So I'm very happy to work with you this semester.
So I'm your instructor, and people just call me Ge.
As in Google, you call me Ge.
OK, easy to remember.
And I will first present a few things.
Then our TA will teach about basic programming
skills in MATLAB.
This is our BB schedule, bio-instrumentation, bio-imaging.
Actually, I really like to call it medical imaging.
And we mainly focus on medical imaging.
At RPI, we have undergraduate level course, this one.
And also, we have a graduate level.
And I will teach more in that class.
We finished the last semester medical imaging.
We focused on X-ray CT, nuclear imaging, MRI, ultrasound,
and optical imaging, five modalities.
And at the graduate level, the depth
will be substantially deeper.
And here, we give you general exposure to medical imaging.
But the tricky thing here, we are undergraduate students.
And this is a very much diverse population.
Some students will be prepared with linear system,
Fourier analysis.
But many of you are not so familiar with Fourier transform,
fast Fourier transform.
And the convolutions of those concepts
are very important, fundamental to understanding of each
of the five imaging modalities.
So specific to undergraduate level medical imaging
or bio-imaging education, we always have a foundational part.
So foundational part shown in this block.
So from next lecture on, we talk about system,
particularly linear system.
Convolution, then we move on Fourier series.
As a limiting case, we have Fourier transform.
Then we go a step further, dealing
with digital signal processing.
So you collect the information, store it in your computer,
your process, you do Fourier analysis,
the filtration, and so on.
Then you would need to discretize the Fourier
transform into so-called discrete FTE.
And you can have a very fast algorithm for the FFT
and a few other things.
And once you finish the foundational part,
the mathematical foundation, you are
prepared to learn CTE, MRI.
Fourier transform convolution will come back to you.
Again and again, without solid understanding,
you will be in real trouble.
And this part we usually teach using PowerPoint slides.
And then we copy material from internet, textbook, and so on.
And I explained that in the first lecture,
and I see a gap existing in BME education.
So for mathematical preparation, on one hand,
we do have a textbook and a very detailed explanation,
linear system, Fourier analysis.
But you do not have time to go through one semester
or two semester workload to cover the ground.
On the other hand, the textbook can give you appendix.
This is a list of formulas.
That would be too abstract for you to digest.
So we have a BME version of linear system
and Fourier analysis.

That is what we are going to explain to you this semester.
And we have been teaching this part for multiple years.
One problem is that the information sources
are multiple.
So the notations and the presentations are not unified.
And later on, you review PowerPoint,
you may still not feel totally clear.
So I see a need to develop an undergraduate-level biomedical
imaging textbook.
And I'm highly interested.
So over the holiday season, I started writing up.
So I uploaded to the LMS.
It's not totally finished yet.
It's almost done.
But I need to make a transition into discrete Fourier transform.
I need a few more days.
But you know RPI professors always
do teaching and research.
February 5th, we have a major federal NIH deadline.
And I'm involved in multiple grant writing, so very busy.
And also, I have other obligations.
Last time, I gave a talk at Michigan State University.
So I see myself too busy before February 5th.
But I would have more time after the deadline.
Then I will finish the first part of the textbook,
called the Mathematical Foundation for Medical
Imaging Whatsoever.
However, you can download the textbook.
And this part of linear system convolution
and tail signal processing are all covered.
So you have that material already in digital form.
You can download, you can read.
And before we teach discrete Fourier transform,
I hope I would have time to finish the text.
And this is the first version.
It's a draft.
And this is my understanding using unified notation.
And as with any first version textbook,
there may be typos and some words still confusing.
Feel free.
And I'm very motivated to work with you
to make sure the text is easy to read.
So this is about the overall schedule.
Thanks to our TA custody and the support
according to our academic calendar,
so something like this.
And the one thing I assure you, I'm
usually very prompt with email.
If you couldn't see me in office hour, you have a question.
Send me email.
And 99% chance you got my reply same day or maybe even quicker.
So this is the overall schedule.
And before we really go into the content,
let me give you a RPI formula for effective learning.
I just made it up this morning.
So first, you need to do preview.
And most students, based on my experience,
you do not read.
You only sit in the lecture.
And you may not totally convince that when you read the textbook
or do your homework, particularly
before examinations, you spend a lot of time.
That's a bad habit.
And you are going to spend the same amount of time.

Why not just do a little earlier?
So you do preview.
And the good thing is that I'm one of a few professors
who recorded my lecture in the system.
So you can just watch all the videos.
You can read textbooks.
And it is in plural form.
So I have my own write-up for the foundation.
Also, you have the green textbook.
You can buy.
You can download to have PDF version.
You do have enough material to read
before sitting in the lecture.
So in class, this is very important.
I wouldn't penalize you if you escape the class.
But that's not to your best interest.
If you do preview, then you have a few confusing things.
And that may be clarified in the classroom.
And you can interact with me freely.
And after the class, you need to do review.
No one just does smart.
You just read vines or license vines.
You understand everything.
Foreign analysis and medical imaging modality
are highly interdisciplinary and complicated.
It's very important for your academic career
and also for your family life, this medical imaging
directly related to health care.
So you need to do review as well.
So review and in-classroom licensing
and proactive participation and the preview before lecture.
That's the RPI formula, very important.
And we do have some advantages for this class.
We have digital properties from spring last year.
So we have a class video, PPTs, the class video.
If you play, so the PowerPoint will
be displayed side by side.
So you'll have all the information available.
And the green textbook available,
also my own draft textbook available there.
And the schedule, and we just, the class schedule,
we just explained, and we follow the schedule.
Where can you find last year's video?
So if you go to IOMS system, so here on the left-hand side,
you see the notation, I-17.
You click, and you will see last year's video recorder.
Majority of the lectures were recorded, not all by me.
And the last few lectures were told by my fellow instructor.
And I went back all the way to my home country.
And then my mother suffered from colon cancer.
And she did surgery very successfully.
And in the process, I show myself
how important actually CT, nuclear imaging, ultrasound,
all play a critical role.
And they save life, and it's just a wonderful thing.
But anyway, you click this button,
you see all those videos.
And at this moment, the system is not streamlined.
It's not listed in order, say, first and second.
It's just a mess up together.
And you need to go through to find the video
you really want to listen.
And one of you sent me email, I think, this morning.
And I just show you, and she has a problem

with the MATLAB installation.
I assume some of you suffer from same problem.
RPI has an official license.
So this problem ought to be solvable by help desk,
or somehow that's an RPI infrastructure support issue.
I myself am not an IT expert.
However, let me give you some suggestion.
So in parallel to your effort solving MATLAB installation,
there are alternative things, and equally convenient,
in my opinion.
So if you do Google search, say, run MATLAB online,
so you're determined enough, and you
will find online interface you can run MATLAB.
And also, another thing I came to know recently,
a software package called Octave.
And what is Octave?
What's the difference between Octave and MATLAB?
And they are essentially the same thing.
And Octave is free version.
And you can download, you can run,
and you can do similar things.
So if you do have trouble with MATLAB,
then we just use free version.
And the MATLAB programming capability,
we are going to use in this classroom
as a minimum, just some data processing.
We don't need fancy programming skills.
And I assume you will listen to the teaching by our TA.
So my book, I don't know how soon I will finish.
It depends on how busy I am.
But I think, within this year, the first draft of textbook
should be ready.
I have been teaching this class for five years in a row.
So I know the key point.
I just need time.
So about two days, I can finish the chapter.
But my day really just fragmented.
So I just see, when could I finish?
But the most important part, or most of that, is part one.
That's the foundation part missing in the literature.
So for BME student, the part one gave you
reasonable exposure to linear system
convolution and Fourier analysis.
So that part should be done either this month is not
possible, should be finished next month.
Then the second part is medical imaging modality.
And the good news is that the green textbook
serve the purpose.
And the text is outstanding.
But why bother I want to write a new book?
The reason is that existing textbook is really out of date.
And I can use state-of-the-art knowledge, update
its imaging modality, and treat them in a unified fashion.
So that part, even I couldn't finish for this class,
you can rely on green textbook.
You can listen to imaging modality teaching later on.
So anyway, so this is where you can find the draft
for first part, dated January 18.
And I will update it for sure.
So the table of content, you see part one,
medical, mathematical foundation.
So I divided it into multiple chapters.
And each chapter, you have four sections.
And the last section is always remarks.

And the first sections, you can treat as three major aspects
under the topic.
And I just try to explain logically.
And not only know the Fourier transform formula,
you know how you come from, how you derive
these very elegant relationships.
So just this part under the green zone
means what I need to finish.
And the second part of medical imaging modality,
and even I couldn't finish, you can use your green textbook.
That's all for today.
And Kathleen, so you can just upload your slides now.
And that's all.
OK, thank you.
Can everybody hear me with this thing?
OK, so we've already met before, but I'm Kathleen.
For those of you who didn't come to the first class,
I'm the TA for this class.
And so after Dr. Wong's introduction,
I'm going to give you guys a little bit of a tutorial
on MATLAB.
And make sure that you already have MATLAB
installed on your computer if you brought it
and would like to work on the code in real time.
But if you don't have it, you can download it
from the .cio software website.
And then you just follow the directions according
to if you have a Mac or if you have a Windows PC or even
Linux, I think.
And then activate it according to the instructions as well.
And it takes a little bit of time.
So I think it took me maybe like 30 minutes just
to install it over the Wi-Fi.
So you might not be able to do it right now in the class,
but you can do it when you get home
if you don't have it right now.
And so just as a brief introduction,
some of you maybe went through the on-ramp MATLAB tutorials
from the MathWorks website, which is the company that
developed MATLAB.
And so in short, MATLAB stands for Matrix Laboratory, or M-A-T-L-A-B.
And MathWorks is based in Massachusetts.
And they've been developing MATLAB for some time now.
And I think now that they have the 2018 version,
they have A and B for part of the year, first part of the year,
second part of the year, they update it constantly.
And there's also a lot of tools online and help forums.
If you ever have any questions, you
can just type it into Google, what is your question
or what is your issue.
And then you can pretty much find a solution that's
similar to what you're looking at,
or you can just keep looking at different forums
to see if you can find your solution.
And so MATLAB is overall used for many applications,
such as data analysis and processing in our lab,
as well as Dr. Wang's lab, the image processing.
Most of the analysis we do in MATLAB,
you can also use it for algorithm development,
not just for image processing, but also
for controlling different things.
And you can also process data from different systems.
And nowadays, they've also started
to develop some methods, some tools for machine learning,

which Dr. Wang mentioned in the class as well.
And so here are some links, some useful tools.
And so first of all is just the general MathWorks website.
And so you can click this link on your own,
but it just links you to the MathWorks website
where they have tutorials.
And again, you can look at the on-ramp one,
the on-ramp tutorials for a more detailed one that
goes over it with a video.
And then also you can code along with it.
And they also have some tutorials just in words
without the video.
And then MATLAB also has a YouTube channel.
And so they have some of their tutorials
online as well on YouTube.
And then they have some YouTube tutorials.
This is just a general one, but there's
multiple ones out there for like, oh, how do I
process this image using technique A
or something like that.
So you can find it on YouTube as well if you
like watching videos and doing it along with the video.
And then also the MATLAB documentation,
which is just mathworks.com slash help slash
MATLAB, which is also the general documentation website
for the MATLAB function.
So you can go to this website.
And then if you have a question about a certain function,
then you can just type in the function in the search bar
and find the documentation pretty easily.
And then also you can use this F1, just your F1 command
on your computer.
Well, on a Windows computer, I guess.
You can just use the F1 when you're
highlighted over the function like this, for instance.
If this was a function and not a link.
You can press F1.
Or in the command line, which I'll
describe a little bit later, you can say help ABC your function
name.
And then it'll give you the inputs and outputs
that you need to put in as well as what this function does
and a lot of other information that you can
use to better use the function.
And so it's pretty good.
So if any of you can either open MATLAB on your computer
or you can just see, look on the screen
if you don't have it installed right now.
But this is in general what the MATLAB window looks like.
And so you can orient these sections any way that you want.
You can move them around.
But this is how I have them oriented.
But you have one section, which is the folders
that you have in your MATLAB folder.
And so I have just some of my folders for my research
as well as BIBI, which is just the class.
And then you also have this editor window
where you can write functions and run functions.
For this one, it's just another function
that I had written before.
And then you also have this command window
where you can type in commands to run your functions
or just to make arrays and that sort of thing.
And then you also have your workspace variables,

which you can make variables in the editor
or in the command window, such as you say X equals five
or something like that.
Then X would appear in your workspace variables.
And then there's this one column that says value here
and it would just say five, minimum, maximum.
And there's other ones that, other parameters
that you can also add there.
If you wanna look at things in more depth,
such as if you have a certain array that you made,
you can see what's the minimum and the maximum value.
And then you can also click them
to look at the variables as well.
And again, your layouts or the positions of your windows
can be different and you can even drag them out.
So if you want to have them separate
in a different window to look at,
and it's pretty flexible, this layout,
but this is just how I have it on my computer.
And so the command window,
which is this bottom one for me here,
the command window is used to execute immediate commands
such as definition of variables, simple operations
and running of functions.
And so I can just show in MATLAB here,
this is a function that I have.
So you can say like, you can just say A equals five.
And so I just said A equals five here.
And then in my workspace variables,
there's A here and the value is five.
And so you can use the command window
for more immediate commands.
You can say like one plus two.
And then it says answer is three.
And then this answer, they always put it into a variable,
but this answer variable is just three.
So you can do a lot of these simpler immediate commands,
but you can also run functions such as the function here,
which I'll describe in a few slides,
how to use that function, how to run these functions.
Oops.
And so then there's also the editor window,
which was the top window for me, this one here.
Oops.
And so in the editor window,
you can write scripts or functions,
and then you can call the function in the command window
and then run it.
And so this function was the one
that was open for me before.
And so if you wanted to run this code yourself as well,
you can just do new and then function here,
and then it'll create a new function,
which a function, output argument one, two, untitled,
it has just a default, some default parameters here,
and so you can change the default parameters
so that it's, change them to what you want.
And so for this one, I just had area,
calculating the area of a rectangle,
which is pretty similar, just the length times the width.
And so function, and then the area is the output here,
which is in the brackets, equals, and then area,
I just named the function area,
but you can call it whatever you want,

like a, I don't know, area calculation
or something like that.
And then X and Y are just my two variables,
my two input variables, and so it can be like two,
three, five, six, any variables that you want.
And then inside here, between this function and this end,
and so you have to have an end at the end of a function
just to show that the function is over, pretty much.
And so within this function and end,
I just have area equals X times Y.
And so X and Y are the variables that you have,
and then it'll calculate the area
and give you back the area as a output of this function.
And so if you wanted to run this function
in the command window, you just do area,
like five, six, or something like that.
So you have X equals five, Y equals six,
and then your answer comes out here, 30.
And so that would be how to run this basic function
in the command window, writing the basic function
in the editor window, and then also running
this basic function in the command window.
Oops.
And so you can test it out for different inputs.
I just had five and six, but here I have 10 and 20,
or two and three, but it'll just,
since this is a very simple function of just X times Y,
it'll just give you the area of the rectangle.
And so here are just some basic commands
and syntax that you can use.
And so I think that I put it in on a previous slide,
but if you have in the command window, help,
and then in this blank line here, your function name,
it'll bring up the help files on the function.
So for instance, if I wanted to help on,
there's this function called image,
where it shows an image, or imshow, which shows an image,
which we'll use a little bit later in the class.
And so if I do help imshow here, oops, oops, sorry, help.
Like this, not the first one.
Help space imshow, and the imshow will appear
in this purplish color, which means
that's what you're trying to find help on.
And so, and so you can see here in this command window,
it has all of the information for imshow.
It has, yeah, the inputs and the outputs.
So you can have the inputs and the outputs here,
imshow file name, or you have the referencing object
and different things that you can put in here.
And then it also says the parameters that you can put in.
And then it also gives you just a summary
of what the function does, which could be helpful
if you find a function online and you don't really know,
or you see it in the homeworks or in the slides,
and you don't really understand what it does,
then it has a summary here.
And then also, if you have the image processing toolbox,
then it has some other certain parameters
that you can also set.
This is just one example of a function, remarks.
And then it also gives you some examples.
So it just has an example of how you would use
this imshow function for each of the different parameters
that you can put in.

This is just a simple imshow of image,
and then you can also have imshow
with a different display range and on and on
for different parameters, input parameters.
And so if you have a function, just simple imshow,
but you want to use it for more advanced applications,
then you can look at the help here
and see how you can improve the quantification
or the display using all the examples
that they give you in the help just within MATLAB itself.
And so also, some other commands that you can use
in the command window, if you type this clear,
it clears the previous variables, references,
and any loaded functions that you have.
And so I think that I did clear already,
but I didn't say anything about it.
But now I have these two variables,
this A that I had, A equals five,
and then this answer variable,
which was the answer for the area calculation.
And so if I do clear here,
then all of these workspace variables are cleared.
So if you end up having like 20 variables here
and you just want to clear up some space,
then you can just type clear and it'll clear all of those.
And so then there's also this CLC.
And so in this command window,
I have all this information here,
but if I type CLC, then everything is cleared away as well.
So sometimes if you end up typing just multiple things
in the command window, if you're trying out a function,
multiple times and it just gets too confusing,
you can just type CLC and then this will be cleared,
the command window.
And then one thing that's important is the semicolon,
which is just the semicolon that you use,
just the regular semicolon.
But it's used at the end of a command
to mute the printed output.
And so as you can see, if I write A equals five here
without a semicolon, it gives me this A equals five,
and it prints it out again.
But if I type A equals five with a semicolon after it,
it doesn't print out the output of A equals five again,
but the variable is still here,
still stored in the memory.
It just doesn't print it out for you.
And so if I had, in this area function,
if I had taken this away, and then save it,
and then if I do this area,
more, four or five, then it'll print it twice
because I didn't mute this output here.
So it says area equals 20 from the original function,
and then it also says answer equals 20.
It just repeats it again.
It displays the output of the function,
but then it also displays what's the answer that you get
from this line in the command window.
And so then also this percent,
this percent you can use to make a comment,
and so this one right here,
it says this function calculates the area of your rectangle.
This is just a comment.
It's pretty much just telling myself what this function does,

and sometimes if you have a script
that's maybe like 1,000 lines or something like that,
it's helpful for you to comment yourself
on the code so that you can know that, oh,
this part of the code does this,
this part of the code does that,
and you just use this percent sign before any of your text
to make a comment, which is very useful for longer scripts.
And so then also there's this control C,
and so that's just like when you press copy.
Sometimes if you have a function
that's running for a long time,
maybe you made an error in the writing of the function,
or it's just sort of crashing MATLAB.
You can just press control C.
I can't really do it here
because I don't have any of those functions right now,
but you can just press control C,
and it'll just stop the operation of that function.
It'll basically just kill that process,
but then MATLAB is still open.
It just stops running the one file
or the one functions that you're running,
and so those are very useful,
some of the pretty useful commands.
And so some of the other basic syntax for operators,
plus, minus, divide, and the times, the asterisks,
those are all just the basic operators
of addition, subtraction, division, and multiplication.
And then this carrot or the hat symbol is the exponent,
so if you wanna do two to the three,
you just do two and then the little carrot sign three,
and then that'll make two to the three.
And so for matrices, these operators also apply.
You can do matrix one plus matrix two,
and then, but if you add this dot here,
if you do dot star, then it'll do element-wise operation,
which I'll describe in a couple of slides as well.
And so there are some specific syntax for arrays,
and so arrays are denoted by brackets,
which are these square brackets here,
and so commas and spaces are used
to separate elements of an array,
such as one, two, and three,
or you can also use a colon to signify
the elements between two values,
such as if you do one colon five,
then it makes an array of one, two, three, four, five.
And you can also do, if you wanted to do
like two, four, six, eight,
you can do two colon two colon eight.
So you can also put a step,
put some information about the step as well.
And then if you wanted to do a matrix with multiple rows,
then you can use a semicolon
to separate the rows of your matrix,
and so you would have, for this one, two, three,
four, five, six, two rows,
you would just do one, two, three,
and then semicolon four, five, six
in order to get the second row,
to get the four, five, six in the second row,
to make this n by m matrix.
And so this is a,

this is a, they called it the MATLAB cheat sheet
that I've just pulled off from online,
which is just from this website directly.
And so this has a lot of the commands
that I just described in the previous few slides,
as well as a lot of other ones,
such as this one has information
about how to deal with tables,
and strings, and also cell arrays,
matrix manipulation, logical operators,
and this is just the second page
where it has information about plotting,
and some other commands, and general comments.
So this could be helpful for you
if you don't have as much experience in MATLAB,
they have some examples too.
So if you wanted to take a look at this,
it's at this website here.
And so this was just an example
of writing a simple function.
So as we already saw the area of just a rectangle,
so you can write a function
for calculating the area of a circle
with the radius as the input,
where just the pi is just pi.
So as I said before,
you can use the new and then function,
and the default format by changing the inputs,
and the outputs, and the name.
And so, oops, I already wrote, showed what it was,
but you guys can either try it for yourself,
or I can go over it on my own,
or do you guys want to try it for yourself first,
or do you want me to just go over it?
Do you guys have a preference?
So who wants to actually try it for themselves?
I guess no one.
So this is what the function looks like.
For calculating the area of a circle,
and so I just modified the previous function,
and so I just have function area,
which is the output, equals area circ,
which is just the name that I made for it,
and then radius is the input.
And so then I just have this comment here for myself,
this function calculates the radius of a circle,
and then I have the area as the output,
equals pi, which is just pi, times the radius squared.
And so I put this dot,
the squared is just the exponent operator.
The little carrot one is just the exponent operator,
so radius squared.
And so I put this dot, dot one here,
because if you have a matrix, or an array,
then you can find the,
you can square each of the values in the array.
And so I just put this here for the next slide.
And then the end again,
which I showed here.
And yeah, this is just the function in MATLAB itself.
So I just showed you all the answers, but that's okay.
So, and then we can test this function,
and so like I just said,
if you make a vector of radius values

starting from 0.1 to five by 0.1 increments,
and so then you calculate the area values and plot them.
And so, oh, okay, I have the code there.
And so this is how, oops, this is how you would do it.
And so making a vector of the radius values
starting from 0.1 to five with 0.1 increment step.
And so, a little small, I guess.
Oops.
So it's a little stretched,
but that's what it looks like in the command window,
and it's the command window here.
And so I just made radius rad as my variable
equals 0.1, which is the starting point,
with 0.1, which is the increment.
So it'll go 0.1, 0.2, 0.3, and then to five,
which is the maximum value that I want.
And then I just did areas equals area circ radius
as the input.
And so I saved those values into this areas variable.
And then I also had that you can plot the values.
And so you can plot this, you can just make a figure
and then plot radius, which is the x-axis,
and then areas, which would be the y-axis.
And then you can see this nice curve here,
which is covered by this, but you can see
that this nice curve here,
where the radius values are on the x-axis,
and then your areas that you calculated with the function
are on the y-axis.
And so then you can plot this nice curve here,
and the area values versus the radius values.
And so in addition to just plotting this normal one here,
you can also plot fancier curves.
So you can have this.
You can just have radius areas.
And then these are some inputs that you can also put
for the plot function, which is just how you,
just a simple, oops, sorry,
a simple plotting function.
And so I have this ro here,
which makes the red circles as the points
instead of just each point connecting into a line.
And then I have this line with three.
So as you can see, the circles,
the line width of each of the circles is three.
And so this is just a simple one.
You can also make a title.
This is a little bit small, but it, and covered,
but it says areas versus radius here.
And then there's also this x-label,
which is labeling on the x-axis, this radius,
and then y-label areas, which also labels the y-axis
of just showing what are the area values.
So you can label your graphs, the axes,
as well as put a title on them.
And then you can also manipulate
how you want to display them,
make them kind of fancy, pretty, different colors.
So that's one of the things that you can do
with not only the plot function,
but there's also, this plot function is just line plots,
but there's also scatter plots, bar graphs,
and 3D bar graphs, 3D plotting.
There's a lot of other different,

different ones that you can use.
And so then, as we made a simple matrix in the first one,
if you wanted to do some matrix multiplication,
so if you wanted to multiply these two matrices, A and B,
where A is this one here, and B is this identity one here,
and so there's matrix multiplication,
which is just this A times B,
and so you just do the general matrix multiplication
algorithm that you guys all know,
where you multiply the rows by the columns,
and then there's also this element-wise multiplication.
And so in this case, you would do this one times one,
and then this two times this zero, three times this zero,
and so you do element by element, the multiplication,
to get a final matrix.
And so if you wanted to do
the element-wise multiplication, for instance,
you would just use this dot,
and so instead of just the star,
if it would be just A times B,
regular matrix multiplication,
if you wanted to do the element-wise,
you would do A dot star B.
And so this is what I just described,
where you do the regular matrix multiplication here,
so it's the identity, you get the same matrix back,
but for the element-wise multiplication,
you only get back the diagonal,
because you do this one times one, two times zero,
three times zero, and so since there's only ones
on the diagonal, you only get the one, five, and nine
back from your matrix element-wise multiplication.
And so this is showing this as well.
And so this part here is just an example
of how you would do this whole operation here in MATLAB.
And so you have, defining the first matrix,
this A equals one, two, three,
and then you have the semicolon here,
so then the second row is four, five, six, semicolon,
and then the third row is seven, eight, nine,
and then your B is the identity matrix
which you make in a similar way,
and then if you do A star B,
then it just becomes the same, this matrix right here,
and then if you do this A dot star B,
then you get this, just the matrix with just the diagonal,
but zeros everywhere else.
And so that would be how you do that in the command window
for yourself if you wanted to try it on your own.
And so another fancy graph that you can make
is this plotting this 3D cone.
And so this, if you have this T equals zero, one, zero,
then you have use this cylinder function
which I just have the help window
about the cylinder function here
where you make this X, Y, and Z,
and then you use this function that's called surf
which is making a surface plot,
and then you can make this 3D cone,
this fancy 3D cone with the height of one
in both directions from the origin.
And so that's an example of one of the fancier
plotting functions that you can use.
And so if you, they also have a lot of examples

where you can make the MATLAB logo as well.
The fancy little, this, this thing.
And they have the examples directly
on the MATLAB website for that
if you wanted to recreate their logo for fun.
And so one of the other things that we can do
is image manipulation.
And so images can be very easily manipulated in MATLAB.
And one of the, since this is an image,
medical imaging class, one of the things that we do
in our own research in biomedical imaging
is we process images very often.
And so you can crop an image or show an image,
plot an image, find the intensity values of an image,
anything like that.
And so if we have this image right here,
which is one of the standard images
that they already provide to you in MATLAB,
and I can crop out this man's head here
if I wanted to just analyze his head region
using a very simple function that's just called IM crop.
And then you can also plot pretty cool,
like nice figures of these different circles
intersecting with each other
to make this yin-yang symbol.
But these are just some simple examples that you can do.
And you can also find the spectra of these.
And we're also going to do some Fourier analysis,
as Dr. Wong mentioned, in a few classes from now.
And you can look at the properties of images
very easily in MATLAB.
And so this is just an example of how to crop the image.
And so in the command window here,
I just use this IM read function,
which is just reading an image.
And so you have IM read, and then this function,
or this image is one of the images that I already have,
that is already given to you when you download MATLAB.
It's one of the standard images
that they provide with the download.
And this is just called cameraman.tiff.
And so I do IM read cameraman.tiff,
and then I make a figure, and then I can show this A,
which is the data that you read
from this cameraman.tiff into the figure.
And so I can say A equals IM read cameraman.tiff.
And then it reads that into this variable.
So you can see that it's a 256 by 256 unsigned integer.
And then you can make a, oops.
You can make a figure here, and then just IM show that,
IM show the data that you read from the figure,
that you read from the picture into your figure.
And then you can just have a figure pop up
and show your image here.
And if you wanted to look at a different one,
such as there's one that's called this trees one.
Oops.
And then you can just IM show that one, too.
And then you can also show any of the figures,
the other figures that you want as well.
This is just a second example.
And so if you wanted to just crop out just his head region,
you can create these.
They just call them data tips,

where they show you where is your pointer on the image.
And so for the images that I have,
there's just this crossbar here.
Oops, crossbar here.
And so if I put it here,
if I put it here, then it tells me the X and Y location,
as well as the index,
which here is just the zero to 255 intensity value.
And then it also gives you an RGB value,
which since this is a grayscale image,
it gives you the RGB values of this grayscale image.
But if it was a color image,
you can also see the RGB values for that.
These are all the same,
because it's different levels of gray.
But if it was a red, just like something red, for instance,
then it would have a higher R and lower G and B,
just for instance.
And so if you select out the region that you want,
if you say that you want to start cropping here,
at this, I don't know, 90, 37 X and Y location.
Well, for here, I said 97, 35.
But this IM crop function,
then it also, you have the starting point X and Y,
and then the second two are the size.
So if you want to have a 40 by 40 pixel cropping region,
then you just do 97, 35, 40, 40,
which we can do here as well.
35.
And then you can show that.
And so I messed up the cropping region,
but that would be a region of the original image.
Yeah, that's a region of the original image,
even though I messed up the region.
But that's one simple example
of how you can manipulate images.
And so in addition to cropping,
you can also resize them.
This is a 256 by 256.
So if you wanted it to be 128 by 128, for example,
you can just resize it.
Or if it's, you can get the intensity values
and a lot of other different things
that you can do with images.
And so, like I said,
Matlab has a lot of demo images
that are already given to you
when you download this image processing toolbox.
And as I used in the function,
in the command window just now,
I just said A equals IMREAD image name,
and then IMSHOW, the A,
which is just the information
that you get out of reading the image
with this IMREAD function.
And so if any of you wanted
to look at the images yourselves at home,
you could do that.
These are the ones that they give you.
And they have TIFF images, PNG, JPEGs.
So you can look at those ones on your own if you want to.
And so one of the reasons
why they give you all of these demo images
is that in a lot of our research,

some of these demo images are used
as a method for testing your algorithm.
So for instance, if I developed a new algorithm
and I want to show everybody that it works,
then I would use one of these images,
like the cameraman one,
to show that, oh, my image processing algorithm
performs pretty well by using this cameraman image
compared to someone else's algorithm,
where they also use this cameraman image.
So it can be used as sort of a comparison
for comparing algorithms,
but also for just some also data processing,
testing of your algorithms.
You can also use these images for your own purpose.
And so these images, as I said before,
are in the image processing toolbox,
which is one of the toolboxes that you can download
along with MATLAB when you download it
from the .cio websites.
And so this is just the MATLAB page
for the image processing toolbox,
but you can do a lot of things,
like they call it exploration and discovery,
where you can see what are the properties
of different images, image enhancement.
So this one is a little bit hard to see,
but you can blur or sharpen the cameraman image,
for instance.
You can also do image analysis,
where you can select out regions that you want
according to intensity values,
such as if you have intensity values
for one region that's higher than the other,
you can select the higher intensity region
and only analyze that region.
You can also do image segmentation,
which some of these examples here,
which would be somewhat simple as an introduction,
would be they have some, especially in microscopy,
they have some images with a lot of different cells,
and it's hard to just manually count them by eye,
like one, two, three, four, five,
but you can do a segmentation,
which is where you can just segment out
each of the cell regions,
and then you can have MATLAB do the counting,
do the counting instead of trying to do it by eye.
Or one of these is also some sort of nerve network,
and so you can see just based on the intensity,
for instance, you can just show what is the nerve,
the network of the nerves just using MATLAB,
and you can also do this image registration,
large image processing,
and there's also, you can implement some of these functions
with a C and C++ on some hardware,
and so there are some tools for linking hardware
and MATLAB software as well.
And then there's also this instrument control toolbox,
where as you can see, you can communicate with instruments,
such as they have some MATLAB supported instruments
specifically, such as I think that they might have
some signal generators.
I don't think that there's any oscilloscopes,

but similar things to that.
So you can analyze the data that's coming
from your instrument in MATLAB directly,
or you could also develop some drivers for them
if they don't already have them made,
so you can use MATLAB to customize your environment,
what you're looking for in terms of the data
you get from your instruments.
And then there's also this optimization toolbox,
where you can solve optimization problems,
such as if you want to do fitting of your data to a line
or something simple like that,
linear and quadratic programming, parallel computing,
data fitting, other types of optimization,
and then nowadays they also have this statistics
and machine learning toolbox,
and so you can use some of the machine learning
algorithms in MATLAB to train the data
and to develop these networks for machine learning,
and they also have a very robust statistics package,
and so you can look at the histograms,
you can also look at probability distributions,
hypothesis testing, and you can also look at these big data,
this big data, and using parallel computing
so that you can speed up your process,
even if you have very big data.
And then there's also symbolic math,
where you can do differentiation, integration,
linear algebra, plotting of the analytical functions,
such as this fancy surface plot like the MATLAB logo,
and you can also do more precise arithmetic,
and you can also generate code for Simulink,
which is one of the tools that you can use
to link a hardware and MATLAB software,
and so in summary, MATLAB has a lot of different tools
that you can use, not just these toolboxes,
but it's very flexible in terms that you can
make your own functions and analyze your images,
or manipulate your matrices very easily on your own,
and we're going to be having a couple more MATLAB sessions
that are more geared towards our class,
where we'll look at some images, some clinical images,
and then analyze them to show you how we can use MATLAB
to analyze, to perform the medical imaging processing
methods, and so that's all I have for today.
If, I don't know if Dr. Wong has any other comments, or?
Yep, so that's all that I have for today.
Does anyone have any questions?
Yep?
Can you suggest going through any of these tutorials
before the next class session?
Can you repeat that?
I didn't hear you.
Can you suggest going through any of these tutorials
and practicing before the next class session?
Yeah, it would be helpful for you to just go over
the tutorials before the next MATLAB session,
which is not next class, but it's in a few classes,
because this was just an introductory one,
but after the introductory one,
the next ones would be more in-depth,
so if you don't understand what's going on,
you might get lost a little bit,
so it would be helpful if you go over it for yourself.

For the slides, would you post it on LMS?
Yes, I'll post these slides on LMS, too.
Okay.
Okay.