

GILBERT

OK. Now, I'm going to have differential equations, systems of equations, so there'll be matrices and vectors, using symmetric matrix. They'll be second order. So second order, second derivative, that y is the vector. And S is the symmetric matrix. And that's the first time we've been prepared for the most fundamental equation of physics, of mechanics, oscillating springs-- so many applications-- rotating torques. It's very important in applications. The finite element, giant finite element codes, are solving equations like that all the time. And we don't have a damping term here, so it-- or a forcing term, so it's the null solutions that I'm going to look for to match initial conditions. I don't have a forcing term.

OK. So the real central equation always looks like that. This is-- Newton's law, is what this is-- mass times acceleration. So M will be a matrix, often a diagonal matrix, telling me the masses. Remember, I have n equations here, so I have n masses, as you'll see. And I have, let's say, a bunch of springs connecting those masses. And then there's a matrix K in multiplying y itself, and that's always called the stiffness matrix.

So, actually, in applications, the first job is to take the problem and create these matrices. I'll give you an example, but let's suppose we've got them, and how do we solve them? We look for, as we always do, solutions where time is separate from the vector x . I substitute that into the equation, so I get M , second derivative will bring down the i ω , twice. e to the i ωt x , right? Plus, this term, K times e to the i ωt x , should be 0. So I'm just substituting the expected form for the solution. In that form, that exponential factor can cancel. And, I see, I have an eigenvalue problem. Let me just look at that eigenvalue problem. I'm going to put that on the opposite side. But what i squared is giving me minus one. I'm just going to be left with Kx . Well, let me put this on the other side, because it's got a minus. And then when I put it over there, will be a plus.

$M \omega^2 x$. That's an eigenvalue problem. Here is the eigenvector. There is the eigenvalue. Oh, but we have two matrices. That's something a little new. Not new to MATLAB however. The MATLAB command to find these eigenvalues, let me call those eigenvalues λ , so λ will now be ω^2 , because two derivatives brought down ω twice. But we have our two matrices, so the MATLAB command would be `eig(K, M)`. If you define the matrices, K and M , and you call that command, it will produce the eigenvalues and the eigenvectors x for this, you could say, generalized eigenvalue problem, two matrix

eigenvalue problem. It's got a K , as usual, and then it has an M . But many, many times, M will be a multiple of the identity and present no problem.

OK, so that-- this is the eigenvalue problem that we reached. And that's the command that would solve it. OK. That's the first step, is to look for solutions of that special form.

Now let's do a little count. How many solutions are we expecting? How many initial conditions do we have? So we initially, we give y at 0 of course, the initial condition, the position. But we also give, in a second order equation, we also give the initial velocity, y' of 0. And those are vectors, because those tell us the initial condition of n masses. And so I have n numbers from y of 0 and n more numbers. Two n , all together, initial conditions. I'm going to need two n solutions. I'm going to need two n solutions to match two n initial conditions and solve the-- solve the equation.

OK, so what do they look like? All right. Here, I've gone ahead to put an application up. So, again, I'm taking the masses to be equal. Here are the masses, M , M , and M . And so I end up with a 3 by 3 matrix. I have three unknowns, n is 3 for this problem. And I've got 4 springs. These are springs-- maybe I make them look a little springier. OK. And they're connected at the top to a fixed support and at the bottom to a fixed support and they're connected to each other.

Do you see what will happen? As I, maybe I, the initial condition, I drag all those masses down? As my initial condition, I let go? Then they will go up and down, up and down, just the way springs always do. And they will solve, their position will be the solution to my differential equation. $M y'' + Ky = 0$. So I'm not, I'm just starting the motion, and then backing off. So I'm not forcing-- it's not forced motion. It's pure oscillation, pure oscillation. Right.

But coupled, several oscillators are a couple, that's what's new. We know all about this equation when y is just a scalar, just one equation. We know that, and that led us to the square root of K over M . That was just a 1 by 1 eigenvalue problem, and now we will have a 3 by 3 eigenvalue problem. The mass matrix is simple. Here's what the stiffness matrix would look like, if all those springs were the same. Just, I wanted to see what kind of a matrix shows up in the problem. And time to write down solutions.

OK, so what are we remembering from solutions? We're remembering that solutions look like this. I have, but I'll have three possible eigenvectors, because I have 3 by 3 matrices. So that

will give me three solutions. But I want six because I have six initial conditions all together. Let me write those six solutions down.

Y, the solution is, sub constant times the cosine of omega t, times the first eigenvector. And another constant times the sine of omega t, times the second eigenvector-- times the first eigenvector, sorry. The first eigenvector, I have three eigenvectors, and for each of them, I get two solutions, one a cosine and one a sine. And the frequencies, there would be omega 1, and you are remembering that lambda, lambda is omega squared. So if I write omega 1 there, it's the square root of lambda 1. That's the square root of lambda 1.

So I've got two solutions so far, coming from the first eigenvector at its eigenvalue, and with the cosine and a sine. And then, I'll also have for the A2 and a B2, and an A3 and a B3. So I just, briefly, A2, B2 with omega, using omega 2 and x2. And then they'll be an A3 and a B3 using omega 3 and the eigenvector x3.

That's pretty stupid looking. This is what I meant to represent. I don't want to rewrite all that with 2's, and then rewrite it again with 3's. That's what the solution looks like.

OK. How do we-- when we match the initial condition, what happens? I said t equals 0, right? When I said t equals 0, the sines disappear. So it's the A's that match y of 0. So A's match y of 0. And when I look at the initial velocity, the derivative, the derivative of the cosine is the sine, and it's 0. But the derivative of the sine is the cosine and it's 1, so I'll see that the B's match y prime of 0.

I'm trying to get the total count to be six. So there are three, three initial positions, and three A's. There are three initial velocities, cause there are three masses, and there are three B's. I get a perfect match, a total of six. Six constants, six numbers to match. It all works.

OK. Now, do I want to try an example? Sure. Let me end with a particular example. I better go to 2 by 2 for an example. So can I do the same problem 2 by 2? So my problem is going to be y double prime plus S-- there's a K, there's a division by m, and there's a 2, 2 minus 1, minus 1 y equals 0. That's my equation.

Now, I'm speaking now, about the problem with a spring, a first mass m, a spring, the second mass m, and a spring. So two masses, two equal masses, three equal springs. That's my equation.

So what's a solution? How do I solve it? I need the eigenvalues and the eigenvectors of my matrix. So here's my matrix S . That's my matrix S . It's symmetric, it's got physical constants there. The stiffness of the springs divided by the masses, k over m , we're expecting that same k over m that always shows up. And we need the eigenvectors there. And what are they? The eigenvalues of that are 1 times k over m , and 3 times k over m . Cause the trace is 4 . 2 plus 2 is 4 . The determinant is 3 . 4 minus 1 is 3 . Those are the two eigenvalues. And these are the ω squareds, remember.

OK. That's how I start the system from rest. So physically I pull down these masses, or maybe I push that one up and pull that one down, whatever I like. I hold them for a moment and I let go. I don't give them an initial velocity. They start from rest, so the B 's will be 0 . So y prime of 0 , y prime of 0 is going to be 0 from rest. And that will give me B 's are 0 .

So my solution will be, my A cosine of, so the eigenvalues cosine, I take the square root of k over m t times the first eigenvector-- one of the eigenvectors. The eigenvectors of this, probably are $1, 1$. And then the other eigenvector, and that will have a cosine of the square root of ω , I have to take the square root of that, so that's the square root of 3 , k over m t , times its eigenvector, which I think is 1 minus 1 . It's going to be perpendicular to that one.

I've solved the problem. The A_1 and the A_2 are determined by the initial condition.

Now do you see what's happening in the motion? That's the last thing, last point for this video. This motion with a $1, 1$ eigenvector, the two masses are in sync. They're growing together, up and down. That's one eigenvector of the problem. And it has a certain frequency that they go up and down-- a square root of k over m , our old friend. But also, with two masses, they can go against each other, like this motion. That's coming from this eigenvector, and it happens at a higher frequency. So those are going-- the final solution is a combination of the masses moving together at a little slower oscillation, and the masses moving opposite each other at a faster oscillation. Some combination of those two is the solution. And then if we had three masses, there would be three oscillations. One where all three are going together, one where the outside ones are opposite, and one where all three are, I see opposite signs.

It's a beautiful subject. Highly developed, and highly important applications. But that's the nicest solution you could hope for.

OK that's a second order system, solved. Good.