**Fast Fourier Transform**

FFT is a very efficient method for computing the DFT coefficients. It reduces the number of complex multiplications from in case of DFT to simply in case of FFT. The only restriction on the algorithm is that the sequence should consist of samples, where is a positive integer – in other words, the number of samples in the sequence should be a power of 2, i.e. etc. If does not contain samples, then we append it with zeros, until the number of samples in the resulting sequence become a power of 2.

There are number of ways in which the FFT could be computed. We shall study radix-2 FFT algorithms, namely,

* Decimation-in-Frequency method
* Decimation-in-Time method

Other types include radix-4 and the split-radix methods.

**Method of Decimation-in-Frequency**

The DFT of a given sequence can be determined by the following relationship,

|  |  |
| --- | --- |
|  | (1) |

The twiddle factor is given by

|  |  |
| --- | --- |
|  | (2) |

In this case must a power of two, i.e. it can take up values

Equation (1) can be expanded as

|  |  |
| --- | --- |
|  | (3) |

The right hand side of Equation (3) can further be split as follows

|  |  |
| --- | --- |
|  | (4) |

This can further be written as

|  |  |
| --- | --- |
|  | (5) |

Consider the second summation on the right hand side of Equation (5), i.e.,

Let us substitute

Therefore,

As

Therefore

Thus

As is a dummy variable (used for summation only), so we can use in its place, i.e.,

Thus the DCT coefficient (from Equation (5)) can be written as

|  |  |
| --- | --- |
|  | (6) |

For as an even number, from Equation (6), we can write

|  |  |
| --- | --- |
|  | (7) |

And for as an odd number, from Equation (6), we can write

|  |  |
| --- | --- |
|  | (8) |

Using the fact that

We can write

|  |  |
| --- | --- |
|  | (9) |
|  | (10) |

where and are given by

|  |  |
| --- | --- |
|  | (11) |
|  | (12) |

Or collectively

|  |  |
| --- | --- |
|  | (13) |

For the overall procedure with some example, see the companion power-point presentation “Lecture12to14 FFT”.

It could be note that the number of complex multiplications are drastically reduced (specially for large N) in case of using instead of. For a sequence with N points,

Number of complex multiplications in

Number of complex multiplications in

A similar approach can be developed for finding out the inverse DFT using “decimation in frequency method of FFT. For the inverse DFT, we have

This equation can be written as,

where as,

Therefore, in this case

where and are given by

**Method of Decimation-in-Time**

In this method, we split the input sequence into the even indexed and odd indexed sequences, as below

where as,

and

Using the fact that

We get

We define new functions

We also note that

So we get

Also keep in view

For the second half, we can write