The Fast Fourier Transform - Part 1

An FFT is an efficient algorithm for computing the DFT. This lecture explains some basic concepts about the FFT.
After this lecture you should be able to: define the FFT-related terms described in this lecture, derive and draw the signal flow diagram for a power-of-two decimation in time (DIT) FFT algorithm and find the number of operations of various types required to compute the FFT. We will not cover the software implementation of the FFT.
This material is covered in Sections 2.4, 2.5 and 2.6 of the text.

Complexity of the DFT

A straightforward computation of the DFT using its definition requires evaluating a sum of N values each of which is scaled by the (complex) value W_n^{kn}. This sum needs to be evaluated for each of the N frequency points. The number of complex additions and multiplications to compute the DFT is thus about N^2.

For large values of N the number of operations required becomes very large. By taking advantage of the periodicity and other properties of W_N it is possible to reduce the number of operations to approximately $N \log_2 N$. Algorithms that do this are called *Fast Fourier Transforms*, or FFTs.

Decomposition

The basis of the most common FFT algorithm is the decomposition of one DFT sum into a sum of two DFTs, each with half of the original number of points. This approach is known as *Decimation in Time* and also as the Cooley-Tukey FFT algorithm after its inventors.

The basic approach is to divide the original sequence into two sequences consisting of the even-numbered and the odd-numbered samples:

$$X_1(k) = \sum_{n=0}^{N/2-1} x_{2n} W_N^{2nk} + \sum_{n=0}^{N/2-1} x_{2n+1} W_N^{(2n+1)k}$$

Using

$$W_N^{(2n+1)k} = W_N^k W_N^{(2n)k}$$

we find

$$X_1(k) = \sum_{n=0}^{N/2-1} x_{2n} W_N^{2nk} + W_N^k \sum_{n=0}^{N/2-1} x_{2n+1} W_N^{(2n)k}$$

This equation is the sum of two DFTs of length $N/2$. The first DFT is the DFT of the even numbered samples and the second is the DFT of the odd numbered samples. However, note that the second sequence has to be scaled by W_N^k before the two $N/2$ DFTs can be combined to form the desired N-point DFT. The factor W_N^k is known as a *twiddle factor*.

This decomposition can be repeated, breaking down the sizes of the DFTs by half each time until we end up with DFTs of length 2. These basic FFTs are known as *radix-2* FFTs.

Signal Flow Graphs

A signal flow graph show the operations involved in a signal processing operation. It is often easier to visualize the operation of an algorithm by using the signal flow graphs than the algebraic representation.

The graph is composed of nodes and vectors. Nodes represent values and vectors represent the "flow" of these values into other values. The value of each node is the sum of all the other values that "flow" into it through the vectors. A value drawn beside a the vector means the value "flowing" along that vector is scaled by that value.

Exercise: Draw the signal flow graph for a radix-2 DFT.