Discrete Weierstrass Fourier Transform
and Experiments

Sheng Zhang

Abstract
We establish a new method called Discrete Weierstrass Fourier Transform to approximate data sets, which is a generalization of Discrete Fourier Transform. The theory of this method as well as some experiments are analyzed in this paper.

1 Introduction

Barnsley, Harding, Vince, and Viswanathan introduced a notion of Weierstrass Fourier Series to approximate rough functions (see [1]).

Key idea of constructing Weierstrass Fourier Series is:
1. Construct a linear operator on $L^p(\mathbb{R})$;
2. Using this linear operator, transform the classical Fourier basis to a new basis;
3. By Gram-Schmidt process, get an orthonormal basis;
4. Using this orthonormal basis, do what we can do in classical Fourier Analysis.

In this paper, we are going to extend their theory by the following steps:
1. Following the key idea in their paper, establish a new expression of Weierstrass Fourier Series, which is more suitable for subsequent discussions in this paper.
2. Deduce the notion of Discrete Weierstrass Fourier Transform.
3. Provide some numerical examples to test this transform.

2 Weierstrass Fourier Series

In classical Fourier Analysis, $\{e^{2\pi ikx}\}_{k \in \mathbb{Z}}$ is a complete orthonormal basis for $L^2([0,1])$, and we can approximate functions with Fourier Series. Barnsley, Harding, Vince, and Viswanathan constructed another orthonormal basis for $L^2([0,1])$, and established a theory for approximating functions with Weierstrass Fourier Series. A key result in their paper is listed below (see Theorem 2.1).
Theorem 2.1 (Barnsley et al.) (see [1], Corollary 3.1). Assume that \( p \in [1, \infty], \) \( a, b \in \mathbb{R}, \) \( b \neq 0, \) and \( |a| \neq |b|^{\frac{1}{p}}. \) For any \( g \in L^p(\mathbb{R}), \) there is a unique solution \( f \in L^p(\mathbb{R}) \) to the equation

\[
f(x) - af(bx) = g(x),
\]

and the solution is given by the following series, which are absolutely and uniformly convergent in \( L^p(\mathbb{R}) \):

\[
f(x) = \begin{cases} 
\sum_{m=0}^{\infty} a^m g(b^m x) & \text{if } |a| < |b|^{\frac{1}{p}} \\
- \sum_{m=1}^{\infty} \left( \frac{1}{a} \right)^m g(x) & \text{if } |a| > |b|^{\frac{1}{p}}.
\end{cases}
\]

The following corollary is a direct conclusion from this theorem.

Corollary 2.2. Assume that \( a \in [0, 1) \) and \( b = 2. \) If \( g \in L^\infty(\mathbb{R}), \) then (1) has a unique solution \( f \in L^\infty(\mathbb{R}), \) and the solution is given by the following series, which are absolutely and uniformly convergent:

\[
f(x) = \sum_{m=0}^{\infty} a^m g(2^m x).
\]

Furthermore, \( f \in L^2([0, 1]). \)

Proof. Let \( p = \infty \) in Theorem 2.1 and the first part of the corollary is proved. As \( f \in L^\infty(\mathbb{R}), f \in L^\infty([0, 1]). \) So, \( f \in L^2([0, 1]). \)

In paper [1], Barnsley et al. introduced Weierstrass Fourier Series by substituting \( g(x) = \sin kx \) and \( \cos kx. \) However, another form shown below is more suitable for subsequent discussions in this paper.

In [1], assume that \( a \in [0, 1) \) and \( b = 2. \) For each \( k \in \mathbb{Z}, \) let \( g(x) = e^{2\pi ikx} \). By Corollary 2.2, there exists a unique solution \( f_k(x) = \sum_{m=0}^{\infty} a^m e^{2\pi ik^m x} \in L^2([0,1]). \) By normalizing \( \{f_k\}, \) a normalized basis \( \{\hat{e}_k\} \) for \( L^2([0,1]) \) is established:

\[
\hat{e}_k = \begin{cases} 
1 & \text{if } k = 0 \\
\sqrt{1 - a^2} \sum_{m=0}^{\infty} a^m e^{2\pi ik^m x} & \text{if } k \neq 0.
\end{cases}
\]

By Gram-Schmidt process, an orthonormal basis \( \{\hat{e}_k\} \) for \( L^2([0,1]) \) is obtained:

\[
\hat{e}_k = \begin{cases} 
1 & \text{if } k = 0 \\
\hat{e}_k & \text{if } k \text{ is odd} \\
\frac{\hat{e}_k - a \hat{e}_k/2}{\sqrt{1 - a^2}} & \text{if } k \text{ is even and } k \neq 0.
\end{cases}
\]

Notice that when \( a = 0, \) \( \hat{e}_k = e_k \) for all \( k \in \mathbb{Z}. \) To sum up:
Theorem 2.3. Given \( a \in [0,1) \), the set of functions \( \{\tilde{e}_k\}_{k \in \mathbb{Z}} \) is an orthonormal basis for \( L^2([0,1]) \).

Weierstrass Fourier coefficients for a function \( f \) can be calculated as follows:

\[
\tilde{\alpha}_k = \langle f, \tilde{e}_k \rangle = \begin{cases} 
\alpha_0 & \text{if } k = 0 \\
\sqrt{1-a^2} \sum_{m=0}^{\infty} a^{m} \alpha_{2m} & \text{if } k \text{ is odd} \\
\frac{(1-a^2) \sum_{m=0}^{\infty} a^{m} \alpha_{2m} - a \alpha_{k/2}}{\sqrt{1-a^2}} & \text{if } k \text{ is even and } k \neq 0,
\end{cases}
\]

where \( \alpha_k \) is the \( k \)th Fourier coefficient. Notice that when \( a = 0 \), \( \tilde{\alpha}_k = \alpha_k \) for all \( k \in \mathbb{Z} \). Now, we can approximate any function in \( L^2([0,1]) \) using Weierstrass Fourier Series.

Theorem 2.4. With the notations above, given \( a \in [0,1) \), if \( f \in L^2([0,1]) \) has Fourier expansion

\[
f(x) = \sum_{k=-\infty}^{\infty} \alpha_k e_k(x),
\]

then it has Weierstrass Fourier expansion

\[
f(x) = \sum_{k=-\infty}^{\infty} \tilde{\alpha}_k \tilde{e}_k(x).
\]

In particular, when \( a = 0 \), two expansions are the same.

Proposition 2.5. With the notations above, given \( a \in [0,1) \), if \( f \in L^2([0,1]) \) is a real function, then the following statements hold:

1. \( \overline{\tilde{\alpha}_k} = \tilde{\alpha}_{-k} \) for all \( k \in \mathbb{Z} \);
2. \( \overline{\tilde{e}_k} = \tilde{e}_{-k} \) for all \( k \in \mathbb{Z} \);
3. \( \sum_{k=-n}^{n} \tilde{\alpha}_k \tilde{e}_k(x) \) is real and \( \sum_{k=-n}^{n} \tilde{\alpha}_k \tilde{e}_k(x) = \alpha_0 + 2Re(\sum_{k=1}^{n} \tilde{\alpha}_k \tilde{e}_k(x)) \) for all \( n \in \mathbb{Z}_+ \).

3 Discrete Weierstrass Fourier Transform

With the notations above, given \( a \in [0,1) \), for any fixed \( n \in \mathbb{N}_+ \), let \( A \) be an \( n \times n \) matrix such that

\[
A_{ij} = \tilde{e}_j(i/n), 0 \leq i,j \leq n - 1.
\]  

Given \( n \) data points \( b_0, b_1, b_2, \ldots, b_{n-1} \in \mathbb{C} \), let \( b = (b_0, b_1, b_2, \ldots, b_{n-1})^T \). Discrete Weierstrass Fourier Transform (DWFT) is defined as the linear operator on \( \mathbb{C}^n \): \( dwft(b) = A^{-1} b \). Inverse Discrete Weierstrass Fourier Transform (IDWFT) is defined as the linear operator on \( \mathbb{C}^n \): \( idwft(b) = A b \).
Theorem 3.1. Given \( a \in [0, 1) \), \( dwft \) and \( idwft \) defined above are inverse linear operators on \( \mathbb{C}^n \). In particular, when \( a = 0 \), \( dwft \) is classical Discrete Fourier Transform and \( idwft \) is classical Inverse Discrete Fourier Transform.

Like DFT, DWFT is useful for data compression. Fix \( n \in \mathbb{N}_+ \). Let \( k \in \mathbb{N}_+ \) and \( 1 \leq k \leq \frac{n+1}{2} \) if \( n \) is odd, \( 1 \leq k \leq \frac{n+2}{2} \) if \( n \) is even. Assume \( b = (b_0, b_1, b_2, \cdots, b_{n-1})^T \) is a set of data. Do DWFT on \( b \) and get
\[
c = dwft(b) = (c_0, c_1, c_2, \cdots, c_{n-1})^T.
\]
Change \( c_k, c_{k+1}, \cdots, c_{n-k} \) to 0 and get
\[
c'(k) = (c_0, c_1, \cdots, c_{k-1}, 0, \cdots, 0, c_{n-k+1}, \cdots, c_{n-1})^T.
\]
Do IDWFT on \( c'(k) \) and get
\[
b'(k) = idwft(c'(k)) = (b'_0, b'_1, b'_2, \cdots, b'_{n-1})^T.
\]
When \( b = (b_0, b_1, b_2, \cdots, b_{n-1})^T \) is real, \( b'(k) \) we got above is not necessarily real. However, we always want to get real approximation for real data. So, if the data are real, a last procedure that discarding the imaginary part of \( b'(k) \) should be taken, say
\[
b''(k) = Re(b'(k)).
\]
Say \( b''(k) \) is the approximation of \( b \) using \( k \) terms using DWFT. Nevertheless, the imaginary part of \( b'(k) \) is usually very small compared to its real part.

Now, if \( b''(k) \) is very closed to \( b \) even if \( k \) is very small compared to \( n \), we can store \( c'(k) \), \( a \), and \( n \) instead of data \( b \). Then we can calculate matrix \( A \) as well as \( b''(k) \) when we need and use \( b''(k) \) as an approximation of \( b \). In this way, plenty of space is saved.

If DWFT and IDWFT are replaced by DFT and IDFT, then the approximation using DFT is obtained.

By Theorem 3.1, DWFT is a generalization of DFT. Since we can choose a suitable \( a \) for each set of data, DWFT will never be worse than DFT in approximation.

4 Experiments

In this section, some data sets are approximated using DFT and DWFT. With the notations in [4, 6, 7, 8], given data vector \( b \) and \( a \in [0, 1) \), error vector for \( k \) terms is defined as
\[
\vec{E}_k = b''(k) - b,
\]
and error function for \( f \) is defined as the function
\[
\mu(k) = ||\vec{E}_k||_2,
\]
which maps \( k \) to the 2-norm of \( \vec{E}_k \), for \( k \in \{1, 2, \ldots, \frac{n}{2} + 1\} \). In particular, \( \mu(\frac{n}{2} + 1) = 0 \) by Theorem 3.1.
4.1 Data from real functions

In this part, the number of data $n = 1024$ for all examples, and each data vector
$b = (f(0), f(\frac{1}{n}), f(\frac{2}{n}), \ldots, f(\frac{n-1}{n}))^T$ for some real-valued function $f$ on $[0, 1]$.

4.1.1 Linear function

\[ f(x) = x - 0.5 \text{ and } a = 0.5 \]

Figure 1.1.1 The figure of the error function for $f(x) = x - 0.5$ with $a = 0.5$. The blue curve represents approximation using DFT, and the green curve represents approximation using DWFT. DWFT is always better.
Figure 1.1.2 Approximation of $f(x) = x - 0.5$ using 10 terms. The red curve represents the original data, the blue curve represents approximation values using DFT, and the green curve represents approximation values using DWFT. See also Figure 1.1.3 and Figure 1.1.4.

Figure 1.1.3 Approximation of $f(x) = x - 0.5$ using 50 terms.
Figure 1.1.4 Approximation of $f(x) = x - 0.5$ using 300 terms.

4.1.2 Triangular function with low frequency and high frequency

\[ f(x) = \sin x + 0.01 \sin 105x \text{ and } a = 0.5 \]

Figure 1.2.1 The figure of the error function for $f(x) = \sin x + 0.01 \sin 105x$ with $a = 0.5$. The blue curve represents approximation using DFT, and the green
curve represents approximation using DWFT. DWFT is better at the first 510 terms.

Figure 1.2.2 Approximation of \( f(x) = \sin x + 0.01 \sin 105x \) using 10 terms. The red curve represents the original data, the blue curve represents approximation values using DFT, and the green curve represents approximation values using DWFT. See also Figure 1.2.3 and Figure 1.2.4.
Figure 1.2.3 Approximation of \( f(x) = \sin x + 0.01 \sin 105x \) using 50 terms.

Figure 1.2.4 Approximation of \( f(x) = \sin x + 0.01 \sin 105x \) using 300 terms.

4.1.3 Discontinuous function

\[
    f(x) = \begin{cases} 
    0 & x \in [0, \frac{1}{2}] \\
    1 & x \in \left(\frac{1}{2}, 1\right] \text{ and } a = 0.5
    \end{cases}
\]
Figure 1.3.1 The figure of the error function for $f(x) = \begin{cases} 0 & x \in [0, \frac{1}{2}] \\ 1 & x \in (\frac{1}{2}, 1] \end{cases}$ with $a = 0.5$. The blue curve represents approximation using DFT, and the green curve represents approximation using DWFT. DWFT is not better.

Figure 1.3.2 Approximation of $f(x) = \begin{cases} 0 & x \in [0, \frac{1}{2}] \\ 1 & x \in (\frac{1}{2}, 1] \end{cases}$ using 10 terms. The red curve represents the original data, the blue curve represents approximation values using DFT, and the green curve represents approximation values using DWFT. See also Figure 1.3.3 and Figure 1.3.4.
Figure 1.3.3 Approximation of $f(x) = \begin{cases} 0 & x \in [0, \frac{1}{2}] \\ 1 & x \in (\frac{1}{2}, 1] \end{cases}$ using 50 terms.

Figure 1.3.4 Approximation of $f(x) = \begin{cases} 0 & x \in [0, \frac{1}{2}] \\ 1 & x \in (\frac{1}{2}, 1] \end{cases}$ using 300 terms.
4.1.4 Rough function

\[ f(x) = \sum_{k=0}^{\infty} 0.42^k \cos(\pi \cdot 2^k x) \text{ and } a = 0.42 \]

Figure 1.4.1 The figure of the error function for \( f(x) = \sum_{k=0}^{\infty} 0.42^k \cos(\pi \cdot 2^k x) \) with \( a = 0.42 \). The blue curve represents approximation using DFT, and the green curve represents approximation using DWFT. DWFT is better at the first 454 terms.
Figure 1.4.2 Approximation of \( f(x) = \sum_{k=0}^{\infty} 0.42^k \cos(\pi \cdot 2^k x) \) using 10 terms. The red curve represents the original data, the blue curve represents approximation values using DFT, and the green curve represents approximation values using DWFT. See also Figure 1.4.3 and Figure 1.4.4.

Figure 1.4.3 Approximation of \( f(x) = \sum_{k=0}^{\infty} 0.42^k \cos(\pi \cdot 2^k x) \) using 50 terms.

Figure 1.4.4 Approximation of \( f(x) = \sum_{k=0}^{\infty} 0.42^k \cos(\pi \cdot 2^k x) \) using 300 terms.
4.2 Discrete data

In this part, data are practical data from some websites. Given data vector \( b \) and \( a \in [0, 1) \), error vector and error function for the data are defined in [9] and [10].

4.2.1 Stock price of Commonwealth Bank of Australia

Daily open prices of Commonwealth Bank of Australia from Sep 30, 2010. Totally 1024 data points. Source from https://au.finance.yahoo.com/.

![Figure 2.1.1 The figure of the error function for 1024 daily open prices of Commonwealth Bank of Australia with \( a = 0.3 \). The blue curve represents approximation using DFT, and the green curve represents approximation using DWFT. DWFT is better at the first 88 terms.](image-url)
Figure 2.1.2 Approximation of 1024 daily open prices of Commonwealth Bank of Australia using 10 terms. The red curve represents the original data, the blue curve represents approximation values using DFT, and the green curve represents approximation values using DWFT. See also Figure 2.1.3 and Figure 2.1.4.

Figure 2.1.3 Approximation of 1024 daily open prices of Commonwealth Bank of Australia using 50 terms.
4.2.2 Water level of Alameda in California

Water level of Alameda in California per hour from May 29, 2014. Totally 1024 data points. Source from http://tidesandcurrents.noaa.gov/.
Figure 2.2.1 The figure of the error function for 1024 water levels of Alameda in California per hour from May 29, 2014 with $a = 0.3$. The blue curve represents approximation using DFT, and the green curve represents approximation using DWFT. DWFT is not better.

Figure 2.2.2 Approximation of 1024 water levels of Alameda in California using 10 terms. The red curve represents the original data, the blue curve represents approximation values using DFT, and the green curve represents approximation values using DWFT. See also Figure 2.2.3 and Figure 2.2.4.
Figure 2.2.3 Approximation of 1024 water levels of Alameda in California using 50 terms.

Figure 2.2.4 Approximation of 1024 water levels of Alameda in California using 300 terms.
4.3 Comments

4.3.1

Figure 1.1.2 and Figure 1.1.3 in Section 4.1.1 coincide with Figure 1, Figure 2, Figure 3 and Figure 4 of [1], which means that the discrete algorithm in this paper gives the same result as the continuous algorithm when the number of the data is much bigger than the terms of approximation.

4.3.2

In the figures above we can see that at some of the data points, DWFT approximates better, while at some other data points DFT is better. For example, in Figure 1.1.3, at the data points near 0.0 or 1.0, DWFT is better while at the data points near 0.5, it is significantly worse, but overall, it approximates the data better in the norm defined above.

4.3.3

We can also see a characteristic that approximation values using DWFT shakes heavily at the points near $\frac{1}{2}$, less heavily at the points near $\frac{1}{4}$, $\frac{3}{4}$, and less heavily at the points near $\frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}, \cdots$.

4.3.4

In the above examples, DWFT is better than DFT for some smooth continuous functions. For some other smooth functions such as $f(x) = x^2$ with $a = 0.5$ and $f(x) = e^x$ with $a = 0.5$, DWFT is also better than DFT. But for $f(x) = x(x-1)$ with $a = 0.5$, DWFT behaves very bad.

4.3.5

For triangle functions with low frequency and high frequency such as the example in Section 4.1.2, $f(x) = \sin x + 0.01 \cos 100x + 0.01 \cos 200x$ with $a = 0.5$ and $f(x) = 10 \sin 0.1x - \cos x + 0.01 \sin 100x + 0.02 \cos 200x$ with $a = 0.5$, DWFT is better than DFT.

However, make sure that the low frequencies are low enough with coefficients big enough compared to the high frequencies, and the low frequencies are not counteracted each other. For instance, DWFT doesn’t work better than DFT for the function $f(x) = 10 \sin 0.1x - \sin x + 0.01 \sin 100x + 0.02 \cos 200x$ with $a = 0.5$ because the low frequency terms “$10 \sin 0.1x$” and “$-\sin x$” are counteracted each other.

4.3.6

For discontinuous functions, DWFT doesn’t seem better than DFT. As we can see in Section 4.1.3, approximation values using DWFT for discontinuous function shake heavily.
4.3.7
As we can see in Section 4.1.4, DWFT has some advantages in approximating self-similar rough functions.

4.3.8
As we can see in Section 4.2, DWFT seems not good for approximating practical data, because practical data are not perfectly self-similar. A small shake in the data can induce shakes everywhere in the approximation values using DWFT. In this case, the approximation values using DWFT look much rougher than the data.

4.3.9
If some data set can be approximated better using DWFT, we have a better way to compress the data. However, the calculation of DWFT is much more complicated than DFT. We might wish to find a fast way to calculate DWFT.
About The Author

Sheng Zhang, a mathematics undergraduate student in Zhejiang University, China, was exchanged to Australian National University in the second half year of 2014, where he met his first teacher in Fractal Geometry, Professor Michael F. Barnsley. His another work, *Five Sierpinski Triangles in A Line Is Not The Attractor of Any Contractive IFS Consisting of Similitudes*, is on ArXiv http://arxiv.org/abs/1504.06801.

Acknowledgments

There are so many people who have helped me in this paper in indispensable ways. I would like to mention especially and express my deepest gratitude to my first teacher in Fractal Geometry, Professor Michael F. Barnsley, who guided me to the fancy world of Fractal Geometry in his wonderful course at the first time, raised the problem in this paper, encouraged me to work it out, and gave me a lot of advice in every aspect; Brendan Harding, who taught me programming in Python, helped me to improve my algorithm, so that I could make beautiful figures in this paper; P. Viswanathan, who discussed theoretical issues about Fractal Geometry and Fourier Analysis with me. I would like to extend my sincere gratitude to Qiushi Elite Program Mathematics major, Zhejiang University, China, and Australian National University Summer Research Program, which financially supported my study in Australia.

References

[1] M. F. Barnsley, B. Harding, A. Vince, and P. Viswanathan, *Approximation of Rough Functions*, ArXiv : 1412.3871v3 [math.FA] 12 Dec 2014.