Slide 1:Fast Fourier Transform

An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition directly; the only difference is that an FFT is much faster.

Let x0, ...., xN-1 be complex numbers. The DFT is defined by the formula

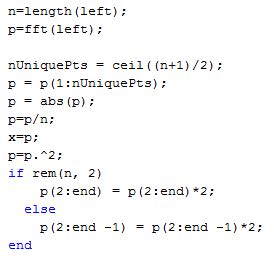
 X_k =  \sum_{n=0}^{N-1} x_n e^{-{i 2\pi k \frac{n}{N}}}
\qquad
k = 0,\dots,N-1. 

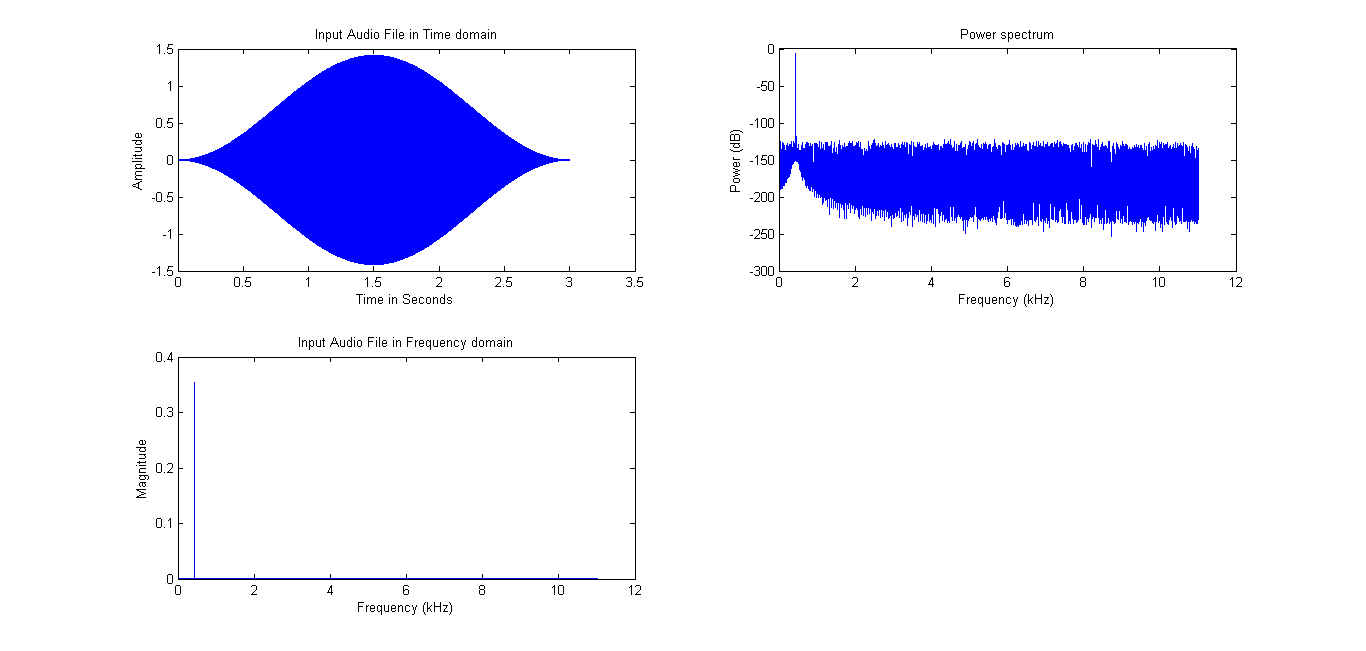
Evaluating this definition directly requires O(N2) operations: there are N outputs Xk, and each output requires a sum of N terms. An FFT is any method to compute the same results in O(N log N) operations.

Slide 2:The first code

1. We have used Matlab to make the prototype.

2. In the code here we are finding the power spectrum for an input audio (I’ll explain)





Slide 3:Monophonic note recognition

1. In the previous code however the whole input signal was taken into consideration and processed all at once. But a given signal can have a number of notes playing throughout the track.

2. We now break the input signal into frames.

Frame size=2048 samples Overlap=512 samples

