Good afternoon.
Last lecture I mentioned that please use my PowerPoint as your standard. There is some typo in the signal processing chapter and also same comments apply here. Please follow this side of PowerPoint when you try to understand how to derive so-called discrete Fourier transform and inverse discrete Fourier transform. In the next lecture, TA will explain homework solutions and so on. Then we talk about network and I still want to finish two more chapters, network and quality performance and put together that will form the foundation part, the part one for the textbook I intended to write. It depends on the timing and hopefully I could finish it but at least I would update what you have and fix a couple of typos and the policy presentation and so on. What you have is really the very first draft and hopefully the next draft will incorporate some feedback from my lab members and my own reading and plus your feedback if any. So far so good that we prepared to take the first examination and I really hope you spend time so you have a very good understanding of the foundation part for medical imaging and if you have a good understanding and the rest of the semester and you learn X-ray tomography, nuclear tomography, MRI, you will feel whole lot easier because we will use the Fourier transformation constantly. So that's why it's important that you need to know Fourier analysis and also discretize the Fourier transform. After all we want to digitize signals and then we perform convolution, Fourier analysis, inverse Fourier transform in 1D, 2D and
so on. So what we learned last lecture is a sampling theorem and we spend a lot of time to explain so-called sampling rate and Nyquist rate and so on. So the idea, by now I hope you will be able to stand up to tell me why you can recover continuous function. In this case let me get laser pointer. In this case continuous function f of t and this is a continuous one and then you sample the continuous function you get a digital signal like this and the sampling theorem says that you can really recover continuous function f of t from discrete version of it. So this is amazing so you'll miss a lot of information between the sampling points and then still you can recover the signal perfectly and this is not done under any condition and then we have a assumption that is this continuous function has a Fourier transform and the Fourier transform is a band limited so that is to say the Fourier spectrum is only significant within a finite interval say minus w to w anything outside of this interval is not significant okay so this is a assumption and oftentimes this works that's why we have a lot of signal processors in our daily life like in iPhone. So if you do digital sampling
then in the frequency domain what you do is a convolution okay. Convolution and it's performed between this continuous Fourier spectrum and the train of delta functions in this case the train of delta functions in Fourier space and is the Fourier transform of the sampling sampling impulses in the time domain so you have a train of delta functions and they'll perform Fourier transformation is a still a train of delta functions just the periods are not the same if you have the period delta t the sampling interval then the period in the frequency for this train of delta function will be reciprocal of delta t so capital P capital P is a period in the frequency domain you've got to make a capital P at least as wide as two times w two times w is this period and we explained the property of convolution so you have a convolution with a delta function a delta function serves us in my word serves as a copying machine you do convolution with this single delta function you basically recover the original original spectrum you get a one copy that here is another delta function
you do convolution this delta function with the original Fourier spectrum you got another copy so you have a many many delta functions in Fourier space so you have many many copying machines you have a multiple copy of a Fourier spectrum okay because I said that the P is greater than or equal to 2w so no overlap between adjacent Fourier spectrum so no overlapping and then looking at this picture and then we know the original information of the continuous function F of t is essentially kept intact so that is very important observation so if you do low-pass field and you can perform Fourier transformation of this sampled signal you got essentially all these things and then you do digital

processing and then you just keep the central copy and then you can recover the original original signal F of t because this is just identical to the original Fourier spectrum of F of t you know Fourier spectrum you perform inverse Fourier transform you get it back so this is not all to be very clear to you now okay okay so you realize that they I put a red box here okay let me make some comments I may ask you to say P it should be greater than or equal to 2w that is to avoid any aliasing so you may wonder what happened if P equal to 2w so in that case the multiple copies of a Fourier spectrum they will just touch at this point right so it doesn't matter and at that point you see the Fourier spectrum Fourier component is zero so your two point touch together really doesn't matter even not not zero the single point single point that contributed nothing to integral okay so this is something something very very obvious from your calculus but under one exception so if this point is a delta function then that will make a significant contribution only when this functional value is a finite number so single point would contribute nothing to the total integral value so looking at this situation suppose a continuous function happened to be a sinusoidal function in this case the frequency is say the frequency is 1 so when P from 0 to 1 the sinusoidal finish of one cycle so frequency is a 1 so if you double the sampling you you use necklace the frequency at P equal to 2w means just like just like the sampling frequency to be equal to twice of the maximum frequency in this case this is a band-limited signal the maximum frequency is a 1 this is a special case so if I do sampling the sampling frequency doesn't make a 2 means within one cycle within this one cycle you do sampling twice and then we say any sinusoidal components as a two unknown and then you need just the just the two samples to determine the unknown each sampling point that gave you a linear equation you have two linear equations and you solve for two unknown sinusoidal thing and that is just the amplitude the phase and then you just try to solve say if you just a sample this way see I satisfy the requirement the sampling rate is equal to twice of the maximum bandwidth now the maximum bandwidth is a 1 1 Hertz and in this case of what happened you see you constantly get a zero you cannot make a reliable inference about original signal this is a case I mentioned at this point is not just a zero it's not a finite number rather at this point is a delta function because this is a continuous function is a single component and it's a Fourier transform it's a delta at the data frequency component so when you have a two frequency components overlapped overlapped because of of this duplication at this point you overlap a two delta function and you end up with the X plus Y equal to C and you know C but you cannot solve for X and Y so that is aliasing problem so from twice like with the sampling rate you still cannot reliably infer the right signal because you just see zero then what will be the right signal you do not know so for that reason P equal to 2W is not a good choice in this case so maybe I in the next version I will just reword the condition just say P greater than 2W mathematically just like the sampling rate slightly a little bit higher than twice of the maximum bandwidth then you're just fine so with that understanding and you will you will have a better picture what's going on with the sampling and the recovery of a continuous signal so mathematical content has been summarized in on one slice shown here and you'll show a number of right lines to underline key step and then in in the previous lecture and each right line and I use one more slice to explain how you get the step so I explain why you have this why you have that why you have this a final signal interpreter so that you can reconstruct the continuous function F of t from this create a copy so this is a function sampled at n point and the sampling interval the delta t is a 1 over capital P so you like the K just keep it increasing and then you have a number of sample discrete samples then you can recover the original function so I'm not going to explain all these things line by line but just review the previous PPT file and then you will get a good understanding that's a very nice to know the derivation so if you know this and you will will agree and from this create this create a version of continuous function you can recover the original continuous function this is an elegant conclusion

and there is a foundation of a digital signal processing and at this point everything is a mathematics or heuristic idea so in the homework I asked you to recover recover this create a signal and then using using this a formula and

then
you can construct the many many signals and there's two to show how to apply this formula and here this is a link and the details can be find from the link and it just gave you some hint and a quite closely related to your homework and you have a real signal that's the right signal so you can see the right trace this is real signal then you do sampling sampling at a blue location so blue this is a blue line this is a blue line so you do this create a sample so on my hand the right signal is a continuous okay a lot of data data point on the other hand of the blue sample that's a discrete locations you have a limited number of point the magic is that using the sampling serum from these blue data and then you do interpolation you can perfectly recover the original right signal so you see sample the data point and after send no interpolation using the formula I explained from a blue you can recover the black black and the right they superimpose together because the satisfy the condition greater than twice maximum bandwidth okay then you can use a formula as a theory predicted everything play out perfectly and in a band-limited signal you can do this trick again and again these different samples so you play around so you have hands-on feeling by the way some student asked what do you mean by by analytically computed it's just the Fourier transform another thing that doesn't mean you really just do derivation like I say what is this value okay one way you do digital computation and I think they I mean you just say it's equal to this one don't see so this is the integral then you have this one zero then you just analytically computed so that is what I mean so just a hope you any confusion with the homework or feel free email me or TA we reply almost immediately at least within one day so this is a good case when the sampling rate is densely is a high enough and this is a better case so that means that the maximum band bandwidth is a given but a sampling rate is not twice as a dance as a maximum frequency indicates so in this case the true right signal and then you sample the sample the blue and the blue signal data and then you interpolated the black signal they do not overlap although they do not overlap at a sampling location the right curve and the and the blue and the black they all agree at a limited point but I go away the between that's a wire that this currency happens and this is not a surprising because you have an alien problem okay so just that just the review we see you and a few more comments so you know what's going on and again let's look at it this is a big picture two parts in this big picture this part is to discretize signal so you've got a digital signal and then we feel this is a okay way and then you do this no information loss and after this discretization you can put data into computer this is all our purpose that we want to use the digital computer but the
story hasn't been finished yet and you have a discretized signal but you still have a continuous for a spectrum you cannot put a continuous function into computer that's a problem and how we deal with that and then we need to discretize the spectrum and this is the central topic of this lecture and the what we did here multiplication that's a convolution okay multiplication you serve the purpose of discretization here you have continuous function we just do the same trigger again so this idea the same trigger you do just that this reader Jason and the base a green train of Delta function you do multiplication here you've got a discretized function and here in the time domain in a symmetric fashion and you will have a green train of Delta function then you do do convolution in time domain and it's a green Delta function it's a copying machine just a copy many many make many copies of this original this created signal profile okay and then in the time domain you have a time period capital T so you just copy signals and every period T and you you just got the same copy so heuristically or visually you see and after this a green sampling process initiated in the frequency domain now you have a discretized signal in the frequency domain as well when you do discretization in Fourier space and that the time space is still discretized the signal so this is a criticism remains but what's changed that in correspondence to this green sampling and it is the duplication so now you have time signal and the frequency signal both discretized this is our purpose but you cannot just simply do that and to fit into our mathematical model so you have periodic function in both time and Fourier space okay this is a big picture

this is so important and that could be confusing if you take a lightly so
signal sampling you see this you have a continuous signal you have a train of
impulse functions so like this you just have multiple Delta function I did it
together and you do sampling okay after you do sampling then this is train of
delta function is a modulated by this continuous function and I call it f of
t and here just another way another worsen the F of X is really the same
thing so it's just that say you'll have a sample the signal and that is really
the envelope of the original function original function applied to weight this
train of delta functions and a point wise so you got this signal sampling and
the sampling in the time or spatial domain is equivalent to to convolution
in Fourier space so this is a one copy of Fourier spectrum you have your max
that's a maximum frequency or bandwidth and here it's nice and this value is not
delta here so you don't have problem in this case you can say if the sampling
frequency is a greater or equal to the maximum frequency here then you wouldn't
have all right okay so this is just the same thing I mentioned it to you if you
do not if you do not satisfy this sampling rate requirement or you do not
admit to the necklace the reader requirement what will happen you see here
you have overlapping things this overlapping will make the hyper you
can say components of both and together then you will not be able to tell what's
going on so this is just a problem and if you don't have this problem you can
use this rectangular function to recover see rectangular gate function covered
these central frequency components get a single copy out of out of the the train
of Fourier spectrum just like a DNA DNA testing you got to just one drop of
blood you have all the information is same thing but if the sampling rate is
not high enough you have overlapping so this is overlapping problem so the
all aliasing problem as I highlighted here is the reason is the area where
you couldn't recover the signal components so let me explain further you
couldn't uniquely recover what's the Fourier component and this part is okay
but this is overlapping reason and the overlapping reason you see this okay
in overlapping reason you know the sum of the point but you do not know you do
not know what's the real value one case is that say at this point the blue plus
right and then let it give you value give you value this is the say this value
called a or you call it a one okay so that gave you the value one and then
likewise so you just got to the problem here the value really really you add it
together you got the value one but if you see if the blue is goes in this curved
way
and the right goes in symmetric to data you added it together so you got the
same value
so here really this is our original system okay the original system and this is
an
intersection point and to show the idea clearly I make a new code in it you see
all these values
blue and the right I did it together always give you same level here the value
is added together
if you think of this part is a positive is negative it will give you zero if you
go back
here is a this is right a part a left a part I did it together will equal to one
so that is
just the picture and then you couldn't uniquely recover the Fourier spectrum one
aliasing is a
problem so if you do not have the aliasing problem so things separate well so
that's not
the issue so we assume from now on the aliasing is not a problem your sampling
rate are always always high enough so it's just that our assumption a practical
sampling
condition always adjusted to be right so in this case that we can talk about how
we we can this
discretize Fourier spectrum okay so this is a Fourier spectrum and then we make
sure the
sampling interval here is a is a high enough so here you got a separation I
mentioned now the
next step and the central task for this central task for for this lecture is how
do you discretize
this continuous periodic Fourier function so we use this green train of delta

function and the
same argument we need to make sure the sampling interval here and there is a density enough so
that the the discretize the signals wouldn't it be wouldn't it be our life after you do
multiplication in frequency domain here you do multiplication in for in Fourier domain that is
equivalent to convolution in time domain so this is a green train in Fourier domain this is a
green train in time domain so they are linked together with a Fourier transformation and now
the delta u equal to 1 over capital T this one over capital T and really you should make sure
same same comment and you need to make sure this is delta T and this capital T is at least as wide
as the period of this discretize the function and then maybe slightly larger all depends on this
extreme functional amplitude you don't want the extreme point to be delta function anyway it is
delta function you go a little bit beyond it so that the and the point of wouldn't overlap in
reason in locations like like these drawings so same idea so if you make sure your frequency
sampling is dense enough then the time domain signal will be converted from a single discrete
copy into multiple copy and then no overlap because we again satisfy this minimum sampling
read requirement this is just a graphical picture so graphical explanation so always remember the
picture and then now we can just the tell you the same idea in a different way so you'll have a
continuous function you'll have a continuous Fourier transform so this is continuous okay
continuous function continuous Fourier transform so in blue domain in right domain right domain
is a Fourier domain for example then you can generate a discretize the function G of T and
also in the time domain the discretize the function has a counterpart in frequency domain
is a capital G of it's not here so to call you so I need to fix it okay it's a frequency domain
and I explained it to you and when when the Nyquist sampling rate is it's satisfied no
information loss from continuous domain to sample the domain so this is equivalent so no information
loss of here either so you perform a Fourier transform no information loss so this is equivalent
to the continuous Fourier spectrum is equivalent to continuous signal in time domain this by
sampling theorem is a equivalent to sample the copy of discrete signal then this discrete signal
can be Fourier transformed this is for equivalent because the Fourier transform is invertible
therefore the continuous Fourier transform and the discrete Fourier transform on high
level should be equivalent and the Fourier space variable should be U and here is a T
so just show this overall relationship then we can comfortably perform digital signal analysis
either in spatial temporal domain or we perform this create a Fourier analysis in the in the
frequency domain so this is just the foundation we can do digital signal processing but what we
do in the digital world is essentially equivalent to what we do in the

continuous domain but our
computer is digital computer is not continuous analog computer but we have the theory as I argued
before so we can still perform the job digitally but which is equivalent to continuous signal
processing so this is a big picture now let me make a few more comments okay so far we have
been quite a general to visualize the process of basically tagging the Fourier spectrum now
let's get a more more specific so we say how many sampling point are we talking about suppose you
have a continuous function f of t then we have a continuous Fourier spectrum I have a height of
U this Fourier variable is U so remember this is U so I need a fixed type of on the previous
slide how many sample you have that depends on really depend on two things and how densely you
sample the continuous function so this is delta t is important thing so delta t is equal to here
delta t is equal to 1 over capital P so this is your delta t and the smaller delta t the more
data you will have okay and how many data point you will you will do those sampling also depends
on the period of the signal how long you recorded the continuous signal the longer signal and the
number n the number of n number of data will be proportionately longer so this is a simple
relationship so t is total length divided by delta t which is a 1 over P that is a total number
of signal you would sample so you do simple algebra so the n equal to T capital T times P okay this
is what how many data you have in the time domain and then now we say time domain sampling is the
first part frequency sampling here is second part and the similar question can be asked how
many data point are you going to get when you sample Fourier spectrum or one period okay in
frequency okay or this period total length is a P because P is a capital P is a period in the
frequency domain okay so this is a capital P so also how many data point capital M you will have
in the frequency domain after discretizing the spectrum and that depends on also the sampling
interval in frequency domain that's a delta u delta u similarly should be equal to 1 over capital T
here is a 1 over capital P here is 1 over capital T so the number of data point in Fourier space
over one period should be should be capital P is total length divided by delta u delta u should
be 1 over capital T you got this one so you do simple algebra again it's P times T okay so we
see that capital M which is a number of data in Fourier space is equal to number of data in
time space because you see this relationship and this relationship they are essentially the same
they are symmetric but the total number of data point in either space is a capital P times the
capital P or you say capital P times the capital T doesn't matter so this is a big picture how
many point we would like to to have so in time domain you have a TP data point in frequency
domain you have TP data point that's the same so if you take a reciprocal for the first line so
you have 1 over capital N then here you have 1 over capital P 1 over capital T

what is 1 over
capital P 1 over capital P is nothing but a delta T that's a sampling step size in the time domain
and the 1 over capital T is a sampling step in the frequency domain so this is a very nice
relationship this 1 over N if you fix the number of N then you can get a finer sampling step in
time domain but at the cost of a larger sampling step in frequency domain this is something fixed
so this is a relationship like what I mentioned to your duality you have something narrow in
white space you must have the counterpart wider in the other space okay so now you just remember
remember this key relationship we will make some use later on in applications of Fourier
discrete Fourier transform now we try to make a transition from a continuous Fourier transform
to discrete Fourier transform and then we first perform a direct Fourier transform of a sampled
digital signal and therefore digital signal after sampling is no longer continuous function rather
is a continuous function times a train of delta function with a period of 1 over capital P which
is a delta T that's a period of delta function in the time space so this is a multiplication so as
sub 1 over P of T is just the train of delta function and the period or adjacent delta function
are separated by amount of delta T equal to 1 over capital P so you got this this is a creative
version of continuous signal if you write it out you really just sample the signal at a time t0 t1
t2 you all together you sample capital N data point so we make indexes go from 0 to capital
N minus 1 so this is tn equal to n and that's the index that coefficient is a 1 over capital P
n equal to 0 1 capital N minus 1 so just all these discretized samples mathematically you
can say you sample the data point I put it just three dots doesn't mean these are samples I think
that's just a look heuristic okay you agree with me that's nice so you just have this multiplication
and then you only have those discrete numbers that we utilize delta function so this sample
the result is summation of a bunch of delta functions at a discrete time the point this
will be the type the point is shown as an over capital P and a weighted by functional value at
that point so this is really we have a train of delta function then we use a continuous function
to modulate that train of delta function so we got this one okay got this one and this is a function
time function any any function of time we can perform a Fourier analysis so we got a Fourier
spectrum so we just perform Fourier analysis is nothing more just directly use a Fourier transform
Fourier transform will we will have a multiple delta functions and the delta function is a nice
only just save the number there so the Fourier kernel stay here then you got a weighted
coefficient here so direct the Fourier transform you see there's a height on top of digital version
of the original continuous function f of t and then you got a Fourier transformation shown here
and this is a Fourier transformation you directly perform the over sample the

signal as you modeled
with the delta function let me make some comments on this this create a Fourier transform I say
this this create a Fourier transform actually is a periodic okay this is periodic why is it
periodic remember the Fourier series this is the formula I copied and what I just showed you this
digital version for a transform is equal to something is something coefficient is really
the functional value sample the functional value here I just a switch variable t with u so is u
in this case it's just a periodic function in time domain but what do we show you here is in
Fourier domain but a functional form is a very similar you see 2 pi 2 pi and you have an is t
over capital T is u over capital P the same thing so this is I mean this right part is a continuous
function in the Fourier space and it is a periodic continuous function it's a periodic because each
sinusoidal component the sinusoidal components by divination is it's a it's a periodic so you see
the basic frequency is a double the frequency triple the frequency you can have many but in
this case we do not have many we just have a n frequency it doesn't matter it's still periodic
even just you have one one component is periodic because sine cosine all periodic so this is
periodic is not surprising because you remember the big picture you see the periodic Fourier
spectrum okay so this is a one argument you are with me then we just say original original
function I have of t has it's a Fourier spectrum and I have had you it's just a statement for a
transform that how about we just sample this Fourier transform in Fourier domain and they
using a train of delta function again in Fourier space so this is a sample the worsen okay sample
the worsen really I should have put a three dots here three dots it mean it's a sample the worsen
okay the sample the worsen will give you a list of Fourier spectrum components this is height u
u0 u1 u capital and minus 1 so you know this is a book writing need the multiple it reasons what
do you have is first the worsen many small things and here I really need to put three dots here to
show that this is a sample the Fourier spectrum and the sampling point will be at u0 u1 so will
be um m is a sampling place and then the interval in the sampling interval is a 1 over capital T
capital T is a period in time domain so you have um equal to m divided by capital T m equal to 0
or 1 and again the same capital n minus 1 because the number of point in frequency space and the
number of point in time space are the same I argued the case and so we have a same capital
N so you do this sampling this sampling and that can be written in this this form so this is just
the next two lines is so you how you represent the sample the Fourier spectrum at position m and
all at a position u m I'm keep changing from 0 to capital N minus 1 you know this is a Fourier
transform this is copied from the previous slide this is the this one so I copy this to here okay
and we know 1 over capital N equal to 1 over capital P times 1 over capital T

okay we know
this here you have an n over P and here you have this you you just become um um equal to m over
capital T so this u is sweet it's changed up to um so this in this u place you have
m divided by T so on top you have m and n this m from here this n from here then you
got you got a P here by this variable substitution you have a capital P also in the denominator so
you have both T capital T and the capital P they come together give you capital N so you have a
capital N here so this is a form and we like you see you have a very nice symmetric thing e to the
power minus j 2 pi so that's a that's a common then you have an either I'm for frequency sampling
and for time domain sampling both normalized by number of samples a capital N this is just
a sampling point now here we say based on these few slices we say this create a Fourier transform
is a well motivated it ought to be defined something like this so this part just see
and move a copy to here so this is the discretize the Fourier transform so this is a Fourier
transform this is respect to sample the sample the time domain signal at N capital N point
indexed by small n so you got a sample the signal the sample the signal you you you just
do multiplication this is a Fourier kernel and for each time and then you do inner product for
small n from 0 to capital N minus 1 so this is summation inner product it seems in like the
continuous Fourier transform this is just the discrete version you perform this created inner
product this is the basic rate of Fourier transform I would like to to introduce it so we know this
ought to be a reasonable form the right the question is how to write it neatly and how do
you perform inverse discrete Fourier transform that will be covered in the next part so I gave
you ten minutes so I drink some water see you later
you
you
you
I have a question.
My question is for tonight's work, but I want to first do the negative and some of the negative to me.
In the homework assignment, are you sure you have this negative?
Yeah, I thought on the PowerPoint page.
Negative to me. I'm pretty sure.
Let me double check. I think it's okay. It's still solvable. It's okay. It's just a problem.
You know the definition of yield?
Yeah, yep. Step function.
Yeah, yeah, yeah, yeah. So then with this yield, it becomes, yeah, it's just a way to deal with it.
Then you just compute this one. It looks right. I didn't check every detail.
Something you would just get half of it. Otherwise, it wouldn't convert. So this B is equal to greater than...
B is positive.
B is positive. So when P goes infinity, the bigger it gets, the smaller it gets. So the whole thing will convert.
Something like that.
I thought since the heavy side is the negative, the integral has to be negative infinity.

So this is...
So this is right. So you just got the integral of the other part. So this just makes it convert.
So we don't use the convolution?
It's asking you to compute Fourier transforms. So it's just an integral that you know the way to remember the transform.
I think this is untracked. I wouldn't check every step. It just looks right. Just make sure you do it right.
Okay.
Several students show me your homework for integral. My best feedback, I just say they look untracked because you do integral by part or substitution.
I wouldn't check. I do not remember exact steps. And not good for me to really fix your small errors.
As long as it is untracked, it's up to you. Make sure derivation will give the same results as your TA has.
So if you didn't do it right, then it may reduce some points. But I'm not here to give you an exact answer.
So just be extra careful so you get the right answer. And all of you show me you're doing the right thing.
Untracked.
Okay.
Okay.
Okay.
Okay. So we pick up a discrete Fourier transform and finish the inversion part. Let me first say a few comments about your learning. This is the foundational part. So I really encourage you to follow the logic step by step so you get a sense how you have the sampling theorem and the inverse Fourier transform, discrete transform.
So you follow the logic step by step. That is really your best homework. If you just read lightly, you wouldn't see what's going on.
After all, this course is kind of more quantitative. A lot of logic steps involved. So you need to spend extra time so you have a better understanding.
So now let's continue. Discrete Fourier transform, and I just decide this form looks very reasonable. So give you a list of data. So sample the functional values.
Then you perform Fourier analysis, not for continuous function, but for this list of discrete data, how you perform Fourier analysis. It ought to be done this way, okay, as I explained before.
So it's just a change in notation, and then we feel this kind of notation kind of messy. So we can just make the transform variables look neater.
We say when we deal with sampling and we have a data point, t0, t1, you have frequency components, u0, u1, and so on. The most important thing is really the integer index.
So you can just say, not call it t0, how about we just call it 0, because this is just to make things easier. In that case, this functional value is not f of t0, t1, just call it f of 1.
And this is an integer index, so I put it into a bracket, just to make it clear. And this is a functional form for integer. It's not for continuous counterpart t1.
Likewise, this lowercase i for bracket capital N minus 1 is nothing but f of t sub capital N minus 1, the same thing.
So if you do this, and then you better do in a consistent way, in Fourier space. So f hat, that's Fourier transform, and then bracket 0, that is f hat u0.
Likewise, f hat capital N minus 1 is equal to f hat u sub capital N minus 1. So this is neater. Then the Fourier transform we already motivated can be put in this form.
The Fourier component for the m's component is a summation of fn, and fn is just the discretized signal, sampled copy, and weighted by this kernel.
This kernel for fixed m, the n would go from 0, n equal to 0, this becomes a constant. n equal to 1 will be the basic frequency component, and all the way up to n minus 1.
So you've got this sinusoidal component from low frequency to high frequency, and weighted with the sampled signal.
And this is inner product form. It's still inner product form. I mentioned

multiple times. So you have functional value and the Fourier component.
For each given Fourier component, you have the sinusoidal waveform, waveform at that frequency, messed up with the functional value.
So you compute all the parts of product, add it together. This is the idea of inner product. You do inner product, you get the coefficient. That's the Fourier spectrum.
So the same thing I explained to you, but now in the discrete case. So you have this Fourier, discrete forward Fourier transform, or simply put, discrete Fourier transform, defined this way.
It looks neat. So you have Fourier components and the discretized function. Then you have this sinusoidal kernel, only you take a discretized value, indexed by m for frequency component, and n for time domain sampling process.
So you've got this one. And this can be really put into a matrix form. So here is nice to think it in the matrix form. And you know you want to find m equal to 0, equal to 1, equal to capital N minus 1.
You can put these capital N values into a vector. This is put here. So you just visualize it. You have f hat 0 on top, then f hat 1, then last one is f hat capital N minus 1.
So you have n frequency components on the left-hand side. And on the right-hand side, this is inner product. So input is always sampled signal.
So sampled signal, again, on this side, you can think the first one is f. Now this is time domain, so you know hat involved here. So f0 on top, then f1 next. The last one is f capital N minus 1. So you have this vector. And this vector needs to be multiplied with capital N by capital N matrix here.
I just put a bunch of e here. Each e is corresponding to this one. And they are not the same. So just different entries determined by mn. The small mn is indexed.
So for each one, say for m hat 0 is m hat 0. So m will be equal to 0. This is equal to 0. Then all these line elements will be the first line anyone can have idea.
So what will be the value for the first row of this square matrix? So m equal to 0 here is equal to 0. So everything is 0. e to the power 0 is equal to 1.
So this line is just 1. Then the first row of the square matrix is 1. You have capital N data point. You do inner product. So this just gives you summation added together.
So you have this f hat 0 just added together all these discretized values. So this is the integer index results. So we say here, say when you do this Fourier transformation, I put a parenthesis up to a scaling factor.
So they have this relationship. But you have a very dense sampling rate. And you see when you do this Fourier transformation, this first component f hat 0 will be larger if you use many, many points.
So the scaling factor will be involved. And we will see that later. But here we just add it together. And this summation kind of like a numerical computation of an integral. Integral is a Fourier transform.
But you didn't put a delta t into account. And we will explain that later. But anyway, so this is a matrix form corresponding to this definition of discrete Fourier transform when we use the integer index.
So you have this one. So now we ask the next very logical question. You have a forward Fourier transform. Then what is the inverse Fourier transform? So this comes to the next slide.
So inverse Fourier transform, you see, the forward is a matrix multiplication. And you go from f to f hat. This is a forward process. You go from f to f hat. So f is sampled data. And f hat is sampled spectrum. The inverse, you should go the other way around, from f hat to f. This is a matrix expression. So naturally you should just go back from f hat, go back to f.
You just use this inverse matrix. It's just as simple as linear algebra. And indeed, this is true. So you can just solve the inverse Fourier transform based on your definition of forward Fourier transform.
And then you do this inversion, you get this relationship. Then you got this 1 over n. The first time I mentioned to you, when you compute f hat 0, you add all the signals together. You didn't do averaging.
So averaging is taken care of now when you do inversion. After row, you do Fourier transform of one function. Then you can go back to the original and perform inverse Fourier transform.

And the weighting factor, you can put on both sides. You can put on one side only. It depends on your definition. It doesn't matter because it's just a pair of Fourier transform.
So here we put 1 over capital N when we perform inverse Fourier transform. And if you like, you want to have a symmetric feeling, you can put 1 over square root capital N.
Then in that case here, you have square root capital N. So you got a perfectly symmetric arrangement. Like in the case of continuous Fourier transform, some definition you have in front of the integral.
You have 1 over 2 pi something. You have 1 over square root 2 pi something. If you have square root thing, the forward and the inverse transform, you have a symmetric thing. Otherwise, you have the coefficient in one form and don't have it in the other form.
So likewise, how you deal with 1 over capital N, this is the way we do it. And that's just fine. So you can use this relationship to go back and forth from discretized time domain function to spectrum.
Or you go from spectrum to function. Either way, both are discretized. That is an essential requirement for discrete Fourier transform and its inversion. And now we have a formula which can do exactly the same.
So now let me make some more comments on this 1 over N. This comes to this slide, about this 1 over N. And this is, you have a minus sign. So this 1 over N is the major place, different from what we have in the forward transform.
For forward transform, you don't have this. Another major difference is the sign. For inward and the transform, you have this Fourier inverse transform. And you have this forward transform and inward.
Inward, one has the minus sign, the other doesn't have the minus sign. So you need to remove the negative sign in the inverse transform. And it's very much the same as the continuous Fourier transform.
For the forward transform, you have minus sign, the continuous version. And then when you perform the inverse transform, so that minus sign is taken off. The same thing here.
So now I explain why you have this 1 over capital N. And earlier I mentioned that this 1 over capital N is equal to delta t times delta u.
Delta t is the sampling interval in time domain. The delta u is the sampling interval in the frequency domain. So with this delta t delta u, so you're really doing the discretized integral.
The integral in this case is forward Fourier transform and the inverse Fourier transform. In either way, you need to do this discretized summation. So one way is delta t, the other way is delta u.
You perform forward transform, then you go back and you perform inverse transform. So you have delta t delta u both involved. So you really need, if you really go from numerical perspective, like shown here, you need to put delta t and delta u into the forward and the inverse transform.
So if you really put them together, what you have will be N. You need to put both together so you have this capital N. That's why you go back and you have this capital N here.
This take care of discretization in the time and frequency domain. Just remember these two major discretization steps. And now with 1 over N, both steps, discretization, are taken into account.
And don't remember, for inverse transform, you need to remove this negative sign so that you can just get the forward and the inverse work together.
So this is why you have this capital 1 over capital N.
And because how you interpret discretized forward transform, discretization is a sampling process. And for each sample, that represents the local situation at a square, small rectangular strip.
At that strip, you have functional value. So I call it, say, F0, F1. So FN uses a bracket in time domain. In time domain, so you have F0, F1, F capital N minus 1.
Then you need to weight the functional value with delta t. And in the frequency domain, the inverse forward transform is a continuous integral.
But you can discretize the frequency thing. And with F height, 0, 1, until capital N minus 1, then in frequency domain, you need to weight the sample values with delta u.
So delta u, delta v, if you put forward and inverse together, then those two

delta quantities play together. And then they multiply together.
That gives you capital N. That's why, again, you have this 1 over capital N. You have this. So now it has been clear to you.
Second perspective is just from, I call it harmonics. Perspective of harmonics. So this transformation like this.
So you have forward transform, really the sampled signal is multiplied, the square matrix multiplied the sampled signal.
Then you got a Fourier spectrum. So that square matrix is really a rotation matrix. In high dimensional space, you perform a rotation.
So our original representation, the sampled signal, is transformed through Fourier transform into the Fourier component.
So Fourier basis functions just form another also normal basis. So matrix rotation. And this inverse, it's just rotated back.
So this is just a matrix perspective. And the similar idea, as long as you have this harmonic sinusoidal component for an interval or for a whole number of axes, or in multi-dimensional, or in some spherical surface.
It's just a spherical surface. You can perform Fourier analysis called harmonic analysis. The same idea. Now the domain is not 0, 1. The domain is a whole spherical thing.
And then you define sinusoidal components. And you have different frequency components all look more complicated and more interesting.
And the essential idea is still the same. You pick up anything, any order, two different harmonics. You do inner products, they will give you 0. So they are orthogonal.
If you single out any harmonics, you do inner product with itself, it will get a non-zero value. It's 1. So exactly the same idea.
So here, it's a particular also normal basis. The function is E, the basis function is E to the power I2 pi MT over capital T is N over capital T.
If I'm not the same, you got 0. That means these basis functions, they are orthogonal to each other. If you do the inner product with itself, it's equal to 1.
So that means this vector, the total length of the vector is 1, or norm of the function is 1, norm and the length is the same thing.
Then this constant, you can use in one part, you can distribute it into two parts. This is what I explained to you. It's just for convenience and to your taste, which way you want to do.
Why you have 1 over T? 1 over T is integral. You want to find the average value. You normalize it.
Why we have 1 over capital N? Because altogether, you have N data points. 1 over capital N is to do the average. It's the same thing.
To summarize what I have been explaining to you so far, this is a summary of what we have.
Again, you see different variables. For Fourier analysis, you read the papers. Different papers use different notation. That's why purposely you mix the notations.
Now this is just another notation. Essentially the same thing. You have a data point, capital N data point.
You have a data point called H. It's small, H, K. K equal to 0, 1, until it's capital N. These are data points.
You want the temperature to change. You want to perform Fourier analysis. In this case, you want to perform discrete Fourier transform.
How do you do it? You do it this way. You don't do the normalization. You save normalization here.
You do Fourier analysis. You do this Fourier analysis. You don't have a minus sign here. You put a minus sign here.
If you put a minus sign here, you put a positive sign here. It's just not the same. Either way you do it.
Once you have capital H, this is your Fourier component, and the index is small n, n equal to 0, again to capital N.
This capital N and this capital N, they are the same because in both domains you have the same number of coefficients.
Inverse transform is used to bring the Fourier components, capital H, N, back to time domain.
This time domain signal, small k is H, K, and I put a weighting factor to take

the sampling delta t, delta u into account.
Altogether. This is a summary. You can have a vector of capital N element, and you only need capital N basis functions.
Indeed, you have capital N basis functions here, indexed by these different components.
You have n components. You have n elements, n basis functions. We are talking about kind of n-dimensional space.
Then you have n harmonic orthogonal basis functions. You need at least n, and you only need n. That's just enough.
You do the transformation from small H to capital H. You do the n-dimensional space rotation. You get a new representation.
That is called Fourier representation, and you can rotate it back. Just recover the original signal.
So you can go back and forth. Forward and inverse transform are symmetric or nearly symmetric.
Because you see the same function, you just have one sign change, and it's just up to a scaling factor.
The scaling factor, as I said, can be distributed to both transformers, so they look symmetric.
But this minus sign, you cannot do that. So one transform with minus sign. The other, you don't.
Just show the geometrical factor. You rotate one direction, then you need to rotate back.
Rotate back means minus, so you cannot rotate, rotate again. That's not the case.
So you need to have the minus sign to show you do rotation, then you do counterclockwise rotation.
Just move forward. You need to come back. Back means just negative. That's just the idea.
Again, from this matrix multiplication, you can understand if you perform Fourier transformation,
how many computations you need to do. One computation means one addition or one multiplication.
You see, for the first line, you have n data points. You have n elements in the first row of the matrix.
You do inner product. You need to perform n times multiplication because you do this matching.
Then you need to do n times addition. And if you are picky, you say not n times, n minus 1.
n is usually very big, so n minus 1, I don't care. Just say n addition, n multiplication.
So you do this, just say you do this capital N times. For each times, you need to do n multiplication.
So the computational complexity is in the order of capital N squared. This is the order of the computation.
Looks like if you want to perform either forward transformation or you want to perform inverse transformation,
the computation involved is in the order of capital N squared by definition.
When we do signal processing, particularly in early years, n is big and n squared computation is very time consuming.
For example, when I was in primary school, 2,000 data point analysis of seismic trace analysis took one day, over one day.
And if you use some smartly designed algorithm called fast forward transform, FFT,
the same task on the same machine took only three seconds, less than three seconds.
That's a big difference when you do signal processing. So you just write airplane or some measure.
And you need to automatically detect, decide you need to do signal processing.
So real time performance is important. And nowadays, machine learning will involve huge amount of computation.
So high efficiency algorithm is very important. That's why we have this FFT algorithm still plays an instrumental role in real time signal processing.
Like why data map PhD thesis, I use FFT to do CT imaging. Basically, I use that

to do convolution.
If you do convolution, like you do Fourier analysis, and then the order of magnitude computational complexity is n squared.
But if I use Fourier analysis and the computational efficiency, so one way is n squared.
Use FFT, it becomes order of log capital N times n.
So this is a big saving. So from n squared to n times log n. So with FFT, you can save a lot of time,
because log n greatly reduces the magnitude of n. So just like if your n equal to 1,000,
and the log n will be about three or something like that. So the larger and the more saving you have.
And you can use a faster algorithm to do Fourier transform, because the inverse transform, as I mentioned, is very similar to forward transform.
So similar algorithm can be designed to do fast inverse Fourier transform. They call it inverse FFT.
So these two are standard MATLAB program you can use. If you implement Fourier transform according to mathematical definition,
you need to do summation for each k. The summation is just the inner product that involves n times, n multiplications, n additions.
And you need to do this n times, that's n by n. But if you use fast Fourier transform, it's a subroutine.
I don't have time to explain how fast Fourier transform works, but you just treat it as your iPhone.
You press a button, and it will call your friend. So you know there is a black box called fast Fourier transform,
and the inverse fast Fourier transform. You just trust it, just like you trust your iPhone.
So now let me give you some examples of why we bother to learn this created Fourier transform.
How it can be used to perform a convolution and to estimate a spectral.
Just two typical applications. With Fourier analysis, as you can understand, you can do signal processing, image processing,
remove noise, detect contours, and many, many things. It's very important.
Let's start with the first example. How do you do a convolution with fast Fourier transform?
You can do so with conventional method. And you did before in the classroom.
I showed you how you do the hands-on example. So you flip one, you're matching up, and the multiplication added together.
You sift a little bit, do the same trick again, again, again. After you sifting, you have an element.
You do an element for one vector, an element for the other vector.
So the total length of the convolution will be n plus m minus 1. That means you need to do so many siftings.
So here, again, it's a very simple example. Example one, you can use MATLAB to do convolution in two ways.
One way is just a direct implementation. You did before. So this is one vector, this is another vector.
You can move them together, so you got a result. X times Y is the first way to compute it, called a convolution, X, Y, Y.
And the second way is indirect way. Indirect way, Y, we go indirect way. Pretty much reason. Indirect way, we save computational cost.
There are many reasons. This is one of the reasons. If you do time domain, straightforward convolution,
the time you take is proportional to capital N squared. But if you do not do the direct way, you do convolution in forest space.
You can save a tremendous amount of time because the time complexity becomes capital N times log capital N.
And this is the second way. And it's not that hard. The MATLAB will program, just have a high-level picture.
And all the basic work has been done by those software engineers. So you have this vector X.
This discretizes the signal. Or they sample that the temperature is 1, 2, 3, 4 degrees and drop to zero.

You can do ice scaling. So this is one example. And this is another example.
For some reason, you want to use this filter to analyze this temperature profile.
What you do instead of directly convolve them together, you perform a Fourier analysis. And yet you use a faster algorithm.
So what results by this factor is a Fourier spectrum returned by FFT, this functional name in MATLAB.
Likewise, you do FFT for this Y filter. So you've got a Fourier spectrum.
So the time domain representations on the first line become frequency representations as FFTX and FFTY.
And then you do convolution in time domain. While you do in frequency domain, you do multiplication.
So this is a multiplication you do in frequency domain. After that, you need to do inverse FFT.
Then you get convolution result. Why convolution result between the first method and the second method should be the same?
Because of the convolution theorem. So you can do convolution in both ways.
And indeed, you just display the result using direct method. This is the result.
Then you display the result obtained using FFT method. You got the result. You look at it, they are the same.
So they just use Fourier transform to do discrete convolution. But there are some tricky things.
When you involve Fourier transform, you do discretized signal processing.
Mathematically, we modeled the whole process, as I explained on the big picture slides.
And then remember, in time and the frequency domain, both are periodic functions.
So when you do discrete Fourier transform, discrete Fourier convolution.
So the convolution theorem talks about circular convolution. So you have a discrete version of Fourier theorem.
Let me explain what I mean. So you got a second example. So you got this one, you got this one.
And you do first way, you do second way. And the first way, you got a result.
Second way, you got a result. They are no longer the same. Why not the same? So look at this.
So this is what I mentioned. You are dealing with discretized signal.
And really, according to your modeling, it's not a single copy. You have really multiple copies.
So this is time domain of one function. Say X looks like this. The other function, Y, looks like this.
So you do the matching, then you find the summation. And for single copy, that's all right.
But if you do saved, you do it again. You see, when you do saved, you got some problem.
Like this way, you do matching. So you match, you look at this is right match, right match. Everything fine.
But here, some part from neighbors into your yard. So you do matching, the neighbor really mess things up.
So this is a convolution. This matching, multiplication, summation involves your neighbors.
So this is really circular convolution. The matching up kind of this part of your neighbor.
And it's not something on fire. And you see, the same part of yourself, you invade your left neighbor.
So what your right neighbor did for you, you already did for your left neighbor.
So this part is really this part. So kind of you just circularly put this single copy around.
So you keep seeing the blue copy. Move out this way, then move in the other way.
So this is called circular convolution. So if you do circular convolution, your neighbor will bother your result.
Mess up your result. Unless you do so-called zero padding. So if you do zero padding, you add enough zero.
So make your infringement area, just make your yard wide enough for your neighbor far away.

The way is to say add many, many zeros. I purposely add more zeros. Only five data points, so I add more zeros.
And for both way, then no overlap. This is to say, see here, I add zeros, make my period artificially long.
I add all the zeros. I do convolution, and they still move away and towards left or right.
And the circular convolution wouldn't work because you just see if you do, say, when you do this,
and the right and the blue copy, only single copy match together. So this is the result.
The circular convolution will be the same as you do single copy right, single copy blue, you do convolution.
That will be the same. And looking at the example here, you get a right result because you add enough zeros.
And the earlier example, you didn't add enough zeros. You mess up.
The first example, you get a pretty good result because you have zeros already here.
And the number of zeros, just many enough, so no mess up.
So this is the zero padding effect. How do you deal with your neighbor?
And you can read more. And there's some very good tutorial examples on MathWorks website.
Just feel free to read. And the second example, called the spectral analysis,
I mentioned, given capital N, delta t and delta v, I use delta u as a frequency sampling interval, is fixed.
And if the original signal has some frequency components and the very reads cannot be captured with discrete sampling rate,
you need to reduce the frequency sampling step. But you fix this delta t, fix the number of N,
you cannot reduce this because of this relationship. So one way is to increase the capital N by zero padding.
You add more zeros, then in that case, you can get a fine spectral sampling interval, get a better spectral resolution.
So this is an example. And really, if you do not fully understand, click this link, you will see MATLAB's tutorial page.
So this is, let me say this, you generate the original function f of x with two frequency components.
One is 100 Hz. The other, something weird, is 102.5. So this is something weird.
So use the sampling interval, not dense enough, and you can recover the first component at 100, because it happens to fit your sampling rate.
The other one, you don't have good spectral resolution. So this is the amplitude you should have. So you got half of it.
Then you add zeros, you add more zeros to the time domain signal. You enlarge the time domain signal by adding many zeros.
As a result, the sampling interval in the frequency domain becomes smaller. So because of smaller, when you estimate,
when you perform a Fourier transform, it gives you a Fourier frequency component. So you got a correct estimation.
So at a frequency around 200, you got good results. So this zero padding for spectral resolution, spectral analysis, is very important, but a little tricky.
So I suggest you read after class. And one thing I want to comment before we finish about this magic thing.
And we say that this is a big picture. And everything looks perfect, but one approximation, and I mention it here.
So we say this is a finite support, a limited line signal. And then we say the Fourier spectrum is also just limited to a finite interval.
But actually, when this is finite in time domain, this cannot be finite. So there must be some infinite long spectrum, just while you get smaller and smaller.
And you could argue when the interval minus w, w, just larger enough, and the other side of this interval, the value will be very small.
So you can safely think this small enough, this right spot wouldn't be an issue.
But there are some mathematical rigor involved here.
It's not that easy to say it's getting smaller and smaller, and then added together will be nothing.

I think the neighbor here, this neighbor involved, gave some contribution. Even this copy, because this copy has infinite support, will contribute a little bit to here.
And the other copies on the right-hand side all contribute to this particular spot, because you have an infinitely many copies.
Just a given copy, the further away from this point of interest, they will contribute less.
But you are adding things smaller, smaller, smaller. But all these things adding together may not be a smaller thing.
Let me just give you an example. So you see this spectrum, it just decays as 1 over x.
X is a frequency component. You move away, it gets smaller and smaller.
Okay, you add the smaller components together, no matter how much smaller. So you have 1 over 1 million, plus 1 over 2 million, plus 1 over 3 million.
You add all these small things, starting from 1 over 1 million. It's a very small number. These things added together is infinity.
So that's something, no matter where you start, you put 1 or you put 1 million here, you just add these small things together.
It can be a very huge thing. So this involves some exponential growth issue.
And like this chess board, 8 by 8, you put 1, 2, 4, you just do this doubling for every grid.
And the result is really a huge number. These rays will cover the whole earth, very thick. This is exponential growth.
So for the argument to work, for this argument to work, for this is really negligible, you really need to make this converging rate faster than 1 over x.
So that means you need to make your function smooth enough. The smoother the function, it decays faster.
And usually, luckily, fortunately, the most practical function after pre-conditioning, after you're smoothing out, it decays greater than this, maybe 1 over x squared.
In that case, this small part added together is still small. Then the whole trick will work.
And so far, the Fourier analysis part is done. So I basically referenced multiple resources, plus some of my own understandings.
And this is a very good textbook, and the instructor allows me to use this textbook for previous lectures.
But this is for one semester class, a very thick book. And what I did, it really comprises the essential things.
I had other points, other ideas, examples from different resources, including my own understanding, and make just a BME version of linear system and Fourier analysis.
So far, you have all of them. Then the next lecture, we talk about MetaLive TA will explain some homework questions and just re-enrich your knowledge and the scale.
And then the next one, I talk about network image quality. So Fourier analysis, the most difficult part is finished today.
And what is your homework today? Have you ever heard of RPI brand name ARTX?
RPI tries to underline the importance of artistic thinking with science, engineering, business, architecture, so on.
So just try to improve your awareness of artistic elements.
Like what we do, how we make our engineering learning more artistic, and the interaction with other fields is an interesting idea.
And I ask you to do homework, make a beautiful, informative posture.
And what you learned, linear system convolution, Fourier series, transform, and signal processing, and discrete Fourier transform.
And something, this is just too rough. You just make key details, put it into a one-seat posture.
Just the knowledge point you added together, make beautiful. What would be what I expected?
It's just an open-ended thing for you to make a posture, summarize all the key relationships, concepts, and so on.
And for graduate level of this class, and I help the students do final review and make a posture for whole medical imaging class.
And something like this, and this is CT part, nuclear part, and MRI, and

ultrasound, and optical.
So all the key relationships, this is my way to summarize medical imaging lecture for graduate student, not for undergraduate student.
Now what I ask you to do something similar, just for the Fourier analysis, linear system part, as your homework.
No standard answer, just do your best. If I am impressed by your design, and I will let you know,
we will let the TA to pick up a few really good ones, and I will try to learn from you.
So much for today, okay?
Thank you.