Homework 10

Print to PDF

Discrete Fourier Transform and the Fast-Fourier Transform

Discrete Fourier Transform

1. Try Exercise 1 and Exercise 2 in section 10.8 of Karris[Kar12] by hand.
2. The eight-point Discrete Fourier Transform (DFT) of the function:

$$x[n] = \begin{cases} n & 0 \leq n \leq 4 \\ 0 & \text{otherwise} \end{cases}$$

   is given by

$$X[m] = [10, -5.41 - j4.83, 2 + j2, -2.59 - j0.83, 2, -2.59 + j0.83, 2 - j2, -5.41 + j4.83]\,.$$

   1. Sketch the time sequence $x[n]$.
   2. Use the DFT formula

$$X[m] = \sum_{n=0}^{7}, x[n] \exp\left(-j\left(\frac{2\pi m}{8}\right)n\right) = \sum_{n=0}^{7}, x[n] W_8^{nm}$$

      to confirm that the value of $X[4] = 2$.
   3. Use the DFT formula given to confirm the DC value of the signal.
   4. Confirm that the correct value of $x[1]$ is returned by the inverse DFT

$$x[n] = \frac{1}{8}\sum_{m=0}^{7}, X[m] \exp\left(j\left(\frac{2\pi n}{8}\right)m\right) = \frac{1}{8}\sum_{m=0}^{7}, X[m] W_8^{-nm}.$$

3. If we wanted to compute a 1024 point DFT from a sampled-data signal, how much speed-up could we expect to achieve using the FFT rather than the standard formula for the DFT?
4. Given that direct convolution of two digital sequences takes order n^2 floating point operations, is taking the FFT of the two 1024 point signals, multiplying them and taking the inverse FFT more efficient than direct convolution? If not, for how many samples, would using the FFT be more efficient than direct convolution?

Solutions

1. See section 10.9 of Karris[Kar12]
2. *Hints*:
   1. Sketch a point or stem plot with x-axis n and y-axis $x[n]$.
   2. Compute $X[4]$ noting that the exponential term for $m = 4$ is -1.
   3. Compute $X[0]$ but note that the exponential term is 1 for all n so this is simply the sum of $x[n]$.
3. FFT takes 9.8% of the time or is 102 times faster than DFT.
4. Convolution of two 1024 real sequences takes 1,048,576 real operations. FFT of each signal takes 10,240 complex operations, multiplication of two signals in complex domain takes 1,024 complex multiplications, inverse FFT is 10,240 more. So FFT method is around 31,000 complex operations. Even if taking into account difference between complex and real arithmetic, FFT is order 10 times faster (and needs less memory) than convolution.

Reference

See Bibliograhy.

By Dr Chris P. Jobling