

MIT OpenCourseWare
<http://ocw.mit.edu>

18.085 Computational Science and Engineering I, Fall 2008

Please use the following citation format:

Gilbert Strang, *18.085 Computational Science and Engineering I, Fall 2008*. (Massachusetts Institute of Technology: MIT OpenCourseWare). <http://ocw.mit.edu> (accessed MM DD, YYYY). License: Creative Commons Attribution-Noncommercial-Share Alike.

Note: Please use the actual date you accessed this material in your citation.

For more information about citing these materials or our Terms of Use, visit:
<http://ocw.mit.edu/terms>

MIT OpenCourseWare
<http://ocw.mit.edu>

18.085 Computational Science and Engineering I, Fall 2008
Transcript – Lecture 33

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high-quality educational resources for free. To make a donation, or to view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu. we he doesn't he's got n e t at you

PROFESSOR STRANG: OK, so. These are our two topics. And, Thanksgiving is coming up, of course. With convolutions, where are we? I'm probably starting a little early. Oh, I am. Is that right? Yeah, OK. So, with convolutions, I feel we've got a whole lot of formulas. We practiced on some specific examples, but we didn't see the reason for them. We didn't say the use of them. And I'm unwilling to let a whole topic, important topic like convolutions, go on with just formulas. So I want to talk about signal processing. So that's a nice, perfect application of convolutions. And you'll see it. You'll see the point, and it's something that you're going to end up doing. And it's quite a simple idea, once you understand convolutions you've got it. OK, but then I do want to go on to the Fourier integral part. And basically, today I'll just give the formulas. So you've got the Fourier integral formulas that take from a function f , defined for all x now. It's on the whole line, like some bell-shaped curve or some exponential decaying. And then you get its transform, I could call that $c(k)$, but a more familiar notation is $\hat{f}(k)$. So that will be the Fourier integral transform, or just for short, Fourier transform of $f(x)$, and it'll involve all frequencies. So we are getting away from the periodic case, and integer frequencies to the whole line case with a whole line of frequencies.

OK, so is that alright? Finishing, I'll have more to say about 4.4 convolutions, but I want to say this much. OK, let's move to an example. So here's a typical block diagram. In comes the signal. Vector x , values x_k . Often, in engineering and EE, people tend to write that $x[k]$, these days. OK, so, as being easier to type. And sort of better. But I'll stay with the subscript. So that signal comes in. And it goes through a filter. And I'm taking the simplest filter I can think of, the one that just averages the current value with the previous value. And we want to see what's the effect of doing that. So we want to see, we want to understand the outputs, the y_k 's, which are just the current value and the previous value averaged. We want to see that as a convolution, and see what it's doing. OK, so that's the - and I guess that here, it's a frequent convention in signal processing to basically pretend that the signal is infinitely long. That there's no start and no finish. Of course, in reality there has to be a start and a finish. But if it's a long, you know if it's a cd or something, and you're sampling it every, thousands and thousands of times. And the input signal is so long and you're not really caring about the very start and the very end. It's simpler to just pretend that you've got numbers for all of k .

OK, then let's see what kind of a filter this is. What does it do to the signal? OK. Well, one way filters are - well, first of all, let's see it as a convolution. So I want to see that this formula is a convolution that the output y , that output of all the averages y , is the convolution of some filter, some, and I'll give it a proper name,

but with the input. And notice that, again, there's no circle around here. We're not doing the cyclic case, we're just pretending infinitely long. So OK, now I just want to ask you what is the h , I want to recognize this as a filter. So that convolution, let's remember the notation. And then we can match this to that. So remember the notation for that would be, for a convolution is, I take a sum. Of all x , of h_l . $x_{(k-l)}$, right? That's what convolution is. This is our famous, and it would be in principle the sum over l . Sum over l 's. So I the filter is defined by these numbers h , these h 's, h_0 , h_1 , so on. Those numbers h multiply the x 's, where this familiar rule to give the k 'th output. OK, let's have no mystery here. What are the h 's if this is the output?

Well, I want to match that with that. So here I see that this takes x_k , multiplies by half. So that tells me that when l is zero, I'm getting h_0 times x_k , so what's h_0 ? So what's the h_0 ? It's $1/2$. That's what this formula is saying. When l is zero, so l is zero, take h_0 , $1/2$ times x_k , ta-da. Now, what's the other h that's showing up here? This is a very, very short filter. It's only going to have two coefficients, h_0 and h_1 what? One. And what's the coefficient h_1 ? What's the number h_1 ? Now you've told me everything about h when you tell me that number. Everybody sees it. It's also $1/2$, right? Because I'm taking $1/2$ of $x_{(k-1)}$. So when l is that one, we have h_1 , the h_1 is $1/2$, times $x_{(k-1)}$. Do you see that that simple averaging, running average, you could call it. Running average, it's the most, the first thing you would think of to - why would you do such a thing? Why is filtering done? This filter, this averaging filter, would smooth the data. So the data comes with noise, of course. And what you'd like, so noise is high frequency stuff. So what you want to do is like damp those high frequencies a little bit, because much of it is not, it hasn't got information in it. It's just noise, but you want to keep the signal. So it's always this signal to noise ratio. That's the key - SNR. PSNR. That's the constant expression, signal to noise ratio.

And we're sort of expecting here that the signal to noise ratio is pretty good. High. Mostly signal. But there's some noise. This is a very simple, extremely short, filter. So this vector h , it's a proper convolution. You could say h has infinitely many components. But they're all zero, except for those two. Right, do you see it? Another way, just at the end of last time, I asked you to think of a matrix that's doing the same thing. Why do I bring a matrix in? Because anytime I see something linear, and that's incredibly linear, right? I think OK, there's a matrix doing it. So these y 's, like y_k , $y_{(k+1)}$, all the y 's are coming out. The x 's are going in, x_k , $x_{(k+1)}$, $x_{(k-1)}$, bunch of x 's. And there's a matrix doing exactly that. And what does that matrix have? Well, it has $1/2$ on the diagonal. So that y_k will have a $1/2$ of x_k , and what's the other entry in that row? I want $1/2$ of x_k , and I want $1/2$ of $x_{(k-1)}$, right? So I just put $1/2$ next to it. So there is the main diagonal of the halves, and there is the sub-diagonal of the half. So it's just constant diagonal.

Now, yeah, let me tell you the word. When an engineer, electrical engineer looks at this, first thing, or this, any of these. First letters he uses is linear time invariant. LTI. So linear we understand, right? What does this time invariant mean? Time invariant means that you're not changing the filter as the signal comes through. You're keeping a half and a half. You're keeping the formula. The formula doesn't depend on k , the numbers are just $1/2$ and $1/2$. They're the same, so if I shift the whole signal by a thousand, the output shifts by a thousand, right? If I take the whole signal and delay it, delay it by a thousand times? Clock times? Then the same output will come a thousand clock times delayed. So linear time invariant. That would be - I mean, linear time invariant is just talking convolution. I mean, that's

what it comes to if we're in discrete problems. It's just that, for some h . Now, our h deserves like, the other initials you see.

OK, that was linear time invariant. Now, the next initials you'll see will be F-I-R. It's an FIR filter. So that's finite impulse response. What's the impulse response mean? It means the h . The vector h is the impulse response. The vector h is what you get if I put an impulse in, what comes out? Just tell me, what happens here. Suppose an impulse, by an impulse I mean I stick a one in one position and all zeroes. Our usual δ . Our impulse, our spike, is just, suppose the x 's have a one here, otherwise all zero, what comes out? Well, suppose there's just x_0 is one. What is y ? Suppose the only input is boom, you know a bell sounds at time zero. What comes out from the filter? Well, y_0 will be what? If the input has just a single x , and it's one, then at that time zero, so x_0 is one. Then y_0 will be? $1/2$, and what will be y_1 ? Also $1/2$, right? Because of y_1 will take x_1 , that's already dropped back to zero. Plus x_0 , that's the bell. It'll be $1/2$. In other words, the output is this. No big deal. The impulse responses is exactly h . So you can say they've created a long word for a small idea, true. And the idea, the word finite is the important number. Finite, meaning that it's finite length, I only have a finite number of h 's, and in this case two h 's. So that's - part of every subject is just learning the language. So LTI, it means you've got a convolution. FIR means that the convolution has finite length.

OK, now for the question. What is this filter doing to the signal? It's certainly averaging. That's clear. But we want to be more precise. Well, let me take some examples. Suppose the input is all ones. All x_k are one. So a constant input. Natural thing to test. That's the constant input, that's zero frequency. Zero frequency. What's the output? From all ones going in wow, sorry to ask you such a trivial question. You came in for some good math here, and I'm just taking $1/2$ and $1/2$. So the output is all y 's equal, right? So, to me, that's telling, just to introduce an appropriate word, low frequencies; in fact, bottom frequencies, zero frequency, is passed straight through. That's a low pass filter. That's telling me I have a low pass filter. So that's an expression. That's so simple that you might as well know those words. Low pass means that the lowest frequencies pass through virtually unchanged. In this case, the very zero frequency the DC term, the constant term, passes through completely unchanged. Now, what about another input all - well, now I want high frequencies. Top frequencies. What's the the highest oscillation I can get would be x equal, say, it starts one minus one, one, minus one, so on. Both directions. Oscillating as fast as possible. I couldn't get a faster frequency of oscillation in a discrete signal than up, down, up, down. What's the output for that? So that's really oscillation. That's the fastest oscillation. What would be the output from my averaging filter for this input? Zero. At every step, I'm averaging this with the guy before and they add to zero. I'm averaging this with the guy before, this with the guy - output is, y equals all zeroes.

OK, so that confirms in my mind that I have a low pass filter. The high frequencies are getting wiped out. OK, so that's two examples. Now, what about frequencies in between? Because ultimately we want to see what's happening to frequencies in between. OK, so what's an in-between frequency? So in between x_k could be $e^{(ikn)}$, let's say. $e^{(ik\omega)}$. $e^{(ik\omega)}$. Where this ω is somewhere between minus π and π . OK, why do I say minus π and π ? If the frequency - so that's the frequency. If ω is zero, what's my signal? All ones, right? If ω is zero, everything is my all ones. This is this k . So I now have a letter for it, $\omega=0$. What's this top frequency? One, minus one, one, minus one, what ω will give me alternating signs? ω equal? π , right? $\omega=\pi$. Because if this is π , I

have $e^{(i*\pi)}$, which is minus one. So when $\omega=\pi$, my inputs are $e^{(i*\omega)}$, to the k 'th power. But this is minus one. $e^{(i*\pi)}$, to the k 'th power, and that's minus one. So that's the top frequency. And also the bottom frequency is π . And the zero frequency is the all one. And this is what happens - ah. Now, comes the point. What's the output if this is the input?

What's the output when this is the input? We can easily figure that out. We can take that average. OK, so let me do that input. Input x_k is $e^{(ik*\omega)}$. And now what's the output? y_k is the average of that. And the one before. Divided by two, right? OK, now you're certainly going to factor out, anybody who sees this is going to factor out $e^{(ik*\omega)}$, right? I mean that's sitting there, that's the whole point of these exponentials is they factor out of all linear stuff. So if I factor that out, I get a very very important thing. I get, well, it's over two, I get a one. And I get, what's this term? $e^{(ik*\omega)}$ is here. So I only want $e^{(-i*\omega)}$. OK, that is called the frequency response. So that's telling me the response of what the filter does to frequency ω . It multiplies the signal. If I have a signal that's purely with frequency ω , that signal is getting multiplied by that response factor. $1+e^{(i*\omega)}$. When ω is zero, what is this quantity? So let me call this $h(\omega)$. what. Is this factor, if ω is zero? Then $h(\omega)=0$ is? One. That's telling me again that at zero frequency the output is the same as the input. Multiplied by one. And at ω equal to π , what is this frequency response? Zero, right. At $\omega=\pi$, this is minus one so I get zero. And it's telling me again that this is the response. And now it's also telling me what the response factor is for the frequencies in between. And everybody would draw a graph of the darn thing, right? So this was simple, let me do its graph over here. So I'm going to graph $h(\omega)$.

Well, I've a little problem. $h(\omega)$'s a complex number. I'll graph the magnitude response. So here I'm going to do a graph from minus π to π . This is the picture. This is the picture people look at. This is the picture of what the filter is doing. All the information about the filter is in here. All the information is in there. So if I graph that, I know what the filter's doing. So you said at $\omega=0$, I get a value of one. At $\omega=\pi$, I get a value of zero. At ω equal minus π , I get a value of zero. And I think if you figure out the magnitude, it's just a cosine. It's just an arc of a cosine. OK, for that really, really simple filter. So any engineer, any signal processing person, looks at this graph of $h(\omega)$ and says that is a very fuzzy filter. A good, an ideal filter, an ideal low pass filter, would do something like this. An ideal filter would stay at one up to some frequency - say, $\pi/2$. And drop instantly to zero. There is a really good filter. I mean, people would pay money for that filter. Because what happens when you send a signal through that ideal filter? It completely wipes out the top frequencies. Let's say, up after $\pi/2$. And it completely saves the in-between ones. So that's really a sharp filter. Actually, what people would like to do would be to have that filter available. And then also to have a perfect ideal high pass filter. What would be an ideal high pass filter? Yeah, let's talk about high pass filters just a moment. Because this is, you're seeing the reality of what people do, and how they - and that little easy bit of math they do.

Do you want to suggest a high pass filter, let me come back to this? And just change it a little? So I plan to do not - I'm now going to do a different filter. That's going to be a high pass filter. And what do I mean by that? A high pass filter will kill the $x_k=1$. I now want the output from - this is now going to be, I'm going to change. Can I just erase, change a lot of things? I'm now going to produce a high pass filter. Sorry, π . And what's the difference? When all x 's are one, the output is going to be? Zero. And when I have the highest frequency, the output is going to be? The input.

And what am I - and then in between, I'll do something in between. OK, what do you think would be a high pass filter, like the simplest high pass filter we can think of? Anybody think of it? You're only getting, like, 15 seconds to think in this class. That's a small drawback, 15 seconds. But, the high pass filter that I think of first is, take the difference. Take the difference. Put minus $1/2$ s on the sub-diagonal. This is the same, this is also a convolution, but now what h_0 is still a half. But now h_1 is? Minus $1/2$. We're still convolving. We're still convolving with - it's still linear time invariant, that just means it's a convolution. It's still a finite impulse response. But the response, the impulse response is now $1/2$ minus $1/2$. So what happens if I, in my picture over here, if I send in any pure frequency, I'm now doing minus $1/2$ here. So I'll just keep the plus. But I'll also add in the minus. So now I'm looking at $1 - e^{(-1 * \omega / 2)}$. And again, let's plot a few points for that guy. So what, at x at ω - so this is ω in this direction. And this is h in this direction. So at $\omega = 0$, what's my high pass guy? When I send in a zero frequency, constant, I get what output? Zeroes, because now - I'll call it a differencing filter.

So I'll just, instead of averaging I'm differencing. OK, so now for this one, maybe I'll put an x to indicate I'm now doing, I'll do x 's for the high pass. So this now, the high pass guy, kills the low frequency and preserves the high frequency. And you won't be surprised to find it's some cosine or something that, well yeah, it's got so sorry that's not much of a cosine. It's the mirror image of the low pass guy. And maybe the sum of squares adds to one or two or something. One, probably. The sum of squares probably adds to one. And they're kind of complementary filters. But they're very poor. Very crude, I mean that's so far from the ideal filter. So how would we create a closer to ideal filter? Well, we need more h 's. With two h 's, we're doing the best we can, with just h_0 and h_1 . With a longer filter, for which we're going to have to pay a little more to use, but we'll get a lot more. We'll get something, we could get a filter that stays pretty close to this, drops pretty fast. There's a whole world. Bell Labs had a little team of filter experts. Creating, and now MATLAB will create it for you, The coefficients, h , that would give you a response, a frequency response, it'll stay up toward, up close to one as long as possible. And drop as fast as possible, and bounce around there. So next week, if I come back to that topic, I can say a little more about these really good filters.

What was I trying to do today? Trying to see how convolution is used. And this is a use you will really make. So now I just have, I think, about two more things to say about this example. Let's see, what are they? Well, first, so all the information is in this $h(\omega)$. Oh, yeah. This simple example gives us a way to visualize convolution. And I think we need that. Right? Because up to now, convolution has been a formula. Right? It's been this formula. That's the formula for convolution, and how do I visualize that? Just think of, may I try to visualize that? Here I have, this is the time line. The different k 's. k equals zero, one, two, minus one. And I have x_{-1} , x_0 , x_1 , x_2 , x_3 . So that would be a little bouncy up and down. And the averaging filter, let me go back to the averaging one. The averaging filter would smooth out the bumps. Because it would take the, like, average neighbors. And that's a smoothing process. As we see here, it's a process that kills high frequency. Now, what is this visualization I want you to think of? I want you to just think of, like, a moving window. So here is the input. Now, I move a window along. And that window, so let's say here's the window, I should have another. So that's the window. When the window is there, it takes the average of those two. That gives me the new output. Now, think of the window as moving along here, taking the average of these. Move the window along, take the average of these. Move the window along. Do you see the sort of, this is what a convolution is doing.

This is a picture of my formula, sum of $h_k * x_{(l-k)}$. So the window is the h's, is the width of the h's, and as that window moves along. I mean, you could write, you could create, design a little circuit that would do exactly this. That would do the convolution. You just have to put together some multipliers, because you have these h's, these, like, halves. And you have to put in an adder, that'll add the pieces. And those are the essential little electronic pieces of an actual filter. Then you just move it along. So it needs a delay. That's about the content of a filter. Is multipliers that will multiply by the h's, so in come the x, multiply by the h's, do the addition, and do a shift to get onto the next one. You see how a filter works? I think that image, or convoluted is a little bit vague, maybe? This window moving along? But it's quite meaningful. And then the final thing I'll say about filters is this. That, what's the connection between $h(\omega)$ and $h(k)$? Or h_k , let me call it h_k ? What's the connection between the numbers in impulse response? Just, which were the h's, and the function, which is the frequency response? Which tells me what happens to a particular frequency? Each frequency, e to the - you notice how the frequency that went in is the frequency that comes out. It's just amplified or diminished by this $h(\omega)$ factor.

So you see the h's are the coefficients here of $h(\omega)$. In other words, $h(\omega)$ is the sum of the h_k 's, e to the, e^{-ik} . Here's the beautiful formula. That's obvious, right? Here you're seeing the formula in the simplest case, with just an h_0 and an h_1 . But of course, it would have worked if I had several h's. So this $h(\omega)$, this factor that comes out, is just this guy. Now, if I look at that, what am I saying? I've seen things that connect a function of ω with a number of filter coefficients. I saw that in Section 4.1, in Fourier series. This is the Fourier series for that function. Right? You might say, OK, why that minus? I say, it's there because the electrical engineers put it there. They liked it. And the rest of the world has to live with it. So, you notice I don't concede on i. I refuse to write j. But they all would. I speak about them, but probably some you would write j. So I'm hoping it's OK if I write i. i is for imaginary. I don't see how you could say the word imaginary starting with a j. And what was the matter with i, anyway?

Current. Well, current used to be i. I mean who, is it still? Well, let's just accept it. OK, they can call the current i, and the square root of minus one j, but not in 18.085. So OK, here we are. So my point is just that we have a Fourier series. Here we have a 2π periodic function. Here we have its Fourier coefficients. The only difference is that we started with the coefficients. And created the function. But otherwise, we're back to Section 4.1, Fourier series. But that fact that we started with the coefficients and built the function, sometimes you could say, OK that sounds a little different from the regular Fourier series, where you go the other way. So people give it the name discrete time Fourier transform. You might see those letters sometime. The discrete time Fourier transform goes from the coefficients to the function. Where the standard Fourier series starts with a function, goes to the coefficients. But really, it doesn't matter. The point is, yeah. So you could say, maybe we have now a force transform. Like the first transform was Fourier series. The second one was the discrete. The third one is the Fourier integral that's coming in one minute. And the fourth is this one. But hey, it's just the coefficients and the function have switched places. In the, which one is the start and which one is at the end. OK, let me pause a minute because that's everything I wanted to say about simple filters. And you can see that this is a very simple filter, and could be improved. Better numbers would give, I mean what would be better numbers? I suppose that $1/4$, $1/2$, $1/4$ would probably be better.

If I took those numbers, I'm pretty sure that this thing would be closer to ideal by quite a bit. If I plotted - so what do I mean by, those are the h's. So I would take $1/4$ plus $1/2 e^{-i\omega}$. Plus $1/4 e^{-2i\omega}$. This would be my better $h(\omega)$. This would be my frequency response to a better averaging filter, sort of this is like averaged averaged, right? If I do a half an average and then I do the average again. In other words, if I just send these signals y_k through that same averaging filter, so average again to get a z_k . I think probably the coefficients would be $1/4, 1/2, 1/4$, and it would be, I've taken out more noise. Right? Each time I did that averaging, I damp the high frequencies, so if I do it twice I get more damping. But I lose signal, of course. I mean, presumably there's some information in the signal in these frequencies. And I'm reducing it. And if I average twice I'm reducing it further. So a better one would be to get a sharp cutoff. OK, that's filters. I guess what I hope is that we have the idea of a convolution, and now we see what we can use it for. Right? And there are many others. So we'll have another. We'll come back to convolutions and de-convolution. Because if you have a CT scanner, that doing a little convolution. I mean, you're the input, right, to the CT scanner? You march in, hoping for the best. OK, CT scanner convolves you with their little filter. And then it does a deconvolution, to an approximate deconvolution, to have a better image of you. OK, let's leave that.

Can I change direction and just write down the formulas for the Fourier integral transform? And do one example? OK. I don't know what you think about a lecture that stops and starts a new topic. Is it - maybe it's tough on the listener? Or maybe it's a break. I don't know. Let's look at it positively. Alright, break. Alright. So let me remember the Fourier series formulas. So I'm just going to break, and now we go to the integral transform. OK, so let me remember the formula for the coefficients, which was $1/2\pi$. The integral of $f(x)e^{-ikx}dx$, right? And then when we added it up to get $f(x)$ back again, we added up a sum of the c_k 's e^{ikx} 's. That's 4.1. We know those formulas. And we notice again. Complex conjugate. One direction is the conjugate compared to the other direction. Now, all I plan to do is write down the formula. And remember, I'm going to use $\hat{f}(k)$, instead of the coefficients. Because it's a function of k , all k 's and not just integers are allowed. And then I'm going to recover $f(x)$. OK, now this integral went from minus π to π , because that was periodic. But now all the integrals are going to go from minus infinity to infinity. We've got every k , every x . So we take, what do you expect here? $f(x)? e^{-ikx} dx?$ Yes. Fine. Same thing, $f(x)$ is there, but now any k is allowed so I have a function of all k 's. And now I want to recover $f(x)$. So what do I do? You can guess. I've got an integral now. Not a sum. Because a sum was when I had only integer numbers. Now, I've got $\hat{f}(k)$, and would you like to tell me what the magic factor is, there in the integral formula? It's just what you hope. It's the $e^{ik\omega}$. Where d what? Now this is where it's easy to make a mistake. d , I'm integrating here. I'm putting the whole, reconstructing the function. I'm putting back the harmonics with the amount $\hat{f}(k)$ tells me how much $e^{ik\omega}$ there is in the function. I put them all together, so I integrate dk . I'm integrating over the frequencies. This was the sum over k . From minus infinity to infinity. Now this is an integral, because we've got, it's all filled in. And it remains to deal with this 2π . And I see in the book that the 2π went there. I don't know why. Anyway, there it is. So let's follow that convention. Put the 2π here. So there's the formula. The pair of formulas, the twin formulas. The transform, from f to \hat{f} , and the inverse transform, from \hat{f} back to f . And it's just like the one you've seen for Fourier series.

Well, I think the only good way to remember those is to put in a function and find its transform. So my final thing for today would be take a particular function, $f(x)$. Here, let me take ever $f(x)$ to be, here's one. f to be zero here, and then a jump to one. And then an exponential decay. So e^{-ax} . OK, so that's the input. It's not odd, it's not even. So I expect sort of a complex \hat{f} of k , which I can compute. So \hat{f} of k is what? Now, let's just figure out \hat{f} of k and look at the decay rate and all the other good stuff. So what do I do? I'm just doing this integral. For practice. OK, so f is zero in the first half. So I really only integrate zero to infinity. And in that region it's e^{-ax} , and I multiply by e^{-ikx} , and I integrate dx , and what do I get? I'll get this, is an integral we can do, and it's easy because this is $e^{-(a+ik)x}$. You're always going to see it that way, right? That we're integrating. And then the integral of an exponential is the exponential divided by the factor that will come down when we take the derivative. So I think we just have this, right? Don't you think, to integrate that exponential, we just get the exponential divided by its little factor. And now we have to stick in the limits. And what do I get at the limits? This is like, a fun part of Fourier integral formulas. What do I get at the upper limit, x equal infinity? If x is very large, what does this thing do? Goes to zero. It's gone. The e^{ikx} is oscillating around, it's of size one. But the e^{-ax} , so I needed a to be positive here. That picture had to be the right one. A positive. Then at infinity, I get zero. So now I just plug in this lower limit, that comes with a minus sign. So what do I get? The minus sign will make this $a+ik$, and what does it thing equal at $x=0$? One. e^0 is one. So there is the Fourier transform. Of my one-sided exponential.

Now, just a quick look at that and then I'll do some more, this example's Example one in Section 4.5, and we'll do more examples. But let's just look at that one. I see a jump in the function. What do I expect in the decay rate of the transform? So a jump in the function, I expect a decay rate of $1/k$. Decay rate, right? Just as for Fourier coefficients, so for the integral transform. So a decay rate in \hat{f} . And it's here. $1/k$ in the denominator. Yeah. So that's a good example. You might say, wait a minute, OK that's fine but what about the second one? Could I put in $1/(a+ik)$ and get back the pulse? The exponential pulse? The answer is yes, but maybe I don't know how to do that integral. So I'm sort of fortunate that these formulas are proved for any function including this function. So this example shows the decay rate. The possibility sometimes of doing one integral but maybe the other integral going the other direction is not so easy. And that's normal. So that's, like, Fourier transforms and inverse transforms, we don't expect to be able to do them all by hand. I mean, I'll just say that actually, anybody who studies complex variables and residues, I don't know if you know any of this, heard these words about, there are ways to integrate. I could put $1/(a+ik)$ in here. And, actually, and do this integral for minus infinity to infinity. By stuff that's in Chapter 5. Can I just point ahead without any plan to discuss it. That some integrals can be done by $x+iy$ tricks, by using complex numbers. But I won't do more. OK, thanks. So that's the formulas. And that's one example. Wednesday there will be more examples and then no review session Wednesday evening.