

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

18.085 Computational Science and Engineering I, Fall 2008

Please use the following citation format:

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

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

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

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

18.085 Computational Science and Engineering I, Fall 2008
Transcript – Lecture 9

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high quality educational resources for free. To make a donation, or to view additional materials from hundreds of MIT courses visit MIT OpenCourseWare at ocw.mit.edu.

PROFESSOR STRANG: Ready? So let me recap the important lecture from last time that gave the framework that we'll see in Chapter two and Chapter three, really a major part of the course. And our example was a line of masses and springs. So that's like, the first example to start with. And you remember, there were three steps. If we want to know how much the masses displace we have to get to the springs. Difference in displacements gave the stretching of the springs. Then comes the material law, Hooke's Law in this case with a matrix C that led to CAu . And then finally, those spring forces are balanced by the external forces. That brought in A transpose. So that's the picture to look for. And I just mention it again because it's so central.

So we'll be doing other examples of that. But I wanted to stay with springs and masses for today. To do another type of problem. A problem in which time enters. A problem in which we're not looking for the steady state. But it's Newton's Law is going to come in. This is actually Newton's Law. You recognize mass. You recognize acceleration. And the force on the spring is $-Ku$. So when $-Ku$, when the force came over to that side, that's the equation we're looking at.

So it's like time-out for a discussion of the very important subject of time derivatives. How do I solve initial value problems now? So I have some starting time, I'm given $u(0)$, the starting position of the springs, and I'm given the velocity at zero. So I'm given two facts at zero. That's very different from the steady state problem where you're given stuff around the boundary. Here we're given stuff at the start. Initial values instead of boundary values. So what does this mean? It means that maybe I pull the springs down and I let go. And what do they do? They oscillate. And you will have studied this equation. I hope you've seen that equation. Because I think of it as the fundamental equation of mechanics. In fact, I've even used those words. The fundamental equation of mechanical engineering. Well, it's Newton's Law for a system.

So the matrix M now has two masses. In this case, it would be a two by two system. The matrix M would have two masses. Suppose mass one may be like, nine and mass two is one. So copying what our problem is we'd have a nine and a one multiplying u_1'' and u_2'' . Can I use prime for a derivative? Rather than dots, because I'm never too sure if the dot is there or not. So prime I can see. And then plus our Ku . And actually you know what K will look like for this system. It's fixed-fixed. Well, let me write down what K would look like. Just because we saw it at the very end. What it looks like if there are spring constants, c_1 , c_2 , and c_3 .

Can I just remind you what that will look like? So this will multiply u and give zero. So the mass matrix is diagonal in the problem. That's diagonal matrices, we certainly expect to be easy. Almost as easy as the identity matrix where we would have two equal masses. Here it's just that little bit more general with two different masses. And in here, the kind of thing that we created just at the end of the last lecture was something like $c_1 + c_2$ on the diagonal and a $-c_2$ and a $-c_2$ and a $c_2 + c_3$. So c_2 for the middle spring is the one that's really there with its full little element matrix, $c_2, -c_2, -c_2, c_2$. The c_1 part is only coming in at the top because it's connected to a fixed support. And c_3 is only coming in at the bottom.

Anyway, that's the matrix K . This is the matrix M . And how do we solve the problem? You've seen before and now ADINA Professor Bathe's code or any other code would offer, like two different ways. One way is because we have constant coefficients here we can expect a pretty clean formula for the answer. And because we're growing, we're oscillating in time, things are happening in time, we expect eigenvectors, eigenvalues to come in. Into that formula. I think it's worth just repeating that formula. How do you approach it? And then the second, the really serious issue is how do you solve equation like this or even more general, time-dependant problems.

I mean this is what finite element codes are created for. How do you study the crash of a car? So this isn't quite the equation for a car crash. What would be different in a car crash? There's a short section later in Chapter two called the reality of computational engineering. And the reality is, cars crash, people drop their cell phones. And those are very difficult problems to compute with. And of course, absolutely impossible to use eigenvectors because, well they're not linear at all. Right? If you crash two cars, if you want to study, everything's happening in like, hundredth of a second or something. But what's happening there is highly non-linear so it's very much more complicated. And you take thousands of time steps maybe within the crash time. And you're going to use finite differences or finite elements. So this is a chance to say something about this special equation for finite difference method.

It's really 18.086 that takes up these questions seriously. If I choose a finite difference method. See, the thing is with finite differences you've got many, many choices. There's only one way to go in step one. Eigenvectors, this thing has got eigenvectors, you find them, you're in. But for the much more general, typical problems that you'll be solving the rest of your lives, finite differences or finite elements, you've got lots of choices. And the issues that come up are the accuracy of the choice. Center differences often gives you that extra order of accuracy. Stability is something we have not seen yet. So what does stability mean? Stability means that as I follow my difference equation forward in time it stays near the true equation. You'll see by example. So some methods are more stable than others, some are completely unstable and unusable.

And then, of course, the other condition, another desired requirement is speed of calculation. So these balance each other. It's a beautiful subject. And let me just open the door to that subject today. May I start with the eigenvector solution. How would I solve this equation by eigenvectors? Well again, I'm going to look for special solutions. So let me write that equation down again. $Mu'' + Ku = 0$. I'm taking zero external force. So the springs and masses are just oscillating. Their total energy won't change, it's a closed system. The kinetic plus the potential energy will stay constant and we can show why. There's a lot in this section, by the way, section 2.2. More than I would be able to do in the lecture. But let me capture the key ideas.

So one key idea is the natural idea when we have constant coefficients. Look for special solutions. So let me say look for special solutions of the form-- well, let's keep in mind always the simplest model of all. Let me put it over here because I'll focus on it particularly, especially. The simplest case would be u'' with a mass normalized to one plus $u=0$. Just a single equation. So that's just a single spring. Single mass. So the mass is just oscillating up and down. And we all know that it's going to oscillate up and down, sines and cosines are going to come into here. In fact, we know that the solution to that would be, this could have cosine, this would have some $\cos(t)$'s and some, u could be a combination of $\cos(t)$ and $\sin(t)$. Right? And the A and B would be used to match the two initial conditions. So that's what we've got in the scalar case. This is one spring. n is one. We don't have a matrix here, just a number. But that's our guide to the matrix case.

So now what am I going to look for? I'm going to look for $u(t)$. I'll look for special solutions of the form \cos , they'll go at different frequencies. So the frequency has to be found. So that's the time dependence. And then some, if I'm lucky all the springs will be going together. So this x -- Am I going to use x for eigenvector? I think so, yes. Let's use x for eigenvector. So this is a vector, this is a constant vector. And this gives the oscillation.

Let me just already start with that. I've got two unknowns here, right? Two things that I'm free to choose. I'm free to choose the frequency ω and I'm free to choose this vector x . And I've like, separated out time. The way we've been doing it with an $e^{(\lambda t)}$. We used an $e^{(\lambda t)}$. So that was sort of right for a first order equation. Cosine is right for a second order equation like this. When is that a solution to my equation? May I just plug it in? So I'm going to just plug it in? Substitute is a better word than plug in, right? So shall I just plug it in? So I get M times that second derivative. So the second derivative of the cosine is the cosine with a minus $M \omega^2$. Is that right? Yes? Times what? So that's what came from the time derivative. Times \cos times this, $\cos(\omega t)x$. Two time derivatives brought down a minus ω^2 .

And the other part is just K times u , $\cos(\omega t)x$. And that should be zero. And this is supposed to be a good solution that works for all time. This is what I want to be zero. Do you see what's left? Kx from here. And putting that on the other side, $M(\omega^2)x$. Can I write the ω^2 that way? $(\omega^2)Mx$. Then if that is satisfied then the differential equation is satisfied. I've got a solution. So I'm looking for K and I'm looking for x and ω . And by the way, I could also have $\sin(\omega t)$ here. As well as cosine just the way over there. Sines and cosines both possible. So $\sin(\omega t)x$. It would be exactly the same except that it'd be a $\sin(\omega t)$ that I'd be dividing out. And again, I'd come back to this. This is key problem.

And do you see that somehow here we have an eigenvector x and an eigenvalue ω^2 ? An eigenvalue ω^2 , right. But there is one little twist. It's not quite our standard eigenvalue problem. What's the extra guy that's present in that box that we don't usually see? M . M is the new person there, new thing. The mass matrix. Which, if all masses were one, the mass matrix would be the identity. We would be back. We would just have our standard problem. I just have to say a word about the case when the mass matrix is there. It's still an eigenvalue problem. If you like, I can bring M inverse over here. I could write it as $(M$

$\text{inverse} \cdot K)x = (\omega^2)x$. So I'm looking for the eigenvalues of $M \text{ inverse } K$. That's really what I'm doing. The eigenvalues and eigenvectors of $M \text{ inverse } K$.

But I'm always trying to be like, aesthetically right. And what's $M \text{ inverse } K$, has just a little something wrong with it. Which is? It's not symmetric. If I take my matrix K , which is beautifully symmetric, and my M , which is beautifully symmetric, but when I do $M \text{ inverse } K$, do see what'll happen? The inverse of that matrix will have a $1/9$. When I do $M \text{ inverse} \cdot K$, that row will get divided by nine. And the second row won't change because I've got a one. So it just like, spoiled things a little. You can deal with it. But actually, in some way it's better to keep it right.

And MATLAB is totally okay with that. So MATLAB would use the command `eig`, and other systems too, of K , M . Just tell MATLAB the two matrices. Then it solves that problem. It will print out the, well if you just typed `eig`, it would print out the eigenvalues. If you ask it for the eigenvector there matrix, it'll print those too. So that has the grand name generalized eigenvalue problem. Generalized because it's got an M in there. I don't know if you've met these problems where there could be an M . It's just a natural, and it's not really a big deal, particularly when M is a positive diagonal matrix. It's not a big deal at all. It's the same codes essentially finding the eigenvalues and eigenvectors.

So suppose we have them. We expect n positive eigenvalues, and those'll be the ω^2 's. And each one comes with its eigenvector. And complete set, everything is good. We're talking about symmetric positive definite matrices. Notice that K is positive definite. That was what our whole last weeks have been about. And M is obviously positive definite. So it's a complete, it's a perfect set-up for the eigenvalue problem. I just want to think through the final step. After we find the eigenvalues, λ , the ω^2 's, then we know the ω 's and we find the eigenvectors, x , how do you write the answer?

So what have I done so far? I've looked for some special solutions. And I will find them. So I've found these ω 's. They can go with sines or cosines, I found the x 's. Now what's the general solution? If I have these solutions to my great equation there, what's the general solution? $u(t)$. So this would be the big picture. What can I do in linear equations? Linear combinations. That's what linear algebra is all about, linear combinations. So I could take a linear combination of these cosines, so $\cos(\omega_1 t)x_1$. So that would be a solution coming from the first time eigenvector and eigenvalue or the square root times any constant. And then, of course, I could have a b_1 . I'll use b 's for the sines. $(\omega_1 t)x_1$. So that's just like that. But it's used the sines, which would also solve the equation. And then all the way down to x_n 's. Right?

So I've got $2n$. $2n$ simple solutions. n cosines times eigenvectors and n sines times those same eigenvectors. Why do I want $2n$? I've got $2n$ constants then to choose, the a 's and the b 's. And how do I choose them? What's the next step? Using eigenvectors, maybe I just mention again the three steps. The three step method. Step one; find a 's and b 's.

Oh, now you have to answer my question. Where are the a 's and b 's going to come from? What's going to determine these $2n$ constants, the a 's and the b 's? The initial conditions, of course. We've got two initial conditions, this is a vector. We're talking about a system here. I've got position of both masses. I've got the initial velocity of both masses. And the a 's, they come from, because the a 's go with the cosines and

so they're going to be the ones associated with $u(0)$, will give this. Will give a combination, will give the a's. $u(0)$ will be $a_1 x_1 + a_n x_n$. This is at $t=0$. So I'm using the initial conditions. $u(0)$ is a combination of those eigenvectors. So this is from the initial conditions.

And the b's of course are going to come from $u'(0)$, the initial velocity because they go with sines. So sines start at zero but they have an initial velocity. So that's step one. That finds the constants. It splits the problem into these normal modes. It finds how much of each eigenvector is in there at the start

Step two? So it's really worth seeing these three steps because they just repeat and repeat for all applications of eigenvalues. Step two is what? Step two is follow each of these guys forward in time. Follow them to any time. What does that mean? That means just put in the cosines and the sines. So the a's go to $a_i \cos(\omega_i t)$. So I'm just saying what happens to those coefficients. Each one. Let's see. a_i , say, goes to a_i is multiplied by the cosine. And the b's, b_i 's go to $b_i \sin(\omega_i t)$. So now we followed the coefficients forward in time.

And step three? The final step is? So here, we're following each one. So the pattern is always the same one. Take your starting conditions, split them up into eigenvectors. Follow each eigenvector. And then step three is? Reassemble. Put them back together. Step three is the solution. u at that later time t is $a_1 \cos(\omega_1 t) x_1$. And $b_1 \sin(\omega_1 t) x_1$. Those are the two guys that are reflecting the fundamental mode. And then we have a_2 's and b_2 's multiplied by cosines and sines and times x_2 and so on. Put the pieces together. So it's split the initial conditions, follow each eigenvector, reassemble. It's the heart of Fourier methods, it's the heart of using eigenvectors. And there's a little code in the book that does exactly that.

So let's just pause a moment. This was the eigenvector solution. And now I want to talk about computational science. For which this problem is a model, but the problem has probably got more complications and things change in time, whatever. This was an exact solution. That's more than we hope for in real computations. We hope for accuracy, but we don't hope for 100% accuracy unless we have this exact model problem. So we use finite differences.

Ready for finite differences? So this is method one, which allows you to understand the model problem. And now comes method two, which allows you to compute much more, many more problems. And if I do it just for this, let me come back to just this model. One spring and one mass. Shall we start the mass at rest? Let me make that the answer. I want that to be the answer. So I'm going to start with $u(0)$ to be zero - to be one, right? I'm going to start with $u(0)$ to be one and $u'(0)$ to be zero. Because the cosine starts at one and its derivative starts at zero.

Apologies that this is such a simple model. I'm just pulling this this mass down and let go. I just pull it down an amount one, I let go, it oscillates forever. How do I draw the motion of a single mass? A picture is important and we have to think what's in the picture. So I think the picture should be, maybe u in this direction and maybe u' in this direction. So it's the u, u' plane. The position velocity plane. Sometimes called the phase plane. I'll just write that word down, phase plane. And what will be our picture here? It starts there, right? $u(0)$ is one, no velocity, starts there. Oh, and then it just follows $\cos(t)$. And of course, I know what u' is. If u is $\cos(t)$, u' is $-\sin(t)$.

Let me put that somewhere where my picture won't run over it, up here maybe. u' is $-\sin(t)$.

Help me out. If as time increases, I follow this point, the x coordinate is $\cos(t)$. The y coordinate is $-\sin(t)$. It's a circle. It's a circle. So I just buzz along it, because of that minus sign, the circle goes this way around. It's a unit circle. Goes on forever. And actually \cos^2 plus \sin^2 is one, of course, that's why it's a circle of radius one. And that represents the total energy actually. I'm trying to draw a circle. Actually how would you get your computer to draw a circle? I'm thinking not just one circle. I want to refresh it. I mean, I want to follow this in time. So I want to draw, I want to follow a point around a circle. Well I suppose what you would do would be to ask it to plot those points. That's not allowed.

I want you to follow the solution to the equation. I want to take that equation whose exact solution is a circle and I want to use finite differences in time. So I'm going to take finite steps. Of course that's what the computer has to do. The computer will take finite steps. So I'd be very happy if those finite steps keep me on the circle. And I'd be even happier if it came around exactly right at 2π but it's not going to. I want finite differences. I want to replace that differential equation by finite differences. And when we do this equation, we're practically doing that one at the same time. So this model will be fine.

So I'm going to introduce finite differences for this equation. And everybody, we've been talking about how to replace second derivatives by second differences. So that looks good. Let me write down, maybe I can write down three or four possible finite differences. For u'' I think I'll write down $u_{(n+1)} - 2u_n + u_{(n-1)}$. That's the second difference that replaces the second derivative divided by what? What do you think goes in the denominator now? Square of what? Yeah, now what's the right name? It's going to be something squared. What do I put down there? The step size. I don't want to call it Δx , right? What should I call that? Δt would be natural, Δt . Or h , I could again use h just to have a shorthand. That's my second difference. It's centered, it's the natural choice.

And not the only choice, by the way. Oh. If I'm an astronomer or like, Professor Wisdom here. So he followed Pluto around, right? And discovered that Pluto wasn't a planet, right? He discovered that the motion of Pluto was not like, regular, like the Earth. I think the Earth is regular. But Pluto has chaotic motion. So it does crazy stuff. Right. Well we're looking for very periodic, circular motion here. So what I was going to say, Professor Wisdom, he would sneer at this second differences, right? I mean, as we'll see that's got decent accuracy, but not accuracy that would allow you to follow Pluto for 100 million years. But it'll do for us. We're not going 100 million years here.

So now comes $-u$. Now, question. I'm taking this, the u term and I'm putting it on the other side. So mass is one. Acceleration is this. The force is $-u$. Now comes the big question. For $-u$, do I use the newest value? Do I use the middle value? Or do I use the oldest value? Or some combination? If I want high accuracy, oh, then I go way up in these differences and I find tricky combinations that get me above first and second order accuracy. But let's not go there. So I've gotta make one of those choices. And that choice is going to decide, so I mean, this is a choice you have to make when you have such a problem. Where are you going to evaluate this? I guess one choice jumps out as natural. What would you think of doing?

Well how many would do that one? So those are people, that's the conservative choice, the stable. And then, how many would do this? Yeah, you would do that, right. I would call that the leapfrog choice. Somehow, this second difference is leaping over this middle point. So that would be the leapfrog method. And then if I use a very low, to use the first value would be. So my point is those are three different choices. And each one has these questions of accuracy. The middle one will be more accurate because it's centered. Each one has a question of stability.

Ah, you don't see stability yet. So that's the point of the rest of the lecture is to see. What is this stability question? The issue of speed, speed would be, this is the slow one. Because it involves, I have to bring $u_{(n+1)}$ over there. And if I've got a system of equations and then I, it would take me some time. This would be called an implicit method. The new $u_{(n+1)}$ is only given implicitly because it's showing up on the right-hand side. I've got to move it over to the left. If it's non-linear I've got a system to solve. It's safer, but more expensive. But this would be the natural choice.

I want to get eigenvalues into this picture. So I'm going to do the step we often do when we see a second order equation. That is, reduce it to two first order equations. Can I do that and see the same thing? So what would be my two first order equations. My unknowns will be u and u' . So it'll be first order, the derivative of u and u' . Shall I call it v for velocity? Let me write down, I don't need to write this fancy. The two equations will be, $u'=v$. And v' is what? So $u'=v$ sort of told me what v was. v is the velocity, the time derivative of u . And now the derivative of the velocity, now that should reflect my true equation. So this is all coming from $u''+u=0$. So v' is what? v' is u'' . Do I want $-u$ there? Yeah, that's my equation. v' , which is u'' is $-u$. Good.

That's my system. So I have a matrix here. I have a two by two system with a matrix $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ I guess. So while thinking about difference methods, so this, I was thinking about a second order equation. I went right to it. Here I'm thinking about a first order system. A lot of problems come in first order systems. Gas dynamics comes as a big first order system for those components of mass, momentum, energy, typically five non-linear equations in gas dynamics. I mean, that's a very, very serious problem, how to solve those. Here we've got a much simpler model problem, linear, constant coefficients. But what do we do?

Can I propose three possibilities here? And actually they are identical to these three possibilities. If I just switch over to a first order system. So possibility one. Possibility one is that $u_{(n+1)}$ and $v_{(n+1)}$, n so the $u_{(n+1)}$ would be, how do I model that equation? $u_{(n+1)}$ is $u_n + \Delta t * v_n$. That would be a natural, right? Let me just pause there, because that's meant to be the natural forward difference. So I replaced u' by $u_{(n+1)} - u_n$ over Δt . And I replaced v by the value I know. And do you know whose name is associated with that? I'm almost going to say his name. It's just I replace a differential equation by a difference equation where each step I just could figure out what the slope is and I go along a straight line. Δt further along. Do you know his name? Euler, Euler, right. It's Euler's method. The very first method you would think of. Right?

So I'm doing the same for v . This is my position I've reached. So Euler's method is replace this by $v_{(n+1)} - v_n$ over Δt equals $-u_n$. So this is Euler's idea. Predict the new value by a forward difference starting from where the slope of the line is this old one. That's the first difference method you would think of. So now if I multiply up by Δt I have a minus $\Delta t * u_n$. Good time to pay attention. This is forward

Euler. Forward Euler. The time step, you follow the derivative, what the derivative is at the start of the step. You follow it for a little step.

Ah, let me draw what the thing would do. Shall I draw what forward Euler will do? So we're here. And what's our first step? Our first step. So we're starting at v is zero at the start. So u won't change in one step. But u is one, so v will go down. I think the first step will take us there. That point was $(1, 0)$. And I think the second step, u came from the zero, so it'll still be a one. And v came from a zero, but minus Δt . I think it'll just be $1 - \Delta t$. It's left the circle, of course. And what do you think happens when I do 1,000 steps? This is two lines in MATLAB. Just write those equations for the next u coming from the previous u starting at $(1, 0)$ and see what happens. And what do you think? It's going to sort of go around. I mean, it's a reasonable, Euler wasn't dumb. Anything true here is the fact that Euler was not dumb. He had a bunch of ideas. This wasn't his greatest. But it's the starting point of any, if you have to code some complicated problem, start with Euler, forward Euler.

What'll happen? It'll spiral out. Exactly. It'll spiral out. And if I was an eigenvalue person, I guess which I probably am, I would look at the, there's some matrix here that's finding the new u, v from the old u, v . Maybe we can even find out, write above what that matrix is. The new u, v come from the old u, v , well there is the identity matrix. Here is a Δt . And here is our minus Δt . That would be my matrix. That's the forward Euler matrix. The growth matrix for forward Euler is that. Now what do you think about its eigenvalues before? Spiral out was the right answer. If Δt is small it'll stay close to the circle, but it'll spiral out from it. So if we were following a planet, it would just be off.

So I can't immediately, well you could almost see the eigenvalues of this matrix. What's the determinant of that matrix? What's the determinant of G ? I'm just asking you to think through. You see a matrix like this. You wonder about its eigenvalues so you take the determinant. And what do you get? One, something bigger than one or less than one? Bigger. So that tells me what? Since the determinant is the product of $\lambda_1 * \lambda_2$, whatever those guys are, at least one of those eigenvalues is bigger than one. This has an eigenvalue, actually, the actual eigenvalues of this happen to be one from the identity matrix and then this guy is what was our example of an imaginary eigenvalue, $i * \Delta t$. Bigger than one in absolute value. It goes out. So that's forward Euler corresponding to that choice.

Can I change to a second, to Euler's other choice? And you can guess what its name is. Instead of forward Euler it's going to be? Backward Euler. Euler said, if you don't like it, I've got another one. And what will happen then? Backward Euler is going to use, instead of the old value, it will use the new values here. So these differences, you see that this is now a backward difference? So I'm at time $n+1$. I'm at the new time. And this goes back from that. So this is at the new time and this is a difference backward. And what happens now? Well, we could follow backward Euler. We could follow its first step. What would its first step be? $u_n + u_{n-1}$. No, I can't follow that first step so easily. Help me out by just making a guess. What do you think backward Euler does? It spirals in.

Right, the cover of the book has forward Euler, I think, is on the cover of the textbook. Spiraling out in the front. But maybe the back cover also has Euler spiraling in. So here is backward Euler. And after I take a bunch of steps it comes back somewhere there, but it's spiraled in. No good. Have we got one minute for the

good method? You're guessing that it's this one? How does that work? Actually, it's pretty neat.

So my question is, where do I evaluate these guys? So this is like forward and backward. The u equation, so I know u_n and v_n , right? Everybody with me? I've got to time in, I know u_n and v_n , the position and velocity approximately at point n . And I want to go $n+1$. what I know is v_n . So that's great. But what's going to change here? Now I know $u_{(n+1)}$ from the first equation. What shall I do, where shall I evaluate u in the second equation? At $n+1$. Why not? I know it already from the first step, from the forward step with this equation. I've taken u forward. So I'll use that forward value of u in the v equation. So it's forward n backward at the same time. And the net result if I go back from my system of two first order equations back to one second order equation, I'll find the centered guy. And what do you think happens with that center difference? I think MATLAB would have to show it. And look in Section 2.2 for the picture that shows the leapfrog method. So what do you think? I'm having to depend on my memory now. It stays almost on the circle. I think it maybe, it stays very close to the circle. And comes back almost at the right time, but not exactly. Because we have an error here. No, we haven't got the real equation. We've got a difference equation.

So maybe that's the point to say. When you start with a differential equation you've got various choices. Backward is actually safer. Probably for a car crash. No, actually I think for a car crash they would use forward differences because they have to take so many. Let me check on that. But for our problem, this one, leapfrog is the winner. Actually, if anybody does computation in computational chemistry, I mean that's a subject that uses giant supercomputers to follow molecular dynamics, to follow what molecules are doing at high speed, very high speed. So this, there's a giant number multiplying that u . And they use leapfrog method.

I'll say a little more Friday if I can about this general subject, because it's so important.