Homework XI

Name: Shao Yanjun, Number: 19307110036

May 22, 2021

Abstract

This is Daniel's homework of "Numerical Algorithms with Case Studies II".

1 Problems

Q1 We just verify this assumption with n eigenvectors $x_j = \begin{pmatrix} 1 & \omega_n^j & \omega_n^{2j} & \cdots & \omega_n^{(n-1)j} \end{pmatrix}^T$ where $j = 0, 1, 2, \cdots, n-1$ and $\omega_n = e^{-i\frac{2\pi}{n}}$

$$\begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ 1 & & & & 0 \end{pmatrix} \begin{pmatrix} 1 \\ \omega_n^j \\ \omega_n^{2j} \\ \cdots \\ \omega_n^{(n-1)j} \end{pmatrix} = \begin{pmatrix} 1 \\ \omega_n^j \\ \omega_n^{2j} \\ \cdots \\ \omega_n^{(n-1)j} \end{pmatrix} w_n^j \tag{1}$$

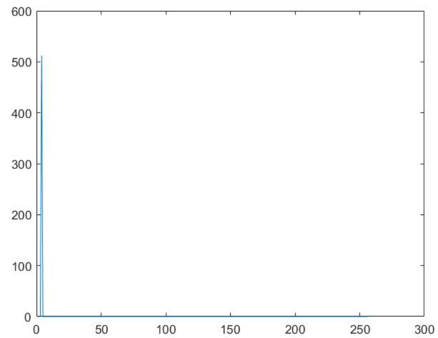
And so we will see that,

$$F_n^{-1} J_n F_n = diag\{1, \omega_n^1, \omega_n^2, \cdots, \omega_n^{(n-1)}\} \tag{2}$$

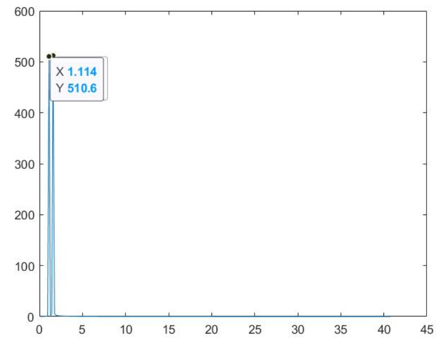
Q2 First of all, we will explain why two-sided power spectrum appear on the plot. Discrete Fourier transformation can be intepreted as follows.

$$X(f) = \sum_{k=0}^{N} x(k) e^{-i\frac{2\pi f k}{N}} = \sum_{k=N}^{0} x(N-k) e^{-i\frac{2\pi f(N-k)}{N}} = \sum_{k=0}^{N} x(k) e^{i\frac{2\pi f k}{N}} = X(-f) = X(N-f) \tag{3}$$

Eliminate the higher mirrored copy of peaks. And check it with Matlab plots, only one peak for $sin(3x)$, while $sin(x) + sin(\sqrt[12]{128}x)$ has two peaks.



(a) $sin(3x)$



(b) $sin(x) + sin(\sqrt[12]{128}x)$

Q3 The convolution theorem based on DFT is describe as follows,

$$\widehat{u * v}_k = \widehat{u}_k \cdot \widehat{v}_k \tag{4}$$

Represent $\widehat{u * v}_k$ using DFT formula,

$$\begin{aligned}
(u * v)_k &= \sum_{j=0}^{N-1} u(j) \cdot v(k-j) \qquad (5)\\
&= \sum_{j=0}^{N-1} (\frac{1}{N} \sum_{n=0}^{N-1} U(n) \cdot e^{i\frac{2\pi nj}{N}})(\frac{1}{N} \sum_{m=0}^{N-1} V(m) \cdot e^{i\frac{2\pi m(k-j)}{N}}) \qquad (6)\\
&= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} U(n)V(m) \cdot e^{i\frac{2\pi mk}{N}} (\frac{1}{N} \sum_{j=0}^{N-1} e^{i\frac{2\pi (n-m)j}{N}}) \qquad (7)
\end{aligned}$$

And we also have the following Lemma that,

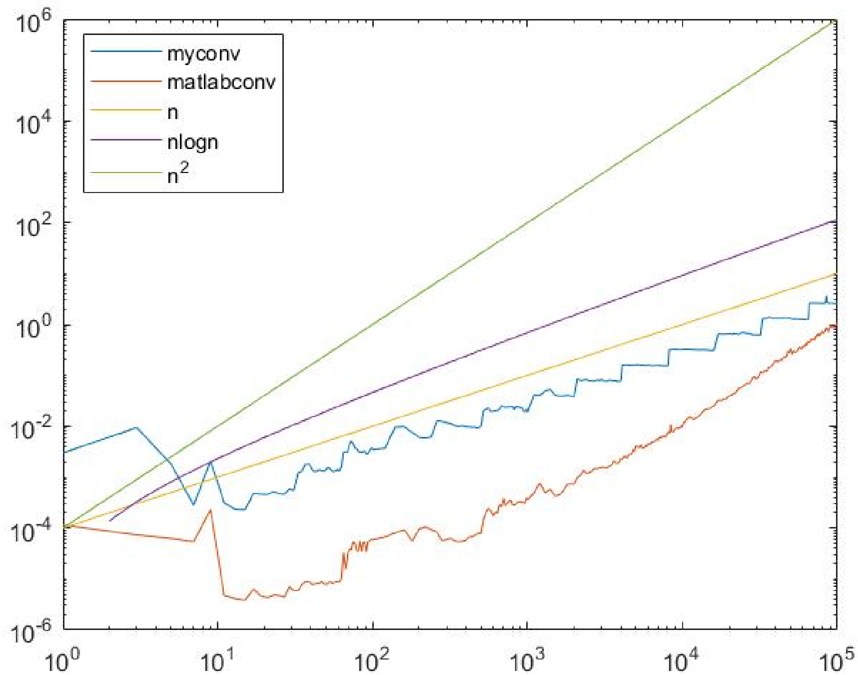
$$\begin{aligned}
\sum_{j=0}^{N-1} e^{i\frac{2\pi(n-m)j}{N}} &= \sum_{j=0}^{N-1} (e^{i\frac{2\pi(n-m)}{N}})^j \qquad (8)\\
&= \frac{1-(e^{i\frac{2\pi(n-m)}{N}})^N}{1-e^{i\frac{2\pi(n-m)}{N}}} \qquad (9)\\
&= \begin{cases} 0 & \text{if } m \neq n \\ 1 & \text{if } m = n \end{cases} \qquad (10)
\end{aligned}$$

Therefore, we also have,

$$\begin{aligned}
(u * v)_k &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} U(n)V(m) \cdot e^{i\frac{2\pi mk}{N}} (\frac{1}{N} \sum_{j=0}^{N-1} e^{i\frac{2\pi (n-m)j}{N}}) \qquad (11)\\
&= \frac{1}{N} \sum_{n=0}^{N-1} U(n)V(n) \cdot e^{i\frac{2\pi nk}{N}} \qquad (12)\\
&= \widehat{\widehat{u}_k \cdot \widehat{v}_k} \qquad (13)
\end{aligned}$$

Which means $\widehat{u * v}_k = \widehat{u}_k \cdot \widehat{v}_k$

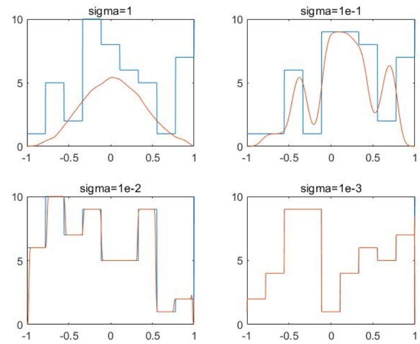
Q4 The correctness of myconv() is guaranteed by conv(). We use tic and toc in Matlab to examine the time consumption of conv() and myconv().



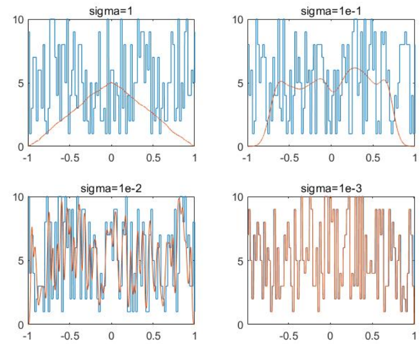
(c) Comparison

Surprisingly, the Matlab conv() is $O(n^2)$ level algorithm, while myconv() is indeed approximately $O(nlogn)$. It's natural to think that myconv() will definitely perform better than conv() for large problems.

Q5 We construct wierd() to generate a insanely and randomly discontinuous function. Use $\sigma = \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$ from Gaussian function family $\phi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$

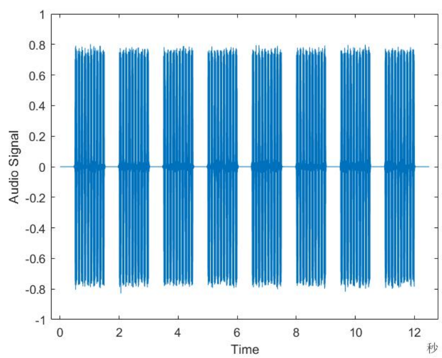


(d) Comparison1

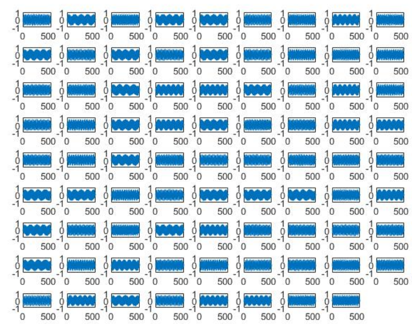


(e) Comparison2

Q6 The audio file and every single tone is divided into small parts using some trivial techniques.



(f) signal

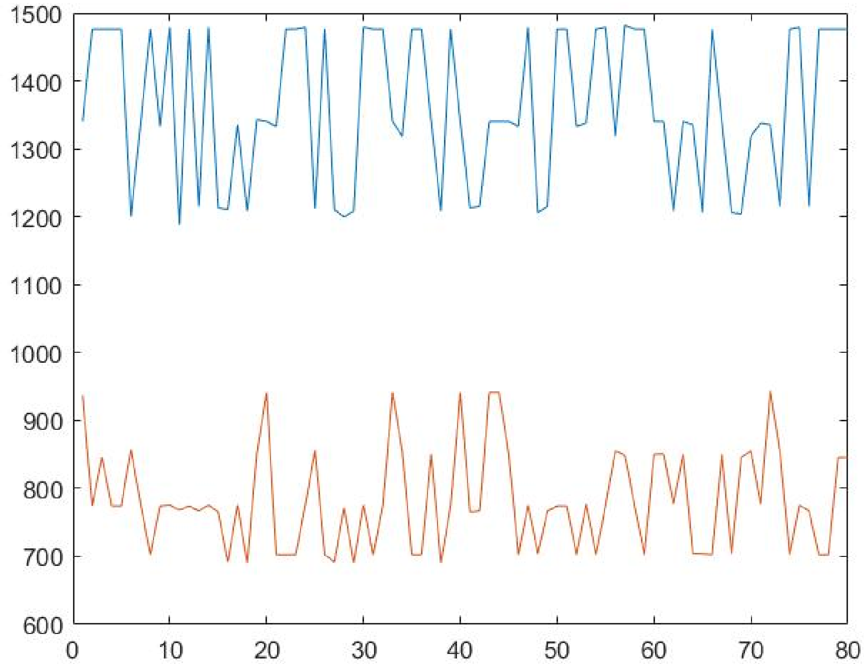


(g) division

Using fft(). Note that the frequency corresponds to the signal frequency, which is $j \cdot fs$.

(h) signal FFT

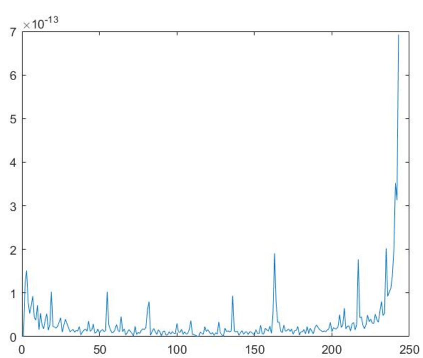
The higher part of frequency can be ignored because they are simply mirror copies of spikes. Determine the two peaks for every sound, and look up for approximation in the table, we have,



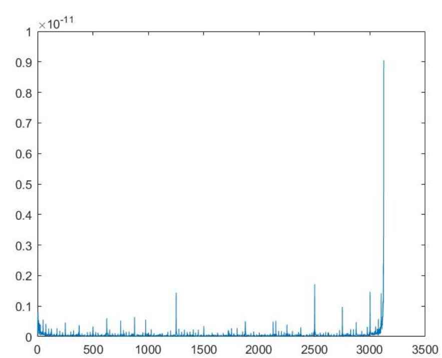
(i) signals in frequency

The recovered keys are "0696675356 4646415180 2336731416 3608338160 4400826146 6253689638 8482138178 5073643399".

Q8 The comparison of myfft3() and myfft5() with fft() in Matlab.



(j) Comparison3



(k) Comparison5