

The following content is provided by MIT OpenCourseWare under a Creative Commons license. Additional information about our license and OpenCourseWare in general is available at [OCW.MIT.edu](http://OCW.MIT.edu).

**PROFESSOR:** OK. So I'll start a little formally. That this is 18086 in the spring semester of 2006. And we're lucky to have it videotaped for OpenCourseWare.

So let me begin by remembering the main topics from what you might call the first half 18085 of fall course. And then most important to us, so it gets a star, are the main topics for this course. So 18.085 began with applied linear algebra. The discrete equations of mechanics, and physics and engineering. And the type of matrices that involved, so we learned what positive definite matrices are. Then the center of the course was differential equations, ordinary differential equations. So that 1D, partial differential equations like Laplace. That was 2D, with boundary values. So that lead to systems of equations. And then the third big topic was Fourier methods. All of Fourier ideas: series, integrals discrete transform. So that's like the first half of a course. Well that's 18085.

OK. And here we are. So this course has also three major topics. The first, the one we start on today is differential equations. That start from initial values. So I'm thinking of the wave equation, where we're given the displacement of velocity. The heat equation. Black-Scholes equation, which comes from finances is a version of the heat equation. Ordinary differential equations that's going to be today. And non-linear equations, Navier-Stokes ultimately. But not heavily Navier-Stokes. So this is where we begin. And then a second big topic is how to solve. Because this is course is really applied math and scientific computing. The second big topic is how do you solve a large linear system  $ax = b$ ? Do you do it by direct methods on limitation with renumbering of nodes. And there are lots of tricks that makes elimination very successful. Or do you do it by iterative methods? Multi-grid is one. So this is really modern scientific computing we're talking about. We're really at the point where people are computing.

And then the third topic which is a little different is optimization, minimization. Linear programming would be one example, but among many, many. So that's a major area in applied math. OK. So this is our topic differential equations. And I was asked before the tape started about my own background. And my thesis was in this topic. Because a key question will be stability. Is the difference method stable. You'll see that that's a requirement to be any good. You really have to sort out what, and that usually puts a limitation on the time step. So we're going to have a time step. And it may be limited by the requirement of stability. And the fact that stability and convergence success of the method are so intimately linked. There was a key paper by Lax and Richtmyer that we'll look at. It's known now as the Lax Equivalence Theorem, stability and convergence. We'll see it in detail.

Actually when I was a grad student by good chance there was a seminar, and I was asked to report on some paper and it happened to be that one. And looking back thank gosh I was lucky. Because that paper, the Lax-Richtmyer paper with the Lax Equivalence Theorem set the question of stability. And then years of effort went into finding tests for stability -- how do you decide is the method stable. So we'll see some of that. OK. Since then I've worked on other things well wavelets would be one that relates to Fourier. And other topics too. And lots of linear algebra. But it's a pleasure to look back to the beginning of [UNINTELLIGIBLE].

OK. So that's the OpenCourseWare site which does have the course outline, and other information about the course. Some of which I have mentioned informally before the class. This is our central website with no decimal in 18086, which will have notes coming up, math labs things, problems. That's our content. It will be the 086 page. OK. So that's the course outline in three lines. And then web page and a handout will give you half a dozen sub-headings under those. OK. So that's the course. And actually it's the center of scientific computing. There would be other courses at MIT covering material like that, because it's just so basic you can't go without it.

All right so now I'm ready to start on the first lecture, which will be ordinary differential equations. And I won't always have things so clearly and beautifully

written on the board, but today I'm better organized.

So here's our problem ordinary differential equations. So we're given initial values  $u$  at  $t$  equals 0. OK. So there's always an initial value here. The question is what's the equation that makes it evolve? OK. So often we'll think in terms of one equation. The unknown will be  $u$ ,  $u$  of  $t$ . Or in reality there are typically  $N$  equations.  $N$  might be 6 or 12 or something in a small problem. But also  $N$  could be very large. We'll see how easily that happens. OK. And often to discuss stability or understand a method and try it the linear cases the natural one. So I'll use  $au$  with a small  $a$  to make us think that we've got a scalar, or a capital  $A$  to indicate we're talking about a matrix. So these are linear. These are likely to be not linear. But there's a natural match between the number  $a$  their corresponds to the partial of  $f$  with respect to  $u$ . And of course we see it. If that was  $f$  then its derivative is indeed  $a$ .

And in this matrix case, there would be a Jacobian matrix. If we have  $N$  right hand side, because I have  $N$ 's, and the matrix that comes in is the derivative of right hand side and equation  $i$  with respect to unknown  $j$ . So it's a  $N$  by  $N$  matrix, and of course this is the case where it's a constant. Yeah I should have said, these are not only linear but constant coefficients. So we know of course the solution, it's an exponential  $e$  to the  $at$  times  $u_0$  the initial bunch. So we know everything about it. Nevertheless, it's the best test case for difference methods.

And let me just comments that we do have to pay attention to the possibility that  $a$  could be complex. Mostly I'll think of  $a$  as a real, and often negative. Often equation will be stable. A negative  $a$  means that  $e$  to the  $at$  decreases. A negative definite matrix, negative eigenvalues, mean that the matrix exponential is the case. And if that matrix is symmetric, those eigenvalues are all real, that's a key case.

Symmetric matrices have real eigenvalues, and they're the most important most attractive class. And a negative definite matrix might come from a diffusion term like second derivative,  $d^2 u / dx^2$ . But a convection term  $du / dx$  from the first derivative will not be symmetric. In fact maybe it'll be anti-symmetric. It will push the eigenvalues into the complex plane. And therefore we have to pay attention to the possibility of complex  $a$ . As you know, the whole point of eigenvalues is that we can

understand matrices to a very large extent by diagonalizing, by reducing them to  $N$  scalar equations with the eigenvalues as the little  $a$ 's in those scalar equations. So anyway that's the set up.

Now I want to say something about methods. I want to introduce some of the names. What am I thinking here? This topic, today's topic, of ordinary differential equations is not going to last long in 18086. I might still be discussing it tomorrow. The variety I mean. But not much after that. We'll soon be on to partial differential equations. So I want to sort of tell you about ordinary differential equations well in kind of an organized m so that you know the competing methods. And also so that you know this key distinction. Nonstiff is a typical equation,  $u' = -4u$  would be a nonstiff still, completely normal equation. And for that, those are they the relatively easy ones to solve. We can use explicit differences. And I'll say what that word explicit means. What I want to do is just like help you organize in your mind.

So nonstiff are the average every day differential equations. For those we can use explicit methods that are fast. And we can make them accurate. And explicit means -- so this is the first meaning -- that I compute the new  $U$ , and I'll use capital  $U$ , at time step  $n + 1$ . So that's capital  $U$  at the new time. Natural to call that the new time. Can be computed directly from  $U$  at the old time, maybe  $U$  at the time before that, maybe  $U$  at the times before that. In EE terms it's causal. The new  $U$  comes explicitly by some formula that we'll construct from the previous computations. It's fast. Of course, I not only use  $U_n$  but I used  $f$  of  $U_n$ . So I should say from  $U_n$  itself and from  $f$  of  $U_n$  at time  $t_n$ , and similarly for earlier times. You see how fast that is? Because the only new calculation of  $f$  is this one. The previous step produced  $U_n$ , and the new step, this step, we plug it in to find out what the slope is. And that may be the expensive calculation, because  $f$  could involve all sorts of functions, all sorts of physical constants. It can be complicated or not. but if it is, we only have to do it once here. that's explicit.

Now what's the opposite? Implicitly. So the point about implicit methods is the difference equation, the formula, involves not just  $U_{n+1}$ , the new value, but also the slope at that new value, at that new unknown value. So it means that an implicit

equation could be non-linear. Because if  $f$  is not linear, then this term could involve cosines, or exponentials or whatever of the unknown  $U_{n+1}$ . So that way you can't do it as fast. So this is definitely slower per stiff. But the advantage and the reason it gets used for stiff problems is that on stiff problems it will allow a much larger stiff. So it's this constant trade-off of speed versus stability. These are less stable as we'll see, and we'll get to see what stability means. These are slower but more stable implicit methods. OK. I didn't get to say what stiff is. I'll say that in a moment. OK.

So explicit versus implicit.  $U_{n+1}$  immediately or  $U_{n+1}$  only by maybe using Newton's method to solve an equation, because  $U_{n+1}$  appears in a complicated formula. And we have an equation to solve. OK.

So now this part of the board begins to identify different methods. And I'll follow the same two column system that non-stiff versus stiff. So this will be explicit and those will be implicit. OK. So that one method that occurs to everybody right away is Euler's method. And that will be the first method we'll construct of course. So that's the first idea anybody would have. It has the minimum accuracy. It's first order, the order of accuracy, will be an issue for all methods. And Euler has  $p$  equal 1, order  $p$  equal 1, first order. So it's too crude. I mean if you wanted to track a space station with Euler's method, you would lose it real fast, or you would have to take  $\Delta t$  so small that it would be impossible. So Euler's method is the first one you think of, but not the one you finish with. And similarly on the implicit side, backward Euler, we'll see is again the first implicit method you think of. But in the end you probably want to do that. OK. So those are two specific methods that are easy. The first ones we'll write down.

Now then comes families of methods, especially these Adams families. So Adams-Bashforth is explicit, Adams-Moulton are implicit, and the coefficients are in books on numerical analysis and will be on the web. And I'll say more about them of course. And see the first ones. So what Adams-Bashforth how does it differ from Adams-Moulton? OK. So those are multi-step methods. Number two was multi-step methods. To get more accurate we use more old values. So Adams-Moulton

and Adams-Bashforth, or especially Adams-Bashforth, would use several old values to get the order of accuracy up high. OK.

Now this third category. Of course actually Euler would be the first of the Adams-Bashforth, and backward Euler might be early in Adams-Moulton. Or maybe backward Euler is early in backward differences. OK. So what I want to say is Runge-Kutta is like a different approach to constructing methods. You might know what some of these already. And it's a one-step method. That's a method that just produces by sort of half-steps you could say. It gets to  $Un + 1$ . But it's explicit. And the code in math lab the code ODE45, or maybe I should say the ODE45, is like the work course of ODEs. I guess I'm hoping you'll try that. You'll find ODE45 and I'll try to get on to the web. But you could just easily discover the syntax to call it, and apply it to an equation. Yeah I'll get examples on the web, and please do the examples.

So the 4 or 5 typically means that it use a 4th order Runge-Kutta, so that's pretty accurate. To get up to a 4th order means the error after a up a time  $t$  equal 1 say is  $\Delta t$  to the 4th. So by cutting  $\Delta t$  in half, you divide the error by 16. And then it also uses a fifth order one. And maybe I can point to these words. What makes ODE45 4 or 5 good, successful. I mean how does it work? It slows down or speeds up. It speeds up when it can, it slows down when it has to. And speed up means a bigger  $\Delta t$ , slow down means a smaller  $\Delta t$  for the sake of stability or accuracy. Yeah. So if the equation is suddenly doing something, if the solution is suddenly doing something, you know, important and quick, then probably at that period  $\Delta t$  will get reduced automatically by ODE45. It'll cut it in half, cut in half again, half again. Every good code is constantly estimating using internal checks to estimate what error it's making. About the error. Just a quick word about the error. So ODE45 it will, unless you tell it otherwise, the default accuracy that it will adjust for is relative. So ODE45 will have a relative accuracy. So can I just squeeze in a few words of  $10^{-3}$ . It'll shoot for that. And an absolute accuracy of  $10^{-6}$ . So weird will try to keep the error. It will plan to keep the error, and it'll tell you if it has a problem below  $10^{-6}$ . And you could set that differently. OK. I hope you experiment a little with ODE45. And you can either plot

the solutions, some exponential. I mean push it. Like let  $\Delta t$  be pretty big, and see where it breaks down. Well I guess ODE45 is engineered not to break down. If you try  $\Delta t$ , it'll decide on what  $\Delta t$  it can do. So we'll also code, just quickly, some methods by ourselves, and see for a large  $\Delta t$  what problems could happen. OK.

And then backwards. This says backward differences. That's the category of implicit methods for stiff equation, and backward Euler would be the very first. And ODE15S is the code that is the most used code perhaps for stiff equations. And it will do the same thing, it will vary  $\Delta t$ . It will vary the order. So if it has to slow down, it may slow. And if things are happening too quickly, it may change to a low order, safe, secure method. When things are, you know, when the satellite is buzzing out in space and it can take giant steps or it can have high order accuracy, say in astronomy of course they would go up to 8th order and high tracking estimating where stars would go or planets, these will do that automatically up to a certain point. And we could write codes, and of course people have, for Adams-Bashforth and Adams-Moulton varying  $\Delta t$  varying  $e$ .

I guess here's a comment. If I had given this lecture yesterday, I would have emphasized number two, Adams-Bashforth and Adams-Moulton, because books do it and I basically learned this subject from the major textbooks on ODEs. But I had a conversation with one of the computational scientists in the math department, and that changed the lecture. Because I learned that although books tend to discuss those at the Adams method, in practice these methods three -- Runge-Kutta and backward differences -- are the most used. And that's reflected in the fact that these two major codes that are available in math lab are in the number three category. OK.

So now that's a picture with name but not details right. I And I haven't even said what stiff means. So somewhere I've written something about stiff. Yeah. OK. So if I had only  $e^{-t}$  in the solution, then the solution would decay at that rate either the minus  $t$ . That time step would adjust to keep it accurate. And if I have only  $e^{-99t}$ , then again that of course is a faster decay, much faster decay.

So the time step would adjust to be much smaller, probably about  $1/99$  or something. Anyway much smaller to capture within each step the correct behavior of either the minus  $99t$  and no problem. So in other words, it's not the minus  $99$  that makes it stiff. If it was only the minus  $99$  -- in other words if I had a equal minus  $99$  up there, I wouldn't call it a stiff equation. What makes it stiff is the combination here. You see what's going to happen as time gets anywhere, this is going to control  $u$  of  $t$ . The decay of  $u$  of  $t$  is going to have that time constant  $1$ . But this is going to control. So I'll say but this would control  $\Delta t$  for explicit methods. And I just picked minus  $99$ , but it could have been minus  $999$  or far worse. So it could be very stiff.

This word stiff and identifying this class of problems, which now is familiar to everybody who has to compute, didn't come, you know, with Gaussian and so on. It came much more recently. But it's become very important now to distinguish stiff from non-stiff. So where does stiff problems arise? Well you can see how they arise in chemical processes with very different decay rates, biological processes, control theory, all sorts of cases where there's a big dynamic range of rates. And they also arise in system, because the eigenvalues could be minus  $1$  and minus  $99$ . I can easily create a matrix that has those two eigenvalues. So it would be a system. Let me create such a matrix. I think probably if I put minus  $50$ s on the diagonal, that gives trace. Sum down the diagonal is minus  $100$ . And that should equal the sum of the eigenvalue. So, so far so good. And now if I want these two particular numbers, I could put  $49$  here. So that matrix has those two eigenvalues.

I mean this is the kind of matrix, well you might say ill conditioned would be a word that you hear for a matrix like this. The condition number is the ratio. So the condition number of this matrix would be the ratio  $99$  over  $1$ . And that condition number is a number that math lab computes all the time every time you give it a matrix problem like this. It estimate, not computes. Because to compute the condition number requires solving an eigenvalue problem, and it does not want to take forever to do that exactly. But it gets an estimate. So that's only a moderate condition number of  $99$ . That's not a disaster, but it makes the point. OK. So that's



what stiff equations are. OK.

Now I got one more board prepared. I won't have this everyday, but let me use it while I got it. OK. It's only prepared in the sense of telling us what we're going to do. So this is what's coming: a construction of methods, what stability is about and what's convergence. So let me begin, let me get Euler constructed today. And backward Euler. So Euler's method will be the most obvious method, equals  $f$  at  $U_n$ . So in the model case it's  $aU_n$ . So that's Euler. So that tells us then that we could rewrite it.  $U_{n+1}$  is explicitly  $1 + a\Delta t$ ,  $U_n$ . Right. I've moved up the  $\Delta t$  and I moved over the  $U_n$ , and it's simple like that. Let me stay with Euler for a moment. After  $n$  steps, every step just multiplied by this growth factor. Let me refer to that as the growth factor even if it's as I hope smaller than 1. So  $U_n$  after and  $n$  steps is this thing has multiplied  $n$  times  $U_0$ . And what is the stability now? Stability is suppose  $a$  is negative typically. That means often we'll just think of that model  $a$  negative, so that the differential equation is completely stable. Right. The  $e$  to the 18 with a negative is perfect. The question is this one perfect? So stability will be is this number below 1? That's the test. Is that we could argue about is probably going to let it equal 1. Maybe I'll let it equal. So  $1 + a\Delta t$  smaller than 1. That'll be the stability requirement for forward Euler.

And what's the  $a\Delta t$ ? What's the boundary? What's the critical value of  $a\Delta t$ ? Remember  $a$  negative. How negative can it be, or how negative can this combination be? You see it's a combination  $a\Delta t$ . So let me put above here what the stability condition is. So the stability condition on Euler is going to be -- so do you see what it is? How negative can  $a\Delta t$  be? Go ahead you. Well let's see. So if  $a\Delta t$ , this is a negative number, so this is dragging us down from 1 always. Right;  $a\Delta t$  is like subtracting away from 1. So we're here with the 1. And  $a\Delta t$  is pulling us this way. And how far, at what point are we going to meet instability? Yeah. When we hit minus 1. This is the  $1 + a\Delta t$  that I graphed. So it starts at 1 with  $\Delta t$  equals 0. And as I increase  $\Delta t$ , it move this way. And eventually it hits there. And the limit will be at a  $\Delta t$  equal minus 2. Because if it carries me 2 from 1, that carries me to minus 1. And if I go further, I'm unstable. And you could easily check. That Euler will blow up like crazy if you just go a little beyond that limit.

Because you're taking powers of a number bigger than below minus 1. OK. So there's the stability limit for Euler. And that's a pretty unpleasant restriction. We will have other stability limits. We'll have some stability limits over here, but we're looking for numbers better than 2.

This Adams-Bashforth will give us numbers worse than 2. We'll see those methods first thing next time. OK. If I drew a picture then. Let's see have I got another board here. Nope. If I drew a picture of good Euler, so Euler doing its job, the true solution is decaying like that, and what does Euler do? Forward Euler says, it looks here, it looks at the slope, that's all it knows, it goes maybe that far. That's a small  $\Delta t$ . That's an OK method. You see it's not very accurate of course. It's lost all the curvature in the solution. So this was the true solution  $e^{at}$ . And this is the  $1 + a\Delta t$  plus a  $\Delta t$  to the power  $n$  over  $\Delta t$  times. Well this was only one step, so that would just be  $1 + a\Delta t$ . OK. where the bad, the unstable one, if I just try to draw that, the true solution is  $e^{at}$ . I started with that value and the right slope. But I take two big  $\Delta t$  and I'm below minus 1. So that was a case of a too large  $\Delta t$ . OK.

Now I've got one minute left to mention backward Euler. Well that won't take me long. I'll just change this to  $U_{n+1}$ . So I'm evaluating the slope at the new time. So this is now backward Euler or implicit Euler. OK. So that changes this equation. Oh let's find out the equation for implicit Euler. OK. I only have one more moment, I'll just pop it in this little corner. So now here's backward Euler. Alright. You can tell me what backward Euler is going to be. So  $U_{n+1} - U_n$  over  $\Delta t$  is  $aU_{n+1}$ . So let me collect the  $U_{n+1}$ 's. So I have one of them. And then minus  $a\Delta t$ . Right. And when I bring it over to the left side, and here I just have  $U_n$ . OK. So what happens at every step now? I divide every step as a division by this. So I could bring this into the denominator. So  $U_n$  is that division times  $U_0$ . And you see the stability? What's the size of this growth factor? Remember I'm thinking of  $a$  as a negative. I'm dealing with a stable differential equation. And so what's the main point about this growth factor, this ratio in parentheses? It is? If  $a$  is a negative, what do I know about that number? Think of  $a$  as minus 2 let's say. So then I have 1 over

1 plus 2 delta t, because it's minus a there. And it's going to be smaller than 1. Right. It's going to be stable. So Euler is actually absolutely stable, backward Euler. Backward Euler has this property of a stability. There is no stability limit on negative a, no problem Or even pure imaginary a, no problem. Always stable. So that shows how implicit methods can be greatly much more stable than explicit, just by comparing backward to forward to backward Euler. But just to remind you the price. The price is that this equation has to be solved -- well it wasn't any trouble in linear case. In the linear case we're just inverting a matrix for a linear system. So backward Euler for linear system, big linear system, is a matrix inversion, well a matrix linear system to solve. But for a non-linear problem, it's serious work. And yet you have to do the work if the explicit method gives you such a small delta t that it's impossible to go with.

OK. Thanks for your patience on this first lecture. So obviously I'm going to have one more discussion of ODEs, in which we construct these methods, see their stability and their convergence. And then we move on to the PDs.