MIT OpenCourseWare
http://ocw.mit.edu

18.085 Computational Science and Engineering I, Fall 2008
Transcript – Lecture 31

PROFESSOR STRANG: So, thanks for coming today. This is a key lecture in the application of Fourier, you could say. So convolution is the big word. And a major application of convolution is filtering signal processing. So we'll develop that application. But it's nothing but convolution. So the key idea is these convolution rules, where they come from. And what these new symbols, that's the symbol for the convolution of two functions. They could be functions here. Here they're long vectors of coefficients, and so these are the rules. So it's just a little bit of algebra. But it just is so central to all this subject. Signal processing is certainly the most important little thing to know. That if I multiply two functions, so that's where we started last time. If I have a function f with Fourier coefficient c, and a function g with coefficients d, then, oh, wrong. I convolve the coefficients, right. So f has coefficient c, g has coefficient d. So if I multiply the functions I definitely do not just multiply each c times the d. I do this convolution operation, which we have to remember. That's our main first step is to remember what that was about. And then this is the other direction. So we didn't see this before. That I mean, there's always this fantastic symmetry between physical space and frequency space. And convolution in one is multiplication in the other. It's so easy to remember. If I multiply in one space, I do a convolution in the other space. If I do a convolution of functions, I do a multiplication of coefficients. So that's the rule to know. And now if we expand on it, by sort of seeing again what it means. So let me do a quick repeat of this first step to remember what this symbol, convolution symbol, means.

And then another thing I have to do. Here, I'm talking about the infinite case, functions and with a whole infinite sequence of coefficients. I've got to do the cyclic case, too. Which goes with the discrete transform. OK, but let's start with the infinite case as we did last time. Maybe, here's something. Here's a suggestion. We had f(x). We started with f(x), as the sum. And remember to have nice formulas, we're doing the complex version. c_k*e^(ikx). Let me suggest something. Let me write z for e^(ikx). So I'm going to just write that at the sum of c_k*z^k. z to So z is e^(ix). This is like a good point to make anyway. When we have this e^(ikx), it's natural to think of that as a complex number on the unit circle. Right? That's always the message, e to the i real is on the unit circle. Absolute value one. And this is great for periodic function. Because if these functions have period 2pi, and if we think of our function as being on the circle, it obviously has period 2pi. I mean, the picture says, yes. If I go 2pi, I come back. So x is the angle, right? x is the in this picture, a little bit unusual maybe, x there would be the angle. And it's just a little bit easier to write. And maybe it even has an official name, the z transform. How do you like that? You learn a transform just, in fourteen seconds. z transform. It'll make it easier to multiply by g, so g is going to be the sum of d_k*z^k. So just think of these as long, well I'm tempted to say, long polynomials. I mean, very long, because they can be infinite series. Infinite negative powers and positive powers. But just think of it as a bunch of powers of z times a bunch of powers of z. And if you multiply a couple of polynomials, you're doing convolution. I guess my message is, you've been doing convolution since the second grade. That's the real message.

Here, let me show you. This is convolution, too. Suppose I have to multiply a 123 times 456? OK, so what do you do? Remember back, it's a long way back but we can do this. One, two, three times four, five, six. So I multiply the three, oh there's a little point. Where that second-grade teacher's going to panic. I'm going to write that as 18. And that's 15, and that's 12. Sorry about that, yeah. And 12, ten and eight, right? Four, or five and six. So you see the nine multiplications that you have to do? Nine multiplications, three times three. OK, right now imagine those were, they could have been longer. But they were finite length filters, we could say. And now, what does convolution do? Just what you did in multiplication. When I add 12, 10 and 6, what am I doing? I'm putting together the three times the four, the two times the five, and the one times the six. That's what convolution, those are the things that convolution puts together. So we got an 18, a 27, 28. 13, and four. So I guess I'm saying that, alright, here's what I really want to say. I want to say that the convolution of those, with these, four, five, six, is this sequence here - oh. Yeah, that's right. Four, 13, 28. 27, and 18. If you just look at that. Where did that 13 come from? Let's just remember, where does that 13 come from? That came from, this was z^0. This 13 is 13z to the first power, right? We're just checking all the powers here. There's 13z to the first power. Where do we get a first power? We get a z^0 times $5z^1$. So that's $5z^1$. And we also have $2z^1$, times $4z^0$. Right? Two times four gave the eight. So the eight from there and the five from there produce that 13. And that's just what you did in multiplication. Right?

So that's multiplication of two series. That's not cyclic. This is definitely not yet cyclic, and but we'll make it cyclic in a minute. This is the infinite one, except that we had all zeroes beyond. OK, so that if you in non-cyclic convolution like this, if I have length m and length n, then I get length m+n, maybe m+n-1. OK, so that's convolution. Without carrying numbers. Without doing it right. OK, so and what does that correspond to? Let me just, so you see it every way. That corresponds to $1+2z+3z^2$, multiplying $4+5z+6z^2$. And it gave, that's times. And it gave this thing up to $18z^4$. Just, exactly the multiplication that you've always done. OK, so that's the idea. Over on that board I'm going to put a formula for this convolution operation. But my point on this board is, you've done it always. When you multiply a couple of polynomials, you collect powers. And that's all convolution is doing, collecting each power separately. OK, let's do it. So then f(x), g(x), is, when I multiply that polynomial or series by that polynomial, I get some polynomial in, with coefficients, oh I was changed to l, just to have a different symbol there. And what was the formula for l? For h_l? What is the coefficient of z^l? If I multiply that by that, do you remember the story there? When I multiply that by that and I looked for the terms that gave me z^l? OK, well that means that this power times this power is going to be z^l. So that the index, do you remember what happened? It was a lot of different, just the way this h_l is here. Here's h_2 or something. I've got to do an addition. Because a bunch of c's come in with different d's, and what's the deal then? c_k comes in with which d? This is the magic number there. What's the subscript that if I look at the coefficient of z^l, I look at each of these. And then I pick out the one of these that will give me a z^l. And which one is it?

$d_{(l-k)}$. It's that magic quantity that the eye spots perfectly. k and l-k, adding to l simply because z to the k is $z^{(l-k)}$, multiplies to z^l. Same thing. Right? OK, this is, now I'll use that notation. This is c convolved with the d. c convolved with d is h. c convolved with d is h. So this is the l'th component. c convolved with d is my symbol. This h is the convolution. And that's the convolution rule. It's just whatever operation you have to do to get the right answer. The right answer when you multiply. So that one. Ready for the discrete case? The finite case? The case when, you have power, when z becomes w. The discrete case. The cyclic case, sorry. Maybe emphasize the cyclic case, meaning it circles around, is the case when z becomes this very special z, on the unit circle that we know as w. OK, and we have to say, and of course it's w_N, I have to tell you in the cyclic case how long the cycle is. So this would be a case.

Watch what you do here to make this cyclic. OK, I have three inputs, so N is three here. Now I'm going to do the cyclic. So instead of z's, I should be putting w's. I will. Just to emphasize, it's good to think of it with the w there, because the w has this special property that's critical to everything, OK so now I'm going to, I think of this is as 1w^0, 2w^1, and 3w^2. I'm thinking of the same multiplication here, but now w's. OK, so I'll end up with 18w^4, and four was the constant and 3w's and so on. All these numbers. What's the difference? Ready for the key point? Now, what's happened in this cyclic case? Well, the difference is what is w^4? If we're in the cyclic case, N is three now, our guys have length three, our circle is w now is 1/3 of the way around. So that's my w, here's my w squared. Here is my one. But here it is also w cubed. So w is the same as w^4, w^2 is the same as w^5. So, what's the difference? What can I do now? If I'm in this discrete case, then my inputs are a vector of length N, three. A vector of length N, three. And I want to get out to a vector of length three. I'm not happy with that in the cyclic case, because I'm not happy with w^4. So now tell me again that last, when I do the multiplication and I just do it, there's no difference except in how I write the answer. 18w^4 is the same as? w. w^4 is the same as w, when N is three. So that 18w^4 cycles back in, with this 13. So now if I do the, can I show you the symbol? I'll just do a little circle there. To say this is now the cyclic convolution.

Then this isn't the answer any more. The answer now for cyclic convolution, I only want three numbers. And if you can tell me what those three numbers are, we've got it. Move those over a little to make room for the three numbers. So there's the answer, the space for the answer. What do I write in? How many, what's the constant term? It's 31, right. It's 31. Where did 31 come from? It came from, you could say, cycling that 27w^3 back with the four. Because there's no w^3 is the same as one. So I've gone around the circle when I come around to w^3. So that 27 and four combined into that 31. And let's see so so what multiplications am I doing here? One times four gives me the constant. Two times six, that's 2w's, and six w squareds, that's 12w^3, that's 12. And then 3w^2, and five w's is 15w^3. But that's the same as 15. So that's why we get 31. We got four, we've got 12 and we've got 15. OK, now what's the second component, the w component of the cyclic convolution? Tell me what number do I write in, in that middle position? How many w's do I have? 31 again. This 28, this 18 is coming back by three. To 13. You see, I could have done that multiplication. So coming back to 31, am I going to get another 31? No, what's the w^2 guy? 28. Yeah. 28, because there's no to the fifth to come back. So 28 uses three multiplications. The 31 there used the four and these two. This came back over to here. And this 31 used, this 18 came back. I could have put the 18 here. You know, I could have lined it up just three. So I'll write a formula for it. So that's the answer. 31, 31, 28.

Could I just suggest a little check on that? Just to check on the numbers. I think that if I add up these numbers, I get six. And if I add up those numbers I get 15. And if I multiply that, I get 90. And if I add those numbers I get 90, right? So those add to 90. So I'm just saying the miracle check is add these, multiply by the sum of those. And you get the sum of those. Why is that? Somehow seems right, doesn't it? Because somehow I've taken all nine products here, and so when I add all the results, I'll have the sum of all nine of these possible products. So I'll have six times 15. Actually, here's a good way to look at it. In doing this, I just set $w=1$. I just set $w=1$ in the polynomial. When I set w to one, this becomes six, this becomes 15 and the answer becomes 90, when w is one. Yeah. So that's another way to see. And actually, this multiplication, the second grade version, had w, well, I'm almost going to say $w=10$, but not quite that. Because it's written in the opposite order, right? If w was ten, this would be one. Well, anyway, w can't be ten, it's got to stay on the unit circle, so. So somewhere in the non-cyclic case, it's something like $w=10$, or $w=1/10$, maybe. Whatever. OK. Could you take the convolution now, let me give you just another example. Do it mentally. What's the convolution of , let me make a little longer.

So I take the, first of all, the non-cyclic convolution of . OK. What's the ordinary - how long is the answer now? This is just practice. How many components am I going to have in the ordinary convolution of those two guys of length four? I think it's seven. I think it'll be seven. Because we'll have here one, we have no, we have it z^0, z^1, z^2, z^3. And here we'll have again the same, it would go up to z^6 but remember there's a z^0, so that's why we have seven. And what will it be? What will it be? I guess, actually, this wasn't a brilliant example, was it? But let's finish it. So I'm just multiplying z by z squared, so what do you get for an answer? I think the one shows up in the z^3. Is that right? Yeah. z^1, here z^2 here, z^3 here, and we always have to remember everything in Chapter 4 starts at zero. Zero, z^0's the first one. OK, that's not too great an example, because what happens if I do the circular cyclic convolution? What would be the cyclic convolution? Now I'm expecting four guys only, right? A cyclic keeps the same length. And what would be the answer? Well, there's nobody to fold back, so it would be just . So let me update this a little bit with one here. OK, just to practice. So suppose I do that convolution. Un-cyclic, first. What do I change here now? I've now got a z^2 and I've also got a z^3, but I only have a single one there. Let's make it a little more interesting.

OK, make it like so. Alright, $z+z^2$ is what we're looking at, $z+z^2$ is multiplying z^2+z^3. And in the long form, what do I get? Let me make space, tell me what numbers to put in. If I multiply $z+z^2+z^3$, I get what? $1z^3$, how many z^4? Two of them. How many z^5? One, and nobody there. OK. And now the cyclic version would be what? What's what's my answer now for the cyclic version? Let me take those out. So the cyclic version would bring the two back to the zero. Would bring that one back so there. That zero will still be zero. And I checked that I haven't missed anything by adding those up to get four, and adding this up to get two times two. Yeah. OK, so that's the rule. And it's a lot cleaner to see these answers than to see this formula. And I need, actually of course, now mentioning that formula, I need a cyclic formula. So can I write above it the cyclic formula? What do I get when I'm, instead of this sum, which went from k equal minus infinity to infinity, in the cyclic case, h_k is just going to be a sum from zero to N-1, and there'll be a c_k, and a d something. And now this is the cyclic case, so I guess this makes us, I think what our situation now is we understand the cyclic case from examples. And now we just have the job of how do I put it into algebra. How do I put it into symbols? What's the point? c_k d_n, let's say. But oh no, I'm looking for h_l, yeah. So what's the deal?

Here l was k. That one is the sum of that one and that one. So here, l is the sum of k and n, but. What's the but? I mean somehow I've got some wraparound to do, right? When I'm doing the cyclic multiplication and I'm doing the wraparound because w^n, the wraparound comes from that, right? That's why I never get as high as n, because when I get to n I go back to the zeroth power.

OK, so what's the relation of k and n and l here? We just need the right word to express it. What's the word? Mod. So that's the word I'm looking for. $k+n$ is l. With wraparound and wraparound means that the nice notation that people use is a mod m. Let's practice. What is two plus two mod seven? Four. Two plus two is four, even in 18.085. Right, OK. But two plus two mod three is, two plus two mod three is? one. Everybody sees it? I'm taking z^2 times z^2, z^4, but I'm doing with $N=3$, so z^3 is on3, so that z^4 is really just z to the first power. So this is the little nifty notation that says make it cyclic. Bring it back so that l only has the values here, zero up to $N-1$. And then stops. OK. OK, so we'll have more practice with examples when we do some filtering. Have you got that fundamental, so we've talked about this rule one here. f times g goes to those coefficients. And if it's the cyclic case then I put a circle around that star. And I do the wraparound. But it's just, it's the z transform, it's polynomials in z or polynomials in w, and when it's polynomials in w, you use that special property that w_N is one. Yeah, OK.

Now, I see I've written another line, there. That I could convolve functions. Let me do a couple more examples. Couple of examples. First, before I go to that line, OK. So I'm up to this line. A couple of examples here. Let's see, what example would I want to do? Let's see, OK, I want to do one example with a delta function. One example with a delta function. One example with the delta vector. Yeah, let me take the function $g(x)$ identically one. OK, constant function. In this rule. I want to see what happens with the rule. OK, then $f(x)$ $g(x)$ is the same as $f(x)$, right? Because this function $g(x)$ is so simple, it's just one. Now, what about the coefficient? So I have the coefficients of c, what are the Fourier coefficients, what are the d's? Ah yes, what are the d's? So I'm testing my rule on a really really simple case, $g(x)$ identically one. What, you have to tell me, in order to check the right side of the rule you have to tell me the Fourier coefficients for that very special function. What would be the Fourier coefficients? If I expand the function one in a Fourier series, what do I see? I see a one. Yeah, that's it, I see one. So what are its coefficients? d_0, right, is one? And the other d's are all zero, right? So my vector of d, my vector of d's is a whole lot of zeroes on the negative side. A one right there in the center, and then a lot of zeroes. And now I want to convolve that with c. I'm practicing the convolution rule on a case that's so simple it's confusing, right? I mean, it's a big mess, this multiplication.

What do I get out of this? If d is this vector, if d has this property that d_0 is one and others are zero, all others are zero, so this is my little example, what does this sum boil down to? Well, I only get something when $l=k$, right? I only got something when $l=k$, because then I have d_0 and that's the only d that's around. So in this sum, something happens only when k and l are the same. And then what happens? Then I have a one, I have c_l, and that's h_l, so that's all I'm concluding then. That this h is the same as c. I'm sorry, it's so dumb. My point is that in convolution, this is the thing that acts like one. Because in multiplication, that's the thing, that's the function that acts like one. That's the function that is one. So this is the one in, oh, would you allow me to do this? I'm going to create a matrix with these d's. There's another way to see convolution. Yeah, there's another way to see convolution and discrete convolution. Maybe the discrete one's the better. Yeah can you stand one more way to write the formula? One more way to write, now I'm going to do, I'm going to do discrete convolution. Discrete cyclic. Just So how am I going to write it? I'm going to write it by a matrix multiplication. Because you know that in this course a matrix was going to show up. So it's going to be a matrix multiplication. So I just have to tell you the matrix, so this is going to be some matrix. Let me take N to be four. So then I have, you watch. So I have four d's, and the output is the four h's. And the rule I'm following is this rule. Is this, the same old rule but with the cyclic part.

And now I want to show you the matrix that'll just do this. Look, I've put the c's in the first column. And then I go, yeah, here's another. So it's a cyclic matrix. So let me finish it up. It's going to be four by four, it's going to be cyclic. So I have a c_0, c_0, c_0, c_0 on the diagonal. That's fine, that's because z to the zeroes multiplying all the d's and leaving them in place. And then I have c_1's, and then I think I come around again here for a c_1. And I have c_2's, see where see c_3, c_2. And I come around again, to a c_2 and a c_2. And c_3 comes around to a c_3, a c_3 and a c_3. Well, can you, I hope you can see, this is cyclic matrix. It's only got one, it starts with a vector c, and those are on the diagonal and the diagonals wrap around. That's the other word that you often see when you see the word cyclic, wraparound. It's because you think of a circle. If you go the second time around, it's wrapped around the first time. OK, just can you look and see that this is the right formula for h_0? h_0 is c_0*d_0. Where does that come from? Remember, h_0 is the coefficient of z^0 in the answer. So it comes from c_0*d_0 to the zeroth power in the input. And then why is there is a c_3*d_1? Why is there a c_3*d_1, and then a c_2*d_2 and then a c_1*d_3 all piling up into h_0? Tell me now, why is there a c_3*d_1? Because we're doing mod four is one way to say it. Three and one add to four. Because c_3 is the w cubed guy, and d_1 is the coefficient of w^1 t and w^3 times w^1 piles back into the constant. And you see the pattern of that matrix? So these matrices are very important. So they circle around.

Oh, we've actually met a matrix of this type, the first day of 18.085. What was that matrix? It was one of our four great matrices. And now here it is back again. Which one was it? Well, you remember the letter for it. Which isn't going to change. And do you remember the particular matrix? Well, everybody does remember that matrix, right? Twos were on the diagonal, minus ones were on the diagonal, and the diagonal curve continued. Minus one was on this diagonal and that, continued and zeroes was on this diagonal. So this is cyclic convolution, the circulant matrix, cyclic convolution by c, what's the c that produces that convolution matrix? It's just, it's got - well there it is. The first column is it. Right, right. And somehow I would say that that's an even vector. It's sort of, I associate it with cosine. It's an even vector. Here is the zero term, and then these are the same, not to worry about that part. Do you see that we've seen that matrix before? And the cyclic convolution means you take its second differences, of course. We're taking second differences, but our everything in our world is cyclic. So the result, the x_4 is x_0. So we're taking second differences - well, maybe I should say d - we're taking second differences d_i, $2d_i$'s, $-d_{(i-1)}$, $-d(i+1)$, I don't know if this is. So there's a minus one, two, minus one. And we're cycling around so that d_0 is d_4. And d_1 is d_5, and d_{-1} is d_3, whatever. OK, I'm just reminding you, we've seen these before.

OK, so this is another way to remember the formula. OK now can I ask you a practical question? A practical question. Let me bring back this second grade multiplication. Well, I have a granddaughter named Elizabeth, I'll have to admit I didn't think about mentioning Elizabeth. She's six. And she delights in sending me long multiplications. I mean, really long. And then every time I talk to her on the phone, she says have you done that one yet? And I say, I'm working on it I've got MATLAB at work. Because they're ridiculous and I haven't figured out how to tell her. I mean, she just writes page after page. Times 100, plus three, minus seven, just whatever she things of. OK, now I need help from the convolution groupal, here, actually. So let's suppose that Elizabeth has given me a multiplication in which I have a thousand digits times a thousand, right. Which Mathematica is prepared to do exactly, right? MATLAB will mess up, but Mathematica and Maple and symbolic packages will do exact computations. So what would be the right way, well let may make it 1,024. 1,024 digits times 1,024 digits. Let's do the cyclic version first. Elizabeth doesn't know about cyclic. Maybe I could teach her that. That'll keep her busy while I'm doing the multiplication. OK, right. Only her older brother would explain it to her, that's the trouble. OK, so how am I going to do, or how are you going to do on the quiz, multiplication of a 1,024 digits times 1,024? And I'll make it easy by making it cyclic, so I just want 1,024 digits in the answer. OK. How would you do it? Well, before today, you would have just multiplied, right? You would have written down 1,024 two, lines of 1,024, done an addition. And you would have had a million multiplications to do. But how would you do it now? Apart from giving it to Mathematica. What's a faster way to do it? What's a faster way to do a convolution?

The fast way to do a convolution is to use the convolution rule, go this way. So take these numbers, these 1,024 numbers, in c and these 1,024 numbers in d, and, well what do I have to do? I want to use the convolution rule, because multiplying is fast. Now I've got functions. But I'm in the cyclic case. So I'm in the cyclic case, so what should I do? How can I change this to be the cyclic case? This is like f, j, g, g, j. So multiplication of components of things in function space is convolution of coefficients. So now, this is the cyclic. So let me make it cyclic. So again, what's your problem? The problem is to do this cyclic multiplication. What's the idea? The idea is to transform c back to f, to transform d back to g. Do the multiplications, now I have only 1,024 multiplications. Not 1,024 squared. That's the point. And if I do this directly, I've got 1,024 squared multiplications to do. Much better. Transform back to here, do just 1,024 - what's the MATLAB command for that, when you're multiplying each component by itself? It's not the dot product, notice. It's not the dot product because I'm not summing. Do you know the MATLAB command, if I have a sequence of numbers of vector f, of length 1,024 and I want to get that result? What's the result? It's a vector of length 1,024 that takes each f times its g. But doesn't do any adds. That's what's there. What's the MATLAB command for that? Dot, yeah. Dot star, right. So that dot says component by component.

OK, so what's the plan here? I do c's back to F. By the Fourier matrix. d back to g, by the Fourier matrix, then I do a very quick multiplication. And then what? Then I mustn't forget. That I'm in frequency space, and what do I have to do? I've got to get back into coefficient space. So I do the inverse transform - here's the formula, then. I'm doing the inverse transform of F, so the transform of c dot star, the transform of d. To get c, d. Is that right? So I took c, and I got back into the function. I took d, and got back to its function, with the Fourier matrix. OK, I'm in the Fourier and now I'm in this space. I've added up coefficients to get in this space. Now I do the dot star, the fast one. And then I transform back. So why is that faster? Than just doing it? Because what's the cost of F times c? And how am I going to do that? I'm going to do with the fast Fourier transform, right. That's the point. I can multiply by F, or by F inverse faster so I have three of these transforms. I've got to get two guys into the other space, and the answer back out. So I have sort of three of these n log n, but that will easily be n squared. Right? So if you have a

OK, so. Pretty full day again. I had last time introduced the Fourier matrix, the discrete Fourier transform. Well, more strictly, the discrete Fourier transform is usually this one. It takes the function values and produces the coefficients. And then I started with the coefficients added back, added up the series to get the function values. So F or F inverse. So we didn't do examples yet. And one natural example is the discrete delta function that has a one in the zero position. That's easy to do. And then we should do also a shift, to see what's the effect, if you shift the function, what happens to the transform. That's an important rule, important for Fourier series and Fourier integrals, too. Because often you do that, and it's going to be a simple rule. If you shift the function, the transform does something nice. OK, and then I want to describe a little about the FFT and then start on the next section, convolutions. So that's fun and that's a big deal.

OK, about reviews, I'll be here as usual today. I think maybe the 26th, just hours before Thanksgiving we can give ourselves a holiday. So not next Wednesday but then certainly the Wednesday of the following week would be a sort of major quiz review on in the review session. And in class. OK, ready to go? On this example gives us a chance just to remember what the matrix looks like, because we're just going to, if I multiply this inverse matrix by that vector it's just going to pick off the first column and it'll be totally easy so let's just do it. So if y is this one, I want to know about f. What are the Fourier coefficients of the delta function? Discrete delta function? OK, before I even do it, we got a pretty good idea what to expect. Because we remember what happened to the ordinary delta function, in continuous time. Or rather, I guess it was the periodic delta function. Do you remember the coefficients, what were the coefficients for the periodic delta function? You remember those? We took the integral from minus pi to pi of our function, which was delta(x). And then we had to remember to divide by 2pi, and you remember the coefficients are e^(-ikx). This is c_k in the periodic case, the 2pi periodic case the function is the delta function, and you remember that if we want coefficient k we multiply by e^(-ikx). That's the thing that will pick out the e^(+ikx) term and of course everybody knows what we get here, this delta, this spike at x=0, means we take the value of that function at zero, which is one, so we just get 1/2pi. For all the Fourier coefficients of the delta function. The point being that they're all the same. That all frequencies are in the delta function to the same amount. I mean that's kind of nice. That we created the delta function for other reasons, but then here in Fourier space it's just clean as could be. And we'll expect something here, too.

You remember what F inverse is? F inverse 1/N. Instead of 1/2pi, and then the entries of F inverse come from F bar, the conjugate. So it's just one, one, one, minus - well, I've made it four by four here. This is minus omega - no, it isn't minus omega. It's it's omega bar, which is minus i, in this case. Omega bar, so it's minus i, that's omega bar. And the next one would be omega bar squared, and cubed, and so on. All the way up to the ninth power. But we're multiplying by one, zero, zero, so none of that matters. What's the answer? I'm doing this discrete Fourier transform, so I'm multiplying by the matrix with the complex conjugate guys. But I'm multiplying by that simple thing so it's just going to pick out the 0th column. In other words, constant. All the Fourier discrete Fourier coefficients of the discrete delta are the same. Just again. And what are they? So it picks out this column, but of course it divides by N, so the answer was 1/N. . It's just constant with the 1/N, where in the continuous case we had 1/2pi. No problem. OK. And, of course, everybody knows, suppose that I now start with these coefficients and add back to get the function. What would I get? Because just to be sure that we believe that F and F inverse really are what they are supposed to be. If I start with these coefficients, add back to put those in here and reconstruct, so 1/N, , what will I get? Well, what am I supposed to get? The delta, right? I'm supposed to get back to y. If I started with that, did F inverse to get the coefficients, that was the discrete Fourier transform, now I add back to get, add the Fourier series up again to come back here, well I'll certainly get , and you see why? If I multiply F, that zeroth row of F is , times will give me N. The N will cancel. I get the one. And all the other guys add to zeroes. So, sure enough, it works. We're really just seeing an example, an important example of the DFT. And the homework, then, would have some other examples. But I've forgotten whether the homework has this example.

But let's think about it now. Suppose that's my function value instead. OK, so now I'm starting with the delta. Again it's a delta, but it's moved over. And I could ask, I really should ask first, in the continuous case, suppose I, I can do it with just a little erasing here. Let me do the continuous case for the delta that we met first in this course. I'll shift it to a. If I shift the delta function to a point a, well, I said we'd met the delta function first in this course. At a, good. But when we did it wasn't 2pi periodic. So we still have, in fact, the Fourier integrals next week. Will have a similar formula. The integral will go from minus infinity to infinity and then we'll have the real delta, not periodic. Here, we have, and people call it a train of deltas. A train of spikes. Sort of you have one every 2pi. Anyway, that's what we've got. Now, you can see the answer. This is like in perfect practice in doing an integral with a delta. What's the integral equal? Well, the spike is at x=a. So it picks this function at x=a, which is e^(-ika). So not constant any more. They depend on k. The 1/2pi's still there. So it's, but still, the delta function shifted over. I mean, it didn't change energy. It didn't change, it just changed phase, so to speak. And we see that, I would call this like a modulation. So it's staying of absolute value one, still. But it's not the number one, it's going around the circle. Going around the unit circle. So it's a phase factor, right. And that's what I'm going to expect to see here in the discrete case too. If I do this multiplication by one there, it picks out this column, right? That one will pick out this column, so you see it's maybe I come up here now. Shall I just, when I pick out that column, the answer then, I guess I've got the column circle, there it is minus i. Minus i squared, minus i cubed. You see it's like k equals zero, one, two, three. Just the way here, we had k, well we had all integers. k in that function case. Here we've got four integers, k equals zero, one, two, and three, but again it's the minus i, it's the e to the, it's the w bar. In other words, the answer was one w bar w bar squared w bar cubed. Just the powers of w with this factor 1/N. Here we had a modulation. It's the same picture. Absolute value's one.

And and what about energy? Having mentioned energy, so that's another key rule. The key rules for the Fourier series, just let's think back. What were the key rules? First, the rule to find the coefficients. Good. Then the rule for the derivatives. This is so important. These are rules. Let's say, for Fourier series. For Fourier series. Let's just make this a quick review. What were the important rules? The important rules were, if I had the Fourier series of f. Start with the Fourier series of f. Then the question was, what's the Fourier series of df/dx. And now I'm saying the next important rule is the Fourier series of f, shifted. And then the last important rule is the energy. OK, and let's just, maybe this is a bad idea to, since we're kind of doing all of Fourier in, it's coming in three parts. Functions, discrete, integrals, but they all match. So this is what happens to the function. What happens to the coefficient? So this starts with coefficient c_k, for f, what are the coefficients for the derivative, just remind me? If f(x), so I'm starting with f(x) equals sum of c_k*e^(ikx). Start with that. And now take the derivative. When I take the derivative, down comes ik. So you remember that rule. Those are the Fourier coefficients of the derivative. Now what's the Fourier coefficients of the shift? If I've just shifted, translated the function, if my original x was this, now let me look at f(x-a). You'll see it. It'll jump out at us, it'll be a sum of the same c_k's e^ik(x-a). So what are the Fourier coefficients of that? Well there is the e^(ikx). Whatever's multiplying it has got to be the Fourier coefficient, and we see it as c_k times e^ik(-a). e^(-ika), times c_k. Right? And, of course, that's just what we discovered here. That's just what we found there, that when we shifted the delta, we've multiplied by this modulation, this phase factor came into the Fourier coefficients.

And now finally, the energy stuff. You remember the energy was, what's the energy? The integral from minus pi to pi, of f(x) squared. dx is the same as the sum from minus infinity to infinity of the coefficient squared. And somebody correctly sent me an email to say energy and length squared are you really, is there much difference? No. No. You could say length squared here, I'm just using the word energy. Now, I left a space because I know that there's a stupid 2pi somewhere. Where does it come? You remember how to get this? You put that whole series in there, multiply by its complex conjugate to get squared. And integrate. Right? That was the idea. Isn't that how we figured out, we got to this? We started with this, length squared. We plugged in the Fourier series. This is f times f bar, so that's this times its conjugate. And we integrated, and all the cross terms vanished. And only the ones were, e^(ikx) multiplied e^(-ikx) came. And those had c_k squared. And when we integrated that one, we probably got a 2pi. So that's the energy inequality, right. For functions. And now what I was going to say is, you shouldn't miss the fact that in the discrete case, there'll be a similar energy inequality. So we had y was the Fourier matrix times c. Now, if I take the length squared of both, y, so I'm going to right? That's the same as that. Now I'm going to do the same as this. I'm going to find the length squared, which will be y transpose y. No, it won't be y transpose y. What is length squared? y bar transpose y. I have to do that right. That will be, substituting that's c bar F bar transpose times y is Fc. Just plugged it in, and now what do I use? The key fact, the fact that the columns are orthogonal. That's what made all these integrals simple, right? When I put that into there, a whole lot of integrals had to be zero. When I put this in, a whole lot of dot products have to be zero. Rows of F bar times columns of F, all zero. Except when I'm hitting the same row, and when I'm hitting that same row I get an N. So I get this, is N c bar transpose c. And that's c squared. That's the energy inequality, it's just orthogonality once again. Everything in these weeks is coming out of orthogonality.

Orthogonality is the fact that this is N times the identity. Right? Well, OK that's a quick recall of a bunch of stuff for functions. And just seeing maybe for the first time the discrete analogs. I guess I don't have a brilliant idea for the discrete analog of the derivative. Well, guess there's a natural idea, it would be a finite difference, but somehow that isn't a rule that gets like, high marks. But we saw the discrete analog of the shift and now we see the energy inequality is just that the length of the function squared is equal to N times the length of the coefficient squared. OK with that? Lots of formulas here. Let's see, and do some examples. I mean, these were simple examples. And I think the homework gives you some more. You should be able to take the Fourier transform and go backwards. And when we do convolution in a few minutes, we're certainly going to be taking the Fourier, we're going to be going both ways. And use all these facts. OK, I'll pause a moment. That's topic one.

Topic two, fast Fourier transform. Wow. What's the good way to - I mean, any decent machine comes with the FFT hardwired in. So you might say, OK I'll just use it. And that seems totally reasonable to me. But you might like to see just on one board, what's the key idea? What's the little bit of algebra that makes it work? So I'll just have one board here for the FFT and a little bit of algebra. Simple, simple but once it hit the world, well, computer scientists just love the recursion that comes in there. So they look for that in every possible other algorithm, now. OK, let me see that point. So here's the main point. That if I want to take the multiply by F of size 1,024, the fast Fourier transform connects that full matrix to a half-full matrix. It connects that to the half-full matrix that takes the half-size transforms separately. So it's half-full because of these zeroes. That's the point. That the 1,024 matrix is connected to the 512 matrix. And what's underlying that? The 1,024 matrix is full of e^(2pi*i), the w, over a 1,024, right? That's the w for this guy. And then the w for this guy, for both of these, is e^(2pi*i) over 512. So if there's a connection between that matrix and this matrix, there'd better be a connection between that number and that number. Because this is the number that fills this one, and this is the number that fills these. So what's the connection? It's just perfect, right? If I take this number, which is one part of the whole circle, 1/1,024, a fraction of the whole circle, what do I do to get this guy? To get 1/512th of the way around the circle? I square it. The square of this w is this w. Let me call this w_N, and this one w_M. Maybe I'll use caps, yeah I'm using cap N, this is my w. This is the one I want. The w_N, the N by N one, N is 1,024, and the point is, everybody saw that, when I squared it I doubled the angle? When I doubled that angle the two over that gave me 1/512. Fantastic. Of course, that doesn't make this equal to that, but it suggests that there is a close connection.

So let me finish here the key idea of the Fourier transform in block matrix form. What's the key idea? So instead of doing the big transform, the full size, I'm going to do two half-sizes. But what am I going to apply those to? Here's the trick. These two separate guys apply to the odd-numbered coefficients. The odd-numbered component. And the even. So I have to first do a little permutation, and even comes first, always. Even means zero, two, four, up to a 1,022. So this is zero, two, zero, up to 1,022. And then come all the odd guys. One up to 1,023. So this is a permutation, a simple permutation. Just take your 1,024 numbers, pick out the even ones, put them on top. Right? In other words, put them on top where 512 is going to act on it. Put the odd ones at the bottom, the last 512, that guy will act on that. So there's 512 numbers there, with the even coefficients, this acts. OK, now we've got two half-size transforms. Because we're applying this to the y, to a to a typical y. OK. But I've just written F without a y so I don't really need a y here. This is a matrix identity. It's a matrix identity that's got a bunch of zeroes there. Of course, that matrix is full of zeroes. I mean, this is instant speed to do that permutation. Grab the evens, put them in front of the odds.

OK, so now I've got two half-sizes, but then I have to put them back together to get the full-size matrix. And that is also a matrix. Turns out to be a diagonal there, and a minus the diagonal goes there. So actually, that looks great too, right? Full of zeroes, the identity. No multiplications whatever. Well, these are the only multiplications, because sometimes they're called twiddle factors, give it a fancy name. Official sounding name, twiddle factors. OK, so that diagonal matrix D, what's that? That diagonal matrix D happens to be just the powers of w, sit along D. 1 w up to w to the, this is W_N, we're talking, the big W, and it's only half-size so it goes up to M-1. Half, M is half of N. Everybody's got that, right? M is half of N here. I'll get that written on the board. M is 512, N is 1,024, here we have the powers of this guy up to 511. The total size being 512 because that's a 512 by 512 matrix. I guess I can remember somewhere, being at a conference, this was probably soon after the FFT became sort of famous. And then somebody who was just presenting the idea and as soon as it was presented that way, I was happy. I guess. I thought OK, there you see the idea. Permutation, reorder the even-odd. Two half-size transforms, put them back together. And what's happened here? The work of this matrix, multiplying by this matrix, which would be 1,024 squared is now practically cut in half. Because this is nothing. And we have just this diagonal multiplication to do, and of course this is the same as that, just with minus signs. So the total multiplications we have to do, the total number of twiddle factors, is just 512, twelve and then we're golden. So we've got half the work plus 512, 512 operations. That's pretty good. And of course it gets better.

How? Once you have the idea of getting down to 512, what are you going to do now? This is the computer scientist's favorite idea. Do it again. That's what it comes to. Whatever worked for 1,024 to get to 512 is going to work. So now I'll split this up into, well, each 512, so now I have to do, yeah. Let me write F_{512} will be, it's now smaller. An I, an I, and a D and a minus D for the 512 size of F_{256}, 256 and then the permutation, the odd-even permutation P. So we're doing this idea in there, and in there. So it's just recursive. Recursive. And now, if we go all the way, so you see why I keep taking powers of two. It's not natural to have powers of two. Two or three. Three is also good. I mean, all this gets so optimized that powers of two or three are pretty good. And you just use the same idea. There'd be a similar idea here for, if I was doing instead of odd-even, even-odd I was doing maybe three groups. But stick with two, that's fine. Then you might ask, if you were a worrier I guess you might ask what if it's not a power of two. I think you just add in zeroes. Just pad it out to be the next power of two. Nothing difficult there. I think that's right. Hope that's right. And once this idea came out, of course, people started looking. What if the number here was prime? And found another neat bit of algebra that worked OK for prime numbers. Using a little bit of number theory. But the ultimate winner was this one. So maybe I'll just refer you to those pages in the book, which were, you'll spot this matrix equality. And then next to it is the algebra that you have to do to check it. I can just say, because I want you to look to the right spot there, maybe I'll take out the great - this is sometimes called after Parseval, or some other person. Yeah, the algebra.

Let me start the algebra that made this thing work. We want the sum, when we multiply Fourier F times something, we want the sum of w^{jk}, right? That's the coefficient, that's the entry of F, times c_k. Sum from k=0 to N-1. To 1,023. That's the y_j that we're trying to compute. We're computing a 1,024 y's from 1,024 c's by adding up the Fourier series when we multiply by F. This is F, this is the equation y=Fc written with subscripts. So this is what the matrices are doing, and now where do I find that 512 thing. How do I get M into the picture, remembering that the w_N squared was w_M, right? That's what we saw. This is the big number, this is half of it, so this is a little bit of the part around the circle. When I go twice as far I get to the other one. So that's the thing that we've got to use. Everything is going to depend on that. So this was w_N, of course. This is the N by N transpose So now comes what, what's the key idea? The key idea is split into even and odd. And then use this. So split into the even ones and the odd ones. So I write this as two separate sums, a sum for the even ones w_N, now, so now the even c_k, so I'm going to multiply by c_{2k}. These are the ones with even, and the sum is only going to go from zero to M-1, right? This is only half of the terms. And then look on the other half, plus the same sum of, but I didn't finish here. Let me finish. So I'm just picking out, I'm taking, instead of k I'm doing 2k. So I have j times 2k here. And now these will be the odd ones. $c_{(2k+1)}$, and $\omega_N^{j(2k+1)}$. It's a lot to ask you. To focus on this bit of algebra. I hope you're going to go away feeling, well it's pretty darn simple. I mean there'll be a lot of indices, and if I push it all the way through there'll be a few more.

But the point is, it's pretty darn simple. For example, this term. What have I got there? Look at that term, that's beautiful. That's w_N squared. What is w_N squared? It's w_N. So instead of w_N squared, I'm going to replace that w_N squared by w_M. And the sum goes from zero to M-1, and what does that represent? That represents the F_{512} multiplication, the half-size transform. It goes halfway, it operates on the even ones, and it uses the M. It's just perfectly, so this is nothing but the Fourier matrix, the M by M Fourier matrix, acting on the even c's. That's what that is. And that's why we get these identities. That's why we get these identities, because they're acting on the even - the top half is the even guy. OK, this is almost as good. This is the odd c's. Here I have, now do I have w_M here? I've got to have w_M. Well, you can see. It's not quite coming out right, and that's the twiddle factor. I have to take out a w_N to the power j to make things good. And when I take out that w_N to the power j, that's what goes in the D, in the diagonal matrix. Yeah. I won't go more than that. You couldn't, there's no reason to do everything here on the board when the main point is there. And then the point was recursion and then, oh, let me complete the recursion. So I recurse down to 256, then 128, then 64. And what do I get in the end? What do I get altogether, once I've got all the way down to size two? I have a whole lot of these factors down in the middle, the part that used to be hard is now down to F_2 or F_1 or something. So let's say F_1, one by one, just the identity. So I go all the way. Ten steps from 1024, 512, 256, every time I get twiddle factors. Every time I get P's. A lot of P's, but the F_{512}, what used to be the hard part, is gone to the easy part. And then what do I have? I've got just a permutation there. And this is the actual work.

This is the only work left, is the matrices like this, for different sizes. And I have to do those. I have to do all those twiddle factors. So how many matrices are there, there? It's the log, right? Every time I divided by two. So if I started at 1,024, I do ten times, I have ten of those matrices and have me down to N=1. - So I've got ten of these, and each one takes 1,024 or maybe only 1/2 of that. Actually, only half because this is a copy of that. I think the final count, and can I just put in here, this is the great number, is each of these took n multiplications. But there were only log to the base two - oh, no. Each of them took half of N multiplications, because the D and the minus D are just, I don't have to repeat. So half N for each factor and the number of factors is log to the base two of N. Ten, for 1,024. So that's the magic of the FFT. OK. It's almost all on one board, one and a half boards. To tell you the key point, odds and evens, recursion, twiddle factors, getting down to the point where you only have twiddle factors left and then those multiplications are only N log N. Good? Yes. Right, OK.

Now. that's discrete transform. The theory behind it and the fantastic algorithm that executes it. Are you ready for a convolution? Can we start on a topic that's really quite nice, and then Friday will be the focus on convolution. Friday will certainly be all convolution day. But maybe it's not a bad idea to see now, what's the question. Let me ask that question. OK, convolution. So we're into the next section of the book, Section 4.4, it must be. And let me do it first for a Fourier series. I have convolution of series, convolution of discrete. Convolution of integrals, but we haven't got there yet. So I'll do this one, this series. So let me start with a couple of series. f(x) is the sum of c_k*e^(ikx). g(x) is the sum of some other coefficients. And I'm going to ask you a simple question. What are the Fourier coefficients of f times g? If I multiply those functions, equals something. And let me call those coefficients, h maybe. h_k*e^(ikx), and my question is what are the coefficients h_k of f times g? That's the question that convolution answers. Actually, both this series and the discrete series are highly interesting. Highly interesting. So here I wrote it for this series. If I write it for the discrete ones, you'll see, so let me do it over here for the discrete one. Because I can write it out for the discrete ones. My y's are c_0+c_1 e - no, w I have this nice notation w, plus c_N-1*w^(N-1), right? That's the - ooh. What's that? I haven't got that right. Yes, what do I want now? I need, yep. Sorry, I'm looking. Really, I'm looking at y_j, the j'th component of y. So I need a w^j, w^j(N-1). Yeah, OK, let me - OK. Alright. And so that's my f. I'll come back to that, let me stay with this.

I'll stay with this to make the main point, and then Friday we'll see it in a neat way for the discrete one. So I'm coming back to this. f has its Fourier series g has its Fourier series. I multiply. What happens when I multiply this times this? I'm not going to integrate. I mean, when I do this multiplication, I'm going to get a mass of terms. A real lot of terms. And I'm not going to integrate them away. So they're all there. So what am I asking? I'm asking to pick out all the terms that have the same exponential with them. Like, what's h_0? Yes, tell me what h_0 is? If you can pick out h_0 here, you'll get the idea of convolution. What's the constant term if I multiply this mess times this mess, and I look for the constant term, h_0, where do I get the constant terms when I multiply that by that? Just think about that. Where do I get a constant, without any k, without an e^(ikx)? If I multiply that by that. Well tell me one place I get something. c_0 times? d_0. Good. Is that the end of the story? No. If you thought that multiplying the functions, I just multiplied the Fourier coefficients, the first point is no. There's more stuff. Where else do I get a constant out of this? Just look at it, do that multiplication and ask yourself where's the constant. Another one, yep. You were going to say it is? c_1 times d_-1. Right. Right. c_1 times d_-1. And tell me all of them, now. c_2 times d_-2. And what about c_-1? There's a c_-1. It multiplies d_1. And onwards. So the coefficient comes from, now how could you describe that? I guess I'll describe it as, I'll need a sum to multiply. This will be the sum of c_k times d_1. Minus k, right? That's what you told me. Piece at the start, and that's the pattern that keeps going. OK, that's h_0, the sum of c_k times d_-k. Now, we have just time to do the next one.

We've got time but not space, where the heck am I going to put it? I want to do h_k, I guess. Or I better use a different letter h. l, let me use the letter h_l, and God there's no space. Alright, so can I - yes. h_l. OK. So this was h_0, let me keep things sort of looking right for the moment. OK, now you're going to fix h_l. So what does c_0 multiply if I'm looking for h_l, I'm looking for the coefficient of e^(ilx). So ask yourself how do I get e^(ilx) when that multiplies that? When that multiplies that, and I'm looking for an e^(ilx), I get one when c_0 all multiplies what? This is it. dl, right. And what about for c_1? Think of here, I have a c_1*e^(ilx). What does it multiply down here to get the exponential to be l? ilx? What doesn't multiply d_-1. It multiplies, c_1 multiplies? d_(l-1). Good, good. l-1, right. l-1, and what are you noticing here? c minus, I'll have to fill that in. But you're seeing the pattern here? And what was the pattern here? Those numbers added to this number. And now these numbers add to l Those numbers add to l, whatever it is, the two indices have to add to l, so that when I multiply the exponential they'll add to e^(ilx). They'll multiply to e^(ilx). So what goes there? It's probably l+1, right? So that l+1 combined with minus one gives me the l. If you tell me what goes there, I'm a happy person. Let's make it h_l. We're ready for the final formula for convolutions. Big star. To find h_l, the coefficient of e^(ilx), when you multiply that by that, you look at c_k, and which d is going to show up in the e^(ilx) term? l-k, is that what you said? I hope, yeah. l-k. Right, that's it. That's it. So we've got a lot of computation here. But we've got the idea of what, we've got a formula. And most of all we have the magic rule. In convolutions, convolutions are, things multiply but indices add. Things multiply, numbers multiply, while their indices add. That's the key idea of convolution that we'll see clearly and completely on Friday. OK.