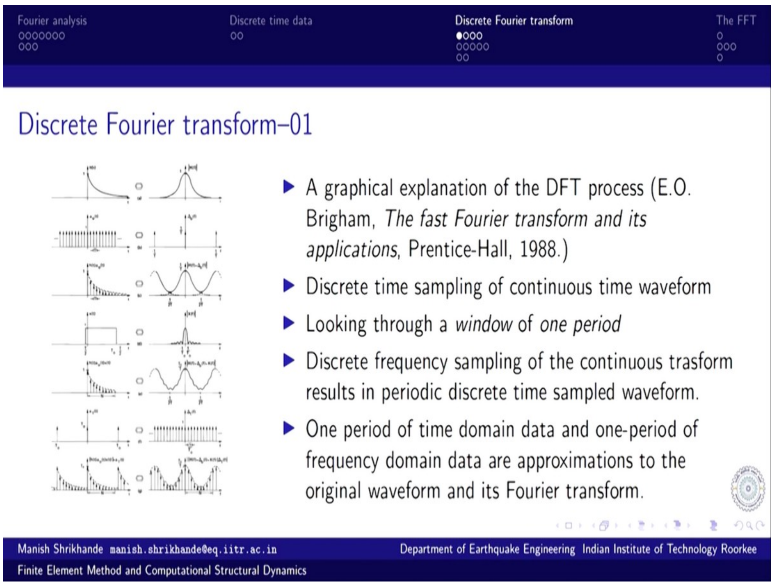
Finite Element Method and Computational Structural Dynamics
Prof. Manish Shrikhande
Department of Earthquake Engineering
Indian Institute of Technology, Roorkee

Lecture - 57
Discrete Fourier Transform-IV

Hello friends. So, we were discussing about the Discrete Fourier Transform; the discretely sampled values in time domain n number of finite number of values they have to be related to another finite set of values in frequency domain. And only then this will be the Fourier transform that we try to compute on for representation in frequency domain only when it is the cast as a map%ping of n set of discrete points on to another n set of discrete points, can it be implemented on a digital computer.

And there are several issues with the whole process I mean it the continuous wave form in time domain and corresponding Fourier transform. So, the continuous domain analogs they do not translate as seamlessly as it may appear to the casual user. There are lots of issues and lots of implications of the process of moving from continuous domain, continuous time and continuous frequency domain to discrete time and discrete frequency domain transform pairs.

(Refer Slide Time: 01:58)



So, we again to reca%pitulate the steps that are involved for transforming this continuous time signal which has its Fourier transform pair as this continuous frequency domain spectrum; frequency domain representation. So, this entire process requires first sampling with in time domain at frequency finite intervals, regular intervals and then it has to be looked through a finite. I mean some kind of finite data length has to be imposed. And that is what is done by this time window, rectangular window in we have chosen and that converts it into a finite set of data, n number of data points.

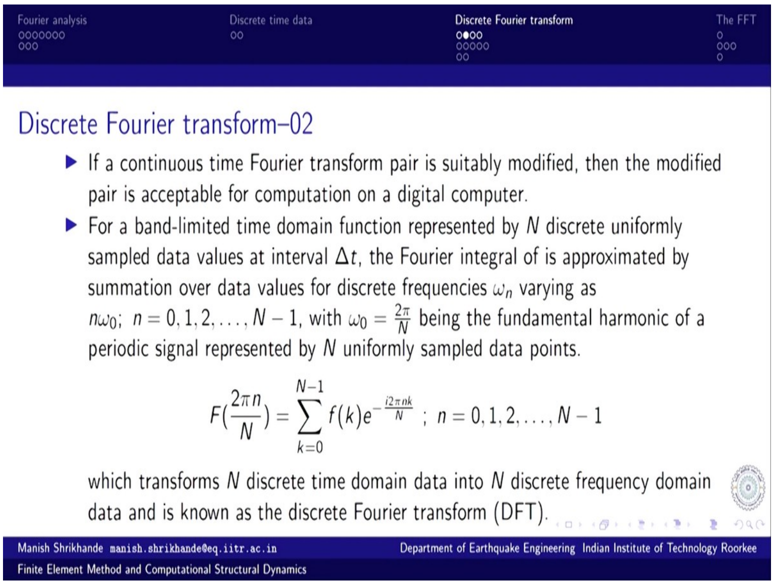
Now, this n number of data points has a continuous spectrum in frequency domain and this continuous spectrum in frequency domain also needs to be sampled discretely on discrete frequencies. And that is what is done. And when we convert this continuous frequency into discrete frequency, the side effect of that is a discrete frequency spectrum corresponds to a periodic function in time.

So, this has a side effect of this waveform that we had gets transformed into a periodic wave form with this segment that we deal with constituting one period of the infinite waveform. But if we look at it, if we look at only one period of the time domain and one period of this frequency domain representation then they, are as far as discrete values are concerned at discrete instance of time or discrete frequencies those values correspond to the values of the corresponding continuous time or continuous frequency domain representations.

So, essentially the process involves discrete time sampling of continuous time waveform and then we look through a window of one period. And then discrete sampling of the continuous transform, which has the side effect of resulting in periodic discrete time wave forms. And this discrete time waveform data is related to this discrete frequency waveform data and that is what is actually implemented on digital computer this transform.

So, if a continuous time Fourier transform pair is suitably modified, then the modified pair is acceptable for computation on a digital computer and that is the end objective. I mean we need to develop; we need to develop techniques, we need to develop algorithm; such that this transform or this information transformation can happen on a digital computer or digital computers can be programmed to implement this transformation of information.

(Refer Slide Time: 05:19)



So, for a band limited time domain function which is represented by n number of discrete uniformly sample data values let us say at interval delta t, the Fourier integral is approximated by summation over data values for discrete frequencies %omega n, which is given as a multiple of the fundamental frequency Ω_0 ; fundamental harmonic. So, n ranging from 0 to N - 1 with N = number of data points.

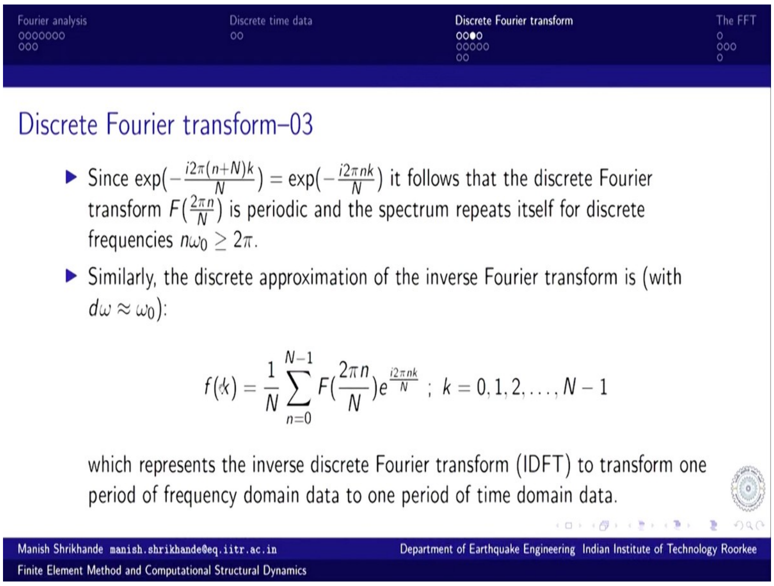
So, this representation is this discrete representation. So, on the left hand side, we have this discrete frequencies and Fourier amplitudes, Fourier transform at these discrete frequencies related to discrete time sample values, sampled at k-th instant and multiplied by of course, this is complex exponential.

Now, the point is the time is all both the time as well as frequency they are dependent on the sampling interval. And the total duration one period that is. So, one period is given by N times delta t and the time sample is given by k times delta t with the time instant. So, delta t happens to be in the numerator and denominator and it cancels out.

So, essentially we end up with just having a relationship or map%ping between discrete pairs discrete set of pairs; one for time domain signal for index k ranging from 0 to N - 1. And another set the frequency domain representation with index n ranging from 0 to N - 1.

So, this entire expression transforms N number of discrete time data into N number of discrete frequency domain data, and this is known as discrete Fourier transform or DFT.

(Refer Slide Time: 07:39)



So, since the complex exponentials, they are periodic with the period of 2π. So, the discrete Fourier transform is of course, periodic and the spectrum the frequency domain, in the frequency domain repeats itself for discrete frequencies n times %omega_0 greater than 2π.

So, 0 whatever the ordinate is amplitude Fourier spectrum value is at n %omega_0 = 0, the same Fourier amplitude is Fourier spectrum is expected at 2π that is the completion of the one period or beginning of the next period.

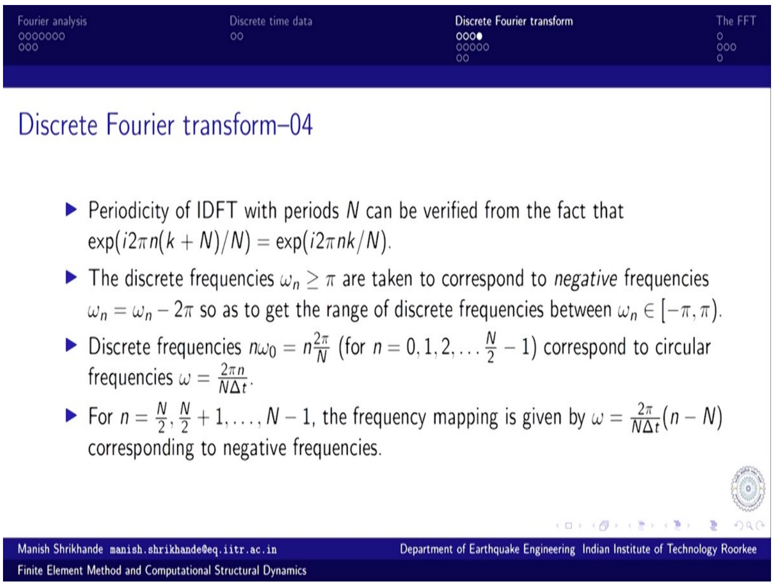
So, similarly, I mean we talked about going from time domain to frequency domain and we can implement the inverse Fourier transform also in the same way with representation d %omega the in the integral being represented approximated as the fundamental harmonic the increment in frequency so, %omega_0. So f(k), the k-th sample, time domain sample can be obtained as linear combination.

So, complex exponential again. So, $f(\frac{2\pi n}{N})e^{\frac{i2\pi nk}{N}}$. Summation over index n and for each value of k ranging from 0 to N - 1. So, this inverse Fourier transform leads to gives us the time domain signal back.

And this represents the inverse discrete Fourier transform or IDFT to transform one period of frequency domain data; period in frequency domain. So, 0 to 2π. That one period of frequency domain data is mapped on to one period of time domain data. So, perfectly fine.

So, we have n number of time domain data which gets mapped on to n number of frequency domain data. So that, obviously, suggests that I mean it is a simple map%ping. And obviously, that seems to be a kind of transformation and we do emphasize here that, this transformation can be represented; I mean this entire process of Fourier transform, forward Fourier transform or inverse Fourier transform it can be represented as a linear transformation, simple linear transformation as a matrix vector multiplication. How? We will let us see.

(Refer Slide Time: 10:33)



So, periodicity of IDFT with periods N can be verified from the fact that this, if we multiply by increase the period k + N and by considering that complex exponentials they are periodic with integer power of 2π. Then this can be verified that this rolls back onto this original expression.

So, this period N that indicates that the time domain, the signal that we get from inverse discrete Fourier transform is periodic with period of N samples. Now, discrete frequencies ω_n greater than π so; that means, $n \times \omega_0$ if it is greater than π then, they are taken to correspond to negative frequencies.

So, negative frequencies that will be $\omega_n = \omega_n - 2\pi;$ so, as to get the complete range of discrete frequencies. So now, we I mean because, if Fourier transform of course, it is defined from positive to negative frequencies and it has to be defined for one complete period of 2π .

So, that range I mean once we take this representation that frequencies above π ; discrete frequencies is greater than π they are actually negative frequencies. And then when we map it back into the range $-\pi$ to π by subtracting 2π from these values and we get the discrete frequencies as between $-\pi$ to π .

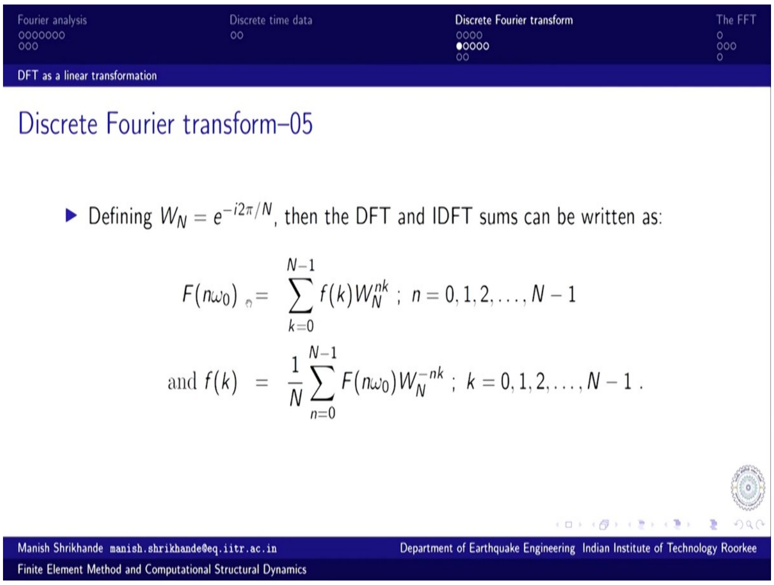
And the discrete frequencies n times ω_0 , where ω_0 is $\frac{2\pi}{N}$. So, the entire period 2π in frequency domain is divided into equal intervals so, n number of intervals. And each of these frequencies discrete frequencies, designating or delineating these intervals they are defined by n times 2π by N where n ranges from 0 to $\frac{N}{2}-1$.

So, that is one half of the spectrum. And they, these correspond to the circular frequencies if I need to go back to; I mean what is the physical frequency unit in terms of radians per second the wave frequency of the wave form that we are also used to.

Then that radiance per second frequency, the circular frequency is related to these discrete frequencies as $\frac{2\pi n}{N\Delta t}$. So, this delta t is the additional factor that comes here other and that gives you the radians per second units for circular frequency.

And for points I mean this is only for positive frequencies or defined for 0 to $\frac{N}{2}-1$. So, this is $\frac{N}{2}$ number of data points. For greater than I mean the other half of the data that corresponds to the negative frequencies and the frequency mapping is given by $\omega = \frac{2\pi}{N \times \Delta t}(n-N)$. Where small lower case n is the sample number and capital N uppercase N is the total number of data points, and these correspond to the negative frequencies.

(Refer Slide Time: 14:39)

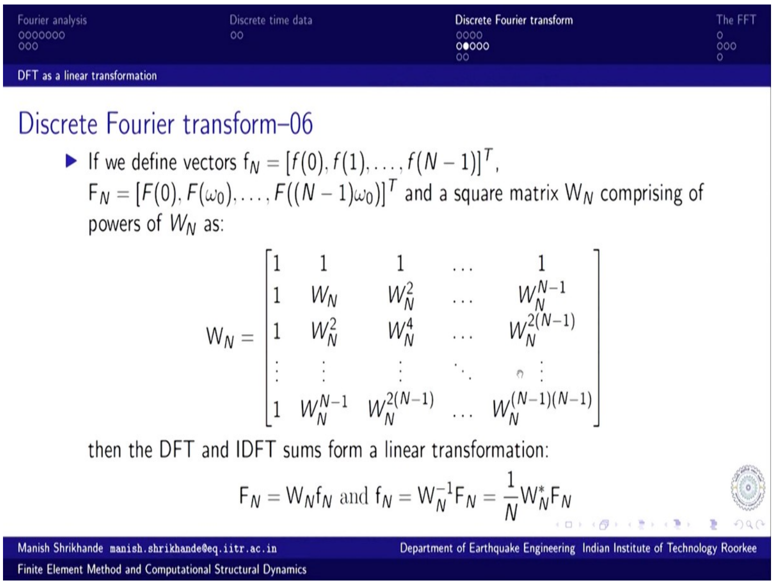


And now, defining the complex exponential e raise to the power $\frac{-i2\pi}{N} = W_N$ then the DFT and IDFT sums can be written as simply powers of this a complex exponential W. So, $\sum_{k=0}^{N-1} W_N^{nk} f(k)$. And that gives us the frequency domain representation Fourier transform as a function of discrete frequencies $n\omega_0$.

And similarly the inverse Fourier transform, if I substitute this complex exponential as W_N then it the Fourier the summation can be represented as F of the Fourier transform, the Fourier amplitudes F($n\omega_0$) the discrete frequencies n-th discrete frequency multiplied by W_N^{nk} . And summation extending over index n ranging from 0 to N - 1 and the entire sum is divided by N.

And this gives us the samples at k = 0, 1, 2 and up to N - 1 so, perfectly fine. So, this is a linear summation and we can always cast this I mean n and k there are two indices here. So, this W_N^{nk} can be cast as matrix; I mean, there are rows and columns. So, we can always represent this as two indices n and k; and this entire representation of forward transform and inverse discrete Fourier transform that can be represented as a linear transformation as this matrix.

(Refer Slide Time: 16:44)



So, if we define vectors f_N as the sample discrete time samples, represented in one vector. And similarly, if we define these vectors I mean the samples of the frequency domain representation, Fourier transform so, I have complex numbers at discrete frequencies then that is a another set of vector.

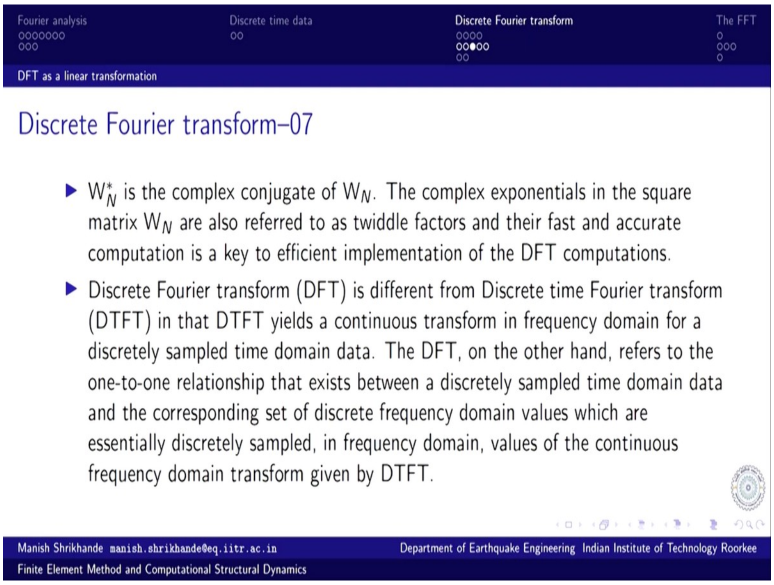
And we define this matrix W_N, which is actually the powers of W_N that is $e^{\frac{i2\pi}{n}}$ different powers of that. So, we can this matrix leads to this square matrix it leads to the square matrix. And then using this, this discrete Fourier transform and inverse discrete Fourier transform they form a linear transformation. In the sense that F_N the frequency domain representation = simply this matrix W_N multiplied by the time domain samples.

So, N number of complex numbers that we have here, frequency domain data they are related to N number of time domain samples through this transformation matrix W_N and this is N by N transformation matrix square matrix. Quite obviously, F_N if this is all the linear transformation then this time domain signal F_N can be recovered from the frequency domain signal F_N through this inverse relationship.

So, $W_N^{-1} F_N$. And if we take this inverse of this complex exponential, we find that the inverse is defined by the conjugate of this original matrix. So, the matrix is essentially, because not a surprise nothing to be surprised here, because the matrix W_N all the terms are complex exponentials and complex exponentials essentially trigonometric functions they are orthogonal functions.

So, if I take the conjugate of them that is going to be an orthogonal matrix. So, $W_N W_N^*$ is going to be identity matrix. So, conjugate matrix is going to be the inverse of this matrix of complex exponentials. So, that saves our trouble. As I said inverse is never ever computed and in this particular case, because the matrix is orthogonal. So, this is defined as the simple conjugation of the operation. And this inverse operator actually effectively implements as the conjugate of the complex numbers that we are dealing with.

(Refer Slide Time: 19:57)



So, W_N^* is the complex conjugate of W_N. The complex exponentials in the square matrix W_N they are also referred to as twiddle factors. And their fast and accurate computation is a key to efficient implementation of DFT computation. So, all that depends is now, this evaluation of these powers of complex exponentials and that is what makes this matrix.

So, if we can compute this quickly, the problem is solved the transformation is simply matrix vector multiplication and if we take; I mean if we somehow incur lot of effort and cost in computation of this complex exponentials then obviously, this is not going to be a very useful transformation, if it is very expensive to compute.

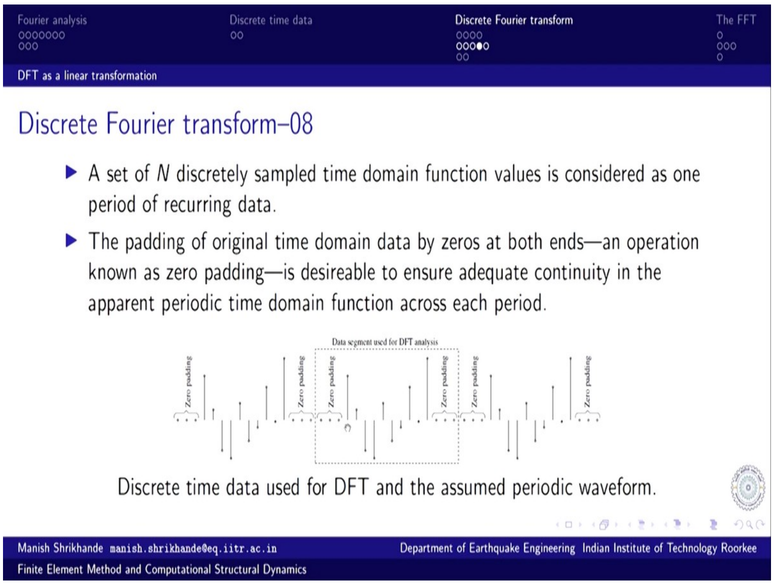
So, discrete Fourier transform as we have just discussed just to reemphasize that it is different from another similarly, similar sounding term that is Discrete Time Fourier Transform. So, Discrete Fourier Transform DFT is different from Discrete Time Fourier Transform that is DTFT in that DTFT; discrete time Fourier transforming that is the only the time domain waveform is sampled discretely.

The Fourier transform of that time domain discretely sample time domain data is going to be a continuous transform in frequency domain, right. It is not in frequency domain just because the signal is sample discretely at discrete times does not imply that in frequency domain that will be defined at discrete frequencies, it is still a continuous frequency signal.

So, that is what discrete Fourier transform differs from discrete time Fourier transform. So, DFT is a one to one relationship between one set of discrete data onto another set of discrete data. So, the equivalence between the two; the continuous domain and the discrete domain is that for at these discrete values, discrete frequencies the spectrum value that we compute they correspond to the spectrum value of the continuous transform that would have been obtained for DTFT; discrete time samples or the continuous transform of distribute time samples.

So, that is why it is a useful approximation and it give allows us to compute and make inference draw inference for physical application, but it is to be understood very clearly that DTFT and DFT they are two different things and they are not to be confused with each other.

(Refer Slide Time: 23:22)



So, a set of N time N number of discretely sample time domain data function values is considered as one period of recurring data. So, this is what I try to present here. So, this is a period of recurring data and what we do in case of DFT analysis is look at one period of this. So, this is the data segment that we use for analysis. So, N number of data points.

But what is interpreted or what is the procedure, DFT procedure looks at this one period of data is that this is only a one period of the data window that we are looking at of infinite waveform. And this entire segment which we are working with it keeps on repeating itself at infinitum.

So, this is an infinite waveform and this one period patch keeps on repeating itself. So, this is what the original data I mean the data that is interpreted by the DFT procedure looks like. And effectively we are actually using only one period of the data.

And we the emphasis is that we hope that by just analyzing this one period of data of this discrete in this discrete Fourier transform pair, the inferences that we draw that can be applied to the corresponding continuous time and continuous frequency functions and the useful inferences can be drawn.

So, here we see that in addition to these non-zero values there are zero padding in the data. So, that is because I mean we will see the importance of the zero padding that is a very very crucial, but in this particular case I would it is this process is called zero padding. One way one of the reasons is that for the many algorithms to calculate DFT it is required that the total data length should be integer power of 2. So, N should be an integer power of 2. So, that is what we deal with in radix two algorithm for computation of DFT

So, naturally whatever data we have whatever data length we have. So, we in order to make it integer power of 2 what we do is we just add more zeros to the data length. So, it really does not matter whether we, where we take the padding we can either put it at the beginning of it, we can put it at the end of it, because eventually it is going to be one period and the entire thing keeps repeating.

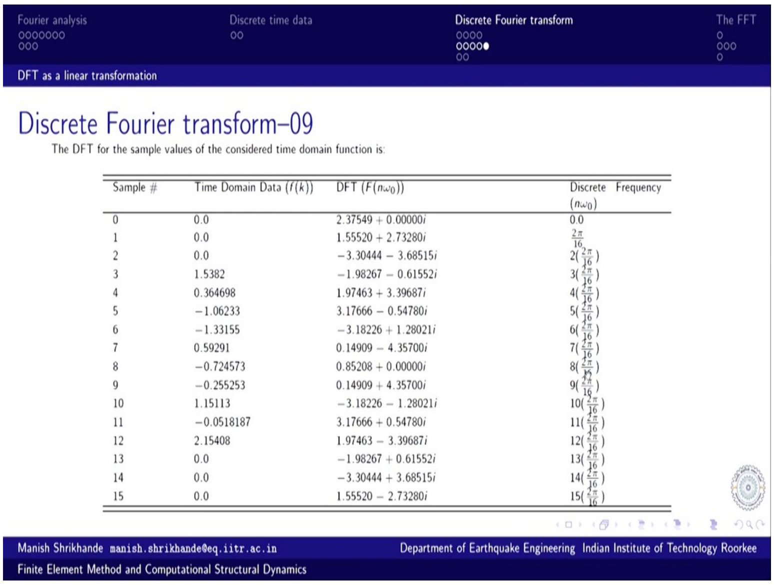
So, but I have my own preferred way. I mean I try to enclose or pad zeros on both sides of the actual data, but it really does not matter where you put the zeros as long as there are enough number of zeros in the data. So, one reason is of course, from the computational point of view the algorithm requires the data length to be integer power of 2, but there is another important reason for this data zero padding.

And that we will see in when we discuss some of the applications the importance of this zero padding. And fortunately, because of the requirement of computational algorithms that data length has to be integer power of 2. So, we eventually end up padding enough number of zeros. And the real reason why these zeros are required they are taken care of automatically without the user even realizing the importance of the zeros.

So, these zeros I mean they are very very important and they need to be retained they just they should not be thrown away lightly. So, zero is really important it is not something that we can discard and without any effect. We will see that what is the significance of these zeros when we discuss some of the applications of discrete Fourier transform the application examples.

So, a simple example of this time domain data this time domain data that we have here. So, I take these as six points here and there are ten points here in between; so, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16. So, total 16 points, 16 data points.

(Refer Slide Time: 28:54)



And these are the data points that we have. So, three zeros and three zeros at the end. So, these are the zero padding before and at the end and then the some random data that I generated. So, one point I mean anything it is just a random waveform and when I compute the Fourier transform by using these linear transformation then this is what the Fourier transform is.

Now, one thing and I also map the sample number time domain sample numbers I also map it to discrete frequency samples. So, here the index variable is k and here the index variable is n on this column last column. And this n represents the essentially the index for Fourier transform frequencies.

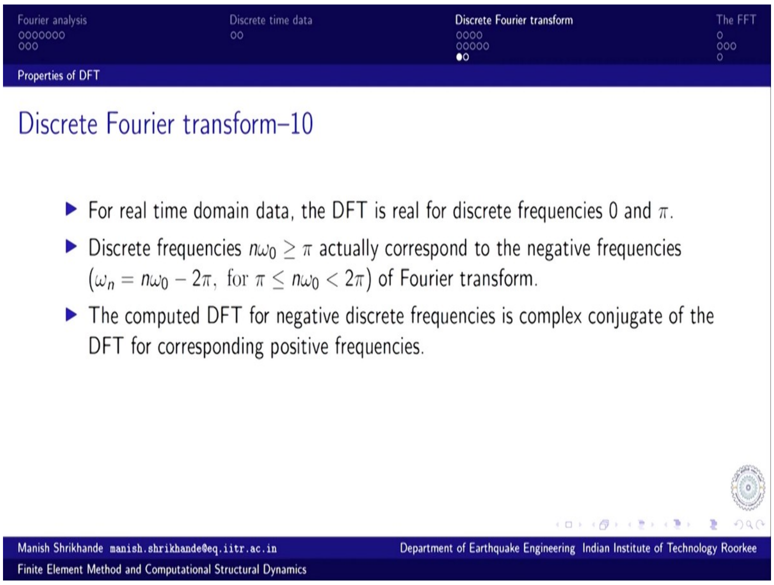
So, two things I mean several things to observe here. The discrete frequency 0 the Fourier transform is real. I mean of course, we are dealing with real data here time domain data is real. So, for 0 frequency the discrete Fourier transform is a real number, it is not complex number.

Generally the Fourier transforms are complex, but only for frequencies discrete frequency 0 and for discrete frequencies that would be π so this k = 8 or 9th sample. So, this is π . So, again for π , it is a real number the imaginary part is 0. And beyond this we see that the Fourier transforms they are complex conjugate of the real number.

So, from here the negative frequencies start, after the discrete frequency π the higher frequency is components they are actually the negative frequencies corresponding negative frequencies. So, we can just substitute subtract 2π from this and to get the corresponding negative discrete frequency.

And we see that the Fourier transform is a complex conjugate of what we have in case of positive frequencies. So, as expected the Fourier transform for positive frequency would be complex conjugate of the corresponding Fourier transform for the negative component negative compliment of that frequency.

(Refer Slide Time: 31:49)



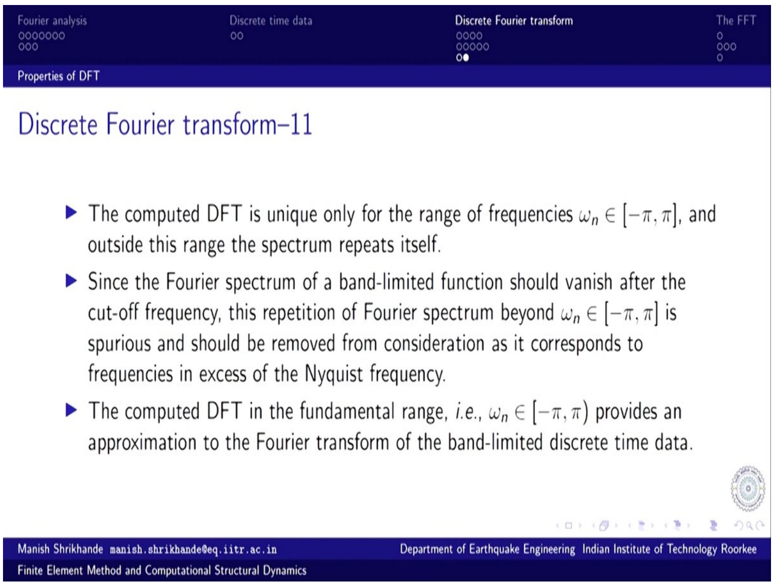
So, couple of observations for real time domain data the DFT is real for discrete frequencies 0 and π. Discrete frequencies $n\omega_0$ greater than or equal to π they actually correspond to the negative frequencies. $\omega_n = n\omega_0 - 2\pi$; for π less than equal to $n\omega_0$ and that will be less than 2π of the Fourier transform. So, from π onwards it becomes negative frequencies.

The computed DFT for negative frequencies negative discrete frequencies is complex conjugate of the DFT for corresponding positive frequencies. So, perfectly fine, consistent with the definition of Fourier transform. Therefore, now comes the important observation; if half of the component half of the representation is merely complex conjugate of the first half then what is the point in storing both halves.

If I know the first half the rest of the information can be deduced and that allows us for a very interesting and very efficient data representation data structures that we can encode, we can design an algorithm and store data in such a way that the first half of n number of data points it stores the real part of the Fourier transform and other half stores the imaginary part of the Fourier transform.

There is just enough space available as required to store this data. And it will be perfectly n number of data points, time domain data and that will be that if it is real then that same set of n number of real data points that can be overwritten if required by it is Fourier transform. First half representing the real part, the other remaining part representing the imaginary part for wherever the Fourier transform components are imaginary except for discrete frequency 0 and π . So, we have these complete; I mean once we have this real and imaginary parts defined then complete Fourier transform can be computed.

(Refer Slide Time: 34:53)



So, the computed DFT, I mean again we reemphasize this point is unique only for the range of frequencies between ω_0 discrete frequencies between $-\pi$ to π , and outside this range this spectrum repeats itself.

And since the Fourier spectrum of a band limited function should vanish after the cut off frequency the Nyquist frequency, this repetition of Fourier spectrum beyond ω_n in the range $-\pi$ to π is spurious and that should be removed from consideration we should not look at any spectrum value that happens that occurs beyond outside this range principle range of $-\pi$ to π.

As it corresponds to the frequencies in excess of Nyquist frequency. If we map these discrete frequencies to circular frequency then that those higher I mean any discrete frequency range outside this $-\pi$ to π that will result in a frequency that is greater than the Nyquist frequency. And obviously, that is in error that should not happen.

So, computer DFT in the fundamental range that is ω_0 in the range of $-\pi$ to π provides an approximation to the Fourier transform of the band limited discrete time data. And that is the main objective of this. As long as we interpret these results correctly our purpose is served.

So, the one period fundamental I mean this fundamental range from $-\pi$ to π, if we look at the data transform data in this range and we look at the one period of the time domain data that provides us a very good approximation to the Fourier transform of band limited time sample data. And we can proceed with our engineering analysis subsequently.

So, this is as far as the definition of discrete Fourier transform and how it is Fourier transform, how it is implemented on a digital computer is concerned. Computation of this is entirely different ball game. We will discuss the intricacies of actual computation of discrete Fourier transform in our next lecture.

Thank you.