

INTRODUCTION: 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: OK. Should we go? Great. Well I hope you enjoyed Patriots Day, and now we'll go for the final weeks of the course. And I wanted start with just a thought about scientific computing, and the people who work in it, and the problems. So just sort of an orientation. And maybe my point is that in part of the subject, the situation started like just give me the matrix, tell me the equation. I don't want to know the physics behind it, just the form of the equation, and I'll discuss how it could be solved. And to contrast that with what you may often be meeting in your thesis even in project, the details, the physical consequences, the actual detail behavior of this solution is what you're looking for. So somehow there are those two parts of the world, and people who know understand the physical problem are on the second side. And people who maybe think just about matrices, like that matrix for example - that saddle point matrix, are more on the first side. Like just give me the matrix and I'll figure out how to solve the linear system. If it's a large linear system, I'll invent preconditioners, study stability, whatever you have to do.

But today's lecture is kind of half and half, because that is the framework that we're in. But the problem comes from a real problem in fluids. And it's called the Stokes problem, not Navier-Stokes. So Stokes alone is dealing with this question when there is a diffusion term, and I've normalized the coefficient there, viscosity term, whatever we call it, to be 1. And there is the constraint of incompressibility, constant pressure divergence $\nabla \cdot \mathbf{p} = 0$. I'm sorry not constant, divergence of \mathbf{p} equals 0 is a constraint. First let's do this. So make some comment on the physical side of this equation. What we're missing that Navier-Stokes would put in is a convection term. A term of $\mathbf{v} \cdot \nabla \mathbf{v}$ that takes account of the fact that there's a flow here, things are moving. We're not in a stationary situation. And that term has been dropped. So that's telling us then we are in the case of very, very slow flow, where that term could be neglected. And certainly neglecting that term makes our

problem easier. It leaves us with this symmetric matrix. It doesn't make it totally easy as we'll see, but it's a lot easier than when the convection term -- which by the way is not symmetric and pushes outside this framework. If that's present then of course in computational fluid dynamics, that's the central problem. You have to deal with it.

So we're taking a model problem that doesn't have the convection, doesn't have that unsymmetric complexity. But actually it's become more and more important. It's more than a model problem now, because there are a lot of medical application. Say probably the flow of your blood pushed by the heart that's moving along at a decent rate. But flow into tiny capillaries is slow, and Stokes is a reasonable step. And then actually Stokes is also going to appear in Navier-Stokes calculations, because often when you have two difficult processes you do a kind of splitting method, and you drop one term for half a step. So you take for example, a Stokes step, and then you do what the fluid forces you to do. You move that process with the flow. OK. So where flowing slowly here. And the nice thing is that this fits our saddle point framework, that's why I'm calling this block matrix S .

And you see really it came from minimization. The minimize was the problem that leads to the Laplacian, that stands for the Laplace operator with a minus sign to make it positive definite. And then this constraint gets imposed by a Lagrange multiplier P . So P is the Lagrange multiplier here. Oh wrong. Divergence V is 0, incompressible. Now that's on the tape. It's now on the tape right. The divergence V is 0. OK. So you remember how our equations looked? I'm going to write the unknowns here, V and P , So that you see what everything is, and it equals f and 0. OK. So this is now S , so I chop off here. So there you see the equation in the saddle point form. So A is the gradient, and it breaks on P . And so P is a scalar. I better say if I put a little arrow here, V is a vector, it has two components say if we're in a 2D problem; f has two components. P the Lagrange multiplier for this constraint, the constraint is just a scalar constraint the divergence is 0. You remember what divergence means; $\frac{dv_1}{dx} + \frac{dv_2}{dy}$, so those are the two components of V , the vector. And its divergence is 0. And these are, well with the minus sign, are adjoints to each other. OK.

So what's new that makes this problem worth some study? I guess that up to now, we've taken problems like this and we've done elimination to get to a single equation. You remember that. So that single equation is $A^T C A p = f$. Let's see. We're multiplying that row by $A^T C$ and subtracting. Let me do the elimination under here. I multiply the first equation by $A^T C$. Just give me the matrix mode for now. I multiply by $A^T C$ and I subtract. So the first equation is still there, $C^{-1} v + A p = f$. But the second equation now when I multiply by $A^T C$, so I just put it up there, and subtract that produces a 0 there, it produced $-A^T C A p = -A^T C f$. It leads us to this crucial matrix. And in applications up to now that was the way to go. That was totally satisfactory. We eliminated v , we got an equation for p , we solved it, and then once we knew p we went back and found v . OK.

So why is this different? Why don't I want to get to $A^T C A$, the stiffness matrix if we were in finite elements. Why do I want to stay with this two field system? Which in finite elements is going to be a mixed method of finite elements. I'm going to keep v and p in the problem for a simple reason, that C in this problem is awkward, C would be the inverse of the Laplacian. C is the inverse of the Laplacian d^2 second by x^2 plus d^2 second by the y^2 . That's what Δ stands for. And that's an uncomfortable operator to work with, the inverse here. In matrix language when we discretize, it won't be sparse, right. The Laplacian is sparse, the Laplacian gives us the five point matrix, the matrix with five non-0's on every row - four on the diagonal and four minus 1's at matrix that we know, the K2D matrix. But this is its inverse, C is now its inverse. It's the inverse of that matrix K2D that we've study, and of course it's full. So taking this step produces a completely full matrix, and all of our efficient algorithms for large problems work for sparse matrices. So whereas this one is sparse. I mean this the Laplacian. So we have the five point matrix sitting here. Here we have a discrete gradient, and a discrete divergence. They're all derivatives, and you think of the derivatives as being replaced by differences.

All these matrices are sparse, but if we take that fatal step of eliminating to get here,

the matrix is full. Because of this fact that the inverse of a sparse matrix tends to be a full matrix. And you don't want to work with that. So we don't want, that's my message, don't want $A^T A$ when this is not sparse. OK. So that's the point about this example, which in the notes on the website is the section called the saddle point Stokes problem. OK. All right. So now we have both a specific example of this Stokes equations, three equations really because this is two equations for v_1 and v_2 and this is the third equation, the constraint. OK. So we have both a specific example and a general case.

So we're now going to work with that matrix as a whole, we're not going to reduce it to $A^T C$, although in principle we could. And the question is one of stability. So like stability is a question that didn't come up for our minimum principles when there were no constraints. And maybe I'll just put a little mention why. So this is the first time we've had to face the question of stability. What do I mean that S inverse is under control? I don't mean that it's easy to compute, the inverse of that block matrix. But I have to know that the matrix is invertible. And stability is always like one slight step more than just being invertible. It means it's invertible and you have some control of the inverse. You know, it's not just on the verge of disaster. So we want to know about the inverse of that matrix. That's essentially it, we want to know about the inverse of that matrix. And in the process, we really want to know about the inverse of this matrix. So we may not want to compute it, that was my point, but stability is telling us we have here a matrix that's not positive definite. That the key issue here. Our saddle point matrix is not positive definite. It has positive eigenvalues and negative eigenvalues. Let me just take some examples. OK.

So here's an S . So I have a positive thing there. Non-0 stuff there. There is a simplest example I could think up. That matrix is not positive definite. It's determinate is minus 1 actually. But I mean they're safely away from 0. it's not singular or close to singular either. So that two by two example no problem. Well I could study the invertibility of this particular matrix, that would be solving differential equations. In other words I could study the case RP and V under control if I have a bound on f . Well they are.

But the issue comes when I go to finite elements, when I look at a sub space. so have to try to make that clear. What I plan to do to solve a continuous problem is to discretize it. This is my continuous matrix, block matrix, symbolic of the differential equations. And it's OK, analysis would show that. But the question is if I approximate V by some combination of finite elements, let me call it V_h . If I approximate the pressure by some pressure element P_h , then I'm sort of projecting the problem down into these finite dimensional, finite elements sub spaces. And of course f will get projected down to f_h . And the matrices, these operators, will turn into finite as far as matrices. And the question is that projection safe? So that really it. It's the stability. S inverse is under control, this is OK. But is the projection of S inverse under control? Is the projection of S . So if I project it, is the inverse of that under control? That's the question. What comes up in applications, this is a projection of the continuous problem, the differential equation, onto finite element spaces, finite element trial function. OK.

So we've spoken a little about finite elements in this course, and more in 18085, and you're probably have seen them elsewhere. You know that we can approximate velocities. Here would be a really important example, because it's an especially simple finite elements. If I take my region, suppose it happened to be that rectangle, I divide that rectangle into small squares, there's a small square. I'm using squares rather than triangles. And in each square I make some low degree approximation, low degree trial function, for the velocities and the pressures. And the question is what functions can I use? For example, one I would love to be able to use would be, so in each square I would like to be able to use piecewise constant pressures. Piecewise constant pressure, so p_h equal constant in each element. Element is the square. So a different constant in each element. So that would give me my vector of p_h s. It would be one component for every little square. And now what about V_h ? What trial function would I like to use for the velocities? Constant velocities wouldn't be good, that's further than I'm willing to go. I'll go piecewise linear in each square. Now piecewise linear in each triangle meant that it had to form a plus b_x plus c_y . Strictly speaking that's what I should mean by piecewise linear with a different coefficient abc in each piece but because we're in squares now, we've got four

nodal values, and we want a fourth coefficient. This was appropriate on triangles, on a square we really include the bilinear term dxy . OK.

So we have four coefficients which determine and can be determined from the four values of the node. You see that I'm describing the whole space. Now I cut the region into more squares, I've described a way of discretizing the differential equation, the weak form of continuous problem. When I plug in these functions would give me a matrix problem, a discrete problem, that will have this form. And the question is do I have any control? So it will give me this projected matrix now, finite dimensional instead of a continuous problem. And the question is Have I still got any control? Well the answer for this particular choice of pressure space and velocity space is no. Unfortunately the matrix S that appears instead, the C inverse h shall I say it would be like our $K2D$ inverse. That was pretty close of what it would be. And the Ah and the Ah transpose h and the 0 that matrix has a famous null vector for this specific example.

I'm taking big jumps here, because I just want to say the main point. The main point here is that a null vector appears in this problem. S fails to be invertible, and that null vector is like finite element people know its name. It has a couple of names actually, maybe checkerboard. The null vector is a vector when the pressure is 1 minus 1 , 1 minus 1 in a checkerboard pattern over the whole region, and a corresponding velocity. Well the trouble is Ah times this P is 0 . That's what kills you. It turns out that A times that particular pressure is 0 . And if you lose control of A that way, you've lost control A transpose CA . You've lost everything.

So I guess coming back to my opening comments. Here's a case in which we have a physical problem, but it falls into a pattern, where the mathematics actually decide are we able to use the trial spaces that are most convenient. I say convenient rather than most accurate of course. If I wanted to improve the accuracy, I would go up in the degree, I would take pressures to be piecewise linear and velocities to be piecewise quadratic with more points. And I could check the stability or not, is the matrix invertible or not, and the answer might be better there. In fact the notes and any book on finite elements has a list. And I don't plan on making that list here. But

it has a list of press spaces, pressure element and velocity elements, on which the first guy that I've mentioned would be like Q of 0, degree 0 and velocities Q_1 degree 1, and the letter Q being used for quadrilateral. If I divided it into triangles, I would use the letter p for polynomial in that case. And this is a fail. This one fails.

And then another list would be how about Q_1 , Q_1 ? I don't know. I wouldn't bet much on Q_1 , Q_1 , and of course I can look at the notes to see what the answer. Or Q_1 , Q_2 that's probably got a better chance. You know people are inventing finite elements all the time, and they're having to check the stability or not, the uniform invertibility or not of S . OK. I just want to say that this one is so attractive, because it's so simple that people don't give up on it. They want to go with this, this is a fail, but so stabilized. That one is so attractive and, you know, specifically that it's this checkerboard mode that's in the null space. So you just think, how am I going to get that mode out of the null? And the stabilizing method is you stick a matrix here for purely numerical reasons, and it'll be a matrix that's exactly has this checkerboard pattern, so that the checkerboard is not any longer in the null space. Maybe I call it minus some small number times some E matrix, it's often written E . And α you keep as small as you dare. But the effect is that it takes this checkerboard pressure out of the null space. You see the problem was that without that the columns of A were not independent. A times P could be 0. And if those columns are not independent you're lost. Let me put the 0 back. Why was that a bad thing? I want to now sort of connect you to the matrix theory.

If I have this particular pressure and AhP is 0, so that tells me that those columns of Ah are dependent. Then the columns of S are dependent, right. If I have something in the null space of Ah , then I just put a pressure, this pressure, in the null space of Ah , and I just take the v to be there 0, then these guys don't do anything. And that A times that bad pressure gives 0, so I have vp , a velocity pressure, that's in the null space. OK. So the way it gets fixed is to stick in some small stabilizing factor there. OK. All right now that's the specifics of this particular example. Well I didn't give all the specific. If I were really on the ball I would show in detail why that checkerboard appears and kills this choice. I would check each of these other choices, and beside each one I would put fail or success. And when it fails, I would try to identify the

failure mode, so that if you wanted you could stabilize and get rid of that failure mode. OK.

What I'll do instead is to come to the general question here. So the general question is to deal with this matrix S . And I write it again $C^{-1}A^T$. I want to look if it's invertible? Can I get some bound on its inverse? What do I have to know? And I'll just emphasize here, what's the real problem? Because we saw it right there. The real problem is if the column of A are dependent. In other words i.e. in other words, $AP = 0$ has a non-0 solution. Then you're wiped out. I'll just repeat. If the columns of A are dependent, and then the columns of S are dependent, the matrix is singular, and you're finished. So somehow we have to get some muscle behind A . But remember A is a rectangular matrix. This C is m by m , this A that comes up in the constraint is m by n . It has, we hope, more rows than columns. But still the columns could be dependent, and in this situation they are. OK. So the question is we have to get some muscle behind A . And this is the famous condition so called in soup condition that you would find in Professor [? Batiste ?] finite element book. But maybe these notes kind of focus on the heart of the idea.

And again in some way it's conditioned on A really, and on the pressure spaces, and the velocity spaces that you choose. OK. So these you could really say are A_h , P_h , and V_h , but I'll skip writing h 's on everything now. So I'm down in the discrete case, but I have a sequence of discrete problem which are growing as h gets smaller, and the question is do I have a fix β that control this? And it turns out that this the requirement this in soup condition says that for every pressure there is a velocity city, every discrete pressure now, there is a discrete velocity such that this holds. So it's sort of forcing A to stay away from 0. This β is a fix number, and this sort of is the right normalization that goes with it. So you see why $v^T A$ and P ? You know if I look at the quadratic form, $v^T P$, you know, the quadratic form with the matrix is always multiply by a vector here multiplied by its transpose here, where is the A term? A will multiply P , and it'll be multiplied by v^T . In the quadratic part, in the energy, is this $v^T A P$. And that's what we can't let go to 0. Because If that part fails us, if we lose these and we have

a 0 there, this part on the sub matrix can't save us. Right.

So before I say anything more about that. let me just emphasize what stability was in the symmetric case. So suppose I have a positive definite symmetric matrix. it's got some eigenvalues, λ_1 to λ_n . OK. What you see on this board is going to explain why stability is automatic if I'm dealing with positive definite things. So suppose I project? Let me just project in the direct way. I'll throw away the last row and column. Suppose I throw away the last row and column. That part is now gone, so these are the eigenvalues of K . And of course being positive definite, they're bigger than 0. OK. So that was the whole matrix, but now what I'm going to do is I'm going to project on the first $n-1$ rows and columns. So I'm going to look at the sub matrix, K_{n-1} . And it has some eigenvalues. Are they the same as those? No way. And you see I'm not in this situation, this was not positive definite. So I'm looking here at the positive definite case. This has some eigenvalues μ_1 up to μ_{n-1} . These will be the eigenvalues of the sub matrix, K_{n-1} . And linear algebra is just like the right place to look for some information on the connection between the μ s and λ s. They're certainly not equal.

Our first question is if I take a positive definite matrix with positive eigenvalues, I throw away the last row and column. Do I still have a positive definite? Is that sub matrix positive definite? And the answer is yes. The eigenvalues are all positive. And more than that, how about that smallest eigenvalue? Because the eigenvalue is measuring like how close are we to disaster, to singular. Can I say anything about that smallest eigenvalue? And the answer is yes. The smallest eigenvalue of K_{n-1} is never below the smallest eigenvalue of the full matrix. So if the full matrix is OK, all the projections, all the sub spaces, all this sub matrices are even more OK. And up at the other end, at the top, the top eigenvalue in sub matrix never passes the top eigenvalue in the big matrix. So you see at both ends I have what I want, I have control. If I have some range of eigenvalues for the big matrix, then all its projections are inside that range. And therefore the condition number, which would be the ratio of this to this, is smaller than the condition number of the whole matrix. So the condition number of K_{n-1} is less or equal the condition number of K . OK. And that's exactly why working with positive definite matrices is so

comfortable. The stability is just automatic.

But here we're not working with positive definite matrices. The stability is not automatic, and we need this in soup condition. And I don't know whether to try. Maybe I'm a little hesitant to go through the steps that lead from there to there, because this lecture is already asking you to put a lot of things together without details. And the notes do the best job I could do in making those steps simple and clear. And this is what you actually check. You check this condition for any proposed finite element, and any different problem. It wouldn't have to be the Stokes problem. Other problems would give us a constraint matrix A . A sort of stiff matrix C inverse. And we could check the condition again. OK. So maybe I'm going to leave that statement for the notes. OK. So that's this in soup condition.

And of course, you see right away that if A had dependent columns, which is the big risk, if A had dependent columns that would mean that Ap is 0 for some p . And then there would be no v . No v could possibly be found that would save it. If A had dependent columns, that's bogey man that we're avoiding above all. If A had dependent columns, then there would be a solution to Ap equals 0, this left side would be 0, the right side wouldn't, and the in soup condition would fail. But the in soup condition tells us well what about if there are pressures, that where Ap get smaller and smaller as the mesh size changes. And the question is can I find a v and so on. So that's the full condition but not to lose sight of the fact that the central question, the key question is the independence or dependence of the columns of A . OK. Good. So let's see, maybe that's as much as I want to say about the Stokes example, and what it leads to in a finite element world. I guess I'm going to stay for another lecture or two with these minimizations. So maybe I'll just anticipate slightly, the start of the next lecture. Because I'm quite pleased about getting this next topic into the series of lectures.

Let me say what it is. So it's going to be on minimization of an Au minus b square. OK. Just as this lecture is seen, the possibility comes up at the columns of A not independent, that this problem is in trouble. So I want to regularize this problem. So this is a major application in which A , maybe A transpose is the easiest thing to say,

A transpose A is nearly singular or just plain singular. I'll say or singular. How do we rescue the problem? Because numerically and theoretically it's a disaster as it stands. OK. It's ill posed. So let me introduce that word ill posed problem. And let me mention, since I'm just using these last few minutes as a kind of getting the big picture, where do ill posed problems come from, where does these things come from? They come typically from what are called inverse problems.

So the application is to inverse problems. What's an inverse problem? An inverse problem for a differential equation is you know some inputs and outputs, you know some solutions, but you don't know the equation. You don't know the coefficient. So you're trying to determine what are the material properties by looking to see what the outputs are for experimental inputs. And very frequently you don't have enough inputs to really determine what these -- so it's inverse problem to find the coefficients of the equation given some information on its solutions. some measurements. And of course they're probably noising measurements. I mean this is an enormous source of applications growing an importance, and how do we deal with it. Well you regularize it. So this is typical of this type of problem. You add on some multiples of something that's safe, like for example u square. So the results is here that this A transpose a that comes from this problem there's an α times the identity coming from this problem. I mean it's an ordinary least square problem, and it leads us to the matrix A transpose a , comes from the first part. But I'm just talking about the standard normal equation. It will produce an A transpose a from here. But from here it will produce an α times the identity.

And of course just the way this minus E worked up there, here I'm doing it even more directly, I'm pushing the problem away from 0. This α , this parameter and the choice of that parameter, is the million dollar question in this whole subject. But it's purpose is to regularize the problem, to take his nearly singular matrix and push it up. See this matrix, A transpose a , never has negative eigenvalues, the problem is 0 or very small, small positive. So by adding on αI , I push those away from 0. Over there we had the problem of eigenvalues being positive or negative, and by pushing in one direction we might just bring a negative eigenvalue right on to 0.

That's not our problem in this upcoming lecture. In this upcoming lecture we know that nothing is negative, but we need to make it properly positive. OK. So that's what's coming.

Can I just say the application that I want to describe is the first one I've given in life sciences, and I hope not the last. I don't know if any of you can suggest, and if you can an email will be very much appreciated. Applications of this whole framework in like sciences. So you'll see a particular application of this ill posed problem in trying to determine gene structure from expressions.

OK. So that's what's coming Friday. And I realize here I'm talking about material that probably is not going to go in your projects, because of the timing. I'm thinking that you're projects are more likely to build on what you had in project one, and the basic material about solving large systems and basic minimizations. OK. Thank you. See you Friday.