

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: All right, let's get started. So today we're going to talk about boundary value problems. But I want to try to start back with a motivating example. So unfortunately, like most technologies, the biggest driver for differential equation solutions was from military technology. And the military problem-- I'll tell you in a second. Let me first remind you.

So we have ODE-IVPs. And they are-- right. So that's the ODE-IVP is we have n differential equations, and we have n initial conditions. And you guys are really good at these now. Had a lot of practice.

But you can have problems where some of the conditions are not specified at the t_0 . They are specified somewhere else. So if all of them are specified at t_0 , we call it an IVP. And we call it a BVP is if the conditions are like this. This is not so good.

And not all the t_m 's are the same. If all the t_m 's are the same, we call it t_0 , and we're back to IVP. So IVP is like a special case of the ODE-BVP problem. And here I'm ranking them as if everything is explicit. So we know an explicit equation for dy/dt .

We have an explicit equation for the boundary conditions. In fact, often, we actually only have implicit equations. So I'll show you an example in a minute. But that's it. And we still need to have n equations.

Well, I guess I could do this. Take this back. Basic goals, and just be just be careful. These ends are-- it's the same number of them. It's the same number of conditions. This is OK to write it this way, but just be careful.

Is that OK? Just to make it confusing. All right. I guess I can call this 0 now. See, but now it's not really 0 , right? It's really maybe-- call it star at the boundary. There's some special value-- some special place where you know the value of y . All right.

So let's do the motivating example, which comes from military technology. And this is quite an old problem. So, because, in the old days, people didn't have motor vehicles, and, basically,

the roads were no good, and there were a lot of bandits out on the roads, people liked to ship everything by water transportation.

And so all the towns grew up on rivers, or on ports on the ocean. And so, if you lived in a nice little town, you know, up in the hills there, and then, at the base of the hills, you have your cute little town. You have your house. You have some warehouses where you keep all your nice goods.

You have a dock that goes out onto the water. Here's the land. Here, these nice ships come and park at the dock. They unload nice goods. You trade your stuff to them. You make a lot of money. Everyone's happy and prosperous and having a great time living. This is a port city.

But then you look out on the horizon in the water, and you see a ship is coming. And sometimes that a ship is coming is really good news for you.

But sometimes the ship coming is not good because on the back, it has a flag, and on the flag, you see skull and crossbones. And you realize it's a pirate ship coming. And now you're in serious trouble because you really don't want the pirates to come to your nice little town because who knows what they'll do there. But it won't be good.

All right, so the people who live in this town, after this has happened to you a few times, you're really alarmed, and you're really happy, and you don't want to rebuild your town again because it just got burned down the last time and the pirate came and pillaged it and took everything. And you guys all had to run away into the hills.

So instead, they've invented cannons or catapults, put one of the cannons, say, at the top of the hill above your town. And then you say, the next time the pirates come, I'm ready for them. So drawing the cannon.

And so the question is, you want to shoot the cannonball so it hit the pirate ship at the waterline, when the pirate is distance L from the shore-- hopefully, pretty far, so they're not going to shoot at you first. And you're shooting from up at the top of the hill, so you can shoot further than they can.

So you're going to shoot distance L . And you're shooting from an elevation H . And you want to make sure your cannonball hits them, ideally, right at the waterline. Boom, makes a nice hole in their ship, and it sinks. So that's the plan.

Now the problem is, getting the cannonball to go from here to there is not so easy. It's a big ocean. There's a lot of places your cannonballs can fall into the ocean, and it might not be where the pirate ship is.

And reloading canon is pretty slow. And you guys live in this prosperous port. You're not military guys, and so you don't have that much practice doing it anyway. And you're cheap because you're merchants. And you don't want to spend your money buying lots of cannonballs anyway.

So you only get a few of cannonballs up here. And there's a little pile of cannonballs next to the cannon. And then, when you see the pirate ship coming, you guys run up the hill quick, and you get to take a couple of shots, and that's it. And if you hit them, then your town is safe, and, otherwise, everything you love and know in your world is going to be destroyed by the pirates. OK?

So this was the motivating sample for that element of differential equation solvers. OK? So let's see if we can write down the equations that correspond to this example.

So it's going to be Newton's laws. F equals ma on the cannonball as it travels. So we can write that also as $ma = F$ -- well what's the F ? F is going to be negative mg in the z direction. That's gravity pulling the cannonball down.

And then you'll have some air friction on the cannonball. And I'm going to call the air friction coefficient γm times v . OK? So that's the simplest model for what the forces are on a cannonball just flying from my cannon, hopefully, to hit the pirate ship.

And I can rearrange us by dividing through by m . And acceleration is velocity, so dv/dt is equal to negative g . So you have negative γv -- sorry. My notation is not so good. OK. Everyone is all right with this?

Yeah. So then we can write the whole differential equation system, d/dt of xz v_x v_z equals dx/dt is v_x , dz/dt is v_z . dv_x/dt is negative γv_x . dv_z/dt is negative g minus γv_z .

So that's a fine ODE system. It's not too hard to solve. It's actually linear. It's a piece of cake, right? You guys just probably just did this already when you were in high school, or college, anyway, for sure. And let's write down the initial conditions. So we know x_0 is equal to 0. z_0 is equal to h , the height of the canon. v_x of 0, oh, what's that equal to? We don't know. v_z of 0,

we don't know that either.

But we bought this canon with a certified muzzle velocity v_0 , so we actually know that v_x of 0 squared plus the v_z of 0 squared is equal to v_0 squared-- whereas, when I bought the canon, the merchants went out to the local canon dealer and bought a canon, and the guy said, you put this much gunpowder in it, you'll get a muzzle velocity of v_0 . So they know that that number is.

So we have a condition that's not really the way we expect it to be. And we don't have enough conditions. We need four conditions.

But of course, the most important condition is what happens at L . We really want to know that x of t final is equal to L and y of t final is equal to 0 at sea level. So that way, our cannonball is going to into the right location at t final.

So we have two conditions specified way, way away from where we are. And we have two conditions specified initially. And now you have some oddball thing in here, some oddball fifth condition. Now, OK.

One problem here is, we don't actually know what t final is because before we shoot it, we really don't know how long it takes the cannonball to get out there. So this is a kind of weird condition where t final is unknown, whereas, the ones you've done before, I think you've always known t final. Is that right?

So let's talk about how to deal with that. So one problem, one way to deal with this is to notice that, if we have an equation system $dy/dt = f(y)$, we could change variables instead of using t . Since we really don't know much about time in this problem, t is not the best variable for us.

So we want to change it to some new variable u . So suppose that we want to change to u . And, if we change to u , we can write something like this-- $dy/du \cdot du/dt = f(y)$. Can you see that?

So I can try to get the differentials in terms of u , and I just have to know what my du/dt is. And you guys might have even changed the variables before, in calculus sometime, you probably did this-- a long time ago, and chain rule, and stuff like that. All right, so we can sort of pick whatever idea we want, and as long as we put in the du/dt in the equation, then we'll be OK. So let's just choose that u is equal to x because we know du/dt would then be with the v_x .

So we could we could write this out this way. We could write d -- I'll keep it as u for a minute, so the notation won't be so confusing. So du/dt , dy/du is equal to f of y over du/dt . And as long as du/dt is never zero, we're good. So we could any u we want as long du/dt is never zero. Then we're still going to have a nice simple question here.

And so I'm going to choose u is equal to x because du/dt is v_x . And I know that v_x is never zero. They're kind of also always moving to the right. So I'm going to divide by something. It's never zero.

So I can write it this way-- dy/du is equal to-- I can go down that list and divide each of these things by v_x . So it's 1 , v_z over v_x , negative γ , negative g over v_x minus γ v_z over v_x . That's my new [INAUDIBLE]. OK? OK with this?

How about energetic? Keep nodding. Yeah, yeah, sure, keep going faster. It's going way too slow. That's what I want you to say. Yeah? Apparently not. OK. All right.

So the advantage to changing here is that now I can write the initial conditions, and all these conditions, in terms of u , which is x . So I really-- I want to go from u is equal to 0 , i.e. x is equals to 0 , to u is equal to L , which is x equal to L .

So now, I could also look at this say, this is kind of stupid. dy/du , the first component of y is dy/du is equal to one. And that first component of y never appears in the rest of these equations. So that's kind of not a very useful one. So I can change, and I could write a simpler equation, dy_tilde/du is equal to v_z over v_x negative γ , negative g over v_x minus γ v_z over v_x , where my y_tilde is just equal to the last three variables-- z , v_x , v_z .

So I did a lot of algebra, but I ended up simplifying my life because now I have three different equations to solve instead of four. And I can write that y_tilde of zero of the first element of y_tilde , which is z , is equals to h . So that's a condition I know for sure.

And I know that I really want this cannonball to hit the pirate ship. So I want y_1 tilde of L to be 0 , so I want tilde as z . And I want z to be 0 when x is equal to L . Or that's the point where the cannonball is hitting the ship at the water line there.

So those are definitely two conditions I have. And then I need a third condition. And my third condition is this crazy equation here. All right, so now I have three conditions for three differential equations, and one of them is an oddball looking thing.

These two look kind of normal. However, this one is a t_0 , and this one is some goofball value, but at least I know the value. I know what L is because I know how far away the pirate ship is. So far so good? All right.

But how am I going to solve this? So how do we know how to solve differential equations right now is we know how to do initial value problems. We have these nice programs like ode45 that work great, integrate differential equations.

So we should be able to use that somehow to solve this. But we don't have all the inputs because ode45 would expect us to give values of all three of the y tilde values at 0. I only know one of them.

So how can I approach it? Sorry, I'm going to put my beautiful drawing down below here. We'll save it for posterity and look at it later.

So I could approach it by saying, well, I don't know what these two initial conditions would be, what v_x and v_z should be at 0. I'll treat them as unknowns. I want to figure out what they should be.

And so, I'm going to vary those unknowns and then, for each value I choose, then I can shoot the cannon with those v 's, and I can see where the cannonball went, and if it didn't go to the right location, I can readjust those v 's and then try again.

Just like, if you were running the cannon, you might shoot at a certain angle, and then adjust the angle of the cannon, and then shoot again. And then, when you change the angle, you change the ratio of v_x to v_y initially, right?

So that's called the shooting method-- for good reason. So the shooting method, that's what we do. We guess the missing initial conditions, then you solve using the guess. You solve the ODE-IVP problem.

And then you see what it computes. And if the solution you get doesn't satisfy the conditions you demand, you go back and adjust your guess. And you loop around. So that's the idea.

Now, this thing of adjusting some parameter values, some initial conditions, to try to satisfy some condition, that's like an f solve problem. That's a system of nonlinear equations kind of problem. So we're going to have f solve as the outside loop, and that's going to adjust some

parameters.

And then, what's f solve do? So f solve tries to solve problems of the form g of v -- I usually write x , but I'll call it v here. No. V 's not good either.

What letter have we have not used yet? Any letter. g of w . There we go. g of w equals 0.

I want to figure out what w 's I need to use. And actually, v is right. Because we're trying to adjust the velocities, right? Sorry, I'm messing up here.

It's good to use the eraser in this class-- especially with this nice big eraser. All right, so g of e -- we're trying to adjust the velocities to make something 0. So what's our g that we want?

We really want that the z value at L -- we want that to be equal to 0. And this thing is, like, implicitly a function of the v 's-- the v_0 values we give. If you put the right v_0 values in, then z at L would be 0.

So this is what we're really trying to solve. Now, to get z of L , we have to solve a system of differential equations. So it's a little complicated to go from v_0 to z_L , but we can do because we have `ode45`.

So we can write a function to do this. Running out of words. All right so let's try to write the program. So we want to write the program g .

So we want to write z of L is equal to g of v_0 . And let's try to think what this function is. So the meat of this is, we have to solve a differential equation, an IVP.

So somewhere, we have an `ode45` call. And what does `ode45` computing? It's computing the x positions and the z positions. And it's going to use some function f , which was-- do we still have it on the board? Maybe I lost it here. Nope. Sorry, it's under here. I'm a terrible board worker.

So this is our function f , f of y . So this is really f tilde. That's the function that I'm going to integrate. And I can't remember what are the arguments, but it's going to be something like, from 0 to L , and then I have to give it a y_0 , and then maybe there's some other stuff up there.

All right, do you guys remember the `ode45`? Is that in the right order? I don't remember. OK.

So I need to specify the y_0 . So y_0 is going to be equal to the initial value of z , the altitude. So

it's going to be H . And then it's the initial value of v_x .

So I don't know what that is, but it's-- maybe you just want to call it v_0 . v_{x0} . And then I do have one more condition here, that v_z is related to v_x . So I can use that. So I can have the last one be the square root of v_0 squared minus v_{x0} squared.

So that's the line for what y_0 is. And then I solve this. And then I write the number of steps in the time stepping program that did, in order to get here. It was x stepping in this case. Num steps is equal to the length of x vec. And z of L is equal to z vec at N step. That's it, right?

So that's the function that you would feed to f solve. And your correct value of $v_x 0$ is the one that makes $z_L 0$, which is what f solve tried to do. Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: You can, but in the physical problem, usually cannons always shoot upwards. So we'll just shoot upwards. But yeah, it could be either way. Yeah, and, in fact, in this problem, you might do better to change-- instead of using v_{x0} as your unknown, you might want to change to θ , the angle, the elevation angle of the cannon, and that would be more physical of what the actual user of the cannon is using. They're adjusting θ , not adjusting v_{x0} .

It's no problem, actually. From the point of view of the f solves, you can make this θ instead. And then you'd have this be v_0 times \cos of θ , and this be v_0 times \sin of θ , and it would work just as well to get the solution.

All right, so that's the shooting method. So this function g is the g that's going-- you'll have f solve, so is going to say, v_{x0} is equal to f solve of g . And then you have to have a guess, right? For f solve.

So that's how you would invoke this function. All right, and so if we wanted to figure operation count of how many operations this is to find a solution this way, how's it go? How many iterations does it take to solve an ODE IVP.

That depends on the step size, right? It's adaptive. Usually adaptive stepping. So you have some experience with that, maybe 100 steps? Something like that? Is usually enough? Maybe 1,000 if it's not stiff?

And then how many iterations does it take f solve to solve something? 10? So you have

maybe 1,000 time steps, maybe 10 iterations, means 10 different initial values that you try before you convert to a solution.

And then you may have to actually compute the Jacobian of g , inside f solve, if it's going to use the Newton-Raphson step, it's going to need the Jacobian of g . The Jacobian of g takes the number of unknowns times the number of function evaluations to compute it by finite differences-- maybe 2 times that number.

So it's going to be something like n squared. It'll be n squared to evaluate the Jacobian. It takes n operations to evaluate f . So order of n , where n is the number of variables, n times that, to the Jacobian, so it would be like order of number variables squared times 10 times 1,000 is kind of an order of the cost.

And it's only that cost because we can do an explicit solver. If you had to do an implicit ODE solution, it would be a lot more because you'd have to compute the Jacobians more often. All right. Is this fun? OK.

So this is the shooting method. You guys should be able to do this no trouble. Conceptually, this is not so different than the homework problem you did where you did an optimization of the parameter values. Do you remember? So you had some differential equations, and you optimize the parameters to try to find the best solutions. Here, instead of optimizing them, you're doing f solve. You're trying to solve them to make something happen.

Now how can this run into troubles. Why would this shooting method not always work? Ready why this might not work? Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: And a common case where that might happen would be if your ODE system was unstable, Or your ODE solution method was numerically unstable. In either of those cases, who knows what you're getting. And actually, you don't know if the solution is even real.

You don't know if the computed final values z_L is really what corresponds to that input value of v_{x0} . Because, if you have a big kind of numerical sensitivity and instability. So that's one kind of problem.

So this is just like all the ODE IVP problems. If the ODE's problem, intrinsically, is unstable in a big way, you're in trouble because your errors are going to amplify. And then also, if you have

a problem where the delta t's are too large compared to this numerical stability of the solver, you'll also have a problem. So those are two kinds of problems that we always have in ODE solutions.

And so, you might actually want to use this method for some regular ODE IVP problems. So for example, if you had a problem-- I don't want to wreck my beautiful artwork. You guys know this. All right.

If you had a problem that said dy/dt is equal to 10 to the 6th times y . Yes?

AUDIENCE:

I have a question about the Jacobian. When you calculate the Jacobian do you have to solve the ODE system?

PROFESSOR:

Yes. Well, two ways. You can either solve it for repeated values, different initial values, of v_0 , and by finite difference, get the Jacobian. Or you could use the sensitivity method that we did in the second problem, a problem set ago, and analytically get the derivatives without respect to the parameters either way. But either way, it's a lot of work.

So the from I guess this is a very simple problem, and suppose you wanted to go all the way out to t equals 1 . Now, the problem is that the solution of this is Ae times 10 to the 6th times t or something. So this is really not looking so great. As a problem to solve, this is about as unstable as you can get.

The tiniest little deviation anywhere along the way is going to make the thing blow up. And so this is not so great and so even if initial condition is that y of t_0 is equal to 10 to the minus 12, so actually this is not so bad, this is still hard to solve, numerically.

But you could flip it around, and treat it as a shooting problem, and solve it in reverse. So you could change variables that u is equal to negative t . And then you'd have dy/du is equal to negative 10 to the 6 times y .

And now this is really stable. And now you might want to integrate in. So supposed the original question was, what is y of 1 in the differential equations? You could do it in reverse and say, I'm going to guess, y of 1 is equal to y guess. And I'm going to integrate it and demand that this is my final condition.

So now this is actually y -- oh geez, now I'm confusing myself. y of negative 1 , I guess. So I just flipped the sign, t . And to make it easier, I would do it this way. 1 minus t . Same derivative.

So there we go. y_0 is y guess. And y of $t = 1$ is $10 - 12$. So this is my final condition that I'm trying to shoot to, and I'll vary the y guess. And they're great. And now I have a perfectly stable differential equation I can solve. OK.

So this same idea of shooting comes up in multiple applications. And this whole thing about flipping the change in the variables is often an extremely useful trick to just recast the equations in a way that's more numerically stable to solve, has fewer number of equations, you can write explicitly what you want. But this takes some cleverness to know how to recast them.

All right, so this motivating example is a good one. Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: You could, but I was just trying to get back in the form of t_0 equals zero. I guess I could have t_0 equals negative 1 do it. I just couldn't do the arithmetic on the board.

AUDIENCE: [INAUDIBLE]

PROFESSOR: y of 0 is known, but then I have to start from t_0 equals negative 1. So I could do-- so suppose I did u equals negative t dy/du is equal to negative $10 - 6y$. And I know y of 0 is equal to t_0 -- no, what do I know? Yeah, y of 0 is equal to $10 - 12$. And y of negative 1 is equal to y yes, and I can integrate from negative 1 over. So this will t_0 if I was using the shooting method, and this would be t final. Or you could shift it over by 1 if you want, and do it the way I did. Either one's fine. Is that all right? Sorry to confuse all the arithmetic. I was getting confused too.

All right, now I wrote this-- another situation where this will not work is you can have systems where you have both positive and negative eigenvalues of the Jacobian, and that means that either way you integrate-- in the plus t direction or the negative t direction, either way it's always unstable. So no matter what you do, you're doomed.

The IVP method is not going to work because you're numerically unstable in both directions. And so that's really a bummer, and it means our ODE IVP method's not going to work for us, and so then we have to find a different method. But before I go into that, let's just write the more general version of this.

OK, so this is-- I think you guys all know this. So let's try a more general version of the ODE

IVP problem. And now we'll allow for the fact that things don't have to be explicit, and they have to be as simple as we thought. And you can see it even in the problem I just did-- we had a v_x squared plus v_z squared had to be equal to something. And so that's not a simple boundary condition like we saw before.

So you get a more complicated conditions, and what you really can have is a set of equations, g_n of dy/dt y , t , and is equal to 0. And corresponding, q_n of dy/dt evaluated at t_n , y of t_n , t_n is equal to 0.

So this is the general ODE system that's written in implicit form, and this has to be true for all t in the domain. All t that are a member of t_0 to t_f . So all the t 's that are inside this interval, the domain we are about, this is the differential equation, written as just as general can be, and n is equal to 1 and N in these equations.

AUDIENCE: Is there a reason you switched to q of n for the--

PROFESSOR: I just want to make it clear that this is the conditions, and this is the general equation. I don't know. You can write it anyway you want. Do you like some variable letters instead of q ? I can change this.

AUDIENCE: No, I was just checking.

PROFESSOR: This one, I'm just trying to emphasize. This one is only true at t_n only. This equation, this is the condition at one particular time I demand something be true, and this is the general difference equation that has to be true at all times. If I have n differentials, I need n differential equations, and n initial conditions.

By the way, you can write the DAEs this way as well-- the differential algebraic equations, and then you just won't have as many initial conditions because every time you have an algebraic equation instead of a differential equation, then you don't need the initial condition because the algebraic equation has its own condition. Does that make sense? All right, so you can write this in general, and just by depending on how many differentials you have, that's how many conditions you need.

A lot of times, you can write them explicit. This thing is equivalent to dy/dt is equal to f of ty . Many times, you can solve for dy/dt , and write this way, but not always. OK All right, and the reason why we need to have n conditions is just like when you're doing integrals, you'd have n

constants of integration. Every time you do an integral, you get one, now you have three. We have n differential equations. We'll have n conditions.

All right? Now, a bad thing about these problems, in general, is that the solutions depend a lot on the boundary conditions. And in fact, you can write down boundary conditions that are not achievable, which means there's no solution. So this is really a bummer if you don't realize you did it because you could try to solve it all day, and you'll never solve it because it doesn't have a solution.

So for example, in this problem, if I set L to be too large, the cannonball can't go that far. I can't set L to be 3,000 kilometers and get a solution because the muzzle velocity is not high enough that the cannonball is going to travel 3,000 kilometers.

But when I wrote the problem, you didn't see that, right? You didn't come and tell me, whoa, watch out. There's no way L could be bigger than such as such. And so you might not know it. You might specify a condition that's just not achievable.

I used to work an industry, this happened a lot. My manager would specify boundary conditions which were not achievable. I want 99% yield and a 10% cost reduction. Go! And I'm the engineer, I'm trying to work it out, and then sometimes it didn't even come out.

So in some ways, it's a lot better to just say max. Please maximize this as best you can, subject to your constraints, instead of specifying this. But of course, I don't know why, there's a school of management that says we weren't definite targets. We don't want to be just the best, we want to be perfect.

So the no child left behind. Every single student in this class is going to really achieve. Anyway, that's the way targets gets set. It doesn't have to be achievable. And there's actually a big branch of mathematics about trying to detect if the boundary conditions are indeed consistent so that they can be achieved, and it's not so obvious to do it.

So I'm not going to get into that at all. You can take a PDE class. There's a math department, and that's all they talk about all day is about how to figure out are these boundary conditions consistent or not consistent. You also have boundary conditions that are repetitive, and don't have enough information. In those cases, you might have an infinite number of solutions.

And I think you can also have cases where you have multiple solutions. So like in the cannonball problem, if you could have a negative v_y , and if the ship was close enough, you

might be able to shoot it two ways. You can shoot it down at it, or you could just blow up high and come down, and they'll both be solutions that satisfy all the conditions.

This is like general non-linear equation problems, that you don't know how many solutions you got, you don't know if you have a solution, you might have an infinite number of solutions, you might have some finite number of solutions, you don't know how many, and it's just that's the way life is, and you'll have to live with it. But it's kind of annoying sometimes.

For now, let's just assume that we somehow know that we have a unique solution. There's a unique physical solution. We're just going to try to find that, and we're not going to worry about this other problem about the fact there could actually be multiple solutions, or no solution, or whatever. When you actually run into a case like that, then you'll really worry about it a lot. Is there a question? No? OK.

All right, so we had an example in actually the zeroth homework, the very beginning, that had a problem like this. The homework 0P1, we had a problem where it gave you-- Do you remember this? It was a cylinder, and you were trying to compute t of r . Do you remember this? And it gave you some conditions about here.

It gave you t when r was equal to r , and it gave you dt/dr -- actually, it gave you some equation. It was actually a function of this and t , all evaluated at r . It was equal to something. Do you remember this? So we gave those two, but then we also told you in the most recent homework, oh, by the way, dt/dr has got to be equal to 0 at r equals 0.

So this is one of these ones where we should only have two conditions, but we told you three conditions. So were any of you able to achieve all three conditions? No. OK, so this is highlighting the problem. So we gave you three boundary conditions that were apparently incompatible, that you couldn't actually solve them all at the same time. So beware, I guess.

Now, in that problem, it might have been more natural to give you this equation, which we are sure is exactly true by the symmetry of the situation as one of the conditions, and then only one of these other conditions outside. That whichever one we were more sure was correct, might be the thing to do. And so we kind of gave you some goofball conditions. I guess, in some ways. Yeah?

AUDIENCE: So if all three of those conditions are true, why wouldn't they converge to a solution?

PROFESSOR: They could all be true, and if they were all true, then if you solve the numerical exactly, you should get the real solution. It should satisfy all of them.

AUDIENCE: So does that mean one of them wasn't true in our case? If we weren't able to do--

PROFESSOR: Yeah, I think one of them wasn't true, and I think it's maybe due to round off. But it's highlights a problem. If you have three conditions that should be true, but they involve numbers in them, then often they won't all be exactly consistent with each other.

But there's actually a worse kind of problem because it's close to being consistent to all of those conditions, but you're not exactly consistent. But then if you get into it, trying to figure what's going on, a lot of times you'll get near singular matrices and stuff because you have things that are nearly linearly dependent in some way.

So really, it's better if you just have the right number of conditions that you're really sure really specify everything correctly, and just go with those, and treat the ones that you think would be nice if they were true as approximate things that are sort of physical checks to see whether your solution is reasonable.

And we did it sort of like that in the problem you did. We said well, dt dr should really be equal to 0, but we didn't demand that in your solution. Your solution didn't actually satisfy that, and we use that as a measure of how far off your solution is. That's actually pretty useful in practice. You get a solution.

It's good to have some additional condition that you can test for physical reasonableness of the solution, and particularly because there's a problem that you might have multiple solutions, and maybe one of them is a physical solution, and one of them is some other crazy solution. And so you need to have some way to detect the crazy solution by some other condition that you can test. Does that make sense?

All right, so I said you could have cases where the Jacobian of your equation-- so you could be trying to solve this, and it could be, for some problems the Jacobian always has both positive and negative eigenvalues together. And in that case, using IVP is not going to be a smart idea because no matter which direction you integrate in, you're going to have a positive eigenvalue, and so then the IVP is not good.

So instead, we're going to use a different method, and we'll talk about that next time, and those are called relaxation methods. And the concept of these is different than it was for

shooting. So the relaxation method is you have a wide guess that you're going to have a guess for the entire y throughout the whole domain.

Actually, a lot of times you might set this up so that y guess actually exactly satisfies the boundary conditions. So you know it satisfies those equations right from the beginning just from the way you set up y guess, but it won't satisfy the differential equation. And then what you'll do is you'll say how badly does it fail to satisfy that g is equal to 0?

Because we really want g is equal 0 for the true solution. And we'll call this failure of our y guess to solve this, we'll call that the residuals. So we can actually plug in y guess in here, and y guess in here, and evaluate this. And normally it will not be 0 because our y guess is not perfect, and so it's going to deviate from 0.

But how big it deviates from 0 is going to be our measure of how bad that guess was, and then we're going to try to change y guess to try to make it closer and closer to 0. So the idea is relax it. You get an incorrect solution, and you adjust it to relax it to make it become the correct solution. All right, and so that's the general idea of the relaxation method, and we'll talk about several different forms of that.

Normally, what we have to do is you have to write y guess in terms of some parameters. So y guess of t depends on some parameters, and we're going adjust the parameters, and that's going to change what y guess is. And so for example, we could do a basis set expansion as we did in one of the earlier classes.

We talked about how you could expand y guess as a sum of some basis functions weighted by p 's, and then you can adjust the p 's, and find the best possible set of p 's to make this g as close to 0 as possible, for example. We'll give examples of that next time. There's several choices about that there. All right.