\mathcal{O}(N)$ algorithms process the non-periodic signal with zero boundaries, the overall magnitudes of their errors do not grow larger compared to when they process the periodic signal. Hence the accuracy of them is robust to different types of signals. The M02CWT achieves the best accuracy among them.

(iii) For the GDW, CW-GDW and MW, A19CWT is the least accurate algorithm. But for the CBSW, the A19CWT is just as accurate as the M02CWT, which is because both the cubic B-spline and piecewise polynomials represent the CBSW exactly.

(iv) At the sampling number we consider, i.e. $N \lesssim 10^6$, the algorithms with the complexity of $\mathcal{O}(N)$ per scale do not exhibit an overall speed advantage over the FFTCWT. Only the V97CWT with real wavelets shows a speed comparable to it.

(v) For the non-periodic signal with zero boundaries, the FFTCWT and V97CWT are less efficient due to padding zeros to the signal. However, the efficiency of the M02CWT and A19CWT is not affected by the type of signals.

Therefore, the FFTCWT and V97CWT are suitable for the periodic signals. In particular, the V97CWT using real wavelets will perform better than using complex wavelets, e.g. the MW. The M02CWT is suitable for the non-periodic signals with zero boundary condition. We do not refer to the A19CWT algorithm because it is not accurate enough.

As a demonstration of the usage of the CWT, we then apply the FFTCWT to perform wavelet analysis of the 1D density fields. Acting like a ‘mathematical microscope’, the CWT allows us to zoom in on complex structures of the density fields at various scales and locations (see Fig. 14). We also introduce the wavelet-based statistic, env-WPS, which is a bivariate function of the local density environment and the scale. As shown in Fig. 15, the env-WPS tells us that for the initial field, there is no any environment dependence of matter clustering on all scales. However, for the late time field, the matter clustering is dominated by the matter in overdense environments. Clearly, the env-WPS contains more information about the matter clustering than the usual two-point statistics. The env-WPS can also be generalized to analyse other signals by replacing the local density environment by other attribute.

To analyse the multidimensional data, such as the 2D gravitational lensing maps and the 3D cosmic fields, the next natural step is to extend the 1D CWT algorithms to multidimensions. It is easy to develop 2D and 3D FFT-based CWT algorithms, since there are publicly available FFT libraries to use. However, multidimensional extensions of the rest 1D algorithms are not straightforward. In the future, we will plan to develop the fast multidimensional CWT algorithms without the use of FFT.

[4]Wang & He (2022) makes it more explicitly.

ACKNOWLEDGEMENTS

YW especially thanks Dr. Vrhel and Mr. Arts for helpful discussions. PH acknowledges the support by the National Science Foundation of China (No. 12047569, 12147217), and by the Natural Science Foundation of Jilin Province, China (No. 20180101228JC).

DATA AVAILABILITY

The double precision Fortran 95 codes of the FFTCWT, V97CWT, M02CWT, and A19CWT algorithms are released in https://github.com/WangYun1995/FortranCWT. The corresponding Python WRAPPERS are available in https://github.com/WangYun1995/pyFortranCWT.

REFERENCES

Addison P. S., 2017, The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance, 2nd edn., CRC Press, Boca Raton, FL
Addison P. S., 2018, Phil. Trans. R. Soc. A, 376, 20170258
Arizumi N., Aksenova T., 2019, in IEEE International Symposium on Signal Processing and Information Technology. IEEE Computer Society, Los Alamitos, CA, p. 1
Arnalte-Mur P., Labatie A., Clerc N., Martínez V. J., Starck J. L., Lachièze-Rey M., Saar E., Paredes S., 2012, A&A, 542, A34
Arshakian T. G., Ossenkopf V., 2016, A&A, 585, A98
Arts L., et al., 2022, Nat. Comput. Sci., 2, 47
Bacon D. J. et al., 2020, Publ. Astron. Soc. Austr., 37, e007
Berkner K., Wells R., 1997, in Conf. Record of the 31st Asilomar Conf. on Signals, Systems and Computers. IEEE Computer Society, Los Alamitos, CA, p. 1235
Briand T., Monasse P., 2018, Image Process. On Line, 8, 99
da Cunha D. C. N., Harnois-Deraps J., Brandenberger R., Amara A., Refregier A., 2018, Phys. Rev. D, 98, 083015
Daubechies I., 1992, Ten Lectures on Wavelets. SIAM, Philadelphia, PA
Daubechies I., Lu J., Wu H.-T., 2011, Appl. Comput. Harmon. Anal., 30, 243
Davé R., Anglés-Alcázar D., Narayanan D., Li Q., Rafieferantsoa M. H., Appleby S., 2019, MNRAS, 486, 2827

Escalera E., Mazure A., 1992, ApJ, 388, 23
Escalera E., Biviano A., Girardi M., Giuricin G., Mardirossian F., Mazure A., Mezzetti M., 1994, ApJ, 423, 539
Flin P., Krywult J., 2006, A&A, 450, 9
Frick P., Beck R., Berkhuijsen E. M., Patrickeyev I., 2001, MNRAS, 327, 1145
Frick P., Stepanov R., Beck R., Sokoloff D., Shukurov A., Ehle M., Lundgren A., 2016, A&A, 585, A21
Frigo M., Johnson S., 2005, in Proc. IEEE Vol. 93, The Design and Implementation of FFTW3. IEEE, Piscataway, NJ, p. 216
Garzilli A., Bolton J. S., Kim T. S., Leach S., Viel M., 2012, MNRAS, 424, 1723
Gu J., Xu H., Wang J., An T., Chen W., 2013, ApJ, 773, 38
Hernández-Aguayo C. et al., 2022, preprint (arXiv:2210.10059)
Hudgins L., Friehe C. A., Mayer M. E., 1993, Phys. Rev. Lett., 71, 3279
Kaiser G., Hudgins L. H., 1994, A Friendly Guide to Wavelets. Birkhäuser, Boston MA
Labatie A., Starck J. L., Lachièze-Rey M., 2012, ApJ, 746, 172
Laureijs R. et al., 2011, preprint (arXiv:1110.3193)
Levi M. et al., 2013, preprint (arXiv:1308.0847)
Li W. et al., 2019, MNRAS, 485, 2628
Lidz A., Faucher-Giguère C.-A., Dall'Aglio A., McQuinn M., Fechner C., Zaldarriaga M., Hernquist L., Dutta S., 2010, ApJ, 718, 199
Martínez V. J., Paredes S., Saar E., 1993, MNRAS, 260, 365
Muñoz A., Ertlé R., Unser M., 2002, Signal Process., 82, 749
Muzy J. F., Bacry E., Arneodo A., 1991, Phys. Rev. Lett., 67, 3515
Omachi M., Omachi S., 2007, in Int. Conf. on Wavelet Analysis and Pattern Recognition. IEEE Computer Society, Los Alamitos, CA, p. 1688
Pérez-Rendón A. F., Robles R., 2004, Signal Process, 84, 55
Pillepich A. et al., 2018, MNRAS, 473, 4077
Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 2007, Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd edn. Cambridge Univ. Press, Cambridge, USA
Proakis J. G., Manolakis D. K., 2007, Digital Signal Processing: Principles, Algorithms, and Applications, 4th edn., Pearson Education, London
Ren H. X., Cerruti M., Sahakyan N., 2023, A&A, 672, A86
Robitaille J. F., Joncas G., Miville-Deschênes M. A., 2014, MNRAS, 440, 2726
Roh S., Ryu D., Kang H., Ha S., Jang H., 2019, ApJ, 883, 138
Rozgacheva I. K., Borisov A. A., Agapov A. A., Pozdneev I. A., Shchetinina O. A., 2012, preprint (arXiv:1201.5554)
Schwinn J., Baugh C. M., Jauzac M., Bartelmann M., Eckert D., 2018, MNRAS, 481, 4300
Shensa M. J., 1993, An inverse DWT for nonorthogonal wavelets, Final Report Naval Command. Control and Ocean Surveillance Center, RDT and E Div., San Diego, CA
Shi X., Nagai D., Lau E. T., 2018, MNRAS, 481, 1075
Slezak E., Bijaoui A., Mars G., 1990, A&A, 227, 301
Slezak E., de Lapparent V., Bijaoui A., 1993, ApJ, 409, 517
Tabatabaei F. S., Berkhuijsen E. M., Frick P., Beck R., Schinnerer E., 2013, A&A, 557, A129
Tarnopolski M., Żywucka N., Marchenko V., Pascual-Granado J., 2020, ApJS, 250, 1
Tary J. B., Herrera R. H., van der Baan M., 2018, Phil. Trans. R. Soc. A, 376, 20170254
Tian H. J., Neyrinck M. C., Budavári T., Szalay A. S., 2011, ApJ, 728, 34
Torrence C., Compo G. P., 1998, Bull. Am. Meteorol. Soc., 79, 61
Turner M. S., 2022, Annu. Rev. Nucl. Part., 72, 1
Unser M., Aldroubi A., Schiff S., 1994, IEEE Trans. Signal Process., 42, 3519
van Milligen B. P., Sánchez E., Estrada T., Hidalgo C., Brañas B., Carreras B., García L., 1995a, Phys. Plasmas, 2, 3017
van Milligen B. P., Hidalgo C., Sánchez E., 1995b, Phys. Rev. Lett., 74, 395
Vrhel M., Lee C., Unser M., 1997, IEEE Trans. Signal Process., 45, 891
Wang Y., He P., 2021, Commun. Theoret. Phys., 73, 095402
Wang Y., He P., 2022, ApJ, 934, 112
Wang Y., Yang H.-Y., He P., 2022, ApJ, 934, 77
Wolfson M., Hennawi J. F., Davies F. B., Oñorbe J., Hiss H., Lukić Z., 2021, MNRAS, 508, 5493

APPENDIX A: DERIVATION OF THE SIMPLE INVERSION FORMULA FOR THE CWT

In our previous works (Wang & He 2021; Wang et al. 2022), we demonstrate that there exists a single integral inverse formula for the real-valued wavelet derived by smoothing window function which is shown below

$$f(x) = f(w \to 0, x) + \int_0^{+\infty} \frac{W_f(w, x)}{\sqrt{w}} \mathrm{d}w, \tag{A1}$$

where $f(w \to 0, x) = \lim_{w \to 0} \int f(u) S(w, x-u) \mathrm{d}u$, $S(w, x) = wS(wx)$ is a smoothing function with scale w, and $W_f(w, x)$ is the CWT of $f(x)$ based on the wavelet $\psi(w, x) = \sqrt{w} \partial S(w, x)/\partial w$.

In fact, we can generalize equation (A1) to hold for more general real wavelets. According to the convolution theorem, the CWT $W_f(w, x)$ can be expressed as

$$W_f(w, x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(k) \frac{1}{\sqrt{w}} \hat{\psi}\left(\frac{k}{w}\right) e^{-ikx} \mathrm{d}k,$$

where $\hat{f}(k)$ and $\hat{\psi}(k)$ are Fourier transforms of $f(x)$ and $\psi(x)$, respectively. Divide the L.H.S. and R.H.S. of the above equation by $\sqrt{w}$ and integrate over w, we obtain

$$\int_0^{+\infty} \frac{W_f(w, x)}{\sqrt{w}} \mathrm{d}w = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\int_0^{+\infty} \frac{1}{w} \hat{\psi}\left(\frac{k}{w}\right) \mathrm{d}w \right) \hat{f}(k) e^{-ikx} \mathrm{d}k. \tag{A2}$$

Let's observe the value of $\int_0^{+\infty} \frac{1}{w} \hat{\psi}(\frac{k}{w}) \mathrm{d}w$:

(i) If $k = 0$, it follows from the oscillatory nature of wavelets $\hat{\psi}(0) = \int_{-\infty}^{+\infty} \psi(x) \mathrm{d}x = 0$ that $\int_0^{+\infty} \frac{1}{w} \hat{\psi}(\frac{k}{w}) \mathrm{d}w = 0$. Hence, the zero frequency component of $\hat{f}(k)$ is subtracted by CWT.

(ii) If $k > 0$ and let $u = k/w$, we have $\int_0^{+\infty} \frac{1}{w} \hat{\psi}(\frac{k}{w}) \mathrm{d}w = \int_0^{+\infty} \frac{1}{u} \hat{\psi}(u) \mathrm{d}u$.

(iii) If $k < 0$ and let $u = -k/w$, we have $\int_0^{+\infty} \frac{1}{w} \hat{\psi}(\frac{k}{w}) \mathrm{d}w = \int_0^{+\infty} \frac{1}{u} \hat{\psi}(-u) \mathrm{d}u$.

It is clear from the above that if the real-valued wavelets satisfy

$$\psi(x) = \psi(-x) \tag{A3}$$

which is equivalent to $\hat{\psi}(k) = \hat{\psi}(-k)$, and

$$0 < \left| \mathcal{K}_\psi \equiv \int_0^{+\infty} \frac{1}{k} \hat{\psi}(k) \mathrm{d}k \right| < \infty, \tag{A4}$$

then we get

$$\begin{aligned} \frac{1}{\mathcal{K}_\psi} \int_0^{+\infty} \frac{W_f(w, x)}{\sqrt{w}} \mathrm{d}w &= \frac{1}{2\pi} \int_{k \neq 0} \hat{f}(k) e^{-ikx} \mathrm{d}k \\ &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(k) e^{-ikx} \mathrm{d}k \\ &\quad - \lim_{\delta k \to 0} \frac{1}{2\pi} \int_{-\delta k/2}^{+\delta k/2} \hat{f}(k) e^{-ikx} \mathrm{d}k \\ &= f(x) - \lim_{\delta k \to 0} \frac{\delta k}{2\pi} \hat{f}(0) \\ &= f(x) - \lim_{L \to \infty} \frac{1}{L} \int_{-L/2}^{L/2} f(x) \mathrm{d}x. \end{aligned}$$

Finally, we arrive at the simple inversion formula for the CWT, which is

$$f(x) = \bar{f} + \frac{1}{\mathcal{K}_\psi}\int_0^{+\infty}\frac{W_f(w,x)}{\sqrt{w}}\mathrm{d}w, \tag{A5}$$

where $\bar{f} = \lim_{L\to\infty}\frac{1}{L}\int_{-L/2}^{L/2} f(x)\mathrm{d}x$ denotes the average of $f(x)$ over all space. For example, in the case of periodic functions, $\bar{f}$ is equal to the average of the function over a period. In the case of compactly supported functions, $\bar{f}$ is equal to zero.

APPENDIX B: B-SPLINE FUNCTIONS

The B-spline function of degree zero $\beta^0(x)$ is defined as

$$\beta^0(x) = \begin{cases} 1, & -1/2 \le x \le 1/2, \\ 0, & \text{otherwise}, \end{cases} \tag{B1}$$

and the B-spline $\beta^n(x)$ of degree n is constructed from the n times convolution of $\beta^0(x)$:

$$\beta^n(x) = (\underbrace{\beta^0 * \beta^0 * \ldots * \beta^0}_{n\ \text{times}})(x). \tag{B2}$$

Obviously, the B-spline $\beta^{n_1+n_2}$ can be calculated by convolving the B-splines β^{n_1} and β^{n_2} as follows

$$\beta^{n_1+n_2}(x) = (\beta^{n_1} * \beta^{n_2})(x). \tag{B3}$$

B-splines have many useful properties, which are listed below

(i) They are compactly supported functions with support interval $[-(n+1)/2, (n+1)/2]$ (Briand & Monasse 2018).

(ii) They satisfy a two-scale relation (Vrhel et al. 1997), which is

$$\beta^n(x/2) = \sum_m h(m)\beta^n(x-m), \tag{B4}$$

where the coefficients $h(m)$ are given by

$$h(m) = \begin{cases} \frac{1}{2^n}\binom{n+1}{m+(n+1)/2}, & |m| \le (n+1)/2, \\ 0, & \text{otherwise}. \end{cases} \tag{B5}$$

(iii) The rescaled B-spline of degree n is (Muñoz et al. 2002)

$$w\beta^n(wx) = w^{n+1}\left(\Delta_w^{n+1} * D^{-(n+1)}\delta^D\left(\cdot + \frac{n+1}{2w}\right)\right)(x), \tag{B6}$$

where $\delta^D(x)$ is the Dirac delta function, D^{-1} is the antiderivative (or integral) operator defined as

$$D^{-1}f(x) = \int_{-\infty}^{x} f(u)\mathrm{d}u, \tag{B7}$$

Δ_w^{n+1} is the rescaled finite-difference operator defined as

$$\begin{aligned}(\Delta_w^{n+1} * f)(x) &= \sum_{m=0}^{n+1} a(m) f(x - m/w) \\ &= \sum_{m=0}^{n+1}(-1)^m\binom{n+1}{m} f(x-m/w),\end{aligned} \tag{B8}$$

and Δ^{-1} is the inverse finite-difference operator defined as

$$(\Delta^{-1} * f)(x) = \sum_{m\le x} f(x-m). \tag{B9}$$

For the discrete signal $f(n)$, its inverse finite-difference $s(n) = (\Delta^{-1} * f)(n)$ can be implemented recursively by

$$s(n) = s(n-1) + f(n). \tag{B10}$$

(iv) The n_1-th antiderivative of the B-spline of degree n_2 is (Muñoz et al. 2002)

$$D^{-(n_1)}\beta^{n_2}(x) = \left(\Delta^{-n_1} * \beta^{n_1+n_2}\left(\cdot - \frac{n_1}{2}\right)\right)(x). \tag{B11}$$

APPENDIX C: IMPLEMENTATION OF THE IIR FILTER

The computation of equations (16), (19), and (23) is essentially to perform IIR filtering on the signal:

$$f_{\text{out}}(n) = \left(f_{\text{in}} * [(\beta^n)^{-1}]_{\uparrow m}\right)(n), \tag{C1}$$

where $f_{\text{in}}(n)$ is the input discrete signal, $f_{\text{out}}(n)$ is the filtered signal, $m = 2^i$ for equation (19), and $m = 1$ for equations (16) and (23). By performing the z-transform[5] on the equation (C1), we have

$$\mathcal{F}_{\text{out}}(z) = \mathcal{B}_{n_0}(z)\mathcal{F}_{\text{in}}(z), \tag{C2}$$

where $\mathcal{F}_{\text{out}}(z)$, $\mathcal{F}_{\text{in}}(z)$, and $\mathcal{B}_{n_0}(z)$ are the z-transforms of $f_{\text{out}}(n)$, $f_{\text{in}}(n)$, and $[(\beta^n)^{-1}]_{\uparrow m}(n)$, respectively. According to Vrhel et al. (1997), the formula of $\mathcal{B}_{n_0}(z)$ is

$$\mathcal{B}_{n_0}(z) = d_0\prod_{j=1}^{n_0}\mathcal{B}(z; z_j), \tag{C3}$$

in which $\mathcal{B}(z; z_j)$ is defined as

$$\mathcal{B}(z; z_j) = \frac{1}{(1 - z_j z^{-m})}\frac{-z_j}{(1 - z_j z^m)}, \tag{C4}$$

and n_0 is

$$n_0 = \text{Floor}(n/2). \tag{C5}$$

Values of the constant coefficients d_0 and z_j are given in Vrhel et al. (1997).

Therefore, equation (C2) can be expressed as follows

$$\mathcal{F}_0(z) = \mathcal{F}_{\text{in}}(z), \tag{C6}$$

$$\mathcal{F}_j(z) = \mathcal{B}(z; z_j)\mathcal{F}_{j-1}(z), \quad \text{for } 1 \le j \le n_0, \tag{C7}$$

$$\mathcal{F}_{\text{out}}(z) = d_0\mathcal{F}_{n_0}(z). \tag{C8}$$

Combining equation (C4) and (C6)–(C8), we obtain the following recursive filter equations:

$$f_{\text{tem}}(n) = f_{j-1}(n) + z_j f_{\text{tem}}(n-m), \ (n=m, \ldots, N-1) \tag{C9}$$

$$f_j(n) = z_j\left(f_j(n+m) - f_{\text{tem}}(n)\right), \ (n=N-1-m, \ldots, 0) \tag{C10}$$

for the input $f_0(n) = f_{\text{in}}(n)$. Then the output is $f_{\text{out}}(n) = d_0 f_{n_0}(n)$.

To calculate f_j recursively, we need to know $f_{\text{tem}}(n)$ for $n = 0, \ldots, m-1$, and $f_j(n)$ for $n = N-1, \ldots, N-m$. By assuming that f_{j-1} is periodic over N samples, the initial values can be calculated by

$$f_{\text{tem}}(n) = \sum_{l=0}^{N_l} z_j^l f_{j-1}[\text{Mod}(n-lm, N)], \ (n=0, \ldots, m-1), \tag{C11}$$

$$f_j(n) = -\sum_{l=0}^{N_l-1} z_j^{l+1} f_{\text{tem}}[\text{Mod}(n+lm, N)], \ (n=N-1, \ldots, N-m), \tag{C12}$$

[5]Please refer Proakis & Manolakis (2007) for the details of the z-transform.

where $N_l = \ln \epsilon / \ln |z_j|$, $\epsilon = 10^{-16}$ is the pre-specified level of precision, and Mod(a, b) returns the remainder of the division of a by b.

If $n_0 = 1$ and $m = 1$, which is the case for the M02CWT and A19CWT algorithms, then it is also convenient to assume that the signal is zero outside the sampled range. Thus the initial values are

$$f_{\rm tem}(0) = f_{\rm in}(0), \tag{C13}$$

$$f_1(N-1) = -f_{\rm tem}(N-1) \sum_{l=1}^{(N_l+1)/2} z_1^{2l-1}. \tag{C14}$$

To calculate the convolution between f_1 and β^7 (e.g. equations 24 and 35), we also need to know

$$f_1(n) = z_1^{-n} f_1(0), \ (n = -6, \ldots, -1) \tag{C15}$$

$$f_1(N+n) = -f_{\rm tem}(N-1) \sum_{l=1}^{(N_l-n)/2} z_1^{2l+n}, \ (n = 0, 1). \tag{C16}$$

APPENDIX D: ACCURACY TESTS OF THE M02CWT ALGORITHM

Both the M02CWT and A19CWT algorithms calculate cumulative sum of the coefficient sequence $c(l)$ four times (see equation 25). However, repeated cumulative summation can produce floating-points with huge values, which are less precise. Therefore, if we use the sequence of coefficients $g(l)$ that is computed in one go before the scale-dependent operations, i.e. equations (24) and (35), then the algorithms will be terribly imprecise, which is not emphasized in Muñoz et al. (2002) and Arizumi & Aksenova (2019).

As an example, we use the M02CWT algorithm to illustrate the accuracy issue, and denote its variant with computing $g(l)$ globally as the ill-M02CWT. In Fig. D1, we show that especially for the large sampling numbers of $N = 512$ and 1024, the ill-M02CWT yields very high errors at small scales. The M02CWT reduces the errors to a great extent. Although the errors of the long double precision ill-M02CWT are lower, the cost of using it is extremely expensive. As shown in Fig. D2, the CPU time consumed by the M02CWT is almost the same compared to the ill-M02CWT, while the long double precision ill-M02CWT takes tens of times more CPU time than the ill-M02CWT.

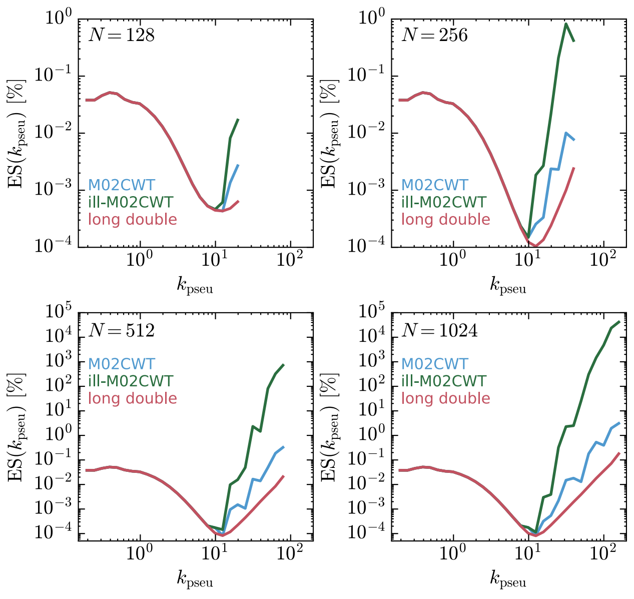


Figure D1. The error spectra of the non-periodic signal $f_2(x)$, based on the CW-GDW at sampling numbers of $N = 128$, 256, 512, and 1024. The blue lines show the results obtained by the M02CWT algorithm with computing the sequence $c(l)$ locally at scale levels $i \geq 1$. The green lines, labelled as 'ill-M02CWT', show the results obtained by the variant of the M02CWT with computing the sequence $c(l)$ globally. The red lines, labelled as 'long double', show the results obtained by the long double implementation of the ill-M02CWT.

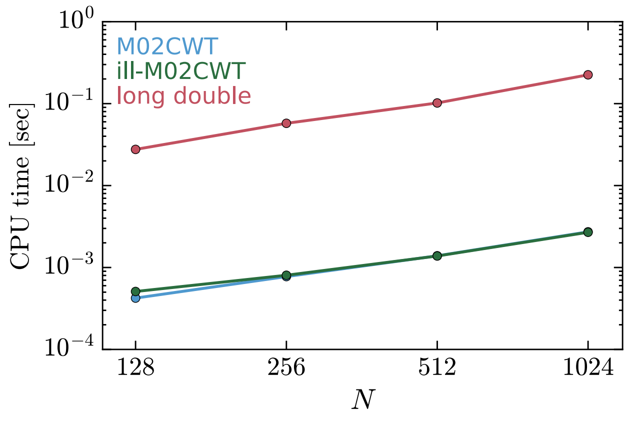


Figure D2. The CPU time per scale of the M02CWT, ill-M02CWT, and the long double precision ill-M02CWT to compute the numerical CWT of the non-periodic signal $f_2(x)$ with the CW-GDW at sampling numbers of $N = 128$, 256, 512, and 1024.

This paper has been typeset from a TEX/LATEX file prepared by the author.