Physics 514 Computational Physics, Fall 2020

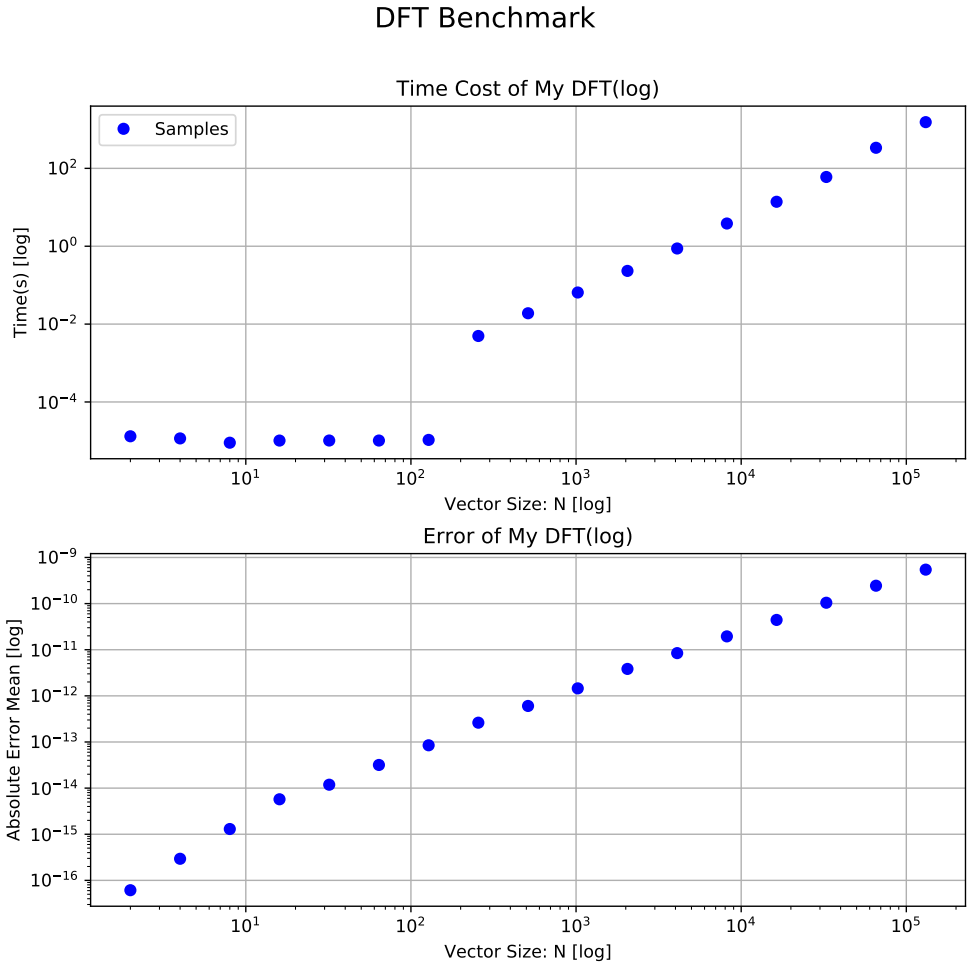
Homework 3: FFT

October 5, 2020

Author: Yunjie Wang

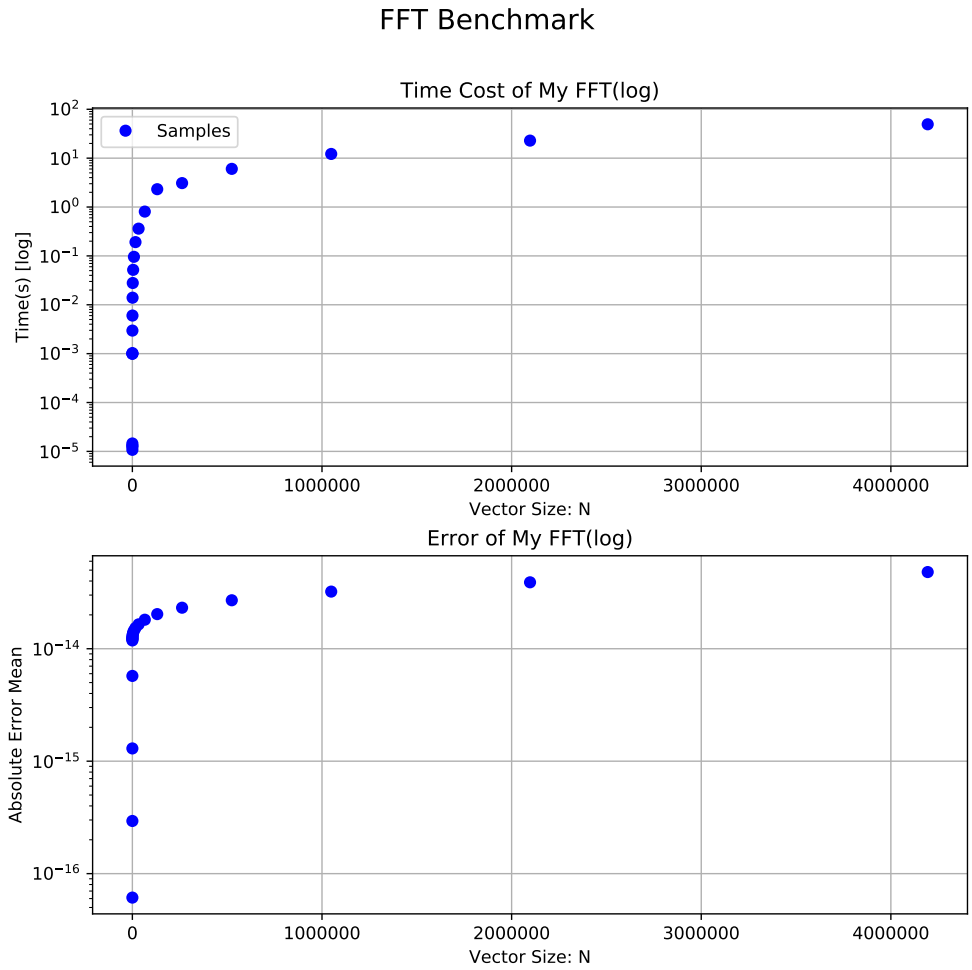
1 Problem 1

For this problem, I have two versions of DFT. One is based on matrix operations, and needs to pre-allocated quite a large space(N $\times$ N, N: length of array). It starts to give errors when N is around 10^5. The second version only allocates size N space for computation. And this method allows us to calculate longer array in order to see the trends of the time complexity. The longest vector that can be calculated within a minute is 10100 for my PC. It is a great way to visualize $\mathcal{O}(N^2)$ in the loglog plot, because the time and vector will be a linear equation in such a coordinate.



2 Problem 2

The basic idea for this problem is to divide the array into odd and even parts and then do the DFT implemented above. Once the size after division is less than 32(2^5), the algorithm will start to caculate the DFT of the array. The longest vector that can be calculated within a minute is 87100 for my PC.



Note: For simplicity, I set the size of the array to be power of 2 for problem 1 and 2.

3 Problem 3

The longest vector that can be calculated via Scipy FFT within a minute is 15905600 for my PC. And the black cross symbol. And here I use symlog for the y-scale, because the linear fit for the complexity function will reach to negative values, which is not properly defined in the log scale. The lengths of the longest vector can be calculated via all three methods are based on the linear complexity fitting curves.