JOURNAL OF
EMERGING INVESTIGATORS

Article

Practical applications of the Fourier analysis to identify pitches and synthesize sounds in music

Anoushka Tandon[1], Evgeny Goncharov[2]
[1] Dhirubhai Ambani International School, Mumbai, Maharashtra, India
[2] Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, Cambridge, United Kingdom

SUMMARY

The fundamental principles of music theory are demonstrably intertwined with mathematical concepts. Mathematical principles have long served as a cornerstone for understanding and manipulating musical elements. Musical notation, the written language of music, also uses concepts such as set theory and ratios to represent notes and their duration. This paper delves into this intricate relationship, specifically focusing on the application of Fourier analysis in music. Currently, while the Fourier analysis is used in audio and signal processing, its direct application in theoretical music analysis and music composition and creation, is less explored. The primary purpose of our research was to explore the potential of the Discrete Fourier Transform (DFT) as a powerful tool for music theory and composition, which is a specific algorithmic implementation of the Fourier Analysis designed for discrete data such as sampled audio. We employed the DFT to effectively analyze and synthesize musical elements with a high degree of accuracy. We hypothesized that pitches of the chords, scales, and intervals can be accurately identified by applying the DFT. Utilizing the DFT command in Wolfram Mathematica, we precisely identified the fundamental pitches in both digitally generated and recorded audio samples. Our results demonstrate the potential for the DFT to serve as a powerful tool for music analysis, enabling precise harmonic transcription and spectral differentiation of audio sources. Moreover, we investigated the potential of DFT for sound synthesis. This opens doors for novel approaches to music composition and sound design. It presents the DFT as a powerful tool for musicologists, composers, and sound designers.

INTRODUCTION

Mathematics serves as a foundational element in the realm of music. Every musical note vibrates at a specific frequency measurable in Hertz (Hz). These frequencies are not random, in fact, the simple, whole number ratios between them create the building blocks of pleasurable music. The construction of musical scales and the fundamental building blocks of melodies, also rely heavily on numerical relationships. While a note is a specific pitch with a name and frequency, a music scale is an ascending or descending series of notes. In Western music, there are 12 possible notes, corresponding to the tones on the chromatic scale (1). The circle of fifths, a visual representation of musical relationships, exemplifies this mathematical connection. Each "step" around the circle corresponds to a specific frequency ratio, often a perfect fifth with a 3:2 ratio (2). Major scales, known for their bright and happy sound, are built upon specific sequences of fifths within the circle. Minor scales, with their melancholic undertones, follow a different mathematical path.

Currently, the applications of mathematics in music are largely confined to the analysis of raw sound signals and audio samples. However, its potential as a tool not only for computation but also for music synthesis and composition remains relatively underexplored. This is important as traditional music theory often relies on human perception to identify music structures. Further, there is limited research surrounding mathematical tools to differentiate between digitally-generated and recorded audios, which can transform audio-forensics by aiding in distinguishing real audios from deepfake music.

Fourier analysis, a mathematical framework for decomposing complex periodic signals, offers a powerful lens for understanding musical sounds (3). By deconstructing a sound wave into its constituent sine waves, each with a specific frequency and amplitude, Fourier analysis allows for the identification of a note's pitch and the presence of its intervals and harmonics, which are integer multiples of the fundamental frequency. This contributes to an instrument's timbre. The identification of the fundamental frequency of audios is a useful tool in music composition, editing and analysis. Pitch, one of the most fundamental concepts in music, is how the human ear hears and perceives frequency (4). A higher pitch corresponds to a sound wave with a higher frequency, meaning it oscillates more rapidly, while a lower pitch corresponds to a sound wave with a lower frequency (1). For instance, a perfect octave, where one note vibrates twice as fast as another, has a 2:1 frequency ratio (5). A music interval is the distance in pitch between two notes. Notes in an interval are mathematically related by their pitches (6). For example, if the frequency of C_4 is 261.6256 Hz, the frequency of the octave-higher C would be double i.e. 523.2511 Hz. In the same way, every C found in music is either 32.703 Hz, 65.406 Hz, 130.813 Hz, 261.6256Hz, 523.2511 Hz, 1046.502 Hz, and so on (5). Further, the ratio of pitches in a perfect 4th is 4:3 and in a perfect 5th is 3:2 (7).

This ability to analyze the spectral content of music unlocks numerous applications: pinpointing exact pitches, dissecting the harmonic makeup of chords, and even creating entirely new timbres through sound synthesis. In essence, Fourier analysis acts as a bridge between the complex world of acoustics and the language of music theory (3). The Fourier Transform (FT) is a specific tool within the broader

https://doi.org/10.59720/24-155

field of Fourier analysis which allows one to pass from the time-domain representation of a function *f(t)* to the frequency-domain representation *F(ω)*. An interesting application of the FT in music is that for 40 years the opening chord to the famous Beatles song, "A Hard Day's Night", remained a mystery. Musicians and scholars had all developed their own theories to identify the chord until Dr. Jason Brown used the FT to solve the Beatles riddle (8). As part of our research, we particularly explored how the DFT can be harnessed to advance the fields of music theory and composition. While the FT is designed for continuous signals, making it impractical for analyzing real-world digital data, the DFT is suited for discrete signals, which is the form in which audio and digital signals are processed. The DFT is an approximation of the continuous FT that operates on the discrete numerical samples of the signal to compute the frequency-domain representation and does not require the functional form of the signal (9).

By applying the DFT, we demonstrated its potential to serve as an objective, mathematical tool for music analysis, enabling automatic transcription by extracting accurate harmonic structures from audio files. Previous research has demonstrated the utility of the DFT in various aspects of music analysis, such as pitch detection, spectral analysis, and time-frequency domain representation. Past studies introduced the idea of using DFT to decompose sound signals into their constituent frequencies, paving the way for further developments in audio processing (10, 11). More recently, the DFT has gained traction as a powerful theoretical and analytical tool in music, providing a mathematically robust way of modeling various musical phenomena. Advancements in computational methods have allowed for more refined applications of DFT, enabling detailed harmonic and timbral analysis (12). Moreover, the section of this paper on music synthesis opens up a range of possibilities, allowing for the creation of unique sound structures that traditional tools may not achieve as well as new AI-driven music generation techniques based on the DFT.

We hypothesized that the DFT can effectively identify the distinct pitches and amplitudes in music, thereby distinguishing between recorded and digitally generated tracks and aiding in the synthesis of new audios. By employing DFT using Wolfram Mathematica, we achieved highly accurate analysis and synthesis of musical elements. We successfully identified the fundamental pitches within recorded and digitally generated audio samples, including chords, scales, and intervals, supporting the hypothesis. Furthermore, we successfully demonstrated the potential of the DFT in sound synthesis by replicating and altering an audio sample of an oboe, suggesting its role in creating entirely new timbres and harmonies. The result highlights the possibility of using DFT as a valuable tool for both understanding and creating music.

RESULTS

We explored the practical applications of the Fourier analysis on music by using Wolfram Mathematica to plot the DFT of audio samples. We showed that the DFT can effectively identify the distinct pitches and amplitudes in music to aid in the accurate analysis and synthesis of audios. The purpose of the following tests was to evaluate the accurate identification of pitches in digitally generated chords using the DFT. The frequencies of all the notes in the C-major scale ranging from C4 to C5 (based on the International Pitch Notation) have been included for reference (**Table 1**). In Test 1, we plotted the DFT of a cosine wave with frequency 300 Hz – *cos (300*2πx)* (**Figure 1a**). The DFT showed a spike at 300 Hz (**Figure 1b).** In Test 2, we obtained the waveform of the function of the digitally generated C-major chord (**Figure 2a**). Its DFT showed spikes at approximately 261 Hz, 330 Hz, and 391 Hz, which correspond to the frequencies of the notes of the C-major chord, namely C, E, and G (**Figure 2b**). In Test 3, we obtained the waveform of the function of the digitally generated C-minor chord (**Figure 2c**). The DFT showed spikes approximately at 261 Hz, 311 Hz, and 391 Hz, which correspond to the frequencies of the notes of the C-minor chord, namely C, E♭, and G (**Figure 2d**).

Note in C Major Scale	Pitch/Frequency of the Note (Hz)
C_4 (middle)	261.63
D_4	293.66
E_4	329.63
F_4	349.23
G_4	392.00
A_4	440.00
B_4	493.88
C_5 (higher)	523.25

Table 1: Frequencies of notes in C-major scale. The notes have been named as per the International Pitch Notation (35).

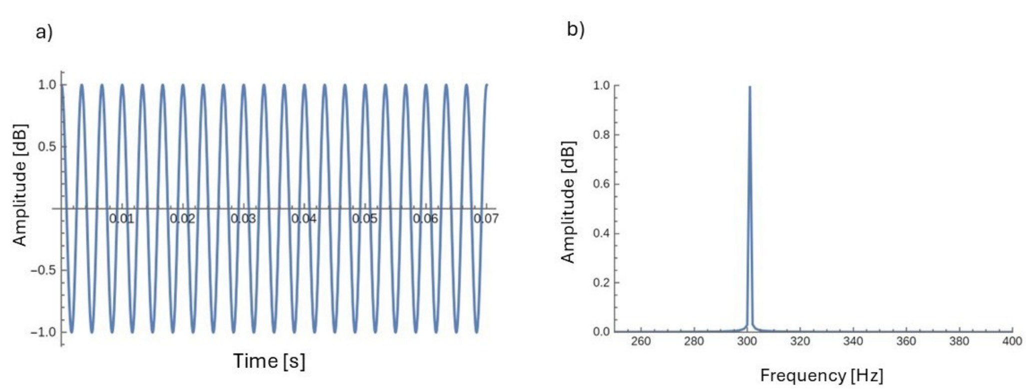

Figure 1: The function cos(300*2π). a) Waveform and **b)** The Discrete Fourier Transform (DFT).

https://doi.org/10.59720/24-155

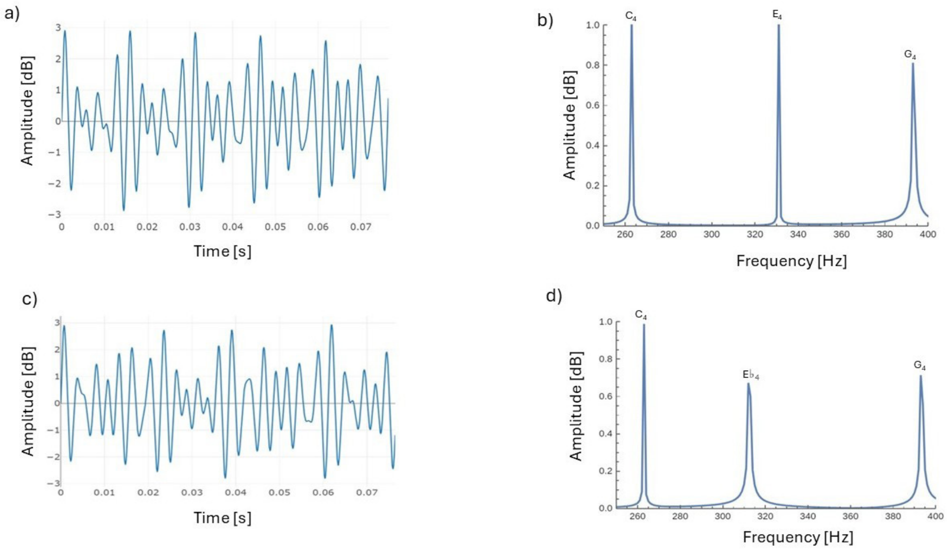

Figure 2: C-major and minor digitally generated chords. a) Waveform and **b)** DFT of C-major chord. **c)** Waveform and **d)** DFT of C-minor chord.

Test 4 was conducted to identify accurate ratios of pitches in intervals using the DFT. We obtained the waveform of the function of the C_4-C_5 octave interval (**Figure 3a**). The DFT showed 4 spikes, 2 of which were at approximately 263 Hz and 523 Hz, which corresponded to the frequencies of C_4 and C_5, respectively (**Figure 3b**). Note that the graph showed four spikes as it was symmetric around 900/2 = 450 Hz, as per the formula in Mathematica.

The next tests were conducted to determine the effect of background noise in pitch identification of recorded audio samples using the DFT. In this context, "background noise" refers to extraneous ambient sounds that are commonly captured during audio recording, such as those produced by environmental factors or unintended physical disturbances (e.g., hand movements, fan noise, air movements). For these tests specifically, we generated periodograms of the signals, instead of DFTs, as they provide a better understanding of the multiple frequency components and changes over time. They are estimates of the power spectral density (PSD) of a signal, which measures how the power of a signal is distributed across different frequencies. The power of a signal at each frequency is proportional to the square of the magnitude of the DFT coefficients. In Test 5, we plotted the periodogram of a recorded C-major chord and set the range from 0 to 2,000 (**Figure 4a, b**). On examining a range of 328 Hz to 331 Hz, we obtained the spike of the second note, E_4 (**Figure 4c**). In comparison, the spike of the digitally generated audio of the C-major chord, when examined for the same range, was much clearer because extraneous frequencies introduced by ambient noise and physical disturbances interfered with the precise identification of the signal's fundamental components (**Figure 4d**). This noise added complexity to the frequency spectrum, reducing the clarity of the distinct harmonic peaks. Further, there were greater variations among the amplitudes of the different frequencies in the recorded C-major chord as compared to the digitally generated C-major chord, which is also an indication of background noise (**Figures 2b, 4b**). In Test 6, we plotted the periodograms of loud and soft recorded C-major chords and examined the frequency of G_4 which is 391.9954 Hz (**Figure 5a**, **b**). In Test 7, we plotted the periodogram of a recorded ascending C-major scale (**Figure 6a**). The graph is examined from 520 Hz to 525 Hz to showcase background noise (**Figure 6b**). Note that the spikes with the greatest amplitude represented the eight notes in the scale. Further on examining the graph around the C_5 note, we observed the effect of background noise that caused several

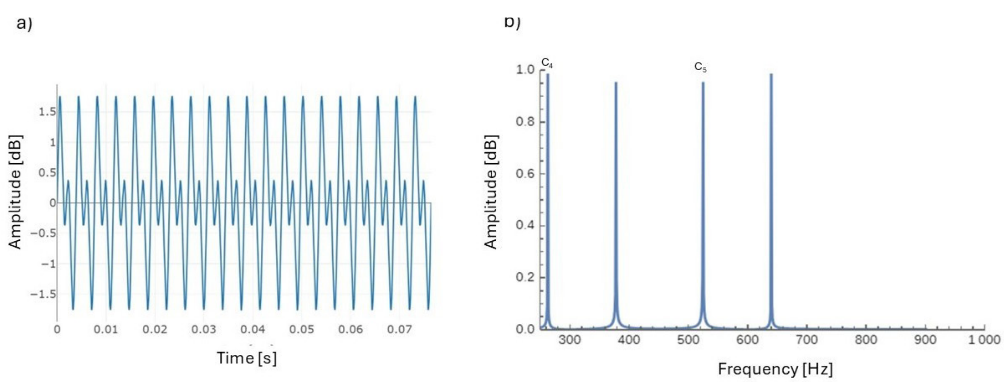

Figure 3: C_4-C_5 octave interval. a) Waveform and **b)** DFT.

JOURNAL OF
EMERGING INVESTIGATORS

https://doi.org/10.59720/24-155

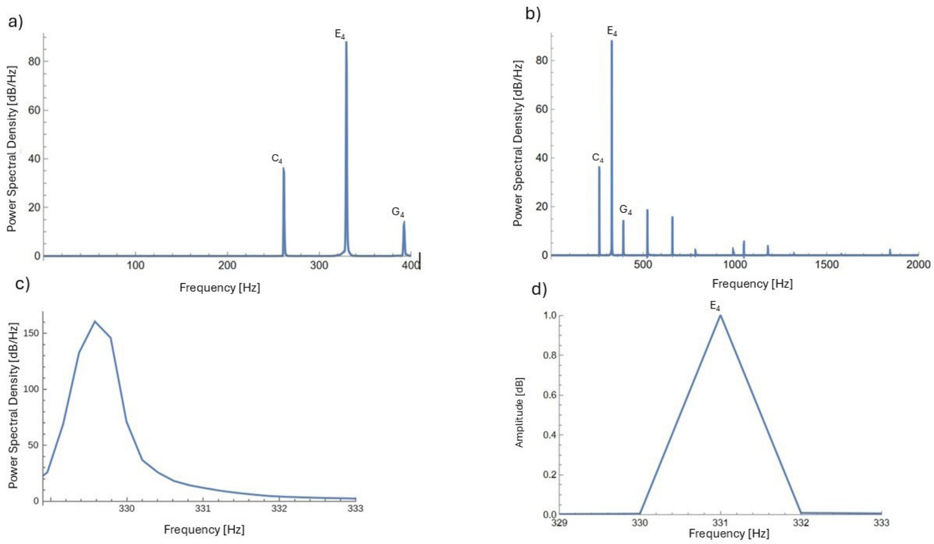

Figure 4: Comparison of DFT of digitally generated and recorded C-major chords. a) DFT of recorded chord. **b)** DFT of recorded chord (range: 0-2000 with background noise)**. c)** E_4 note of recorded chord with background noise. **d)** E_4 note of digitally generated chord without background noise.

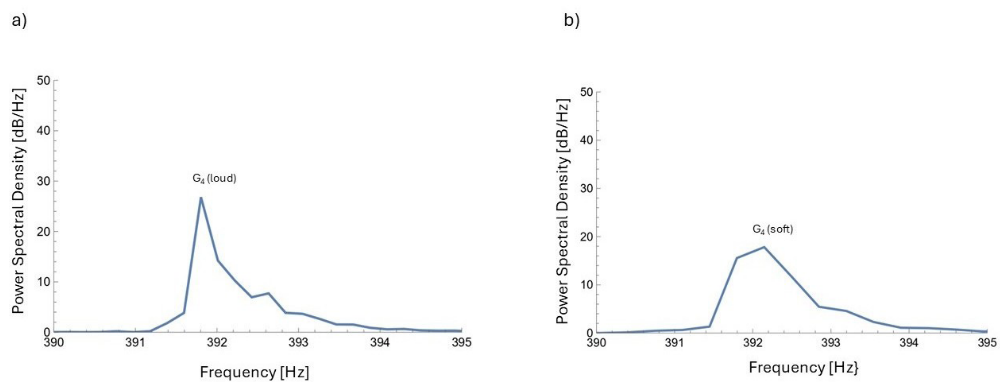

Figure 5: Representation of amplitude in DFT. G_4 note of a **a)** loud and **b)** soft recording of a C-major chord.

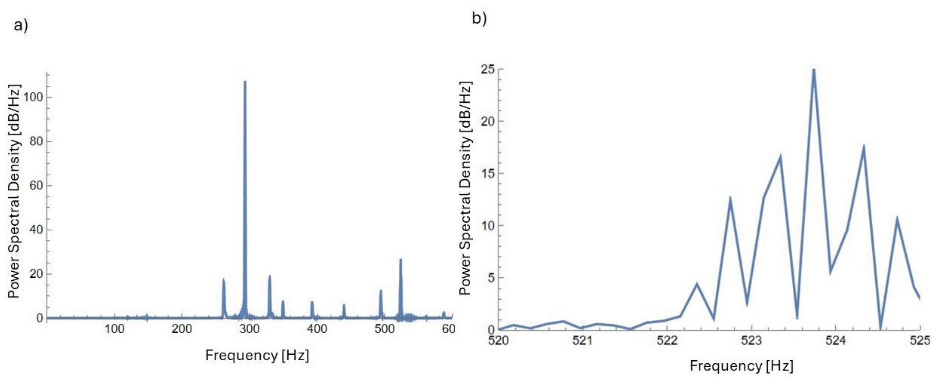

Figure 6: C-major scale (ascending). a) DFT with background noise. **b)** C_5 note (not precise).

smaller spikes, rather than simply one sharp spike.

The following tests were conducted to demonstrate the potential of the DFT in synthesizing audio by replicating and altering an audio sample. In Test 8, we replicated the Mathematica Oboe sound (13). We generated a synthesized plot of the frequency and PSD values obtained from its periodogram by inputting the numerical values to create the corresponding graph (**Figure 7a**). Thereafter, we generated a mathematical expression for its waveform and plotted it (**Figure 7b**). On playing both the synthesized and original audios, they sounded the same, which was objectively represented by the identical waveforms. The original list of

https://doi.org/10.59720/24-155

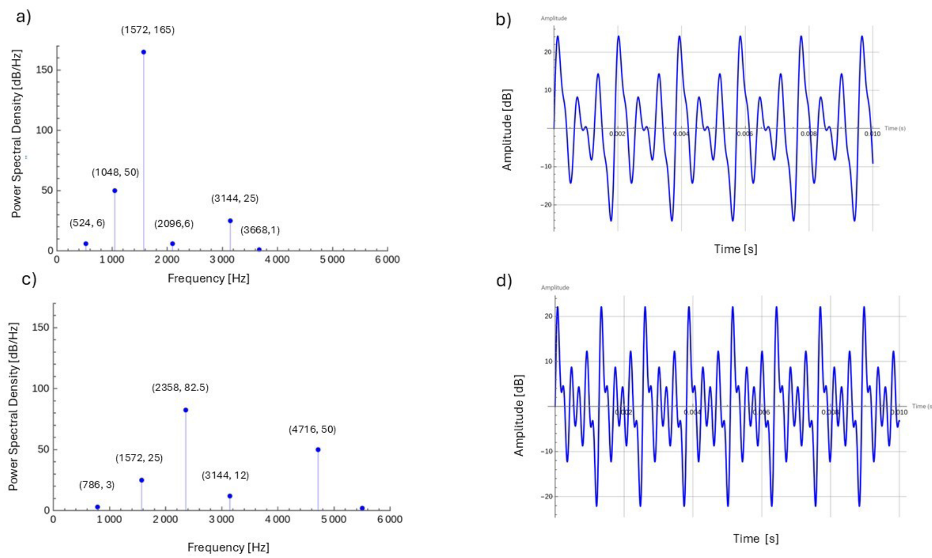

Figure 7: Replication of the oboe audio sample. a) Synthesized plot and **b)** waveform of replicated audio values (same as that of original audio). **c)** Synthesized plot and **d)** waveform of altered audio values.

frequencies and their corresponding amplitudes have been included for reference (**Table 2**).

Lastly in Test 9, we altered the audio obtained in Test 8. We multiplied each frequency by 1.5 to shift the pitch while preserving harmonic relationships. We scaled down the lower frequencies (524 Hz to 1572 Hz) by 0.5 to reduce their dominance and balance the spectrum. We amplified the higher frequencies (2096 Hz to 3668 Hz) by 2.0 to enhance clarity and brightness, compensating for their natural attenuation in perception. This approach allowed us to analyze how frequency and amplitude scaling affected the timbre and perception of the synthesized waveform. We generated the synthesized plot of the altered frequency and amplitude values (**Figure 7c**). We plotted the altered waveform (**Figure 7d**). The difference was clearly visible in the waveform itself, which was not the same as that of the original audio. When played together, the new sound no longer sounded the same as the original, with a noticeably higher pitch and weaker low-end presence. The altered list of values that we used have been included for reference (**Table 2**).

Original Frequencies (Hz)	Original PSD (dB/Hz)	Altered Frequencies (Hz)	Altered PSD (dB/Hz)
524	6.0	786	3.0
1048	50.0	1572	25.0
1572	165.0	2358	82.5
2096	6.0	3144	12.0
3144	25.0	4716	50.0
3668	1.2	5502	2.4

Table 2: Original and altered frequencies in Test 9.

DISCUSSION

Our study provides a detailed analysis into the applications of the DFT in music analysis and synthesis. Our results demonstrate the power of the DFT as a tool for music theory, music analysis, and composition by showcasing its ability to accurately identify pitches in both simple and complex musical structures. By analyzing digitally generated tones and chords (Tests 1-4), we employed the DFT to successfully identify the fundamental frequencies of notes in C-major and C-minor chords, as well as the octave relationship between notes. Furthermore, we utilized the DFT to effectively capture the specific pitch difference in E vs E♭. The experiments, specifically Tests 5, 6, and 7, also highlight the impact of background noise on real-world audio analysis (**Figure 4a - d**). While the DFT can still identify the dominant pitches in a recorded chord (C-major), background noise introduces additional spectral components, making the results less precise compared to clean, digital audios. We also explored how the DFT can capture not just pitch information but also changes in volume over time. One could potentially use this information to differentiate between articulation techniques used by musicians. Finally, the successful replication and alteration of an oboe sound using the DFT from an audio sample in Tests 8 and 9 underscores the potential of this technique for sound synthesis. By analyzing the harmonic structure of a sound, the DFT paves the way for not only replicating existing sounds but also creating entirely new timbres through manipulation of the identified frequencies. In essence, these tests reinforce the DFT's valuable role in the sound engineering industry.

Tests 1 to 4 strongly depict the accuracy of the DFT in identifying particular pitches and intervals. This demonstrates its ability in being used for automatic transcription, autotune, and other music tools that require extraction and analysis of audio components. Test 5-7 demonstrate the effects of background noise on the DFT of recorded audios. It was interesting to note that even a small amount of noise can distort a frequency spectrum, making it difficult to distinguish between true signal frequencies and noise artifacts. Further, this has a very powerful real-world application in differentiating original audios from deepfakes. When the DFT is applied to both original and deepfake recordings, the frequency spectrum of a deepfake might exhibit unusual patterns, such as unnatural harmonic structures and odd frequency spikes that wouldn't be present in the original audio. Those generated by neural networks may inadvertently introduce unnatural residual noise artifacts, detectable by a spectral

https://doi.org/10.59720/24-155

analysis using the DFT. In Test 6 specifically, we examined the difference in the amplitudes of the soft and loud audios for the G4 note in two recorded audios of the C-major chord. This highlights the ability of the DFT to accurately represent not only frequencies but also amplitude. Note that the sample rate of our recorded audios is 48,000 Hz, which is suitable as it captures human hearing range. As the Nyquist-Shannon sampling theorem states that the sampling rate needs to be at least twice the highest frequency to be captured, 48 kHz (well above 20 kHz) ensures all audible frequencies are accurately represented in the digital audio (14).

Lastly, in Tests 8-9, we utilized the DFT to synthesize an entirely new sound by altering the frequency and amplitude values of a sample oboe track. These tests demonstrate the ability of the DFT to help musicians create new soundscapes and effects that would not be possible with traditional acoustic instruments alone.

There are technicalities that are important to note as they may have affected the accuracy of the experiments. The accuracy of the DFT calculation depends on the specific software being used (15). Different algorithms and implementations can introduce slight variations in the results. The software might limit the frequency resolution of the DFT plot. This would make it difficult to distinguish between closely spaced frequencies, potentially leading to missed information about the audio content, especially in the recorded audios. Notably, a signal with a length that is not an integer multiple of its fundamental period will suffer from spectral leakage. Therefore, to effectively prevent spectral leakage, an audio should be long enough to contain an integer number of periods of the lowest frequency present in the signal. Shorter samples might not provide enough detail to distinguish between closely spaced frequencies. When a finite audio sample is used, the frequencies in the DFT plot can “leak” into neighboring frequencies. This is called spectral leakage and can make it difficult to precisely identify the true frequencies present in the audio (16). Especially in real world signals which are non-periodic over short-time intervals and have background noise, the leakage effect is worsened.

The DFT plot typically shows frequency on the horizontal axis and amplitude on the vertical axis. While it appears as spikes, these spikes aren’t infinitely thin lines. They have a certain width depending on the frequency resolution of the DFT and the windowing function used, if any (17). The DFT plot will show the fundamental frequency of a note and its harmonics. Without additional analysis, it might be difficult to distinguish the fundamental frequency from its harmonics, especially for complex sounds. Therefore, we have zoomed in the graphs to the correct ranges. In reality, the ideal DFT output for a single pure tone (like the 300 Hz cosine wave in Test 1) is not a single point but a sinc function (18). This function has a central lobe at the actual frequency, 300 Hz in this case, and it also has side lobes that extend to positive and negative frequencies. Due to the limited resolution of the plot and the finite width of the displayed spikes, these side lobes might not be readily visible, especially for closely spaced frequencies. However, their presence can still affect the amplitudes of neighboring frequency bins, leading to inaccuracies in the perceived sharpness of the spikes. For example, in Test 2 and 3 with chords, the presence of spectral leakage might make it seem like the spikes are slightly wider than they truly are. As a future step, we will use windowing functions, like the Hanning or Blackman windows to reduce spectral leakage by tapering the edges of the frequency components in the time domain, which translates to narrower main lobes and lower side lobes in the frequency domain (19). Our research highlights the challenge of background noise in real world audio analysis. Therefore, one of the potential future experiments is advanced noise reduction techniques. One could delve deeper into this area by investigating and implementing advanced noise reduction techniques specifically tailored for music signals. Techniques like spectral subtraction or Wiener filtering could be explored to isolate the desired musical content from background noise, potentially improving the accuracy of pitch identification and DFT-based analysis in noisy environments (20, 21).

Another potential direction for future research using the DFT is musical onset detection and segmentation. The DFT provides a snapshot of the frequency content at a specific time. However, music is a dynamic art form with notes that begin and end over time. One could investigate using the DFT in conjunction with other techniques like Short-Time Fourier Transform (STFT) for musical onset detection. STFT analyzes short snippets of audio over time, allowing one to pinpoint the exact moments when notes start and stop within a musical piece (22). This information is valuable for music transcription, rhythm analysis, and further music information retrieval applications.

Further, one can explore automatic genre classification (23). Different musical genres often have distinct characteristics in their frequency content. By analyzing the DFT output of various music genres, one could explore the development of an automated music genre identification system. This system could learn the characteristic frequency fingerprints of different genres and then use the DFT analysis of an unknown song to classify it into the appropriate genre.
Lastly, music emotion recognition is another line of research and future application that stems from this paper (24). Music can evoke a wide range of emotions in listeners. Research suggests that certain musical features like tempo, rhythm, and spectral characteristics can be linked to specific emotions. One can investigate the potential of using the DFT- derived spectral information, along with other musical features, to develop a system that can automatically recognize the emotions conveyed in a piece of music.

In this study, we conducted a two-fold analysis of the applications of DFT in music. Firstly, we successfully proved the ability of the DFT to identify scales, pitches, intervals, and other components of an audio, essential for music analysis. Building on these results, we were able to differentiate between digitally generated and recorded audios of the C-major chord owing to the noise artifacts and their resulting disturbances in the DFT of the recorded audio. However, we not only proved the potential of the DFT as a computational tool, but also as a powerful mathematical approach to synthesize and compose music. Our results are of great importance as they highlight the revolutionary impact the DFT can have in audio-processing, serving as a significant aid to musicians, composers and sound engineers in their pursuits.

MATERIALS AND METHODS

The FT is a continuous version of the Fourier series, which allows one to pass from the time-domain representation of a function $f(t)$ to the frequency-domain representation $F(\omega)$.

The general form of the FT for bounded functions where the following integral exists is:

$$F(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} f(t)dt$$

However, this formula does not apply to periodic functions. Therefore, for a function with period 2, we define the FT as:

$$F(\omega) = 1/2\pi \int_{-\pi}^{\pi} e^{-i\omega t} f(t)dt \quad (25).$$

Note that the real part of the above formula is:

$$1/2\pi \int_{-\pi}^{\pi} cos(\omega t) f(t)dt$$

and the imaginary part of the formula is:

$$-1/2\pi \int_{-\pi}^{\pi} sin(\omega t) f(t)dt$$

The real and the imaginary part of the FT can now be used to recover the coefficients of the Fourier Series which are as follows:

$$a_0 = 1/\pi \int_{-\pi}^{\pi} f(\theta)d\theta,$$

$$a_n = 1/\pi \int_{-\pi}^{\pi} f(\theta)cos(\theta)d\theta,$$

$$b_n = 1/\pi \int_{0}^{2\pi} f(\theta)sin(\theta)d\theta \quad (26).$$

Setting ω = 0 for the real part of the formula, we get a_0 = F(0) and setting ω = n, for the real part we get a_n = 2F(n). Setting ω = n, for the imaginary part we get bn = -2F(n). For more information on the FT, please refer to the references below (27 - 30).

We used the DFT technique in Wolfram Mathematica for the experiment. The DFT is commonly used to perform the FT in the absence of a formula for the function. It uses the computed Fourier coefficients to construct the frequency-domain representation of the input data, using the following formula:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i\frac{2\pi kn}{N}}$$

where X_k is the k-th Fourier coefficient, xn is the n-th sample of the input sequence in the time-domain, N is the total number of samples, and k is the index of the frequency component (31).

The magnitude $|X_k|$ indicates the amplitude of the corresponding frequency component, and the phase of X_k gives the phase of the frequency component. The phase conveys the starting position of the waveform of a frequency relative to time. Mathematically, it is represented by the angle of X_k and is derived from the real and imaginary parts of the coefficient, using the formula:

$$\phi_k = atan2(Im(X_k), Re(X_k)),$$

where *Re(X_k)* is the real part of the Fourier Coefficient, *Im(X_k)* is the imaginary part, and *atan2* is the two-argument arctan function that gives the phase angle in radians (32).
The following function was used to convert chords to their functions:

$$Sound = sin(2\pi * frequency_1 * time) + sin(2\pi * frequency_2 * time) + \ldots \quad (33)$$

For Tests 1 to 7, we used the function *Plot[]* of Wolfram Mathematica to plot the waveform of the chord/scale/interval. We used the function *Fourier[]* to obtain the DFT of the audio. In the visualization stage of the analysis, we employed the *ListLogPlot[]* function to create logarithmic-scale plots of the magnitude spectrum obtained from the DFT.

For Tests 5, 6 and 7, we record the chord/scale on the piano and import the corresponding .wav file to Wolfram Mathematica using the *Import[]* function. For Test 7 specifically, we used the *Periodogram[]* function to obtain an estimate of the PSD of the signal, which is directly related to the DFT. The periodogram is computed from the DFT by squaring the magnitude of the DFT coefficients to estimate the power at each frequency, using the formula:

$$P_k = \frac{1}{N}|X_k|^2 \quad (34)$$

For Test 8, we first plotted the periodogram of the original audio in the form of a linear scale in the vertical axis by using *Scaling Function – "Absolute".* Then, we extracted the frequency and amplitude values from it and store them in the form of a list. We synthesized this list into a plot using the *ListPlot[]* function and then obtain its waveform using the following formula, where *synthet* stands for the list of frequency peaks and their corresponding amplitudes:

$$\text{Sum}\left[\sqrt{synthet[[j,2]]} * Sin[2 * Pi * synthet[[j,1]] * t], \{j, 1, Length[synthet]\}\right]$$

For each frequency *synthet[[j,1]]* (the first element in each sublist), its corresponding amplitude *synthet[[j,2]]* (the second element) is multiplied by a sine wave function *sin(2π*frequency*t)*. This process effectively reconstructs the waveform of the audio by adding together all the individual frequency components. Thereafter, the audio was played using the *Play[]* function.

For Test 9, we altered the frequency and amplitude values in the list, and then followed the steps mentioned for Test 8. We multiplied each frequency component by 1.5. We scaled down the amplitudes of the lower frequencies by 0.5 (524 Hz to 1572 Hz) and scaled up the amplitudes of the higher frequencies by 2.0 (2096 Hz to 3668 Hz).

Received: May 8, 2024
Accepted: November 11, 2024
Published: May 29, 2025

REFERENCES

1. "Physics Tutorial: Pitch and Frequency." *The Physics Classroom*, www.physicsclassroom.com/class/sound/Lesson-2/Pitch-and-Frequency. Accessed 04 Apr. 2024.
2. DeVoto, Mark. "Key Signature | Musical Notation." *Encyclopædia Britannica*, 2019, www.britannica.com/art/key-signature.
3. Morin, David. "Fourier Analysis." 28 Nov. 2009. Faculty

https://doi.org/10.59720/24-155

Webpage, Harvard University. https://scholar.harvard.edu/files/david-morin/files/waves_fourier.pdf. Accessed 04 Apr. 2024.
4. Chase, Samuel. "What Is Pitch in Music? A Complete Guide." *Hello Music Theory*, 26 Mar. 2024, https://hellomusictheory.com/learn/pitch/. Accessed 04 Apr. 2024.
5. Abbott, David. "Frequency Ratios and Pitch Perception." *Understanding Sound*, https://pressbooks.pub/sound/chapter/pitch-and-frequency-in-music/. Accessed 04 Apr. 2024.
6. Kendall, Hannah. "What Are Intervals in Music Theory?" *Hoffman Academy*, www.hoffmanacademy.com/blog/what-are-intervals-in-music-theory-piano-intervals-explained/. Accessed 04 Apr. 2024.
7. Everbach, Carr. "Musical Scales and Frequencies". 2006, Engineering 5, Engineering Methodology, Microsoft PowerPoint presentation. www.swarthmore.edu/NatSci/ceverba1/Class/e5_2006/MusicalScales.html. Accessed 04 Apr. 2024.
8. Rousseau, Christiane. "Mathematics, Physics and a Hard Day's Night." *Canadian Mathematical Society Notes de La SMC*, vol. 36, no. 3, Oct. 2004, https://notes.math.ca/archives/Notesv36n6.pdf. Accessed 30 Jan. 2025.
9. Cimbala, John M. "Fourier Transforms, DFTs and FFTs." Penn State University, 22 Feb. 2010
10. O. Smith iii, Julius. *Spectral Audio Signal Processing*. https://ccrma.stanford.edu/~jos/sasp.
11. Roads, Curtis. *The Computer Music Tutorial*. MIT Press, 6 June 2023,
12. Harding, Jennifer Diane. *Applications of the Discrete Fourier Transform to Music Analysis. Fsu.edu*, 2021, purl.lib.fsu.edu/diginole/2020_Summer_Fall_Harding_fsu_0071E_16452. Accessed 28 Sept. 2024.
13. Luis Gómez-Muñoz, José, and Out. *Fourier Synthesis.* 29 Sept. 2017. https://matecmaticaacustica.weebly.com/uploads/1/9/1/8/19187949/synthesis.pdf.
14. Shannon, C.E. "Communication in the Presence of Noise." *Proceedings of the IRE*, vol. 37, no. 1, Jan. 1949, pp. 10–21, https://doi.org10.1109/jrproc.1949.232969.
15. Schatzman, James C. "Accuracy of the Discrete Fourier Transform and the Fast Fourier Transform." *SIAM Journal on Scientific Computing*, vol. 17, no. 5, Sept. 1996, pp. 1150–1166, https://doi.org/10.1137/s1064827593247023.
16. M. Blackledge, Jonathan. "The 2D DFT, FFT and FIR Filter." *Digital Image Processing - Mathematical and Computational Methods*, Woodhead Publishing Series in Electronic and Optical Materials, 2005.
17. National Instruments. *Understanding FFTs and Windowing*. National Instruments, n.d.
18. Daneglo, Antonio. "Lecture 3: Fourier Transform.", 2013, University of Udine. Microsoft PowerPoint Presentation.
19. "Understanding FFTs and Windowing." *NI*, 11 Oct. 2024, www.ni.com/en-us/innovations/white-papers/06/understanding-ffts-and-windowing.html. Accessed 26 Apr. 2024.
20. Zhang, Shenghuan, and Ye Cheng. "Masking and Noise Reduction Processing of Music Signals in Reverberant Music." *Journal of Intelligent Systems*, vol. 31, no. 1, 1 Jan. 2022, pp. 420–427, https://doi.org/10.1515/jisys-2022-0024.
21. Upadhyay, Navneet, and Rahul Kumar Jaiswal. "Single Channel Speech Enhancement: Using Wiener Filtering with Recursive Noise Estimation." *Procedia Computer Science*, vol. 84, 2016, pp. 22–30, https://doi.org/10.1016/j.procs.2016.04.061.
22. "STFT - Short-Time Fourier Transform." *Help Centre, MATLAB*, www.mathworks.com/help/signal/ref/stft.html. Accessed 04 Apr. 2024.
23. Tzanetakis, G., and P. Cook. "Musical Genre Classification of Audio Signals." *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, July 2002, pp. 293–302, https://doi.org/10.1109/tsa.2002.800560.
24. Bilal Er, Mehmet, and Ibrahim Berkan Aydilek. "Music Emotion Recognition by Using Chroma Spectrogram and Deep Visual Features." *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, 2019, pp. 1622–1634, https://doi.org/10.2991/ijcis.d.191216.001. Accessed 12 June 2020.
25. Kulkarni, Sanjeev. "Frequency Domains and Fourier Transforms." ELE201 Introduction to Electrical Signals and Systems, 2002.
26. Fussell, Don. "Orthogonal Functions and Fourier Series." CS384G – Computer Graphics, Spring 2010, University of Texas, Austin.
27. Bevelacqua, Peter. "Fourier Transforms." *Fourier Transform*, www.thefouriertransform.com/. Accessed 04 Apr. 2024.
28. "Fourier Series." *A Handbook of Fourier Theorems*, by Champeney D.C., Cambridge University Press, 5 June 2012.
29. Fan, Zhou. "Seminar Notes" The Mathematics of Music, Yale university. Received 1 Sept. 2010. Course Handout.
30. Kessler, Bruce. "A "Sound" Approach to Fourier Transforms: Using Music To Teach Trigonometry." *Bridges Donostia*, Spain, Jan. 2007, The University of The Basque Country.
31. Lecture 7 -the Discrete Fourier Transform. University of Oxford, www.robots.ox.ac.uk/~sjrob/Teaching/SP/l7.pdf
32. Harley, Joel B. "Frequency/Magnitude/Phase Responses." EEE 5502: Foundations of Digital Signal Processing, Fall 2022, University of Florida.
33. Greminger, Michael. "Musical Chord Waveform Visualizer." *The Official EngineeringPaper.Xyz* Blog, Michael Greminger, 28 Sept. 2022, https://blog.engineeringpaper.xyz/musical-chord-waveform-visualizer. Accessed 04 Apr. 2024.
34. F. Chaparro, Luis . "Fourier Analysis of Discrete-Time Signals and Systems." *Signals and Systems Using MATLAB*, by Luis F. Chaparro.
35. "Music Note Frequency Chart - Music Frequency Chart." *MixButton*, 3 Aug. 2021, https://mixbutton.com/mixing-articles/music-note-to-frequency-chart/.