

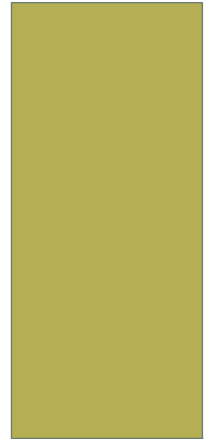
$$\mathcal{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft} dt$$

$$\mathcal{F}^{-1}\{G(f)\} = g(t) = \int_{-\infty}^{\infty} G(f)e^{i2\pi ft} df$$



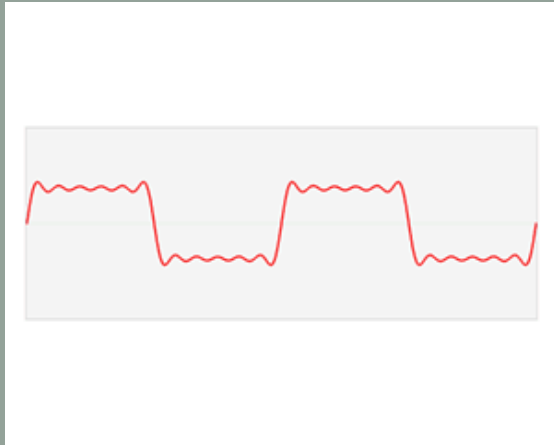
Fast Fourier Transforms

By Jacob Riedel and Deniz Celik



What is a Fourier Transform?

- Converts a signal between the time domain and frequency domain.



$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx$$

- The Fourier Transform is a result of the study of a Fourier Series.

Fourier Series

- The Fourier Series is an approximation of complex signals by summing simpler sine and cosine functions
- The transform is produced when these approximations are allowed to continue until time goes to infinity.

Discrete Fourier Transform

- A Discrete Fourier Transform take a finite number of points from a complex sinusoid and transforms it using summation, rather than integration.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N}, k \in \mathbb{Z}$$

Fast Fourier Transform

- A Fast Fourier Transform is an Algorithm which makes computing a Discrete Fourier Transform computationally easier.
- A FFT will produce the same values as a DFT in less time.
- FFT can actually be more accurate than DFT in the presence of rounding errors.

FFT Algorithms

- There are many different algorithms used in computing FFT.
- Cooley-Tukey Algorithm is a divide-and-conquer method, using recursion and breaking down DFTs.
- Their algorithm is inspired by I. J. Good's Prime Factorization Algorithm.
- Their formula was actually a rediscovery of Carl Friedrich Gauss's algorithm

Code Demo

- Slow:

- Uses Matrix Multiplication

- $\vec{X} = M \cdot \vec{x}$, where $M_{k,n} = e^{-i2\pi kn/N}$

- Fast:

- Uses Cooley-Tukey Algorithm

- Symmetry of Fourier Transform – Odd is Even???

- Numpy:

- Uses Optimized Cooley-Tukey Algorithm

- Tweaked and Written in FORTRAN

Applications

- Solving Differential Equations
- Signal Processing
- Image Processing
- Nuclear magnetic resonance and MRI's
- Tomography – Delay in time is phase shift
- Convolution: $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$
 - ▮ Is multiplication in Frequency

Work Cited

- Brigham, E. Oran. *The fast Fourier transform and its applications*. Englewood Cliffs, N.J.: Prentice Hall, 1988. Print.
- "Fast Fourier Transform." -- *from Wolfram MathWorld*. N.p., n.d. Web. 29 Apr. 2014. <<http://mathworld.wolfram.com/FastFourierTransform.html>>.
- "Fourier Transform." -- *from Wolfram MathWorld*. N.p., n.d. Web. 29 Apr. 2014. <<http://mathworld.wolfram.com/FourierTransform.html>>.
- "Fourier Transforms." *Fourier Transform*. N.p., n.d. Web. 29 Apr. 2014. <<http://autotransformer/>>.
- Loan, Charles F.. *Computational frameworks for the fast fourier transform*. Philadelphia: SIAM, 1992. Print.
- Smith, Steven W.. "Chapter 8: The Discrete Fourier Transform." *The scientist and engineer's guide to digital signal processing*. San Diego, Calif.: California Technical Pub., 1997. Print.

Our code in Python 2.7 can be found at:

<https://github.com/dcelik/olin-linearityfftproject/>

in the src folder. Run the file gui.py in any Python 2.7 interpreter and it will display a window where an expression of sins and cosines can be entered to create graphs of the Fourier Transform.