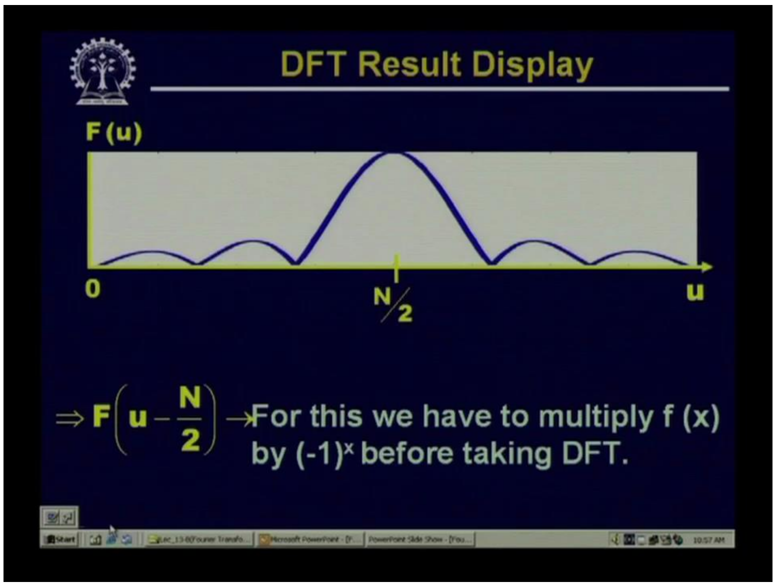
Digital Image Processing.

Professor P. K. Biswas.
Department of Electronics and Electrical Communication Engineering.
Indian Institute of Technology, Kharagpur.
Lecture-27.
Rotation Invariance Property.

(Refer Slide Time: 0:49



DFT Result Display

$\Rightarrow F\left(u-\frac{N}{2}\right) \rightarrow$ For this we have to multiply f (x) by $(-1)^x$ before taking DFT.

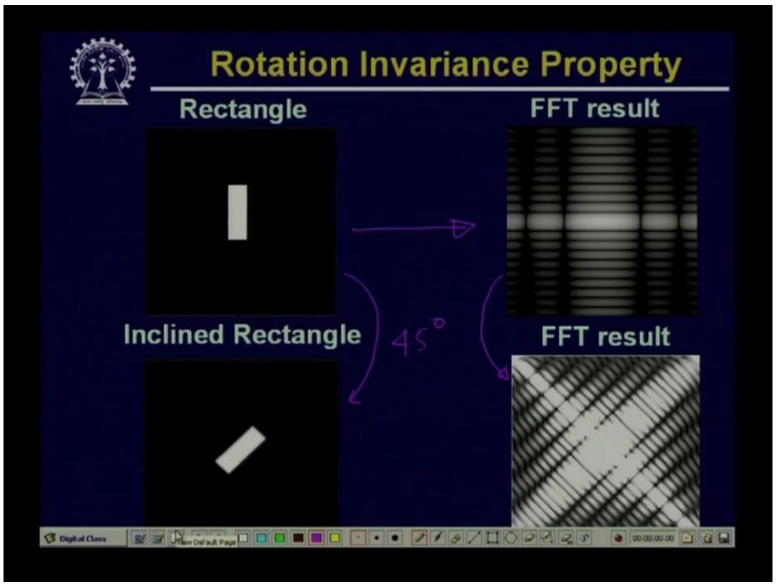
Hello. Welcome to the video lecture series on Digital Image Processing. Now Rotation property. Rotation property of the discrete fourier transformation. So to explain this rotation property, we will introduce the polar coordinated coordinate system, that is we will now replace x by r cosine theta, y will be replaced by r sine theta. Ok? u will be replaced by omega cosine phi and v will be replaced by omega sine phi.

So by this, now our original 2 dimensional signal, 2 dimensional array in the plane f(x,y) gets transformed into f(r,theta) the fourier transformation F(u,v) the fourier transform coefficients F(u,v) now gets transformed into F(omega, phi). Now using these polar coordinates, if e find out compute the fourier transformation, then it will be found that, f(r,theta + theta naught), the corresponding fourier transformation will be given by capital F(omega, phi + theta naught).

So this will be the fourier transformation there in the polar coordinate system. So this indicates our original signal was f(r,theta), if I rotate this f(r,theta) by an angle theta naught then the rotated image becomes f(r,theta + theta naught) and if I take the fourier transform of F(r,theta + theta naught) that is the rotated image which is now rotated by an angle theta naught, then the fourier transform becomes F(omega,phi + theta naught) whereF(omega, phi) was the fourier transform of the original image f(r,theta).

So this simply says, that if I rotate image f(x,y) by an angle say theta naught, its fourier transformation will also be rotated by the same angle theta naught and that is what is obvious from this particular expression because f(r,theta + theta naught) gives rise to the fourier transformation F of capital F((omega +-) omega, phi + theta naught) where if omega uhh phi was the fourier transformation of f(r,theta).

(Refer Slide Time: 3:42)

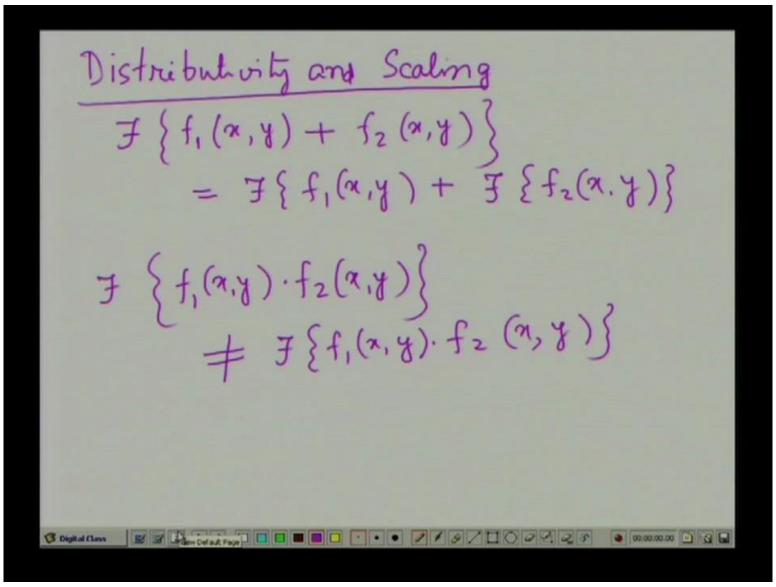


So by rotating an input image, by an angle theta naught, the corresponding fourier transform is also rotated by the same angle theta naught. So we will illustrate this, let us come to this particular figure, so here we find that we had a rectangle, an image where we have all values uhh we have, pixel values equal to 1 within a rectangle and outside this, the pixel values are equal to 0.

And the corresponding fourier transformation is this, so here the fourier transformation coefficients are uhh or the fourier spectrum is uhh a represented in the form of intensity values in an image. The second pair shows, that the same rectangle, is now rotated by an angle 45 degree, so here we have rotated this rectangle by an angle 45 degree and here you find that if you compare the fourier transformation of the original rectangle and the fourier transformation of this rotated rectangle.

Here also you find that the fourier transform coefficients, they are also rotated by the same angle of 45 degree. So this uhh illustrates the rotation property of the discrete fourier transformation. The next property that we will talk about is what is called distributivity and scaling property.

(Refer Slide Time: 5:10)



Distributivity and Scaling

$$\mathcal{F}\{f_1(x,y) + f_2(x,y)\} = \mathcal{F}\{f_1(x,y) + \mathcal{F}\{f_2(x.y)\}$$

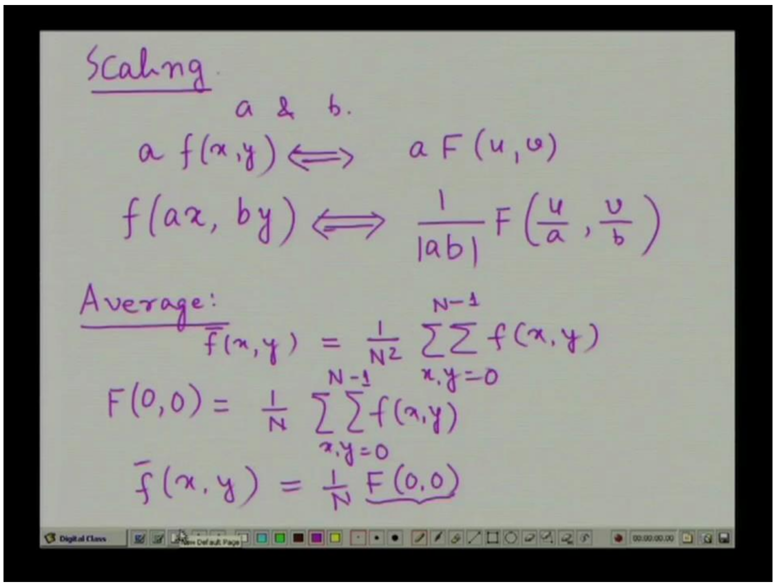
$$\mathcal{F}\{f_1(x,y) \cdot f_2(x,y)\} \neq \mathcal{F}\{f_1(x,y) \cdot f_2(x,y)\}$$

The distributinvity property says that if I take two signals, two arrays f1(x,y) and f2(x,y). So these are two arrays. Take the summation of these two arrays f1(x,y) and f2(x,y) and then you find out the fourier transformation of this particular result that is f1(x,y) + f2(x,y) and take the fourier transform of this. Now this fourier transformation will be same as the fourier transformation of f1(x,y) + fourier transformation of f2(x,y).

So this is true under addition that is for these two signals f1(x,y) and f2(x,y), if I take the addition, if I take the summation and then take the fourier transformation. The fourier transformation of this will be uhh the summation of the fourier transformation of individual signals f1(x,y) and f2(x,y). But if I take the multiplication, that is if I take f1(x,y) into f2(x,y) and take the fourier transformation of this product, this in general is not equal to the fourier transform of f1(x,y) into the fourier transform of f2(x,y).

So this shows, that the discrete fourier transformation and same is true for the inverse fourier transformation; so this shows that the discrete fourier transformation and its inverse is distributive over addition but the discrete fourier transformation and its inverse is in general not distributive over multiplication. So that distributivity property is valid for addition of the two signals but it is not in general valid for multiplication of two signals.

(Refer Slide Time: 7:55)



Scaling.

a & b.

$$a f(x,y) \Longleftrightarrow a F(u,v)$$

$$f(ax, by) \Longleftrightarrow \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

Average:

$$\bar{f}(x,y) = \frac{1}{N^2} \sum_{x,y=0}^{N-1}\sum f(x,y)$$

$$F(0,0) = \frac{1}{N} \sum_{x,y=0}^{N-1}\sum f(x,y)$$

$$\bar{f}(x,y) = \frac{1}{N} \underbrace{F(0,0)}$$

So the next property of the same discrete fourier transform that we will talk about is the scaling property. The scaling property says that if we have two scalar quantities a and b. Now given a signal f(x,y) multiply this by the scalar quantity a, its corresponding fourier transformation will be F(u,v) multiplied by the same scalar quantity a and the inverse is also true.

So if I multiply a signal by a scalar quantity a and take its fourier transformation then you will find that fourier transformation of this multiplied signal is nothing but the fourier transformation of the original signal multiplied by the same scalar quantity and the the same is true for the reverse that is also for inverse fourier transformation.

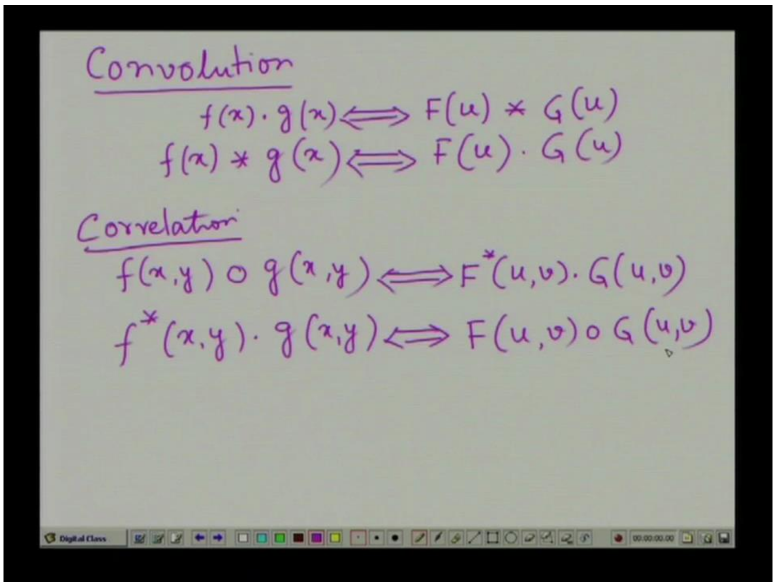
And the second one is, if I take f of (ax,by) that is now you scale the individual dimensions. x is scaled by the scalar quantity a, the dimension y is scaled by the scalar quantity b. The corresponding fourier transformation will be 1 upon a into b then fourier transformation u by a and v by b and this is the reverse. Uhh so these are the scalar scaling properties of the discrete fourier transformation.

Now we can also compute, the average value of the signal f(x,y). Now the average value for f(x,y) is given by, if I represent it like this, this is nothing but 1 upon capital N square into summation of f(x,y) where the summation has to be taken for x and y varying from 0 to capital N - 1. So this is what is the average value of the signal f(x,y). Now we find that for the fourier coefficient, the transform coefficients F(0,0).

What is this coefficient? This is nothing but 1 upon capital N then double summation f(x,y) because all the exponential terms will lead to a value 1 and this summation has to be taken for x and y varying from 0 to capital N - 1. So you find that there is a direct relation, between the average of the 2 dimensional signal, f(x,y) and its zero-eth fourier coefficient DFT coefficient.

So this clearly shows that the average value, f(x,y), the average value is nothing but 1 upon capital N into the zero-eth coefficient zero-eth discrete fourier transformation coefficient and this is nothing but because here the uhh frequency u equal to 0, frequency v equal to 0. So this is nothing but the DC component of the signal. So the DC component divided by N, that gives you the average value of the particular signal.

(Refer Slide Time: 11:50)



Convolution

$$f(x) \cdot g(x) \Longleftrightarrow F(u) * G(u)$$

$$f(x) * g(x) \Longleftrightarrow F(u) \cdot G(u)$$

Correlation

$$f(x,y) \circ g(x,y) \Longleftrightarrow F^*(u,v) \cdot G(u,v)$$

$$f^*(x,y) \cdot g(x,y) \Longleftrightarrow F(u,v) \circ G(u,v)$$

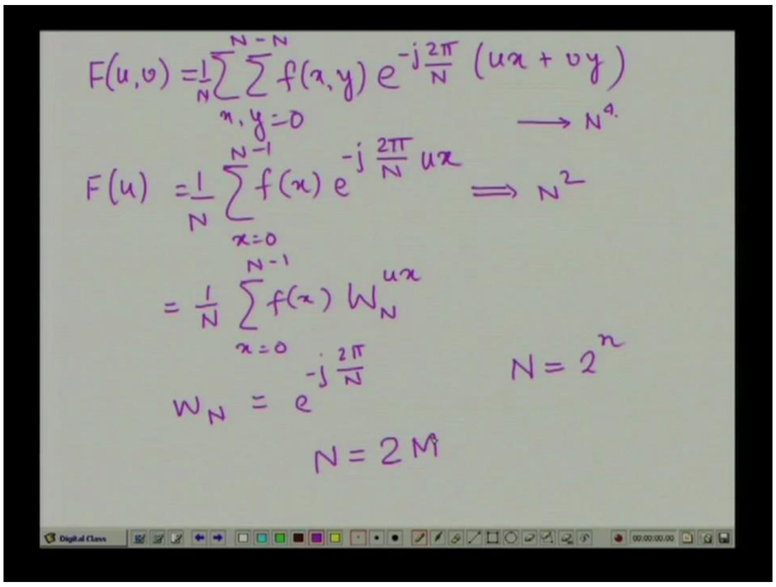
The next property, this we have already discussed in one of our earlier lectures when we have discussed about sampling and quantisation that is the convolution property. In case of convolution property you have said that if we have say two signals f(x), multiply this with the signal g(x). Then the fourier transform in the frequency domain, this is equivalent to F of (u) convolution with G of (u).

Similarly, if I take the convolution of two signals f(x) and g(x), the corresponding fourier transformation in the fourier domain, it will be the multiplication of a F(u) and G(u). So the convolution of two signals, in the spatial domain is equivalent to multiplications of the fourier transformations of the same signals in the frequency domain. On the other hand multiplication of two signals in the spatial domain is equivalent to convolution of the fourier transforms of the same signals in the frequency domain.

So this is what is known as the convolution property. The other one is called the correlation property. The correlation property says that if we have two signals f(x,y) and g(x,y), so now we are taking 2 dimensional signals and if I take the correlation of these two signals f(x,y) and g(x,y), in the frequency domain this will be equivalent to the multiplication F*(u,v) uhh where this star indicates the complex conjugate into G(u,v).

And similarly, if I take the multiplication in the spatial domain, that is f*(x,y) into g(x,y), in the frequency domain, this will be (equi**v**-) equivalent to F(u,v) correlation uhh correlation with G(u,v). So these are the two properties which are known as the convolution property and the correlation property of the fourier transformations. So with this we have discussed, the various properties of the discrete fourier transformation.

(Refer Slide Time: 14:56)



$$F(u,v) = \frac{1}{N}\sum_{x,y=0}^{N-N}\sum f(x,y)\, e^{-j\frac{2\pi}{N}(ux+vy)} \longrightarrow N^4$$

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, e^{-j\frac{2\pi}{N}ux} \Longrightarrow N^2$$

$$= \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, W_N^{ux}$$

$$W_N = e^{-j\frac{2\pi}{N}} \qquad N = 2^n$$

$$N = 2M$$

Now let us see an implementation of the fourier transformation because if you look at the expression of fourier transformation, the expression we have told many times. This is F(u,v) which is same as uhh f(x,y) e to the power - j 2 pi by capital N ux + vy where both x and y vary from 0 to capital N - 1 and this divided by 1 upon N. So if I compute, if I analyse this particular expression, which we have done earlier also in relation with unitary transformation, you will find that this text N to the power uhh 4 number of computations.

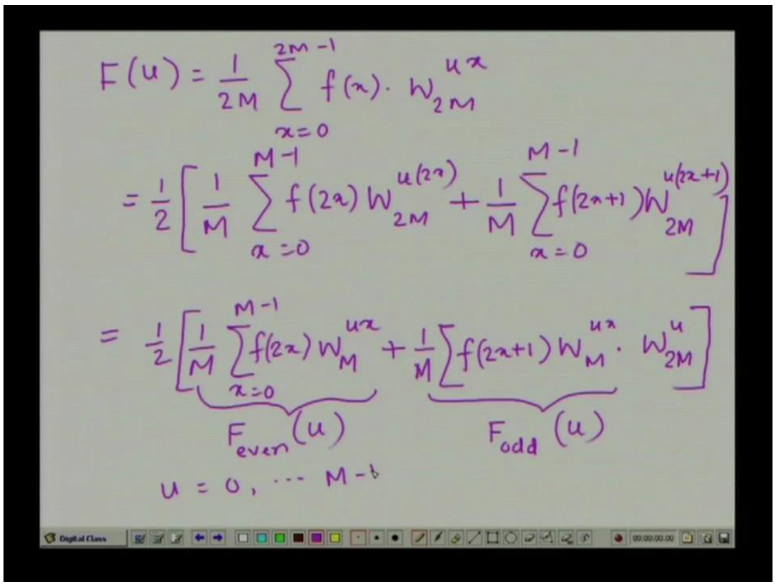
In case of 1 dimensional signal, F(u) will be given by f(x) e to the power - j 2 pi by capital N u x, summation of this over x equal to 0 to capital N - 1 and you have to scale it by 1 upon N. This particular expression takes N square number of computations. So obviously the number of computations and each of these computations are complex addition and multiplication operations.

So we find that a computational complexity of N square for a data set of size capital N is quite high. So for implementation, we have discussed earlier that if our transformations are separable in that case we can go for fast implementation of the transformations. Let us see how that fast implementation can be done in case of this discrete fourier transformation. So because of the separability property, we can implement uhh this discrete fourier transformation in a faster way.

So for that what I uhh do is, let us represent this particular expression F(u) is equal to 1 upon capital N, f(x) e to the power - j 2 pi by N ux, take the summation from x equal to 0 capital N - 1. We represent this expression in the form 1 upon N f(x) now I introduce a term W N to the power ux where x varies from 0 to capital N - 1. Now here this W N is nothing but e to the power - j 2 pi by capital N.

So we have simply introduced this term for simplification of our expressions. Now if I assume, which generally is the case that the number of samples N is of the form say 2 to the power n. So if I assume that number of samples is of this form, then this capital N can be represented as 2 into capital M. And let us see that how this particular assumption helps us.

(Refer Slide Time: 18:22)



$$F(u) = \frac{1}{2M}\sum_{x=0}^{2M-1} f(x)\cdot W_{2M}^{ux}$$

$$= \frac{1}{2}\left[\frac{1}{M}\sum_{x=0}^{M-1} f(2x)\, W_{2M}^{u(2x)} + \frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)\, W_{2M}^{u(2x+1)}\right]$$

$$= \frac{1}{2}\left[\underbrace{\frac{1}{M}\sum_{x=0}^{M-1} f(2x)\, W_{M}^{ux}}_{F_{even}(u)} + \underbrace{\frac{1}{M}\sum f(2x+1)\, W_{M}^{ux}}_{F_{odd}(u)}\cdot W_{2M}^{u}\right]$$

$$u = 0, \cdots M-1$$

And with this assumption now we can represent rewrite F(u) as 1 upon 2M because N is equal to 2M, so now I can write 2M then take the summation f(x)into W 2M to the power ux where x now varies from 0 to 2M - 1. uhh The same expression, I can rewrite as half 1 upon capital M, summation f(2x) W 2M to the power u into 2x + 1 upon capital M summation f(2x +1) W 2M to the power u into 2x + 1 where x varies from 0 to capital M - 1. Here also x varies from 0 to capital M - 1.

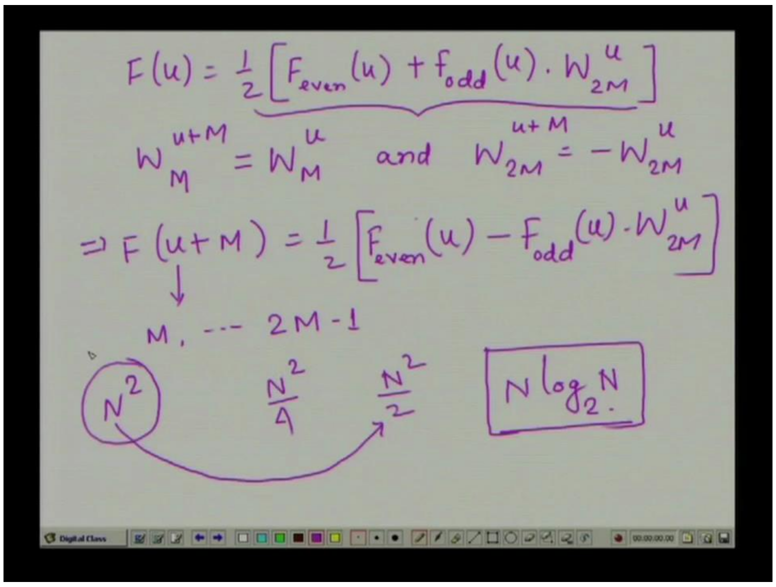
Now by this you see that what we have done. f(2x) as x varies from 0 to capital M - 1, this gives us only the even samples of our input sequence. Similarly f(2x + 1) as x varies from 0 to capital M - 1 this gives us only the odd samples of the input sequence. So we have simply separated out the even samples from the odd samples.

And if I further simplify this particular expression this expression can now be written in the form half into 1 upon capital M summation f(2x) into W capital M to the power ux where x varies from 0 to capital M - 1, + 1 upon capital M summation f(2x + 1) W M to the power ux into W 2M to the power u. So after simplification, after some simplification, the same expression can be written in this particular form.

Now if you analyse this particular expression, you find that the first summation, this one gives you the fourier transform of all the even samples, so this gives you F even (u) and this quantity in the second summation, this gives you the fourier transformation of all the odd samples, so I will write it as odd (u). And in this particular case u varies from 0 to capital M - 1.

Ok? So by separating the even samples and odd samples, I can compute that the fourier transformation of the even samples to give me F even(u), I can compute the fourier transformation of the odd samples to give me F odd(u) and then I can combine these two to give me the fourier DFT coefficients uhh of values from 0 to capital M - 1. Ok.

(Refer Slide Time: 22:34)



$$F(u) = \frac{1}{2}\underbrace{\left[F_{even}(u) + f_{odd}(u)\cdot W_{2M}^{u}\right]}$$

$$W_{M}^{u+M} = W_{M}^{u} \quad \text{and} \quad W_{2M}^{u+M} = -W_{2M}^{u}$$

$$\Rightarrow F(u+M) = \frac{1}{2}\left[F_{even}(u) - F_{odd}(u)\cdot W_{2M}^{u}\right]$$

$$\downarrow$$

$$M, \;--- \; 2M-1$$

$$\boxed{N^2} \qquad \frac{N^2}{4} \qquad \frac{N^2}{2} \qquad \boxed{N \log_2 N.}$$

Now following some more property, so effectively what we have got is, F(u) is equal to half, F even(u) + F odd(u) into W 2M to the power u. Now we we can also show, that W M to the power u + M is same as W M to the power u. This can be derived from the definition of W M and also we can find out that W 2M u + M is same as - W 2M to the power u.

So this tell us that capital F(u + capital M) is nothing but half of F even(u) - F odd(u) into W 2M to the power u. So here again, u varies from 0 to M - 1 that means this gives us the coefficient from M to 2M - 1. So I get back all the coefficients. The first part, this part gives us the coefficient from 0 to M - 1 and this half gives us the coefficients from M to 2M - 1.

Now what is the advantage that you have got? In our original formulation, we have seen that the number of complex multiplications and additions were of the order of M square. Now we have divided the N number of samples into two halves. For each of them, for each of the halves, when I compute the discrete fourier transformation, the amount of computation will be N square by 4 for each of the halves.

And the total amount of computation will be of order N square by 2 taking 2 halves considering two halves separately. So straightway we have got a reduction in the computation by a factor of 2. So it is further possible that this odd half of the samples and the even half of the samples that we have got, we can further subdivide it. So from N by 2 we can go to N by 4, from N by 4 you can go to N by 8 number of samples.

From N by 8 we can go to N by 16 number of samples and so on, until we are left with only two samples. So if I go further, breaking the sequence of samples into smaller sizes, compute the DFTs of each of those smaller size samples and then combine them together, then you will find that we can gain enormously in terms of amount of computation.

And it can be shown that for this first fourier transform implementation, the total number of computation is given by N log N, where log is taken with base 2. So this gives enormous amount of computation as uhh enormous gain in computation as against N square number of computations that is needed for direct implementation of discrete fourier transformation. Thank You.