Recitation 11

DS-GA 1013 Mathematical Tools for Data Science

1. Is convolution associative? That is, is $(\vec{x} * \vec{y}) * \vec{z} = \vec{x} * (\vec{y} * \vec{z})$ for all $\vec{x}, \vec{y}, \vec{z} \in \mathbb{C}^n$?

> **Solution:** By taking DFTs the operation becomes coordinate-wise multiplication, which is associative.

2. 1. Suppose we plot the magnitude of the spectrum of a signal $\vec{x} \in \mathbb{C}^n$. Then we (circularly) time shift it, and plot the magnitude of the spectrum again. How will the two plots differ?
   2. Suppose we have two images I_1 and I_2 and their DFTs $\hat{I}_1$ and $\hat{I}_2$. We create R_1 by superimposing the phase of $\hat{I}_2$ on the magnitude of $\hat{I}_1$ and R_2 by superimposing the phase of $\hat{I}_1$ on the magnitude of $\hat{I}_2$. How will R_1 and R_2 look like?

> **Solution:** The two plots will be the same. Time shifting multiplies the frequencies by a factor with modulus 1. Thus the magnitudes are unchanged.

3. Let $\vec{1} \in \mathbb{C}^n$ denote the vector that is all ones. What is $\vec{x} * \vec{1}$ for $\vec{x} \in \mathbb{C}^n$? Can we deconvolve to get $\vec{x}$?

> **Solution:** $\vec{x} * \vec{1}$ is the vector with every entry equal to $\vec{1}^T\vec{x}$. We cannot deconvolve, since we have only learned the sum of the entries. Alternatively, note that all but one of the Fourier coefficients of $\vec{1}$ are 0.

4. 1. In the homework we blurred the signal by convolving with a Gaussian filter. Explain why this has a blurring effect, and what this suggests about its Fourier transform.
   2. Consider a 1-dimensional sequence that we want to "blur" or smooth by computing a moving average. We will replace entry $\vec{x}[j]$ with the following average:

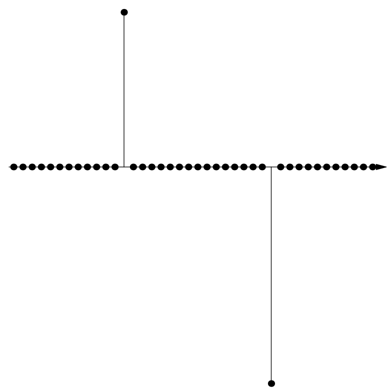
$$\widetilde{\vec{x}}[j] := \frac{1}{2w+1} \sum_{k=-w}^{w} \vec{x}[j+k].$$

   How do you represent this as a convolution, and what do you expect the Fourier coefficients of the convolution filter to look like?

> **Solution:**
>
> *Solution.*
>
> 1. The Gaussian averages nearby values causing the blur. We expect this to remove high frequency (i.e., quickly varying) components, as it does.

2. The convolution filter is $1/(2w+1)$ in a window of width w centered at 0. That is, the filter $\vec{k}$ is given by $\vec{k}[j] = 1/(2w+1)$ for $|j| \leq w$ and 0 otherwise (as usual, we interpret negative indices as wrapping around for a finite vector $\vec{k}$). The Fourier coefficients $\vec{K}$ should look like a Dirichlet kernel, converging toward a single spike as w grows (i.e., the high frequency components are decaying to zero).

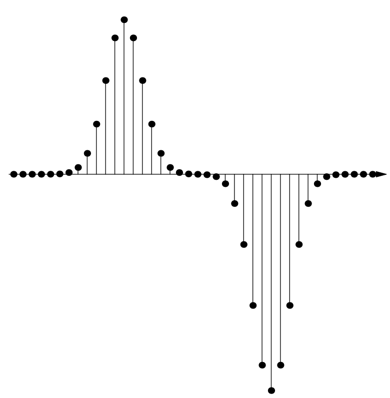
5. Suppose we are given the following vector $\vec{x} \in \mathbb{R}^{40}$:



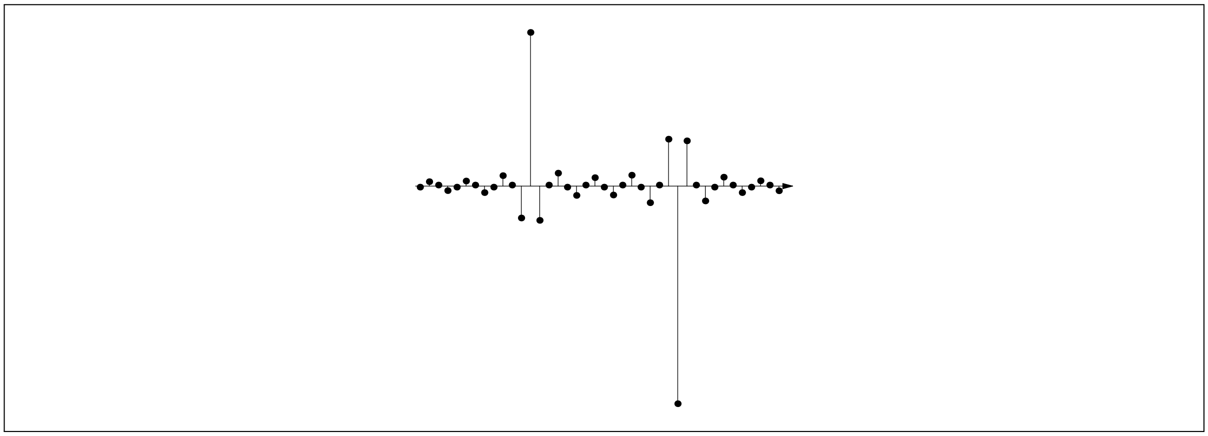
What does it look like when we convolve it with a sampled Gaussian or a Discrete sinc (Dirichlet) ?

Solution:

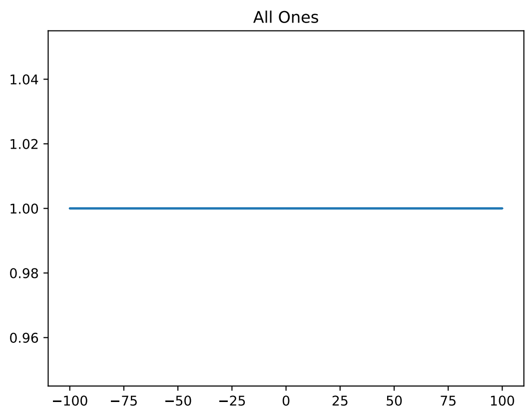
Solution. Gaussian:

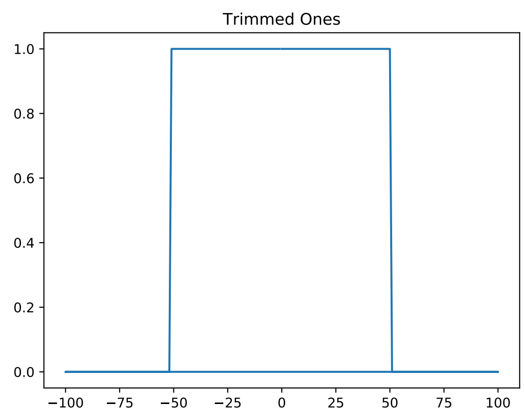


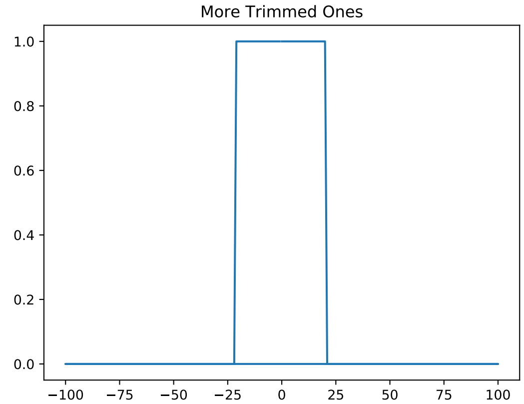
Dirichlet:



6. What are the signals corresponding to the following DFT plots?

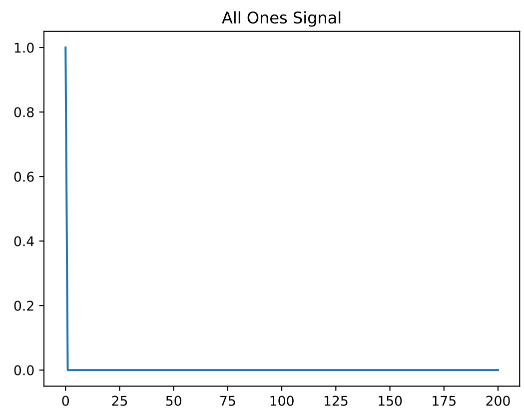


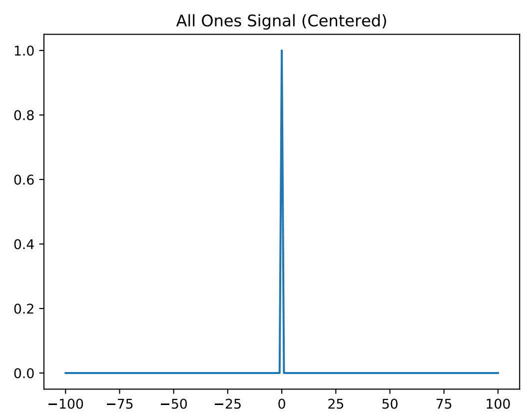


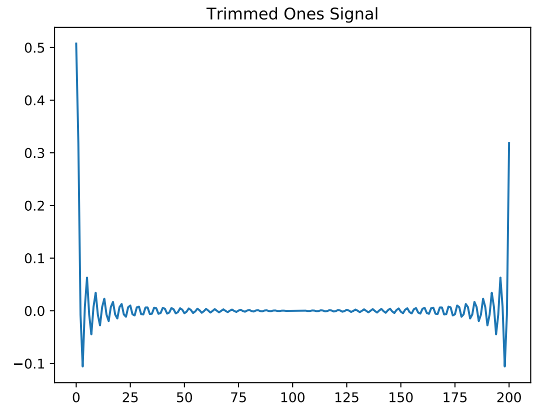


Solution:

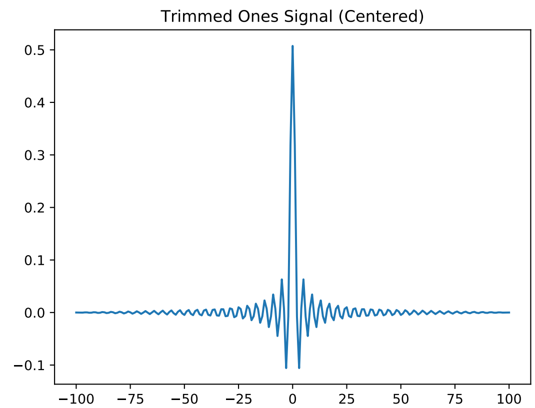
Solution. These are the DFTs corresponding to sampled versions of the Dirichlet kernels: $\frac{1}{201}D_{100}$, $\frac{1}{201}D_{50}$, and $\frac{1}{201}D_{25}$.



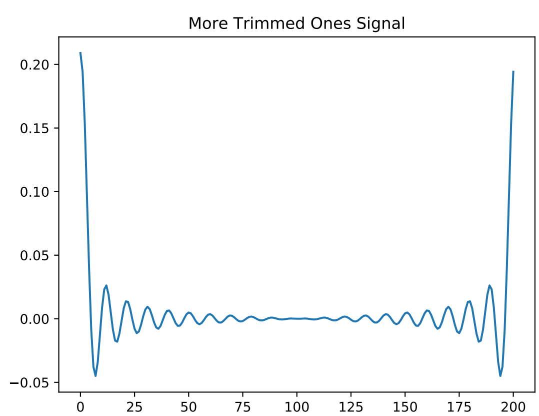




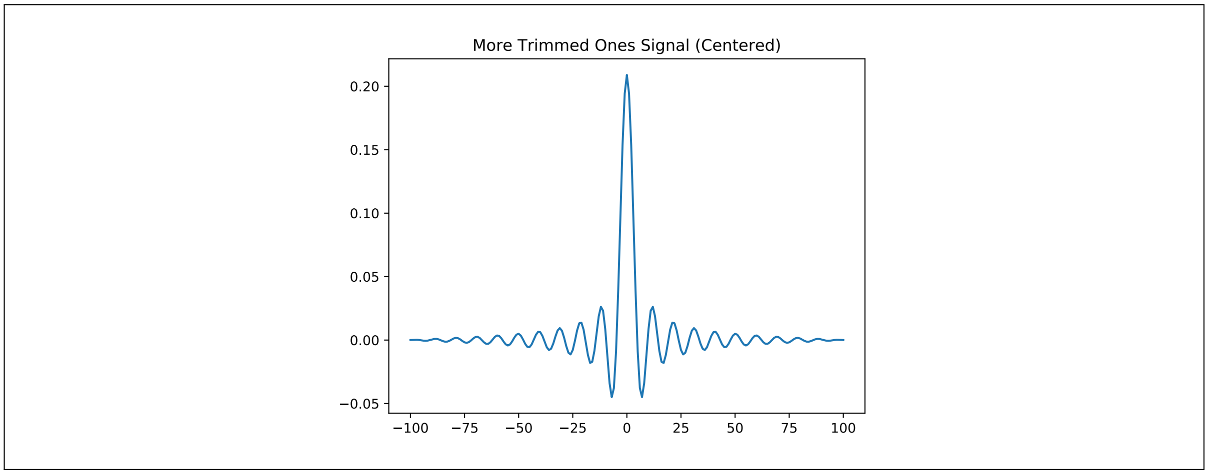
Trimmed Ones Signal
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
0
25
50
75
100
125
150
175
200



Trimmed Ones Signal (Centered)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−100
−75
−50
−25
0
25
50
75
100

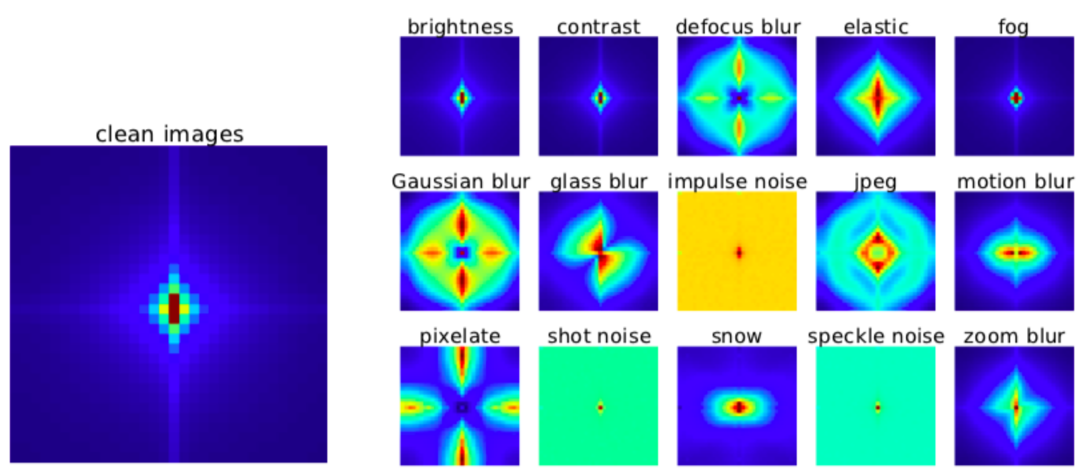


More Trimmed Ones Signal
0.20
0.15
0.10
0.05
0.00
−0.05
0
25
50
75
100
125
150
175
200



7. (Robustness to perturbations and data augmentation) We train a deep network to do image classification. We're interesting in understanding the robustness properties of the network. Particularly, we would like to know, what kind of perturbations in the frequency domain is this network robust to. (Disclaimer: a lot of this question is hand-wavy arguments)

   1. We perturb the image with Fourier basis vectors (of different frequency) with a fixed L2 norm. Which frequencies do you expect the network to be more robust to?
   2. Based on your answer to above, do you expect the network to be robust to the addition of Gaussian noise to the input?
   3. Gaussian data augmentation is a common trick in deep learning where the training data is augmented with input examples with Gaussian noise. How do you expect this to change your answer in your part(a)?
   4. The figure below shows the spectrum of different kind of corruptions found in a dataset called CIFAR10-C (We are plotting the spectrum of $C(X) - X$ where $C(X)$ is the corrupted version of X). When trained with Gaussian data augmentation, how do you expect robustness to change on speckle noise shot noise, defocus blur, Gaussian blur, fog and contrast?



Solution:

1. Low frequencies. Fig 3 and 4 in https://arxiv.org/pdf/1906.08988.pdf
2. Less robust since the error rate in high frequency region is very high.
3. Since we're corrupting high frequency information, model should reply more on low frequency information now. The error rate in high frequency region should go down. Fig 3 and 4 in https://arxiv.org/pdf/1906.08988.pdf
4. Section 4.1 of the paper says: It was observed that Gaussian data augmentation and adversarial training improve robustness to all noise and many of the blurring corruptions, while degrading robustness to fog and contrast

8. For $\vec{x}, \vec{y} \in \mathbb{C}^n$ the (circular) convolution $\vec{x} * \vec{y} \in \mathbb{C}^n$ is defined by

$$(\vec{x} * \vec{y})[k] = \sum_{i=0}^{n-1} \vec{x}[i]\vec{y}[k-i]$$

where we treat negative indices as cyclic mod n (i.e., $v[-j] = v[-j+n]$).

1. What is the DFT of $\vec{x} * \vec{y}$?
2. What is the DFT of $\vec{x} \circ \vec{y}$ where $\circ$ denotes element-wise product?

Solution:

Solution.

1. We show that $\widehat{\vec{x} * \vec{y}} = \hat{x} \circ \hat{y}$. Note that

$$\begin{aligned}
\widehat{\vec{x} * \vec{y}}[k] &= \sum_{j=0}^{n-1} (\vec{x} * \vec{y})[j] e^{-2\pi ijk/n} \\
&= \sum_{j=0}^{n-1} \sum_{p=0}^{n-1} \vec{x}[p]\vec{y}[j-p] e^{-2\pi ijk/n} \\
&= \sum_{p=0}^{n-1} \vec{x}[p] \sum_{j=0}^{n-1} \vec{y}[j] e^{-2\pi i(j+p)k/n} \\
&= \sum_{p=0}^{n-1} \vec{x}[p] e^{-2\pi ipk/n} \sum_{j=0}^{n-1} \vec{y}[j] e^{-2\pi ijk/n} \\
&= \hat{x}[k]\hat{y}[k].
\end{aligned}$$

2. We show that $\widehat{\vec{x} \circ \vec{y}} = \frac{1}{n}(\hat{x} * \hat{y})$. Using the fact that

$$\vec{x}[k] = \frac{1}{n}\langle \hat{x}, \overline{\vec{\varphi}_k} \rangle = \frac{1}{n} \sum_{p=0}^{n-1} \hat{x}[p] e^{2\pi ikp/n}$$

we have

$$
\begin{aligned}
\widehat{\vec{x}\circ\vec{y}}[k] &= \sum_{j=0}^{n-1}\vec{x}[j]\vec{y}[j]e^{-2\pi ijk/n}\\
&= \frac{1}{n}\sum_{j=0}^{n-1}\sum_{p=0}^{n-1}\hat{x}[p]e^{2\pi ipj/n}\vec{y}[j]e^{-2\pi ijk/n}\\
&= \frac{1}{n}\sum_{p=0}^{n-1}\hat{x}[p]\sum_{j=0}^{n-1}\vec{y}[j]e^{2\pi ij(p-k)/n}\\
&= \frac{1}{n}\sum_{p=0}^{n-1}\hat{x}[p]\hat{y}[k-p]\\
&= \frac{1}{n}(\hat{x}*\hat{y})[k].
\end{aligned}
$$

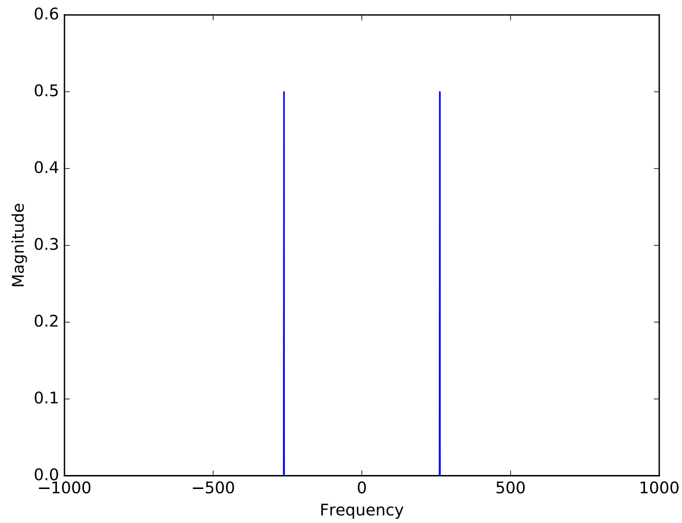
9. Let $\omega_C = 262$, $\omega_E = 330$, $\omega_G = 392$ denote the (rounded) frequencies of the (middle) C, E and G notes (in Hz). An audio signal corresponding to a note is just a sine wave of the given frequency. We sample at a rate of 44100 samples per second.

    1. What will the FFT of a signal for the C-note look like?
    2. What will the FFT of a signal for the C,E,G-chord look like?
    3. Suppose we play C,E,G,E,C. What will the FFT of that signal look like?
    4. Now suppose we filter out only the first E in the sequence above (by setting the signal to zero for the other notes). What will the FFT look like now?
    5. Now suppose we convolve the signal with a Gaussian centered at the second E. What will the FFT look like now?
    6. Suppose we are playing a single C-note again, but we extract the first 400 samples. What will the FFT look like? [Note $44100/262 \approx 168$.]
    7. Suppose we are playing a single C-note again, but we extract the first 30 samples. What will the FFT look like?
    8. In what situation would we experience aliasing?

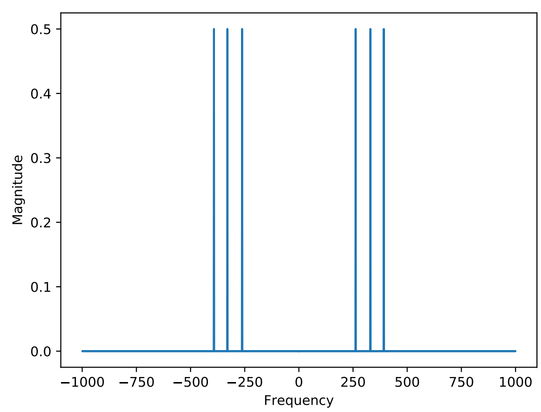
Solution:

Solution. We have scaled all FFT plots below by 1/num samples so that we see the actual amplitudes. In the cases where we window or extract, we divide by the number of remaining samples.

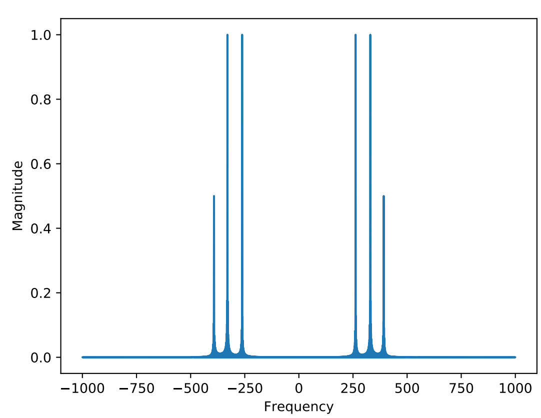
1. Shown below.



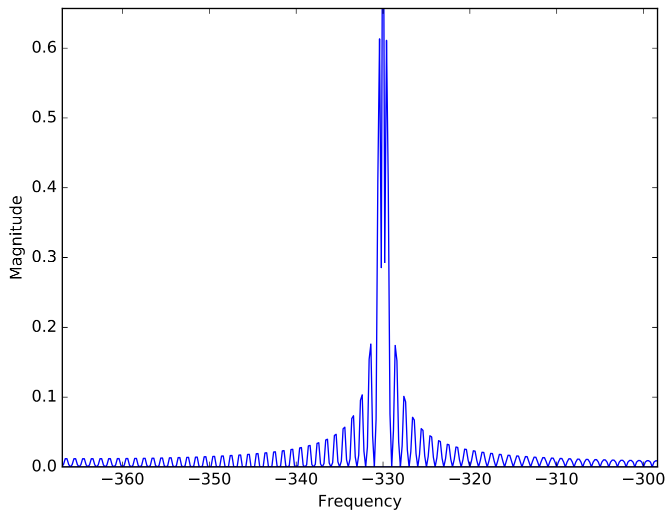
2. Shown below.



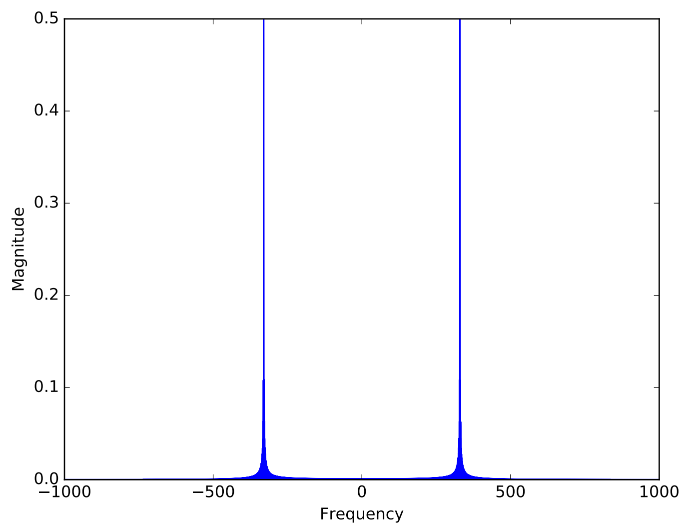
3. Shown below. You can see the effects of the Dirichlet kernels thickening the spikes below. Also note that 2 of the spikes have half the magnitude of the others, since the G note is played only once.



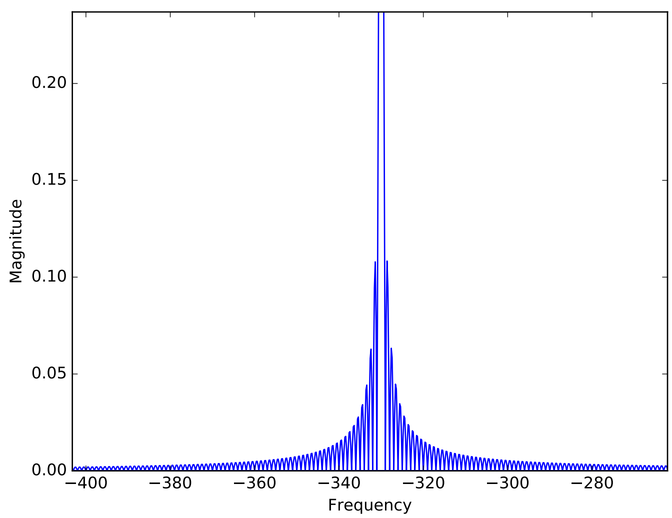
To understand this plot, note that the original signal can be thought of as a sum of 5 windowed notes. Below we zoom in on one of the spikes so you can see the effect of the Dirichlet convolution.



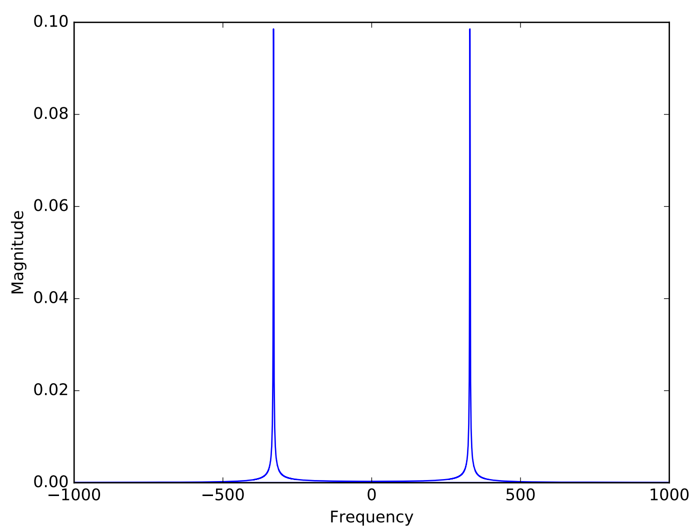
4. Shown below. It is interesting to note that the plot below is obtained from the previous plot by convolving with a Dirichlet kernel. That kernel exactly cancels out the effects of 4 other notes due to its phase.



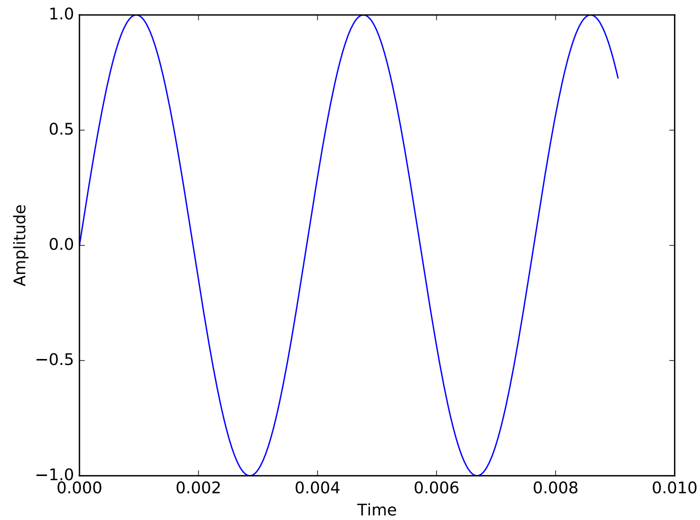
Below we zoom in so you can see the effect of the Dirichlet convolution.



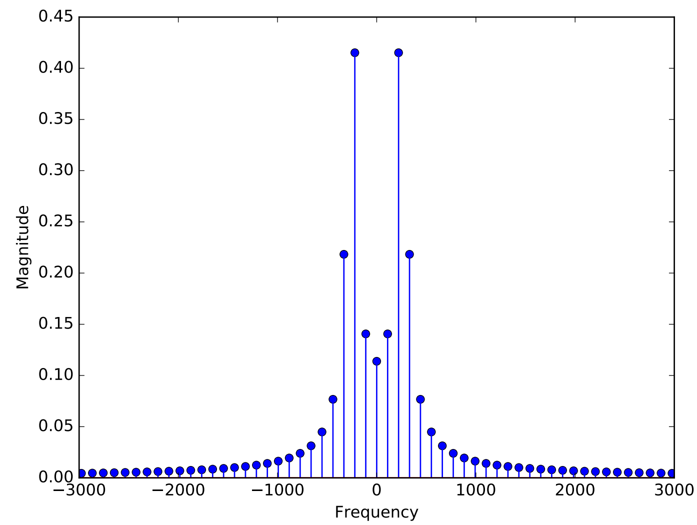
5. Shown below.



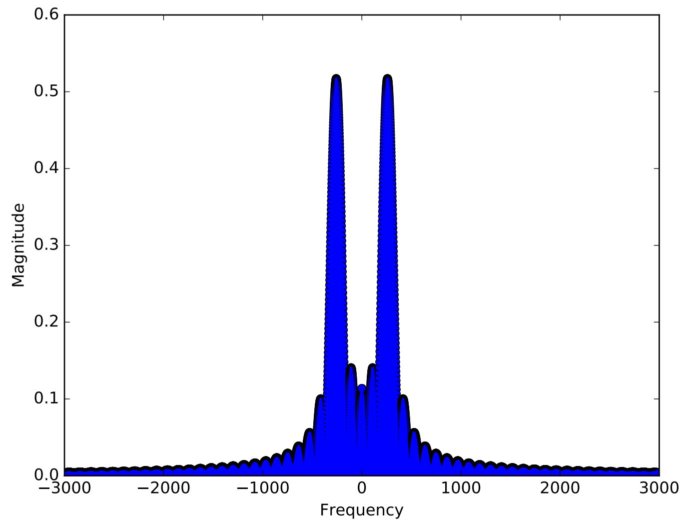
6. We first cut out the first 400 samples and treat it as a vector in $\mathbb{R}^{400}$:



This gives the following frequencies:



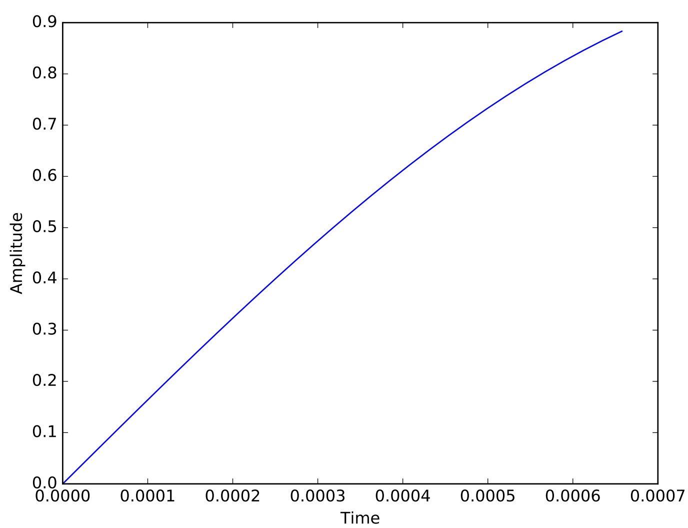
If instead we extract the first 400 samples using a window (zeroing out all other samples) we get:



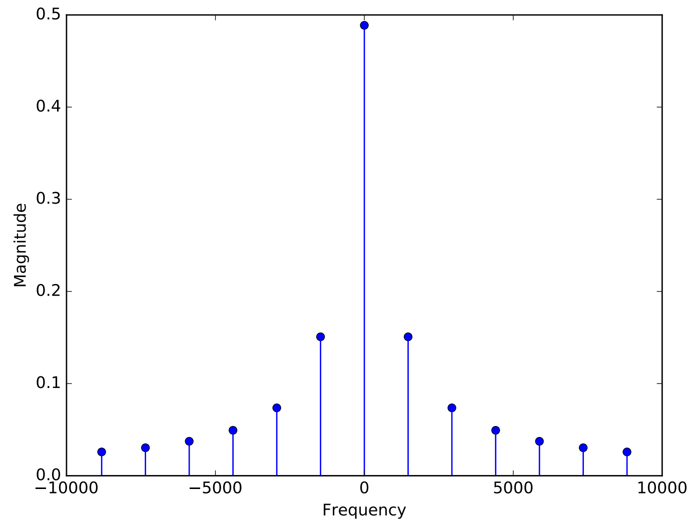
This is also the effect of a convolution with a Dirichlet kernel, but since our window is small (400 compared to the 44100 before), the resulting Dirichlet kernel is very fat (the central lobe has a large width).

Note that the extracted version is a "subsampled" version of this. [Aside: This can be seen by looking at the DFT of a spaced comb, and noting that the extracted version is the periodization of the windowed version. That is, the extracted version is obtained from the windowed version by convolving with the spaced comb.]

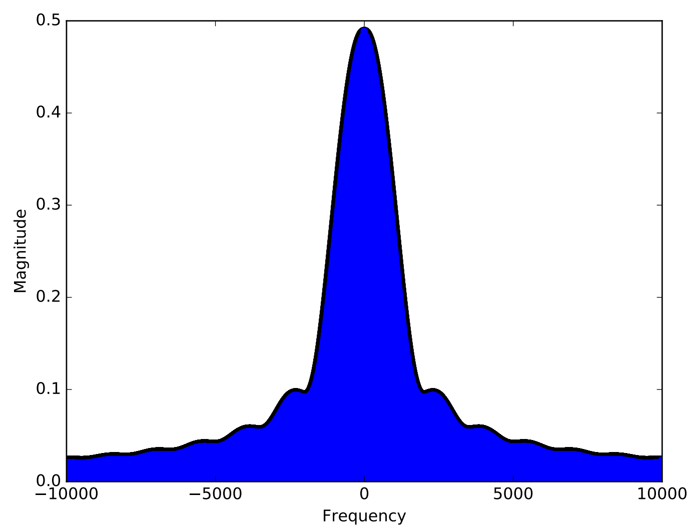
7. We first cut out the first 30 samples and treat it as a vector in $\mathbb{R}^{30}$:



This gives the following frequencies:



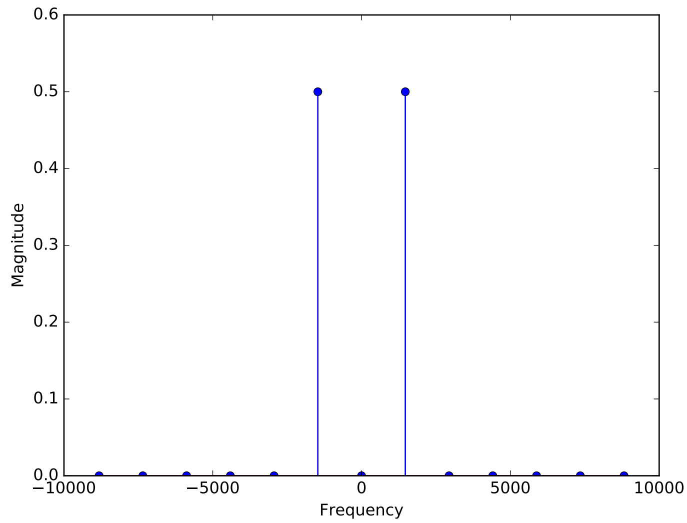
If instead we extract the first 30 samples using a window (zeroing out all other samples) we get:

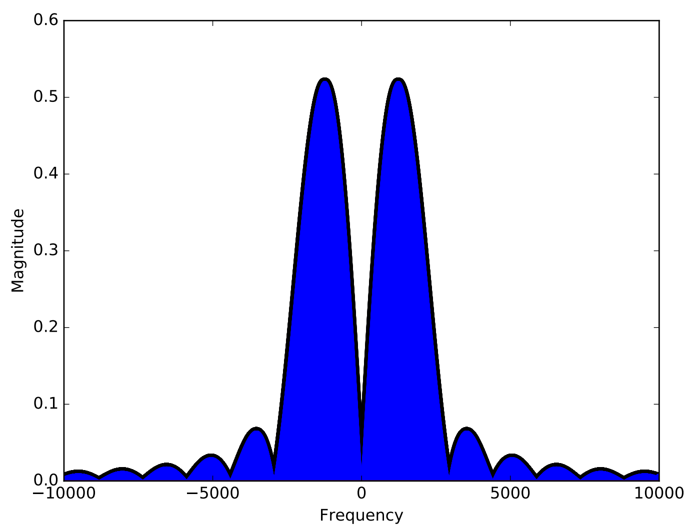


Since our window is so small, we are unable to detect "low" frequency signals that do not oscillate on the time frame of 30 samples. The smallest frequency we can see a full cycle of completes a cycle in $30/44100 = .00068$ seconds for a frequency of 1470 Hz. When you take the inner product of a slowly oscillating signal with much faster ones, you approximately get zero. This is why we only see zero.

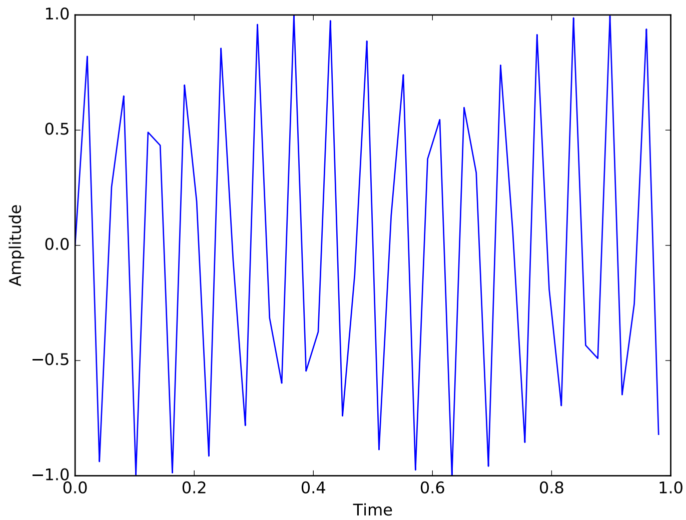
As an alternative way to understand this, the resulting Dirichlet kernel we convolve with in frequency-space has a central lobe that is so wide that it "swallows" the frequency information of the low frequencies (the first zeros of the kernel occur at approximately 1470 Hz).

Below we show the same plots (first extracted, then windowed), but for a signal with frequency 1470 Hz to illustrate what we described above.

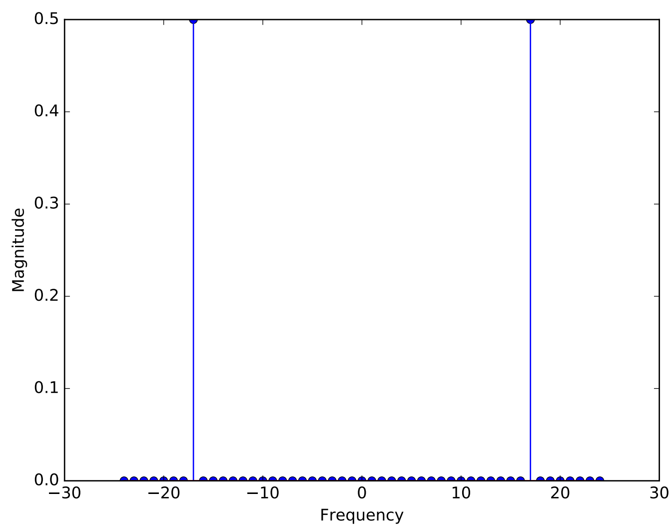




8. To observe aliasing we would need to regularly subsample our signal over the full time interval. This is a different phenomenon from cutting out an excerpt. If we aggressively undersample, we will not be able to differentiate high frequency signals from low frequency signals. Below we draw a curve through our 49 samples of the data:



As you can see, it appears to be oscillating much more slowly than it should be. Below we compute the Fourier coefficients.



We exactly see two spikes at 17 which is 262 modulo 49. This also shows why our plot of the signal above isn't perfectly sinusoidal. It is because 17 and 49 are relatively prime, so our cycles don't line up well with the subsampling.