

The following content is provided by MIT OpenCourseWare under a Creative Commons license. Additional information about our license and MIT OpenCourseWare in general is available at ocw.mit.edu.

PROFESSOR: So I hope in time for you to consider this material also in thinking about projects because there are lots of experiments to do with these. I did a MATLAB experiment that I'll mention. And I guess it was too small because it didn't make multigrid look as good as it is. So can I remember the steps of multigrid and then you'll remember why we take those steps and then make a sort of basic analysis of how do you decide whether this multiple step process is successful? How much does it reduce the error? So you remember the steps. Oh, thanks.

So the first step is really a little bit outside multigrid. It uses an ordinary iteration like Jacobi, a few times and so you remember that the matrix that governs this ordinary step, the matrix that's crucial for this is the-- so what multiplies the error at every step. So we have initial error and if we take a smoothing step with a preconditioner, p , then the error is multiplied by $I - p^{-1}A$. So that's the matrix. That's sort of that one step, one iteration matrix for ordinary iterations where A is of course, we're solving $Au = b$. That's our problem. And matrix p is close to A . Of course if we took p equal to A then the error would be reduced to zero right away because we would be solving the system one shot. But now, we're left with some error and this-- steps 2, 3, 4 reduce it further. And well, our hope is to see that this multigrid operation reduces the low frequency part because this one will act successfully on the high frequency part, the oscillating error, but for smooth error the eigenvalues here are too close to 1 and it's too slow.

OK, so I was just going to track down-- well, first have to remember, what are the pieces here? So there's a matrix R , restriction matrix. A matrix A_{sh} , which is original matrix A or A_h . I could call that. But the original one is on fine mesh. So I need a coarse mesh version. So that's what I solve on the coarse mesh, the smaller problem and then interpolate back with a matrix I back to the fine mesh. OK, so can I just remember also what those matrices-- we have all the pieces here. So if

I remember what R and I are-- OK, so let me write down all those pieces.

So A-- let me tell you the MATLAB experiment I ran. So A is our two's and minus one's matrix. Our second difference matrix in 1D to keep it simple. And it should be divided by h^2 . Well, I guess we took the mesh where the coarse mesh just had two interior points. The value at the boundaries was zero and the fine mesh has 5 [INAUDIBLE]. So this will be 5 by 5. Then we needed the restriction and interpolation matrices. So this was R, the restriction. So the restriction, which we're going to use right away will take us from the fine to the coarse. So that matrix will be-- it'll multiply things with 5 components and produce an output with only 2 components. Can always find either of those mesh points and their numbers were 1, 2, 1, 0, 0, and 0, 0, 1, 2, 1 and needed to divide by $1/4$ because those rows added up to 4. The idea being that if we restricted all ones we should get all one and we need that $1/4$ factor to do it. And the interpolation went the other way and it was essentially the transpose.

That's the nice connection to have and again, we need a factor-- and maybe the factor is $1/2$ there. Good. We've now got all the pieces except A_{2h} and I think just at the very last minute last time I said what A_{2h} should be. And the good choice for A_{2h} -- so that's on the coarse mesh. So that's going to be small matrix. So the good choice is I, which puts it up on the fine mesh, A_h , which is our operator on the fine mesh and then R that restricts back. So R, A, I, is the right construction, it's sort of the natural construction for the coarse mesh operator. And notice, I don't know if you appreciate having this, this is the same as R, A, R transposed with some factor so that it'll come out symmetric, it'll come out positive definite, and actually, it'll come out in this example-- it will come out exactly the second difference operator on the coarse mesh. So this will turn out to be 2, minus 1, minus 1, 2. The 2 by 2 operator and since we're on the coarse mesh the step size is 2. I won't do that multiplication, but MATLAB confirmed it. So that's where we were. So I've sort of done now a recap of multigrid and I'm ready to tackle a little understanding now of what happens to the error.

So at this stage I end up with some error. U_h is not the exact answer. I have an

error e_h , which is $u - u_h$. That's the error at the end. Now the question is, what happens to that error? Can we see how and why it's reduced as we do a v -cycle, as we do a 2, 3, 4 cycle? So since everything's linear that error is multiplied by some matrix that we just have to track those steps, see what that matrix is, and look at its eigenvalues. So that's my goal here, is to find the matrix, call it s that takes a step, a multigrid step and what does it do to the error? OK, so can I just follow that step? Here's the key matrix. Maybe I'll write it underneath. So what happens to the error?

First of all, let me write the steps down. e_h then is $u - u_h$. That's entering multigrid. That's my error. OK, now I compute this residual, r_h . I want to say that it's Ae_h . Can I check that that's true? Yes. A times this is A times $u -$ that's the b minus A_h times u_h . That's the second term. So that's OK. So that's what happened there. Then it got restricted. I'll keep a record of what happens to e_h . So it's multiplied by A to get the residual. We work with the residual and with b because b is the input we know. Then at this step two, that residual is restricted by R . Now we've got r_{2h} . When I multiplied by R I got r_{2h} . OK, next step. To get to E_{2h} . So what do I do there? I multiply by the inverse. I solve this system, so of course, I don't actually find an inverse matrix, but that's the formula. To get this I multiply the inverse matrix by that, so I've got an inverse matrix here. So I write it as A_{2h}^{-1} . This was A_h .

Now when I've multiplied by A_{2h}^{-1} I'm this far and the final step, multiplies that by I . So there you go. It looks pretty awkward, but that's the matrix, s , that produces E_h . So I'm thinking that my matrix is going to be called s . This matrix here is going to be called s . So let me write down what s is. All I want to do is remind us what A_{2h} is by this choice there, it's RAI . And then comes R and then comes A . A is now A_h .

Well, we've got a matrix. And for this example I can compute what it is. Can we say anything about that matrix from its formula? Looks a little messy. But of course, it's built out of these pieces. It's built out of the given matrix A and our R and our I . And there is a neat property hiding there. Which you can see if you square that matrix.

So look at s squared. If we look at s squared, so this is the matrix to understand. And I don't claim to understand it as well as I would like, but-- so this is the matrix for these three steps. We would have a full multigrid v -cycle just before I lose the track on that. A full multigrid v -cycle would do M a few times, say twice. Two smoothers, then it would do a v -cycle and then smooth again. Well, I should've said the smooth again would be the one on the left. This is the original, so there's two smoothers followed by a multigrid cycle, followed by a couple of smoothers. That would be our total overall matrix, but I want to focus for right now on this matrix, s . And I guess, I hope that anybody who does a project or a homework on this topic, you're going to create I , R , and A and compute s . And it should show what s squared is. So could I compute, could I look at s squared? You'll see why. So I just write s down twice. RAI inverse RA . So that's once, now repeat the whole thing. I , RAI inverse RA . I have multiplied s by itself. And what do you see? Well it's nice. If you look at this you see that we've got this combination RAI sitting in the middle because when I did it twice the RA came from there and then an I was at the start, so this is-- so what's left? S . This matrix S is equal to its own square. S squared equals S . You know, we're able to see this without-- this would apply for this example, but for all examples. Now I'm ready to think about eigenvalues a little bit.

What does that tell me about the eigenvalues of S ? So this is a small class of matrices with this property that s squared equals s and maybe I say something about those here. So if s squared equals s , what does that tell us about eigenvalues of that matrix? It's a square matrix. It doesn't happen to be symmetric. Very often we're looking at symmetric matrices, but this is a little lopsided. We have an A off on the right. I might be able to make it symmetric by bringing part of that A off to the left, but this is the main point. What are the possible eigenvalues of that matrix, S ? Well, zero, 1, 1. The only eigenvalues are zero and 1. Why is that? Because if Se equals λe , so I'll put that word if in. If Se equals λe I could multiply by S again so I would get S squared e equals λSe and that's λ squared e . I haven't done anything fancy here, all I'm saying is that any matrix, if it has an eigenvalue of λ and an eigenvector e then S squared has that same eigenvector and it's eigenvalue is λ squared. But now S squared is S . So this

is the same as Se , which is λe . You see that I now have λ equal λ squared. Sorry, I probably made that take more time than it had to. If S squared equals S then λ squared equals λ . So I have λ squared equal λ and the only two possibilities-- that's a quadratic equation that's got two roots and those are the roots. If λ squared equals λ then λ could be zero or it could be 1. So what does that tell me? That tells me that the eigenvectors that have eigenvalue zero, what happens to those in multigrid? They're killed. An eigenvector with an eigenvalue of zero means that Se is zero. So whatever error we started with one cycle through multigrid will remove it. These are in the null space of S . They're vectors that S completely removes and these are vectors where Se equals e . The eigenvalue is 1 and multigrid does nothing.

Well, one part of the message then is that I could use multigrid-- well, I guess I hadn't thought about this, but one message is don't repeat 2, 3, 4. That would be a large waste of time. If you just repeated-- if you just did another v -cycle and didn't do any smoothing or anything in between, then the error that came out of the first v -cycle would survive right through. Nothing good would happen in the second cycle that didn't already happen in the first. Now another message is of course, I could do multigrid forever and I'm not going to get convergence. Actually, that's sort of interesting because we're totally accustomed to working with matrices M , whose eigenvalues are spread out, but below 1. This has its eigenvalues not spread out at all, zero or 1. That's it. I'd better say quickly that the key matrix here is I minus S . Just the way that it was I minus p inverse A . So the multigrid matrix-- yeah, maybe I could've messed up. The matrix is really I minus S , so let me be sure I see where that happens. I guess I got down here to figure out what $E_{sub h}$ is and then I didn't do anything with it. I forgot to write in the key step that at the end of step 4 when I have a piece of the error that I add that to U_h . So add U_h plus this estimated error. See if I knew $E_{sub h}$, if I knew small e_h -- the exact error-- then when I added I would get U , the answer. Of course, I don't know that. What I know is capital E . So that's a cap E , which will agree with e on some of these, but not others. So all right, I guess I'm seeing the problem. I may have headed in reverse because it's I minus S that's crucial. So these are errors. Now what happens to them? Are these the good

ones? I think they're the good ones, is that right? They're the good ones because the errors in this stage, multigrid gets them exactly, so that when I add this in it's right on target. It's accounted for. So these are ones that multigrid fixes because of course, everybody recognizes that $I - S$ minus-- let me just say it, though. That $I - S$ times those errors is zero. So the eigenvectors with eigenvalue 1 are the ones that are perfectly captured-- $E_{sub h}$ has got those components of small $e_{sub h}$ exactly right when I add them on; perfection. But then there were other components which it hasn't got at all. And when I add those on, no change, no improvement.

By the way, so these eigenvalues are repeated. Like here, I've got 5 by 5 matrix. How many zeroes and how many ones? You could say it's a projection matrix. S projects on to these eigenvectors. $I - S$ projects onto these eigenvectors and how many zeros and how many ones for this if A is 5 by 5? Well, what is A ? This A is 2 by 2. So if I look at my S , look at my nifty formula for S , do you see all this here? Let's just get the sizes. A is 5 by 5. R is 2 by 5. It shrinks us down to the coarse mesh, which is only where this is only 2 by 2 and this takes us back up. So the total result, 5 by 2, 2 by 2, 2 by 5, 5 by 5 is 5 by 5. But what's the rank of S ? Small prize for telling me the rank. I'm interested in this count. How many vectors are killed? How many go through unchanged when I multiply by S ? So what's your guess for the rank of S ? 2, exactly.

So S will have 2 of these guys. 2. This has multiplicity 2 in the 5 by 5 case and this one has multiplicity 3. So multigrid is only fixing 2-- getting 2 vectors exactly right. And not getting these right. Well, I mean, 2 is half of 5. Well, in fact 2 is exactly half of 5 because if I went up it would be 3 and 7, 4 and 9. The point is that the number of coarse grid points is about 1/2 the number of fine grid points. So 2 is about 1/2 of 1, 2, 3, 4, 5. Of course, how could I expect better than getting 2 perfect? I can't expect more because the problem I ultimately solve here is on the 2 by 2 size. So when I actually do something useful, I'm only doing it on the coarse grid and that's got dimension 2 and so there will be 2 eigenvectors that get perfectly fixed, but the others don't. I do have to think which are the eigenvectors? What eigenvectors are there? I guess what I hope is, what I believe is that these eigenvectors, which multigrid fixes for me, those E 's-- well, the E 's have 5 components. Our matrix S is 5

by 5. I guess I'm thinking, I'm expecting that the eigenvectors that these 2 are somehow basically on the coarse mesh. And that these 3 that don't get fixed are somehow basically orthogonal to the coarse mesh I would say. That these will be low frequency vectors and these will be high frequency vectors. That's what I expect and that's what somebody-- I mean, there are whole books on multigrid, which where the main step in the analysis is to see that those eigenvectors are as I described, sort of coarse mesh vectors that are fixed and fine mesh vectors that don't get fixed by multigrid and those are the ones that a smoother better deal with.

I mentioned that just before class I did a MATLAB experiment and I actually did it on these matrices with a Jacobi smoother. And I broke down the numbers here somewhere. and actually I was a little disappointed. But I came away saying, OK, 5 by 5 just wasn't big enough. So I'll put down what I've got. What did I get for the eigenvectors? eigenvalues of S I got 1 and zero. That was a good thing. Now did I write them down? I sure hope I did. Yep. So the question is, what about when M is in there? So the exercises were find the eigenvalues-- well, it'd be interesting first of all, to know the eigenvalues of M. This is the Jacobi smoother, weighted Jacobi. Here's one we're really interested in. We're interested in doing a smoother, doing a multigrid cycle, and doing a smoother. I'm hoping that those eigenvalues are below this. So this is one smoother and this is smoothers with multigrid and this would be 2 smoothers and 2 post smoothers. Repeating Jacobi twice and M cubed would be interesting. Here is what I got for these things.

What do I know about the eigenvalues of M? Do you remember that we actually worked those out? For ordinary Jacobi they're cosines and for weighted Jacobi that weight ω , which I chose to be $2/3$ comes into it. Anyway, so in a 5 by 5 case-- well, it's the biggest one that we all really want to know. So this is maximum eigenvalue. Let me just write down the maximum. So the maximum eigenvalue of M was 0.91. So that says that ordinary Jacobi on a 5 by 5 problem reduces the error by that factor at every step. And of course that factor is totally satisfactory. If we could always do that we wouldn't need to do multigrid. But that 0.91, if I go to 7 by 7 or 9 by 9 it's going to quickly move up toward 1. That's an indication that my little

experiment here is a bit small because that's not as near 1 as a true [? vector. ?]
OK, what was the result after this? I'm sorry to say that it was 0.88. Disappointing.
So I did this one. I did 2 smoothers.

Well, of course that's going to give me 0.91 squared in there. But anyway, I got 0.64. Which of course is very good. I'll say very good, but actually for that size I had hoped for better because my impression of multigrid from reading about it is that-- I mean that a good multigrid cycle, but this has an eigenvalue of about 0.1, is the goal. So how is that achieved? Well, I don't want to achieve it by just doing a whole lot more smoothing. Actually it's achieved by doing a whole lot more multigrid instead of doing just a little v-cycle this would be the same for a good, deep v-cycle that went down maybe 4 and maybe a w-cycle; do it twice. Then of course, this is for a big problem, but really making multigrid work like a w-cycle. Or even there's something called full multigrid. Should I draw the picture that corresponds to w for full multigrid? So full multigrid you start on a coarse mesh. Go up, back up, up, back, back, maybe twice, up, up, up to this level, back, back, back, and up again. So that's called full multigrid. That gives the best numbers. Anyway, we get numbers like this for when we really do more and it's very much worth it. I mean, it's completely a success. I would be interested to see a little-- well, I guess what I'm going to do now is, for this model problem that we worked on I could find these eigenvectors, e. And analyze this model problem.

OK, so how to find these eigenvectors. Again, for these particular matrices. Well, let me start with the eigenvectors of A itself. Do we know the eigenvectors of A? I guess if I had one important matrix, whose eigenvectors-- and nontrivial. I can't look at the 5 by 5 there and guess at sight, say what the eigenvectors are. So everybody sees it's 5 by 5 with 4 minus ones below, 5 two's on the diagonal, 4 minus ones above the diagonal, and the eigenvectors of those special matrices have a special form, which allows us to do everything here. So what can I do here? I can do the computation of what S is. Can I report on that, which will be in the notes. Let me tell you what S is. If I take that as a R, that as I, that as A sub 2h I get S to be this matrix. Column of zeroes, 1/2, 1, 1/2. Column of zeroes. 1/2, 1, 1/2 and column of zeroes. So it's rather remarkable in this particular case.

Now what can we recognize from this? Well can we see eigenvalues of S explicitly from the matrix? Well, we certainly see that it has rank 2. It has 2 independent columns so its rank is 2 as we predicted. So it will have 3 zero eigenvalues and we can see what the eigenvectors are actually. What are the eigenvectors if I go up above here? Can I do that? I want to go up and say something about the eigenvectors with eigenvalue zero. What's the null space of that matrix? If I multiply S by vectors I want to get zero vectors. The eigenvector's e for λ equals zero. If I multiply by anything with a first component, with these three components, you see that if I multiply Se is zero there for these e 's. In a way that's exactly what I hoped for. That these vectors, e are the vectors that live on the fine grid, but with no component on the coarse grid. And they're the ones which have Se equal zero and multigrid doesn't help. Multigrid doesn't help. $I - S$ times those vectors is those vectors, no improvement. So they're eigenvectors with eigenvalue zero for S , but with 1 for $I - S$. See my general thought is those would be the high frequency eigenvectors, high frequency vectors. They're sort of high frequency, but they're not low frequency anyway. Because low frequency these would be a smooth vector.

OK, so what are the eigenvectors with the other two eigenvectors-- the ones that multigrid fixes? OK, so I'm looking for the other two eigenvectors here and how do I find them? Let me see if I can even do it for this matrix. I'm looking for the eigenvectors with eigenvalue 1 now. So I should subtract the identity from this matrix. When I subtract the identity the ones become zeroes and the zeroes become minus ones. That's $S - I$ now, trying to keep a running equation that's correct. That's $S - I$. And I'm looking for the null space of that now. Am I going to find it? I don't know. Do see immediately what-- that has a two-dimensional null space. There are 2 vectors and can you see what they are? What do you think? Yeah, we certainly could. I don't know whether-- anybody see a vector that's in the null space of that matrix? Let me just start and try to get one.

So I'm looking at that matrix, looking for a vector in its null space. And that will be an error vector that multigrid totally fixes. Say if I start with a 1 up in that first component then the first row will force me to have a 2 there. Is that right? And then I

fixed the first row, the second row is all zeroes already. Then I have 0.5 of that, which is 1. Oh, well let me start from the bottom here. Well, it's probably symmetric. What do you think? Maybe something like 1, 2, and what's that middle number? Maybe 2. I think that that is in the null space of S minus I because if I multiply by the first row I get minus 1, plus 1 and the third row gives me 1, minus 2, plus 1 and yes - so that's one of them anyway. And the point about it is it's pretty smooth. That's what I anticipate vectors that are smooth, that's the type of vectors we're going to find multigrid fixing. So I've got one more eigenvector to find. Maybe I'll leave that as a puzzle, somewhere there's another eigenvector. Guess I'm not sure where it is, but I think it should be smooth. Somewhere there is one. Can I now go away from this numerical example to give you the answer for this problem, but for any size?

Oh, it's because we know the eigenvectors of A . I started to talk about the eigenvectors of A and I didn't finish. If I want to get formulas for these I need to know the eigenvectors of A . So A , this matrix of twos and minus ones--- the eigenvectors of that matrix are discrete sine functions. So an eigenvector would be for example, sine of π on 6, sine of 2π on 6 up to sine of 5π on 6. It's worth knowing that those are eigenvectors. And other eigenvectors would double the frequency, triple the frequency, four times the frequency and five times. The eigenvectors of this matrix are what I will call, discrete sines. Discrete sine vectors. And why? Because this matrix builds in the boundary condition since there should be a minus 1 there, but it's chopped off. That minus 1 if it was there would multiply the zeroes component. But the zeroes component is removed, so we have to be sure that the zeroes component would have been zero anyway. And that's the point. You see if I follow this pattern the zero component that I'm missing would be sine of zero. The component I'm missing up here would be sine of zero π over 6, which is zero. And the component that's missing down at this boundary would be sine 6π over 6, which is sine π , which is zero. That choice of vectors is the perfect Fourier vector. I expect Fourier when I see constant coefficients there with the right boundary conditions zero at both ends. Anyway, those are the eigenvectors. The k 'th eigenvector will have sine $k\pi$, sine $k^2\pi$, down to sine $k^5\pi$, all over 6. And I'll just, in the remaining moment write down the result of following through those

vectors and finding finally, the eigenvectors of S itself.

So this is like multigrid in a nutshell if I can find it. Multigrid in a nutshell. All right, better be on this final board. I minus S times this sine vector, the k 'th sine vector that I wrote over there and I minus S times the n minus k 'th sine vector. So these will be low frequencies for small k and these will be high frequencies. And it turns out that these are $1/2$, 1 minus the cosine of $k\pi$ over 6 or n plus 1 times y_k plus-- s_k , sorry. S_k plus S_{n-k} . And this acting on the high frequencies is $1/2$ of 1 plus cosine $k\pi$ over 6 . S_k plus S_{n-k} . What am I seeing there? Let's just accept the algebra that went into this and see, does that tell me what the eigenvectors of I minus S are because that's my goal.

Well, it does. Look, if I add these two equations, let me look first just at this equation. So this says that if I start with a low frequency, k small, low value of k that it's nearly removed. If k is very small the cosine of $k\pi$ over 6 is somewhere near 1 . Of course, 6 was a little feeble, so let me take n ; bigger matrices. Then the low frequencies are the ones where this cosine is near 1 . This number is near zero and they're practically removed. Where the high frequencies, not much happens because the cosine is near 1 , this number is near 2 . I divide by 2 and it's near 1 . So the overall picture is right. Now to get to pin it down, if I add that two equations then I get this combination S_k plus S_{n-k} . Oh in fact, I guess I see it. What happens if I add the two equations? I get I minus S times S_k plus S_{n-k} and what do I have on the right-hand side? Well, 1 . These add to 1 . So I get S_k plus S_{n-k} . Those are the eigenvectors of I minus S with eigenvalue 1 . They're the ones I found here. They're the eigenvalues of S with eigenvalue zero. These are the S_k 's plus S_{n-k} . And they live completely on the mesh points that are in the fine, but not the coarse mesh. And then if I play around one more step I can find the eigenvectors for the other eigenvalue. The ones that are basically low frequency and where we have a numerical example.

OK, I've run out of time before I can do that last bit of algebra. That will be in the notes, which we'll get onto the website, Monday at the latest. So I hope you have a sort of picture of what multigrid does and the experiments that you might do to

check it out. OK, I'll see Monday and we'll take a little more time to discuss projects because we'll have material on the web to work with. OK, thanks.