

CMPE 422/Spring 07/Project II: FFT and Convolution

Samir Chettri

March 19, 2007

Abstract

In class we studied convolution and I have repeatedly mentioned the FFT (Fast Fourier Transform) which is a fast way of computing the DFT (Discrete Fourier Transform). In this project we will see how quickly we can do these operations. **NOTE:** I am assigning a project involving the FFT, but this will occur prior to your being taught this important algorithm. Rest assured, there is something very important to learn in doing this project which will become evident when you perform the calculations required and plot the graphs.

Students will write a script in MATLAB to check the speed of the FFT. They will also write a script to compute the convolution using FFT's and will compare the speed without the FFT.

Keywords: FFT, convolution, fast convolution, MATLAB.

Contents

1	The FFT	1
2	Fast Convolution	1
3	What to do?	2

1 The FFT

MATLAB has a function called `fft` to compute the DFT (to be defined in class). Students should familiarize themselves with this function by typing `help fft` within MATLAB. Also do the same for `help ifft`.

Let us study the execution time of the `fft` function for $1 \leq N \leq 16384$ where N represents the size of the input vector which is generated using the function `rand`. In order to time the execution, you will need calculate the `fft` of a vector $x[n]$, $n = 1, 2, \dots, N$ and also simultaneously its execution time. For this, two useful functions are provided in MATLAB - `clock` and `etime`.

Plot your results on one graph using the command `plot`. Put N on the x axis and execution time (sec.) on the y axis. Comment on any patterns you find in the plotted graph.

2 Fast Convolution

In order to fully understand the mathematics behind what we are about to do, students can read example 8.11, page 573 of OSB and also Section 8.7 of the same book. However, in order to do the computations described below and to understand the fundamental principles without recourse to mathematics, it is not necessary to read the above mentioned section though read the example. Once this material is taught in class, students will get an additional “**ah-ha!**”

Consider $x_1[n]$ and $x_2[n]$ each respectively being N_1 and N_2 point signals. Choose N to be

$$N = 2^{\lceil \log_2(N_1 + N_2 - 1) \rceil}.$$

Note that it is permissible for $N_1 = N_2$, the above just represents the most general case. With this the linear convolution is:

$$x_1[n] * x_2[n] = IFFT [FFT [x_1[n]] \cdot FFT [x_2[n]]]$$

The issue is whether the left-hand side of the equation (done with the MATLAB function `conv`) is faster to perform or the right-hand side.

To be more specific, let $x_1[n]$ be an L -point uniformly distributed random number sequence containing values between $[0 \ 1]$ (`rand`) and $x_2[n]$ be an L -point normally distributed random number sequence of mean 0 and variance 1 (`randn`). Determine the average execution times for $1 \leq L \leq 150$ where the average is calculated by 10000 repeated trials.

Plot your results on one graph using the command `plot`. Put N on the x axis and execution time (sec.) on the y axis. Comment on any patterns you find in the plotted graph.

3 What to do?

1. Write a MATLAB script (not function) that computes the `fft` and calculates the execution time. Your script should produce the graph previously described. Comment on the graph. **Note:** In order to make some intelligent comments, you will be required to do some reading.
2. Write a MATLAB script (not function) that computes the fast convolution and regular convolution and calculates the execution time. Your script should produce the graph previously described. Comment on the graph. **Note:** In order to make some intelligent comments, you will be required to do some reading.
3. Place both your MATLAB scripts in a single file with the script name `project02.422`. Any commentary on the graphs should be in a simple text file.

Due date: 11:59pm on 13 April 2007 by email (single script file and commentary on graphs) only to Haleh Safavi.