Good afternoon.
So as we scheduled, today's lecture will be given by TA.
See you in another meeting, will be a few minutes later.
So taking this opportunity, we can chat a little bit.
The homework assignment related to so-called RPI-ART-X program is really free style.
I don't want to make a physical poster, just make a digital one.
You can use the font you like, you use your creativity.
So the idea is to let you summarize what we learned.
Remember we learned the system, linear system, convolution and the delta function.
Then Fourier series, Fourier transform, discretize that, and then you have sampling theorem.
So a lot of interesting things.
And this part is called the foundational part, meaning that the later material, like image modalities, rely on your thorough understanding of these things.
And some students interacted with me.
So if you confused at earlier stage, like convolution,
then you wouldn't be able to understand the convolution theorem,
because it involves the convolution concept in the first place.
So if you understand all these things,
if you still do not fully understand, you still have an opportunity.
So my plan, hopefully, just a busy week, but not that busy, we finish the grant proposal deadline.
I hope by next Monday I will upload the next version of the book, book part one.
So by Monday you can see the book.
But right now you still have PowerPoint files and video records.
So you review, make sure you understand the key stuff.
And let me assure you, after all, so these stuff are not that difficult to understand.
You just need to understand what's a linear system, what's a convolution, how you derive a convolution.
Then the idea to represent a general function, either as a sum of delta functions,
a bunch of particles in pulses, or you represent the function as a number of sinusoidal waves.
That's really the essential idea of Fourier series.
Then you make the period infinitely long, you get a Fourier transform.
This goes through the derivation.
Then you say you want to use modern computer, so you have to deal with digital signals.
You need to discretize and you need to perform quantization.
So you just turn a continuous wave into a bunch of discrete signals that can be coded in binary format.
Then you put it into a computer.
When you do so, you don't want RTAs coming.
You don't want to lose information.
So the sampling theorem is very elegant to assure you.
And you do digital sampling if you do it right, under a very reasonable model.
Then you will have the original continuous information, fully or nearly perfectly reserved.
So this is just a time domain thing, from continuous to discrete signal.
Then you say Fourier analysis is a powerful tool.
You can do denoising, some feature extraction.
So you also want to discretize continuous Fourier spectrum.
So after digital sampling in time domain, you have a periodic continuous Fourier spectrum,
many copies in the Fourier domain.
Then you repeat the same trick again.
You do digital sampling in frequency domain.
Then you've got a nice symmetric arrangement, discretized signal in time domain and in frequency domain.
Then you have the discretized Fourier transform.
So the story is pretty much like this.

You need to go step by step.
So do spend time.
And this is a classic, some very important part of engineering education.
Spend time, you will understand.
Otherwise, I believe our TA and myself are very responsive.
And you send an email, you talk to us, you will get an answer.
And today, Kathieen will explain to you some hands-on things.
And we utilize a very popular MATLAB toolbox and the commands and so on.
You see how Fourier series and Fourier analysis discretize the fast Fourier transform in working.
So this is a hands-on part.
And what I mentioned earlier is some good understanding.
And I really hope you understand.
Not just say, plug in this, use this line, you can do Fourier transform.
And I can teach kindergarten the same way.
You put this, you are doing Fourier transform.
You don't know what's going on.
I really hope you understand.
So that is very important.
We, as engineers, we need to have scales.
We need to have insight and understanding.
So you can do better.
And today, basically, talk about hands-on part and visual understanding.
And some of you got lost.
It's not surprising.
This is always a challenging part.
But that's up to you to spend time to get a good understanding, Katherine.
It's all yours.
You can use mine or any way you want to use.
Thank you.
Can you hear me?
Yeah.
Can you hear me?
Yeah.
Can you hear me?
Yeah.
Thank you.
Okay.
Okay.
Oops.
Sorry.
Oops.
You need to have this.
This is the video part.
Otherwise, maybe you try this one.
When you learn the material, try to understand many things.
Even the linear system for analysis, very classic,
all this stuff.
So many people keep talking about it.
You could still figure out some good ways to understand it.
We'll talk about that later.
Knowledge, please.
Is that okay?
Can everybody hear me?
Is that better?
Okay.
Sorry for being late.
Today I have another MATLAB tutorial to present for you guys
so that you guys can better understand how to use the tools
that Dr. Wong has been talking about in class in MATLAB,
how to better understand using MATLAB.
First, I'll just briefly go over two of the MATLAB-based homeworks
that we have recently, where the one from lecture number five
was for the Fourier series.

You were to determine the Fourier components from the code
that was given in that video that was provided in the slide.
Then use the Fourier components that you got,
which ended up being numerical values,
and then create a curve that's close to the square wave.
I just have a figure here on the right
where we have the square wave plotted in blue
and then the Fourier estimate in red.
I used this equation for f of t that was given in the slides.
Then using the equations from the video,
you were able to get the a n and the b n components.
Then using this as just a summation of the different a n, b n,
with the cosines and the sines.
Then using this code here, which is just a MATLAB version of this equation,
I was able to make this guess vector.
Then when plotting it with the square wave,
you can see that it approximates it pretty well.
Not the best, because there was only, I think it was five Fourier components.
I don't remember in the video,
but there was only a finite number of Fourier components.
If you had an infinite number, then instead of these sums,
it would be the integral.
Then you would be able to approximate this square wave very closely
if you were able to do an infinite number of Fourier components.
From looking at the homework,
most of you were able to get something similar to this.
Not necessarily this exact plot, but something similar.
That was good.
Then for the second one, the Fourier transform part two.
One, there were some people who ended up doing it in MATLAB
using the integral int function.
This is a YouTube video that describes how you do the analytical solution
with the integrals by hand.
Some people did it in MATLAB as well, and both of them are accepted.
This video is a few minutes long, so I'm not going to play it here.
It shows you step by step.
If you didn't understand how to do it analytically using the equations,
then it shows you step by step how to get, for this specific equation,
how to get the analytical solution of the Fourier transform.
Then here's the MATLAB.
I didn't include the code, but I could include the code as well
when this is uploaded to LMS.
Here's just a brief outline of what we're going to go over today.
First, we would look at the discrete convolution.
This is a convolution of discrete signals
and then using zero padding or circular convolution.
Then we're going to go over spectral analysis,
which is one application that's very common with Fourier analysis
and with zero padding and with refined spectral bins.
Then we're going to briefly look at 2D image filtering
using the Fourier transforms for noise removal and edge enhancement.
This is one of the slides from the lecture
where you have the time domain signal on this side on the left
and then the frequency domain on the right here.
It shows the multiplications or convolutions of the different functions
and then the equivalent in the time domain and the frequency domain.
This is one of the slides from the lecture that you've gone over with Dr. Wong.
This was also in the initial slides
with the hands-on example using the two hands in opposite directions.
You can see that when one of them starts at the zero
and the other one ends at the zero,
then you have the convolution of the two end fingers in this case.
Then at n equals eight, they slide in the other directions,
so you have the opposite two end fingers in this case.

Here is a slide depicting a functional example.
We're going to go over this in MATLAB in a few slides later.
Here's an example that I actually got from Wikipedia.
If you wanted to look at it in more detail with more details in the description,
they have this h of n, which is just a square wave,
and then this x of n, which is like three square waves basically in one function,
one red and two blues.
Then the third figure is the linear convolution
when you just do the convolution of this h and this x directly.
You get these three triangle forms, which is as you would expect
when you do two square waves or rectangular waves together.
Then when you have this xn with n number, there's three here.
x is now a periodic function with n components,
and they mention this as circularly extended.
When you do the convolution of this xn and this h,
this is what you get as the linear convolution of those two,
but you can see that instead of looking exactly like this,
there's a lot of these edge effects where the triangles overlap each other.
The red and the blue triangles overlap each other.
But then when you look at the composite output,
when you take away the overlapping parts,
then you can see that the green highlighted section
is the only part that's not affected by the edge effects,
and it's very similar to the linear convolution
of the separate non-circularly extended two functions.
We would need the zero padding here,
which you're going to describe in a few slides as well.
Here is a hands-on result that you can plot in MATLAB yourself,
where if you just make these two vectors, this x and this h,
5, 4, 3, 2, 1, 1, 2, 3, 4, 5, and then you convolve them,
then you get this triangle shape.
You get this triangle shape.
This is just a linear convolution of these two arrays here.
With the circular convolution, as we saw in the graph from Wikipedia,
you have these three periodic, so here it's just three hands instead of one,
as in the linear convolution example.
You can see that there's overlap between the fingers of the bottom one and the top one.
But then if you have the zero padding, you add some space here.
Before, the red and the green lines are very close.
But when you add zero padding, there's some space here,
so there's not as much overlap of the functions in places that you don't want it to overlap.
For instance, you only want to look at this section, the red section, for instance,
but there's still overlap in the green section.
In the second one, now there's only overlap in the red section.
So you don't have to worry about the edge effects,
like it was in the picture from Wikipedia, example from Wikipedia.
Here are some examples of the linear convolution.
You can do this in Matlab on your own here, if you would like.
We have the 5, 4, 3, 2, 1, 0, and then 1, 2, 3, 4, 5, 6 as the second one.
This is just a plot of the two functions.
Then the second one here is the linear convolution just using the CONV,
which is the function that you used for the previous homework with the RC circuit,
where you did the convolution of the RC response and the unit step function.
So you just use the CONV here.
This is the result that you get, this sort of triangle here.
This is basically the code that was on the slide.
When you run this, you have the first figure, which is just the two different lines,
which is the information in the two different arrays, the X and the H.

Then the linear convolution ends up being this triangle.
The result is this triangle here.
If you do the circular convolution without the zero padding,
you do in terms of using the Fourier transform,
then you do the Fourier transform X and H,
and then you do the inverse Fourier transform of X times H,
because we know that the multiplication in the Fourier space is the same as the convolution in the time domain.
Now we just do the FFT of X and H, and then we do the inverse FFT of the X times H,
which is basically doing the convolution here.
Then when you plot that, you can see that the blue line is the figure without the zero padding,
so you just use the two vectors directly,
and you can see that the blue line result is not the same as the red line result,
which is just the linear convolution from the previous example.
So now you know that you would need to have some sort of zero padding here.
Then when you do have zero padding, here in this first N vector is just the length of X and H minus 1,
which would be the final length of the convolution vector,
which you saw in the RC circuit homework from a few lectures ago.
Then you can make this X pad, or whatever you want to call it, with X,
and then the rest, the first values are just the same X vector,
and then the rest will be zeros that are the length of the zeros is N minus the length of the original X,
and then the same for H, where it's going to be H at the start,
and then zeros for N minus the length of H zeros,
and then you do the same FFT of the two, and then the inverse,
and then you can see that when you plot them, the zero padded circular convolution and the linear convolution,
they directly line up with each other,
so you know that for this case, adding the zero padding can give you the correct result for these two functions.
You can see that when you run it in MATLAB,
so this is again the same code that was in the slide,
and so when you run it in MATLAB here,
this is the, if the graphs were a little bit too small in the PowerPoint,
you can see that this is the no padding circular convolution,
and this is the linear convolution from the first part,
but then when I add the zeros at the end here,
and then do the Fourier Transform of X, zero padded, Fourier Transform of H, zero padded,
and then the inverse of the multiplication of the two,
then they line up directly with each other,
and so that's an example of these two discrete signals where you would need the zero padding
in order to get the same result as the linear convolution.
And so now we can go over to some spectral analysis,
which as I said before is a very common application for Fourier Transforms,
and so this is a little bit blurry, but this is an example signal that we're going to look at,
and you can go to this website here from MathWorks,
and all the codes that are used here are from this example,
so you can look at this example when you get home if you wanted to,
if it's not clear from this lecture,
then you can still look at this website here when you get back home.
So here's the function that we have, so we have a frequency,
and then we also have a time vector,
and then you just make your X vector as the sum of cosines and sines with the different frequencies,
this 100 and this 202.5,
and then when you plot it, it ends up looking like this,
and so then you can make the Fourier Transform of this,

and then you can plot it in the plot,
you can plot the Fourier Transform of the function,
and then versus the frequency, which is in hertz as you can see on the X axis here,
and so from here you can see that the 100 frequency from the signal is very well resolved,
as you can see just a clear spike here,
very close to a delta function at the 100 hertz frequency,
which is one of the frequencies that was given in the original function,
but then for the second one, the one that was 202.5,
it's not a very clear discrete spike here,
which is because the 202.5 frequency is not within the default spectral bins,
which in this case were just one hertz,
so if you wanted to see what it looks like with bigger graphs,
and in the code, when you run the code that was in the slides,
you can see here where this is just the maximum value, the one,
and you can see this very clearly at 100 hertz, this first blue spike,
but then the 202.5 is still not very discrete here,
but this is without any zero padding,
but when you do add zero padding for this,
you can increase the length of the vector using this FFT and then X and 2000,
and so here the spacing between the frequency bins instead of one hertz is just the frequency over 2000,
which is 0.5 hertz, and so then when you plot that,
you can see these very clear spikes here for the 202.5 as well as the 100,
which are the two frequencies that are in your initial signal,
and so that's because we changed the spectral bins from one hertz to 0.5,
and since the spectral bins, since the frequencies were 202.5 and 100,
then now we're able to very well distinguish the frequency value
for this 202.5 signal within the combined signal,
and so when you plot, when we run this code,
this is what the graph looks like with the one again as the maximum value,
and then you can see very clearly these two 100 and 202.5 frequencies after the zero padding
that decreases the width of the spectral bins and increases the number.
And so this is just another example with a continuous function,
and so this starts as the same function as before,
but then there's more and more, there's three more components that are added,
three more cosine and sine components that are added with 45, 407, and then this 445.8 frequencies,
and then this is what the plot looks like for that function,
which just from here it looks very messy because there's a lot of things going on here,
but with the frequency for your transform, for the frequency domain,
then you can more easily establish what's going on in terms of the frequency response.
And so here is what it looks like without any zero padding, without any bin refinement,
which is as in the previous example, we refined it to 0.5 hertz bins.
This is without any refinement of the bins.
And so you can see here the 45, the ones that are the discrete values,
not the ones with decimal point, decimal points in the frequencies,
they're pretty well resolved, but then the 200 and the 445.8 are still not well resolved.
You can tell that there is a spike here, but it's not at the maximum value of one,
and so this would be the code that you use for that.
And as you can see, the 202.5, 445.8 are not well resolved,
the frequencies are not discretely given.
And then if you try to decrease, but if you refine the bins by a factor of two,
so if you decrease the bin size by a factor of two,
then that's the same as going from one to 0.5 as in the previous example.
And so here you have, you can see that the 202.5 one is again,
you can see very clearly where it is, but for the 445.8 one,

since the bin is, since it's 0.5, then this one is still not well resolved.
And so we can see that in MATLAB as well.
And so, oh, I didn't run this one, but...
This is the initial function that we had,
and then this is with the unrefined bins here.
And then this is what it looks like with the refining by a factor of two.
And so I cut off the code in the slide,
but it's just you have this bin factor equals two,
and then you increase the length of the FFT by making the zeros at the end,
using this bin factor two times the length of your time vector,
and then you can plot the discrete for your transform
in the same way as in the previous example.
But then if you change the bins to 0.2 by refining by a factor of four,
then here you just have the same code where you do bin factor
times the length of the initial time vector.
And then as you can see here, all of the frequencies are very well resolved,
and they all have maximum values of one for each of them,
the five different ones, even the 445.8 one.
And so as you can see in the MATLAB code here,
if I change this from two to four,
then yeah, this is what you would get.
So you can see that all of these are very nice, very well defined here.
But when you have two, it's not as well, all of these are well defined
except for the last one.
But then when you just had unrefined ones,
then it was only the first three that were integer values
that ended up working well.
And so these were all just 1D cases where it was a sum of cosines and sines,
or just linear values in the convolution example.
But you can also use Fourier transforms in 2D cases,
such as in image processing,
which is pretty important in the image analysis part of the course.
And so this was one of the slides that was also from the lecture,
where you have the 2D Fourier transform in these rainbows here,
and you have the forward and the inverse Fourier transforms.
And so if you have this as the initial function, this sideways rectangle,
then when you do the Fourier transform,
you can have this frequency response that has real and imaginary values.
And so one of the things that you can use the Fourier transform for
is noise suppression.
So if you have, for instance, this image of this woman that's very noisy,
you can see that there's a lot of white noise in the image.
And you look at the Fourier transform here, which is on the bottom.
Then if you remove some of the information,
you make all of these ones zero, for instance,
and you only keep the information in the middle,
you can remove a lot of that noise,
and you can more clearly see the woman in the picture.
And so this is very important for a lot of our applications
where sometimes just inherently there's noise in the system
or noise in the environment.
And so very often we need to do this sort of thing
to decrease the amount of noise that's in our original,
so that we can reconstruct our original images.
And so you can also use the Fourier transform for lower high pass filtering.
And so if this is the original image here of this building,
and then you get the Fourier transform of that,
you get the Fourier transform response of that.
And then if you take only the middle section here as the useful information,
then you have some filtering here.
And then if you make the center of this one dark, but keep the rest of it,
then you have a different type of filtering as shown here.
And so these are some common filters that are used for 2D image processing.
This Unsharp, Gaussian, Sobel, one type of Sobel,

or this is in the spatial domain and this is in the spatial frequency domain.
And so you can have these different types of filters.
These are just examples, but there's a lot of different filters that you can use.
Matlab also has a function, different functions that you can use.
There's one called fspecial3, which is a 3D filtering.
And then there's also like Gaussian filters
and other functions where you can use filters on your signals.
And so here's an example of what it looks like for filtered images.
And so if this is the original image of someone's eye,
and then you look at the 2D FFT of the original image,
then this is what it looks like when you do a convolution with the Unsharp filter.
And then the Gaussian blurred image, which is just using the Gaussian filter.
And then the Sobel image filter, which gets the edges, it's good for edge detection.
And so you can see all the lines within the eye as well as on the contours here.
And this one is Gaussian blurred, and then this one is also somewhat blurred, but sharp in other areas.
And so these are what the frequency domain information might look like.
And so you can see that the Gaussian filter, you see these like sort of squares in the edges.
And then in the center here, it's not as well defined.
But you can use these filters here to, if you wanted to just look at the edges, for instance,
you could use the Sobel filter to get like some sort of thresholding.
Or you can use Gaussian blurring or deblurring.
And there's a lot of different things that you can do with these different filters and functions.
And so this is just an overview of the homework that you have due on Friday.
And so I'm not sure if Dr. Wong has anything to say about the homework poster.
Oops.
Thank you very much, Katherine.
You explained clearly.
Homework-wise, before you came into the classroom, I already explained to students,
it's really free form, and just use your creativity, try a little bit artistic.
And you can select the top three and let me know.
Take a look and see how students will make it.
And I like the poster view graph and the info graph, those things.
So we do things in an impressive way.
It's a very effective communication scale.
So again, the next two lectures are about network and image quality.
The toughest part, I would say, is finished, the foundational part.
So in the next two lectures, you'll learn a little bit of relevant things,
but in terms of logical connection.
So all things we learned up to now, and the hands-on experiment already covered,
this is really your last chance.
And you do hands-on, you review PowerPoint and video record,
and I told the students by Monday I will upgrade my draft,
the part one draft from version zero to version one, some typo will be fixed.
But this wouldn't affect your review.
You'll have a lot of material to read already, particularly the PowerPoint.
The good way could be you just review PowerPoint, slice by slice,
and then somewhere you feel confused and you just check video record.
And you still have problem and you read my version zero,
but after Monday you will have version one.
And then you'll have further questions.
You do Google search, you participate in your group discussion,
and in addition to possible consultation with TA and with myself.
So I strongly urge you to get a thorough understanding.
And this set of scale, linear system concept,
Fourier analysis will go with your whole career.
This is fundamentally important.

And MATLAB is a very good thing, and some commands you're not familiar,
just Google example will pump out maybe even better than I could,
or TA could explain to you.
If you do your style, you do hard work and are motivated,
dedicated to this part, you will be well prepared
to understand medical imaging modalities such as X-ray CT and MRI.
So use some high-tech machine and we can see what's inside you,
crystal clear, dynamically up to sub-millimeter details.
Those wonderful things, and you need to understand
the mathematical stuff we have been trying to explain well.
So so much for today.
In an earlier announcement I mentioned that we have faculty meeting
from 3.30 all the way.
I don't know how long the meeting will last.
So as a result I saved my office hour a little bit earlier.
And also we finished class today earlier.
I will stay here a little while.
If any of you want to just treat this as on-site office hour,
you can ask me right away.
Okay, so much for today.