Ay190 – Worksheet 6
Daniel DeFelippis
Date: January 29, 2014

# 1 The Discrete Fourier Transform

In this worksheet, we compare the performance of a function to compute a discrete fourier transform by matrix multiplication and NumPy's fast fourier transform. Surprise, surprise: NumPy's is much faster.

### a

My dft function is shown below.

```
import numpy as np

def dft(x):

    i = np.complex(0, 1)  # lets "i" be the imaginary number "i"
    w = np.exp(-2*np.pi*i / x.size)
    m = np.zeros((x.size, x.size), dtype=np.complex)
        # initializes transformation matrix

    for j in np.arange(x.size):
        for k in np.arange(x.size):
            m[j][k] = w**(j*k)

    return np.dot(m, x) # matrix multiplication
```

We compare that function to NumPy's own fft function with the code

```
test = pylab.randn(5)
myfunc = dft(test)
npfunc = np.fft.fft(test)
```

and the resulting two arrays are indeed identical.

### b

Using the code from the worksheet, we can plot the time it takes to do 100 fourier transforms on random matrices of sizes $N = 10$ to 100. I chose 100 fourier transforms for each value of N so that the resulting plot looks smoother, but it also doesn't take to long to run.

Shown below in figure 1 is a graph of the time vs the matrix size for my transform function and NumPy's transform function.

It is visually clear that for my function, the time increases as $N^2$ (e.g. double $N$ from 50 to 100, and the time $t$ goes up by a factor of 4).

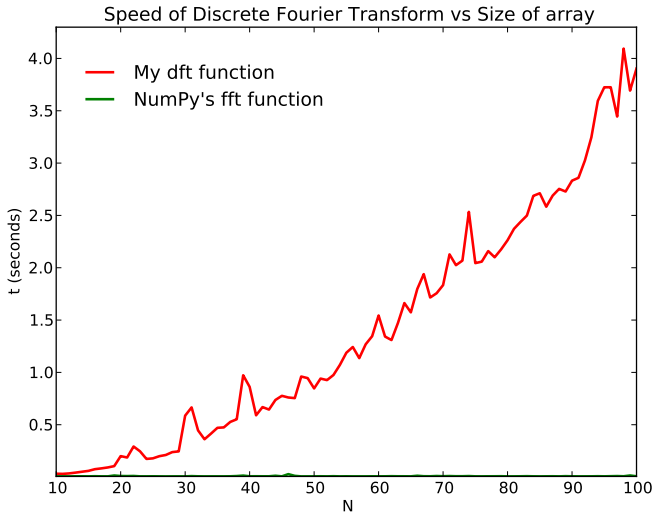


Figure 1: Performance of My dft and NumPy's fft

**c**

Note that it is difficult to see exactly how much faster NumPy's fast fourier transform is because the green line in figure 1 hovers right around $t = 0$. To see it better, we plot $\log_{10} t$ vs $N$ instead, which is shown in figure 2.

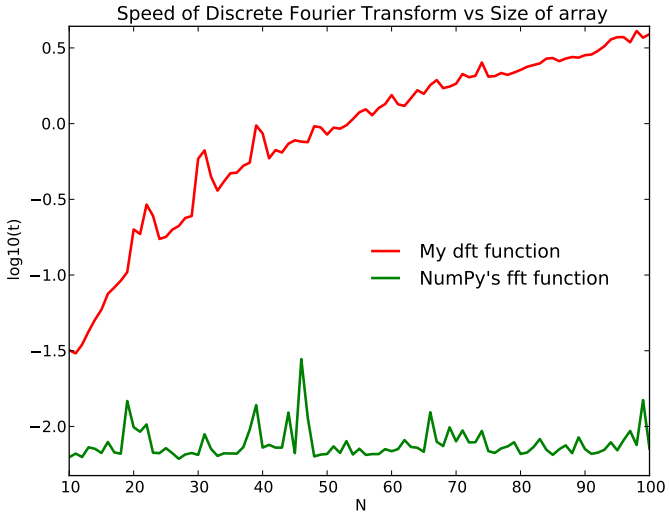


Figure 2: Same as figure 1 but with time plotted on a log scale

For $N = 100$, NumPy's fast fourier transform is about 2.5 orders of magnitude faster than the discrete fourier transform I wrote. I'm unsure why the curve is so spiky, but it probably has to do with the processing power/speed of my computer.