Peer-reviewed version available at Computation 2018, 6, 61; doi:10.3390/computation6040061

Article

Coefficient-of-determination Discrete Fourier Transform

Matthew Marko*

Naval Air Warfare Center Aircraft Division Joint-Base McGuire-Dix-Lakehurst, Lakehurst NJ 08733, USA [1]
* Correspondence: matthew.marko@navy.mil

Abstract: This algorithm is designed to perform Discrete Fourier Transforms (DFT) to convert temporal data into spectral data. This algorithm obtains the Fourier Transforms by studying the Coefficient of Determination of a series of artificial sinusoidal functions with the temporal data, and normalizing the variance data into a high-resolution spectral representation of the time-domain data with a finite sampling rate. What is especially beneficial about this DFT algorithm is that it can produce spectral data at any user-defined resolution.

Keywords: Fourier Transform; Spectral Domain; Resolution; Coefficient of Determination

1. Introduction

The *Fourier Transform* [1–7] is one of the most widely used mathematical operators in all of engineering and science [8–10]. The Fourier Transform can take a temporal function and convert it into a series of sinusoidal functions, offering significant clarity on the nature of the data. While the original Fourier Transform is an analytical mathematical operator, *Discrete Fourier Transform* (DFT) methods are overwhelmingly used to take incoherent temporal measurements and convert them into spectral plots based on real, experimental data.

The author proposes a numerical algorithm to perform a highly-resolved Fourier Transform of a temporal function of limited resolution. The spectral magnitude is determined by finding the magnitude of the Coefficient of Determination of the function as compared with a given sinusoidal function; this represents the independent spectral value as a function of the sinusoidal frequency. Rather than the spectral domain being proportional to the time step, the user defines exactly which frequencies are necessary to investigate. The spectral domain can be as large or as resolved as is necessary; the resolution possible is limited only by the abilities of the computer performing the transform.

2. Fourier Transform

The transform starts by first determining the peak total range of the data in the temporal domain, this range will become the base amplitude of the spectral series. The computer then generates a series of sine and cosine functions at each frequency within the spectral domain, and compares each of these sinusoidal functions to the temporal data to be transformed. In the comparison, a correlation coefficient is found and saved. To accommodate fluctuations in phase, each frequency generates both a sine and cosine function; this ultimately results in real and imaginary spectral components. Finally,

[1] NAVAIR Public Release 2016-755 Distribution Statement A - "Approved for public release; distribution is unlimited"

Peer-reviewed version available at Computation 2018, 6, 61; doi:10.3390/computation6040061

the magnitudes of the correlation factor data is normalized, and the result is an accurate spectral representation of the temporal function.

The Fourier Transform is one of the most utilized mathematical transforms in science and engineering. By definition, a Fourier Transform will take a given function and represent it by a series of sinusoidal functions of varying frequencies and amplitudes. Analytically, the Fourier Transform is represented as [1,2]

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-2\pi \cdot i \cdot t \cdot \omega} dt, \quad (1)$$

where i is the imaginary term ($i = \sqrt{-1}$), $f(t)$ is any temporal function of t to be transformed, and ω (rad/s) represents the frequency of each sinusoidal function. The inverse of this function is

$$f(t) = \int_{-\infty}^{\infty} F(\omega) \cdot e^{2\pi \cdot i \cdot t \cdot \omega} d\omega. \quad (2)$$

Conceptually, the spectral function $F(\omega)$ represents the amplitudes of a series of sinusoidal functions of frequency ω (rad/s)

$$f(t) = \Sigma_{n=0}^{\infty} F(\omega_n) \cdot sin(\omega_n \cdot t). \quad (3)$$

Often in practical application, one does not have an exact analytical function, but a series of discrete data points. If it is necessary to convert this discrete data into the spectral domain, the traditional approach has been to use the DFT algorithm, often known as *Fast Fourier Transform* (FFT). The DFT algorithm is, by definition [11,12]

$$F_k = \Sigma_{n=0}^{N-1} x_n \cdot e^{-2\pi \cdot i \cdot k \cdot n / N}. \quad (4)$$

where F_k is a discrete spectral data point, and x_n is a discrete data point in the temporal domain. With DFT, the spectral resolution is proportional to the temporal resolution, and it is often the case that the limited temporal data will not be sufficient to obtain the spectral resolution desired.

If one wants to obtain frequency information, there is a certain minimum temporal resolution necessary to properly distinguish the frequencies; this is known as the Nyquist rate [13–17].

$$\delta f = \frac{0.5}{\delta t} \quad (5)$$

As demonstrated in Table 1 and Figure 1, two different cosine functions with frequencies of *1* and *9* have exactly the same results when resolved at a temporal resolution $\delta t = 0.1$.

Table 1. Equal values for *f=1* and *f=9* for $cos(2\pi \cdot f \cdot x)$.

x	f=1	f=9
0.0	1.0000	1.0000
0.1	0.8090	0.8090
0.2	0.3090	0.3090
0.3	-0.3090	-0.3090
0.4	-0.8090	-0.8090
0.5	-1.0000	-1.0000
0.6	-0.8090	-0.8090
0.7	-0.3090	-0.3090
0.8	0.3090	0.3090
0.9	0.8090	0.8090
1.0	1.0000	1.0000

Peer-reviewed version available at Computation 2018, 6, 61; doi:10.3390/computation6040061

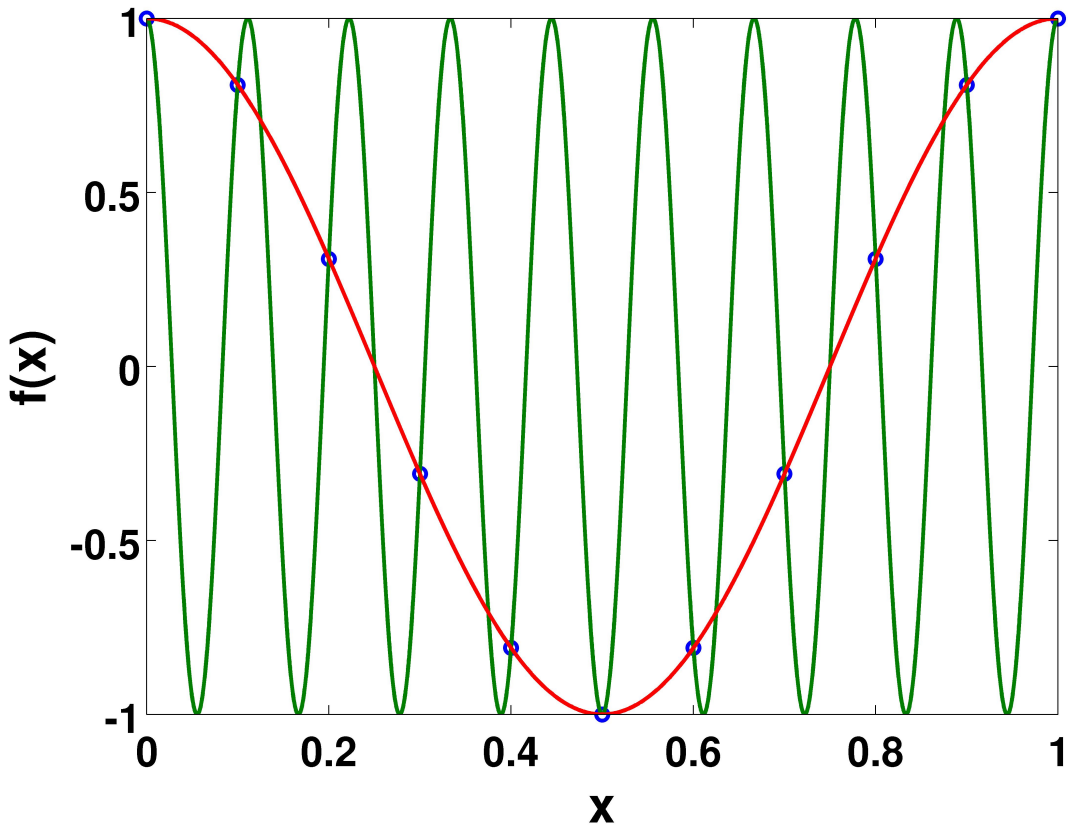

Figure 1. Equal values for f=1 and f=9 for $cos(2\pi \cdot f \cdot x)$.

There are many approaches to implementing Fourier transforms on data of limited resolution. One method is to introduce a scaled coordinate system and identifying the Fourier variables as the direction cosines of propagating light have been used to spectrally characterize diffracted waves in a method known as Angular Spectrum Fourier transform (FFT-AS) [18–21]. Another technique of numerical Fourier Transform is Direct Integration (FFT-DI) [22], using Simpson's rule to improve the calculations accuracy. Finally, one of the simplest approaches to taking the Fourier transform with a limited temporal resolution is to use Non-uniform Discrete Fourier Transforms (NDFT) [23–31]

$$F_k = \Sigma_{n=0}^{N-1} x_n \cdot e^{-2\pi \cdot i \cdot p_n \cdot \omega_k}. \tag{6}$$

where $0 < p_n < 1$ are relative sample points over the range, and ω_k is the frequency of interest.

3. Algorithm

This algorithm, which the author calls the *Coefficient of Determination Fourier Transform* (CDFT), is an approach to obtain greater spectral resolution; the full spectral domain, or any frequency range or resolution desired, is determined by the user. Greater resolution or a larger domain will inherently take longer to solve, depending on the computer resources available. One advantage of this approach is that the spectral domain can also have varying resolutions, for enhanced resolution at points of interest without dramatically increasing the computation cost of each Fourier Transform.

At each discrete point in the spectral domain, the algorithm generates two sinusoidal functions

$$\begin{aligned} F_n(t) &= A \cdot cos(2\pi \cdot \omega_n \cdot t), \\ \hat{F}_n(t) &= A \cdot sin(2\pi \cdot \omega_n \cdot t), \end{aligned} \tag{7}$$

where $F_n(t)$ is to represent the real spectral components, $\hat{F}_n(t)$ is to represent the imaginary spectral components, ω_n is the discrete frequency of interest, t is the independent variable of the data of interest, and A is the amplitude of the function,

$$A = max\{f(t)\} - min\{f(t)\}, \tag{8}$$

defined and the total range within the temporal data.

The next step is to take each of these functions, and find the *Coefficient of Determination* (CoD) between the function and the temporal data, all with the same temporal domain and resolution [9,32–35]. The CoD is a numerical representation of how much variance can be expected between two functions. To find the CoD between two equal-length discrete functions $F_n(t)$ and $f_n(t)$, three coefficients are first calculated

$$\begin{aligned} SS_t &= \Sigma_{n=1}^{N} (F_n(t) - \bar{F_n}) \cdot (f_n(t) - \bar{f_n}), \\ SS_1 &= \Sigma_{n=1}^{N} (F_n(t) - \bar{F_n})^2, \\ SS_2 &= \Sigma_{n=1}^{N} (f_n(t) - \bar{f_n})^2, \end{aligned}$$

where N is the discrete length of the two functions, and $\bar{F_n}$ and $\bar{f_n}$ represent the arithmatic mean value of functions F_n and f_n. The CoD, represented as R, is then determined as

$$R = \frac{SS_t}{\sqrt{SS_1 \cdot SS_2}}, \tag{9}$$

Peer-reviewed version available at Computation 2018, 6, 61; doi:10.3390/computation6040061

and the closer the two functions match, the closer the value of the CoD reaches 1. If there is no match at all, the CoD will be equal to 0, and if the two functions are perfectly opposite of each other ($F_n = -f_n$), the CoD goes up to -1. In practice, the CoD is often represented as the R^2 value,

$$R^2 = \frac{SS_t^2}{SS_1 \cdot SS_2}. \quad (10)$$

This process is repeated for every sine and cosine function generated with each frequency within the spectral domain. The coefficients of determinations can be used to represent the spectral values, both real (cosine function) and imaginary (sine functions), for the given discrete frequency point. These functions of R values for the real and imaginary components are then normalized to the maximum real and imaginary values, and multiplied by the amplitude A determined in equation 8. The final outcome is a phase-resolved spectral transformation of the input function, but with a spectral domain as large or resolved as desired.

Finally, this spectral transformation can easily be converted back to the temporal domain. By definition, the temporal domain is merely the sum of the series of sinusoidal waves, and thus the inverse Fourier transform can simply be defined as

$$f(t) = \Sigma_{m=1}^{N} \{real(F_m) \cdot cos(\omega_m \cdot t)\} + \{imag(F_m) \cdot sin(\omega_m \cdot t)\}. \quad (11)$$

4. Initial Demonstration of Algorithm

To demonstrate the capability of this algorithm, six functions are generated based on the two similar functions demonstrated in Table 1 and Figure 1; the two functions are used with a temporal range of 0 to 1, with the same temporal resolution of $\delta t = 0.1$, and frequencies of both *f=1* and *f=9*. The cosine functions are modified to have a phase shift of $\pm 2\pi/3$.

The Fourier transform was taken of all six of these functions with both the CDFT algorithm, as well as the NDFT algorithm defined in equation 6. The spectral magnitude and phase from both methods are plotted in Figure 2. By using equation 11 to get back to the temporal domain, all six functions matched with the spectral magnitude and phase from the CDFT; there is no coherent match for the NDFT. This is realized by finding the coefficient of determination between the recovered temporal data and the original temporal data; the R^2 results are tabulated in Table 2. While the NDFT may give a clear picture of the spectral domain of the function, it is impossible to recover the function back to the original temporal domain without excessively computationally intensive matrix analysis. The strength of CDFT transform lies in its inverse operator defined in equation 11, with which the true temporal function can be obtained back from the spectral domain obtained with highly resolved CDFT.

Table 2. Correlation between the original temporal function and the temporal function retrieved (equation 11) from the spectral plot obtain with both CDFT and NDFT (Figures 2).

f (Hz)	Phase	R^2 CDFT	R^2 NDFT
1	0	0.9999	0.1197
1	$2 \cdot \pi/3$	0.9967	0.0091
1	$-2 \cdot \pi/3$	0.9964	0.0163
9	0	0.9999	0.1197
9	$2 \cdot \pi/3$	0.9964	0.0163
9	$-2 \cdot \pi/3$	0.9967	0.0091

5. Parametric Study

A parametric study of this transform was conducted, to demonstrate that it can be used for high resolution measurements of the spectral frequency with a limited temporal resolution. To demonstrate this, 15 random frequencies were selected, ranging from 2 to 17 cycles over the duration of the

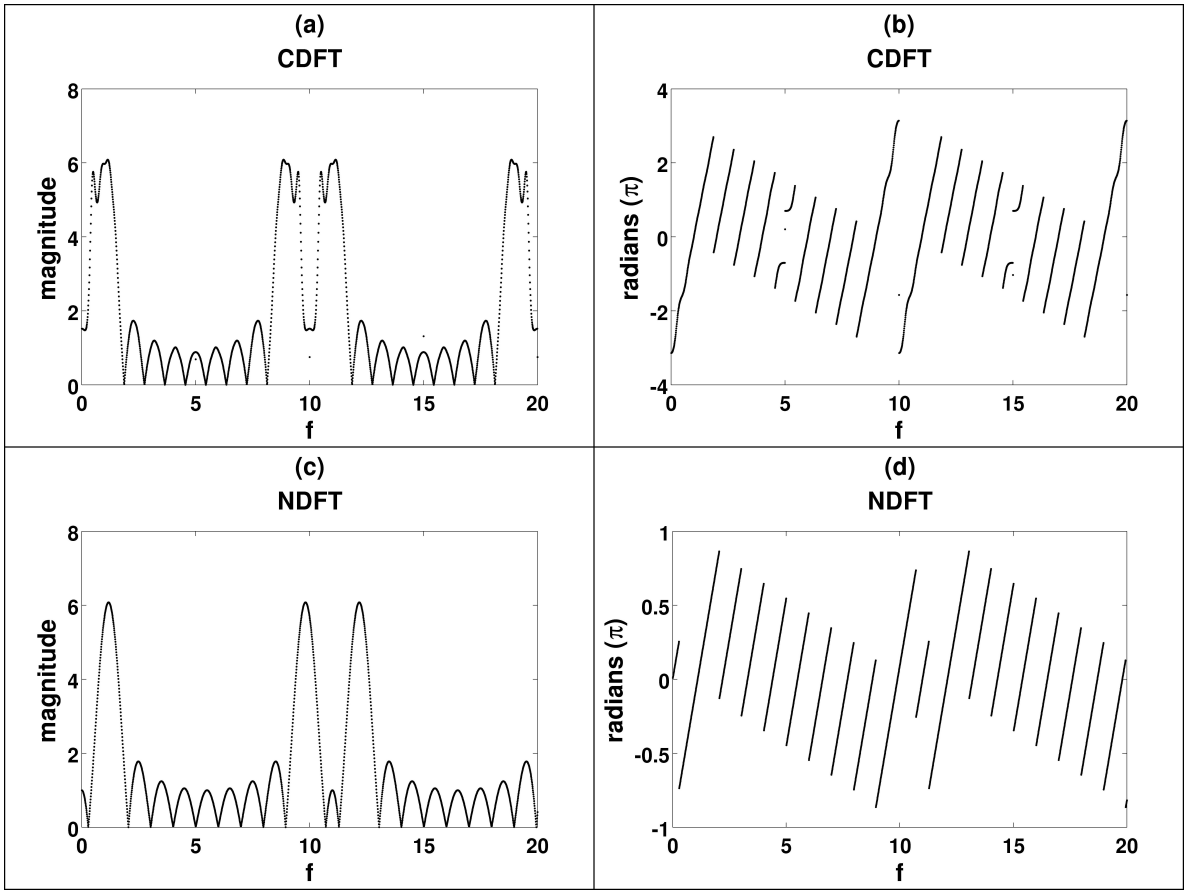

Figure 2. Spectral results of the function $cos(2\pi \cdot x)$ with a δs=0.1, with both the proposed CDFT in magnitude (a) and phase (b), as well as NDFT in magnitude (c) and phase (d).

Peer-reviewed version available at Computation 2018, 6, 61; doi:10.3390/computation6040061

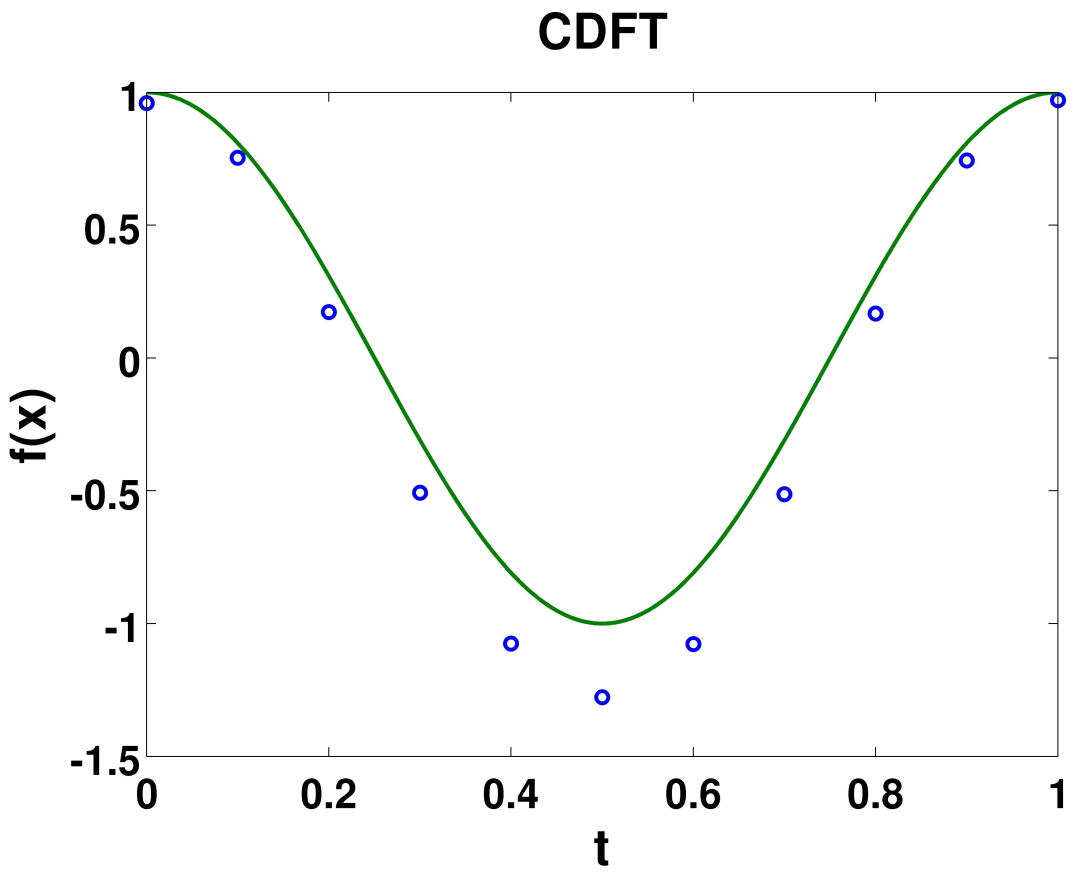

Figure 3. Demonstration of the function $cos(2\pi \cdot t)$, both the original function (solid green lines), and the output (blue circles) obtained from equation 11 and the CDFT spectral results, obtained with a limited initial temporal resolution of $\delta t = 0.1$.

Peer-reviewed version available at Computation 2018, 6, 61; doi:10.3390/computation6040061

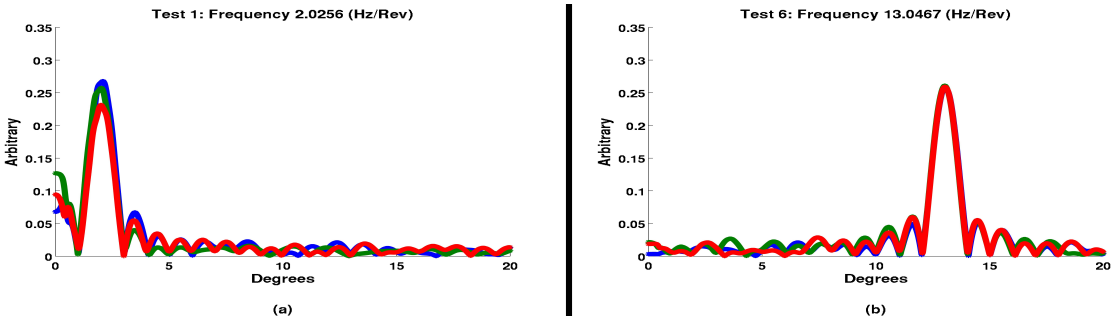

Figure 4. Spectral results of the randomly generated functions, for frequencies of (a) 2.0256 (Hz/Rev) and (b) 13.0467 (Hz/Rev), but for different phases, magnitudes, and random noises.

measured window. Both the independent and dependent temporal variables are arbitrary values to demonstrate the transform function; the independent scale ranges from 0 to 1 and has 180 data points. The arbitrary dependent data had random averages between -1000 and 1000, with an amplitude of 200 and random noise to represent the typical randomness found in typical test data. Each of these 15 random frequencies was phase shifted by three random phases. All forty-five arbitrary functions were transformed into the spectral domain with this transform, with a frequency domain ranging from 0 to 20 cycles per unit time duration, and a frequency resolution of 1 mHz; two examples of these spectral results are presented in Figure 4. As a further test of the robustness of the transform, the spectral data was then converted back to the temporal domain, and the new temporal function was compared to the original function with the coefficient of determination method to ascertain errors from the transform.

This Fourier transform was remarkably effective at finding the peak primary frequency, often with accuracy's down to tens of mHz. The functions of the peak frequencies (Figure 5), both which were used for the initial function and the peak of the Fourier transform, matches with an R^2 value of 0.999991; effectively identical. The functions of the random phase angle at the peak frequencies (Figure 6), both which were used for the initial function and the phase of the Fourier transform at the peak frequency, matches with an R^2 value of 0.9982; demonstrating that this transform can be used to capture both spectral magnitude and phase with great accuracy.

Finally, the inverse of this Fourier transform was conducted for each spectral output, and the errors between the original functions and the transformed-inverse-transformed function are minimal. As expected, not all of the fine random noise is captured; this would require a near infinite spectral domain, which would further increase computational costs, but the overarching shapes, magnitudes, and phases of the functions are consistently captured. Taking the coefficient of determination squared of each function pair, the value of R^2 is never less than 0.92. Two examples of the original function (lines) and the transformed-inverse-transformed function (stars) are represented in Figure 7. The tabulated results of all fifteen studies, for each of the three phase magnitude shifts, are demonstrated in Tables 3 - 5.

6. Conclusion

This effort has demonstrated a practical, working, invertible method of numerically conducting a Fourier Transform, obtaining high-resolution in the spectral domain from limited resolution in the temporal domain, and retaining the ability to go back to the temporal domain from the spectral data with equation 11. While the CDFT transform inherently is more computationally expensive than traditional DFT and NDFT methods, any desired spectral resolution and spectral domain can be used to characterize the input data; the transform can even convert the function to a spectral domain of varying resolution, so that peaks can be accurately identified without too much computational expense. The algorithm was tested at fifteen different random frequencies, all with three different random phases, all with random noises and errors, and consistently the transform was able to characterize the

Peer-reviewed version available at Computation 2018, 6, 61; doi:10.3390/computation6040061

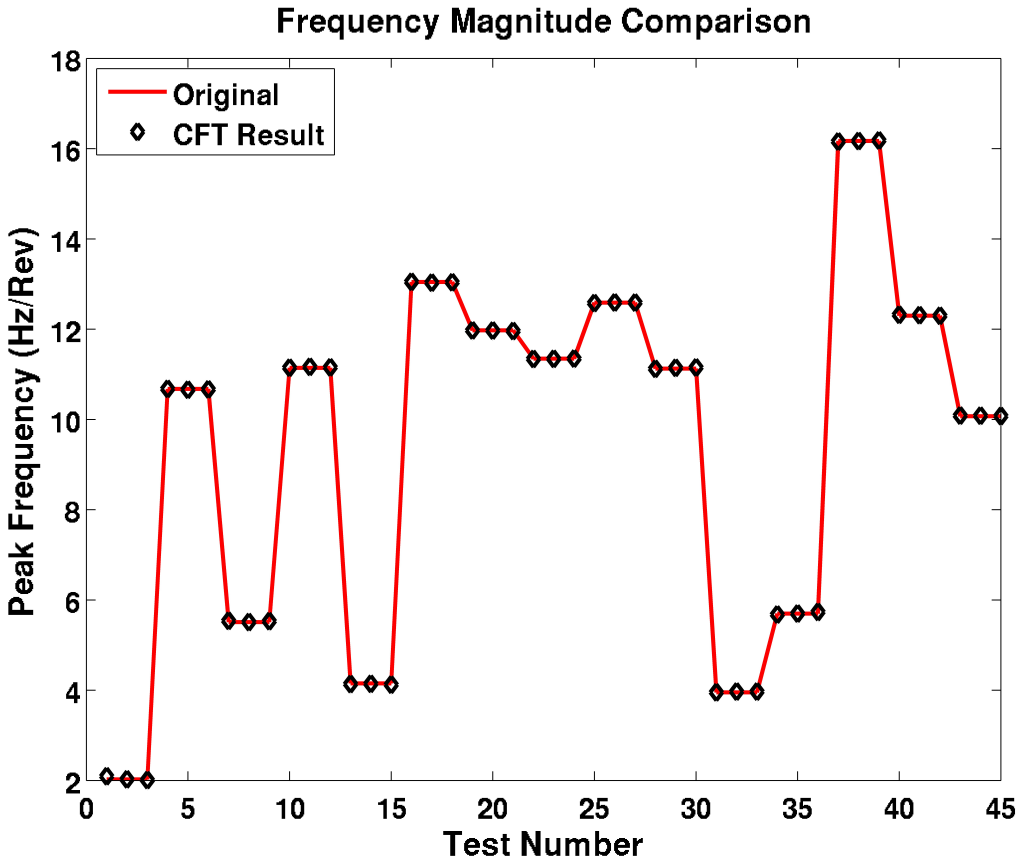

Figure 5. Frequency Prediction Results, $R^2 = 0.999991$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 August 2018 doi:10.20944/preprints201808.0009.v1

Peer-reviewed version available at Computation 2018, 6, 61; doi:10.3390/computation6040061

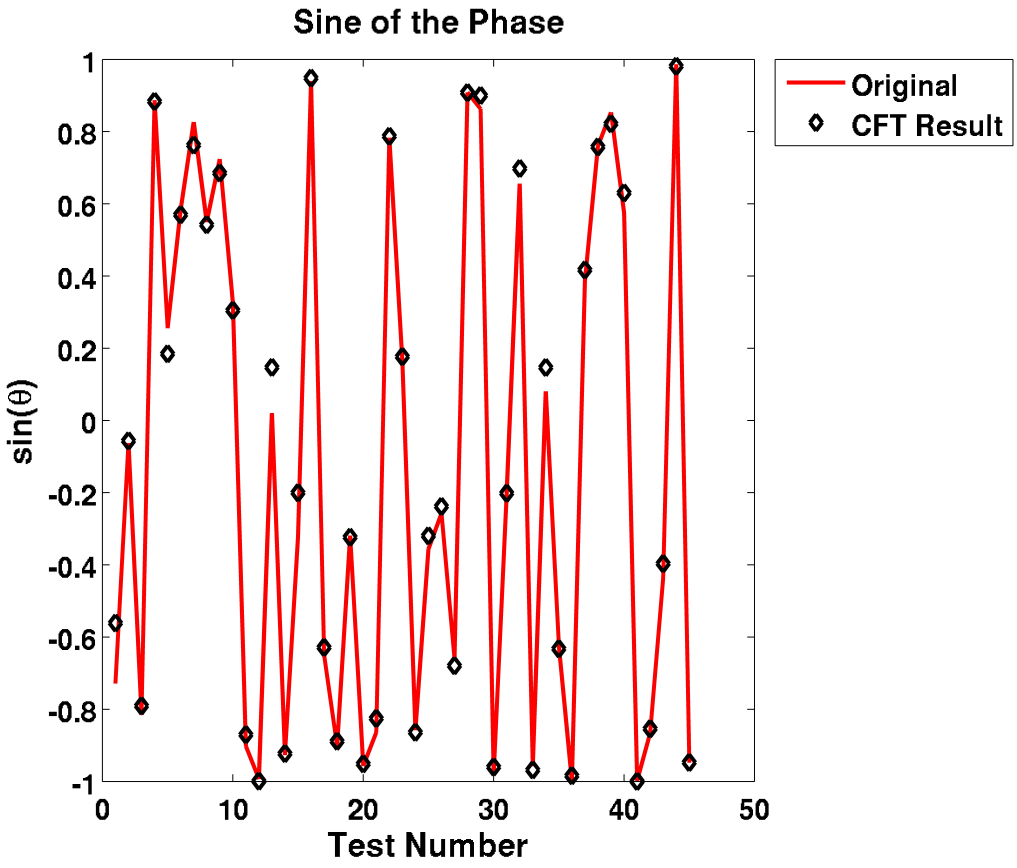

Figure 6. Angle Prediction Results, $R^2 = 0.9982$

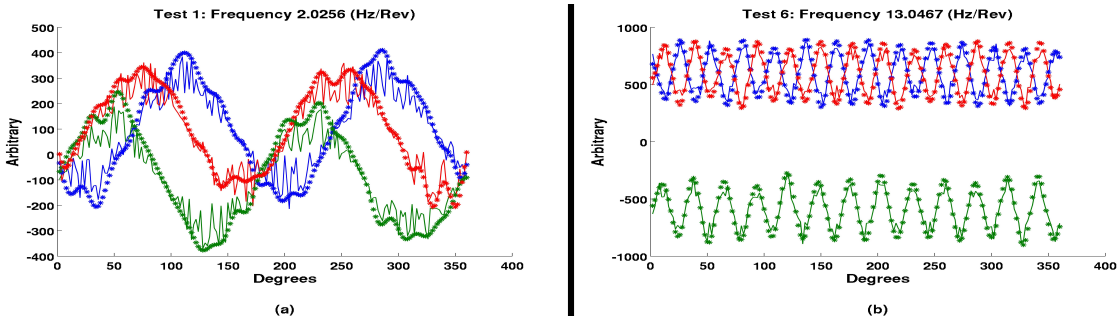

Figure 7. Time results of the randomly generated functions, for frequencies of (a) 2.0256 (Hz/Rev) and (b) 13.0467 (Hz/Rev), but for different phases, magnitudes, and random noises.

Table 3. Comparison of Results, for Phase Shift Angle 1.

Test	Max Freq Original	Max Freq CFT Result	sin(Phase) Original	sin(Phase) CFT Result	R^2 (Temporal)
1	2.0256	2.095	-0.72837	-0.56129	0.92791
2	10.6771	10.678	0.88707	0.88349	0.94843
3	5.5118	5.537	0.82648	0.76277	0.94092
4	11.1441	11.146	0.32585	0.30499	0.93117
5	4.1527	4.137	0.020562	0.14696	0.93132
6	13.0467	13.05	0.94015	0.94829	0.93359
7	11.9742	11.973	-0.31888	-0.32338	0.92769
8	11.3472	11.339	0.78318	0.78725	0.93398
9	12.5892	12.578	-0.35645	-0.31945	0.93956
10	11.128	11.122	0.90871	0.90812	0.92836
11	3.9531	3.954	-0.20131	-0.20166	0.93636
12	5.6981	5.677	0.080619	0.14678	0.93363
13	16.1721	16.158	0.38625	0.41684	0.94243
14	12.2989	12.32	0.57446	0.63023	0.93483
15	10.0715	10.085	-0.43758	-0.3967	0.92398

Table 4. Comparison of Results, for Phase Shift Angle 2.

Test	Max Freq Original	Max Freq CFT Result	sin(Phase) Original	sin(Phase) CFT Result	R^2 (Temporal)
1	2.0256	2.026	-0.062681	-0.056313	0.93311
2	10.6771	10.658	0.25575	0.18466	0.93944
3	5.5118	5.511	0.54959	0.54223	0.9434
4	11.1441	11.166	-0.90216	-0.87023	0.93194
5	4.1527	4.141	-0.92686	-0.92268	0.92885
6	13.0467	13.031	-0.64986	-0.6287	0.92476
7	11.9742	11.97	-0.95715	-0.95146	0.93294
8	11.3472	11.344	0.16368	0.17658	0.92725
9	12.5892	12.593	-0.25732	-0.23808	0.94473
10	11.128	11.145	0.86293	0.90011	0.93841
11	3.9531	3.97	0.65591	0.69795	0.94498
12	5.6981	5.7	-0.62372	-0.63233	0.93151
13	16.1721	16.169	0.76013	0.75789	0.93039
14	12.2989	12.31	-0.99804	-1	0.93751
15	10.0715	10.084	0.9865	0.98098	0.9411

peak frequency and phase angle remarkably, with a higher degree of accuracy than one can expect with traditional DFT methods.

Supplementary Materials: The following are available online, Run_Initial_Study.zip

Acknowledgments: The author would like to acknowledge Lou Vocaturo and Kevin Larkins for useful discussions.

Author Contributions: M.M. is the sole author of this manuscript.

Conflicts of Interest: The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

Table 5. Comparison of Results, for Phase Shift Angle 3.

Test	Max Freq Original	Max Freq CFT Result	sin(Phase) Original	sin(Phase) CFT Result	R^2 (Temporal)
1	2.0256	2.009	-0.81562	-0.79053	0.93367
2	10.6771	10.666	0.58339	0.5697	0.92987
3	5.5118	5.53	0.72393	0.68535	0.93461
4	11.1441	11.154	-0.99286	-0.99909	0.92839
5	4.1527	4.122	-0.3222	-0.20013	0.95071
6	13.0467	13.04	-0.90228	-0.88856	0.94272
7	11.9742	11.958	-0.86461	-0.82531	0.93705
8	11.3472	11.347	-0.85766	-0.86378	0.93519
9	12.5892	12.588	-0.66127	-0.67959	0.92307
10	11.128	11.152	-0.9832	-0.96008	0.9276
11	3.9531	3.969	-0.9574	-0.96818	0.9428
12	5.6981	5.734	-0.99907	-0.9842	0.94156
13	16.1721	16.183	0.85337	0.82142	0.93247
14	12.2989	12.296	-0.86634	-0.85394	0.92086
15	10.0715	10.075	-0.94753	-0.9473	0.94203

MDPI Multidisciplinary Digital Publishing Institute
DOAJ Directory of open access journals
DFT Discrete Fourier Transform
FFT Fast Fourier Transform
NDFT Non-uniform Discrete Fourier Transform
CDFT Coefficient of determination Discrete Fourier Transform
CoD Coefficient of Determination

Bibliography

1. Nagel, R.K.; Saff, E.B.; Snider, A.D. *Fundamentals of Differential Equations, 5th Edition*; Addison Wesley: 75 Arlington Street, Suite 300 Boston, MA 02116, 1999.
2. Haberman, R. *Applied Partial Differential Equations With Fourier Series and Boundary Value Problems, 4th Edition*; Prentice Hall: Upper Saddle River, New Jersey, 2003.
3. Harris, F.J. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proceedings of the IEEE* **January 1978**, *66*, 51–83.
4. Dorrer, C.; Belabas, N.; Likforman, J.P.; Joffre, M. Spectral resolution and sampling issues in Fourier-transform spectral interferometry. *Journal of Optics Society of America B* **2000**, *17*, 1795–1802.
5. Arfken, G.B.; Weber, H.J. *Mathematical Methods for Physicists, Sixth Edition*; Elsevier: 30 Corporate Drive, Suite 400, Burlington MA 01803, 2005.
6. Zill, D.G.; Cullen, M.R. *Advanced Engineering Mathematics, Second Edition*; Jones and Bartlett Publishers: Sudbury MA, 2000.
7. *Numerical Recipies in C: The Art of Scientific Computing*; Cambridge University Press, 1988; chapter 13, pp. 584–591. ISBN 0-521-4310805.
8. Chen, W.H.; Smith, C.H.; Fralick, S.C. A Fast Computational Algorithm for the Discrete Cosine Transform. *IEEE Transactions on Communications* **September 1977**, *25*, 1004–1009.
9. Agarwal, R.C. A New Least-Squares Refinement Technique Based on the Fast Fourier Transform Algorithm. *Acta Cryst.* **1978**, *34*, 791–809.
10. Finzel, B. Incorporation of fast Fourier transforms to speed restrained least-squares refinement of protein structures. *Journal of Applied Cryst.* **1986**, *20*, 53–55.
11. Garcia, A. *Numerical Methods for Physics, Second Edition*; Addison-Wesley: 75 Arlington Street, Suite 300 Boston, MA, 1999.
12. Poon, T.C.; Kim, T. *Engineering Optics With Matlab*; World Scientific Publishing Co: 27 Warren St, Hackensack, NJ 07601, 2006.

13. Nyquist, H. Certain Topics in Telegraph Transmission Theory. *Proceedings of the IEEE* **2002**, *90*, 280–305.
14. Landau, H.J. Necessary Density Conditions for Sampling and Interpolation of Certain Entire Functions. *Acta Mathematica* **1967**, *117*, 37–52.
15. Shannon, C.E. Communication in the Presence of Noise. *Proceedings of the IEEE* **447-457**, *86*, 1998.
16. Luke, H.D. The Origins of the Sampling Theorem. *IEEE Communications Magazine* **1999**, *April*, 106–108.
17. Kupfmuller, K. On the Dynamics of Automatic Gain Controllers. *Elektrische Nachrichtentechnik* **2005**, *5*, 459–467.
18. Harvey, J. Fourier treatment of near-field scalar diffraction theory. *American Journal of Physics* **1979**, *47*, 974–980.
19. Jiang, D.; Stamnes, J.J. Numerical and experimental results for focusing of two-dimensional electromagnetic waves into uniaxial crystals. *Optics Communications* **2000**, *174*, 321–334.
20. Stamnes, J.J.; Jiang, D. Focusing of electromagnetic waves into a uniaxial crystal. *Optics Communications* **1998**, *150*, 251–262.
21. Jiang, D.; Stamnes, J.J. Numerical and asymptotic results for focusing of two-dimensional waves in uniaxial crystals. *Optics Communications* **1999**, *163*, 55–71.
22. Shen, F.; Wang, A. Fast-Fourier-transform based numerical integration method for the Rayleigh–Sommerfeld diffraction formula. *Applied Optics* **2006**, *45*, 1102–1110.
23. Boyd, J.P. A Fast Algorithm for Chebyshev, Fourier, and Sine Interpolation onto an Irregular Grid. *Journal of Computational Physics* **1992**, *103*, 243–257.
24. Lee, J.Y.; Greengard, L. The type 3 nonuniform FFT and its applications. *Journal of Computational Physics* **2005**, *206*, 1–5.
25. Dutt, A. Fast Fourier Transforms for Nonequispaced Data. PhD thesis, Yale University, 1993.
26. Dutt, A.; Rokhlin, V. Fast Fourier Transforms for Nonequispaced Data II. *SIAM Journal of Scientific Ccomputing* **1993**, *14*, 1368–1393.
27. Dutt, A.; Rokhlin, V. Fast Fourier Transforms for Nonequispaced Data II. *Applied and Computational Harmonic Analysis* **1995**, *2*, 85–100.
28. Greengard, L.; Lee, J.Y. Accelerating the Nonuniform Fast Fourier Transform. *SIAM Review* **2004**, *46*, 443–454.
29. Dohler, M.; Kunis, S.; Potts, D. Nonequispaced Hyperbolic Cross Fast Fourier Transform. *SIAM Journal on Numerical Analysis* **2010**, *47*, 4415–4428.
30. Fessler, J.A.; Sutton, B.P. Nonuniform Fast Fourier Transforms Using Min-Max Interpolation. *IEEE Transactions on Signal Processing* **2003**, *51*, 560–574.
31. Ruiz-Antolin, D.; Townsend, A. A Nonuniform Fast Fourier Transform Based on Low Rank Approximation. *SIAM Journal on Scientific Computing* **2018**, *40*, 529–547.
32. Cameron, A.C.; Windmeijer, F.A. An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics* **1997**, *77*, 329–342.
33. Magee, L. R2 Measures Based on Wald and Likelihood Ratio Joint Significance Tests. *The American Statistician* **August 1990**, *44*, 250–253.
34. Nagelkerke, N.J.D. A note on a general definition of the coefficient of determination. *Biomelrika* **1991**, *78*, 691–692.
35. Strang, G. *Introduction to Linear Algebra, 3rd Edition*; Wellesley-Cambridge Press: 7 Southgate Rd, Wellesley, MA 02482, 2003.