PROFESSOR: You get your quizzes back on Monday. Maybe you've had some time to think about them and reflect on your answers and the problems that were posed, maybe how they compared to the quiz that-- sample quiz that was put online. I think it was pretty fair exam overall. It looked very similar, actually, to the exam that we've given in the past, but we tried to-- tried not to put any sort of chemical engineering problems on top of the numerics, which tends to lead to confusion basically. Spent a lot of time reading and trying to understand what the underlying physical problem is instead of just testing the numerical methods.

Are there any questions about the quiz now that you've gotten it back? You've seen your answers, things that were unclear that were asked on there? Now is a good time for reflection if there are. No. I heard this last homework assignment was really long, so sorry about that. I didn't write this one.

[LAUGHS]

I'm not going to Take credit for it. It's hard to judge sometimes how long these things take. A lot of it has to do with how facile you are with coding up some of the problems. So conceptually they can be simple, but the coding can be quite difficult, so it's hard to estimate. I think this week's, which should be posted and is on the topic of DAE, should be easier. It's only got two problems instead of three. You're likely going to be able to leverage some of the code from your solution last week for this week. So I don't think it should be quite so challenging.

There is a last part to the first problem that seems to always give first-year graduate students trouble. Somehow you guys don't know how to write down energy balances on reactors. So you might ask pointed questions about this at your office hours, so you don't spend time trying to solve the wrong equations. This happens. I think many of you maybe were taught incorrectly or just forgot at some point how to write these things down. So the office hour is a good point to bring that up, and that way you don't spend a lot of time troubling yourself with the wrong system of equations. Any questions? Yes.

AUDIENCE: [INAUDIBLE]

PROFESSOR: What's that? I'd-- well, to be fair, almost all the lectures for the rest of the class are going to be taught by Professor Green. I'll step in and do some review for two sessions, review before the second quiz and review before the final. Since he's leading the lectures, he'll be responsible for largely generating the homework assignments. I'll try to help out with that as best I can, but this will be the last formal lecture you get from me this term. Other questions? Well, that's very, very sweet of you. Thank you so much. Least there wasn't any hushed yes!

Today's lecture is our last one on DAE, so we're only going to do two. You're going to see today that differential algebraic equations are pretty complicated actually. And when they get sufficiently complicated, you really need to reach out to existing codes that are designed to solve particular classes of differential algebraic equations. So for you, it's more important to be able to identify in the models you formulate when these complications arise and what are the essential ingredients of the model that can be put into one of these solvers that you get a result that isn't nonsense.

And that's what we're going to do today. There's going to be lots of examples to work through. So make sure you're sitting next to somebody you like because I'm going to ask you to try to think about these things and discuss things as we go through. Let me review where these complications come from.

So last time we talked briefly about semi-explicit and fully-implicit differential algebraic equations. I told you in principle, you could simulate these things with backwards difference formulas in solving nonlinear equations at each time step and just marching forward. Did this in your homework for ODE IVP So you can do the same thing for differential algebraic equations. But there was a catch to that, and I illustrated that at the very end. So maybe your brain was fried at the end, and you missed this. It's good to recap.

So I showed you an example of a stirred tank where you had some-- you had transport of some solute into the stirred tank, and then you're pulling it out at a different concentration. And we're trying to track the dynamics of the system, the concentration in and the concentration out, and we imagined a problem in which we measured and the concentration in and we tried to predict the concentration out. So we had a system of differential and algebraic equations to solve. And I showed you that if I used the backward Euler method, the lowest order backwards difference formula that I can-- of the canonical class of these things that I can craft, where I approximate the derivative of an unknown with a relative error or an error proportional to the time step delta t, that carrying out one time step with this backward differentiation formula would determine c1 in principle exactly-- if I knew gamma exactly, I would know c1 exactly. And it would determine c2 to within order delta t-squared. It's the local truncation error here was order delta t-squared. So you just substitute this formula in, and you ask does this error term change at all. At some point, I wind up multiplying by delta t, and so I go from order delta t to order delta t-squared It's the numerical error that gets carried around in this calculation.

I just switch the model a little bit. It seems like a irrelevant change, but it turns out to be incredibly significant. So now I imagine a different problem in which I'm measuring c2 the outlet, and I want to predict c1 the inlet. I still have a system of DAEs that I have to solve, and if I applied the backwards Euler method, well, c2 is automatically determined by this algebraic equation. So I know that exactly. I got to go up to this first equation and solve it for c1, and when I do, I have to have an approximation for the derivative. And the derivative is proportional to delta t-- carries an error around with it that's proportional to delta t.

So I know c1 to within order delta t, not order delta t-squared like in the previous model. So there's something fundamentally different about these two circumstances. And all I did was go from c1 to c2 in this algebraic equation. So that's peculiar, and that should make you really suspicious about your ability to solve these problems accurately.

And here was the third example I gave you. So I said imagine this system of DAEs instead. So here's the differential equation. Here's an algebraic equation. Apply the backward Euler method. Well, c3 is determined automatically by this algebraic equation. I know it exactly. C2 is related to the derivative of c3, so I need to approximate it with my backward Euler derivative. And that picks up an error, order delta t. C1 is equal to the derivative of c2, so I need an approximation for c2. The derivative of c2 that's the backwards Euler approximation. That has an error that's proportional to order delta t. But c2 itself also has an error proportional to order delta t. And order delta t divided by delta t is order 1.

So I get c3. I get c2. I even know what c1 is. I solve this problem with my backwards difference formula, and I have no resolution of c1, no clue what that value is. So this is some complicated control scheme that I've set up, and I need to know the value of c1 to figure out how to operate this process. I'm lost. This is never going to work.

So these three problems have fundamental differences between them. And I'm going to show you how to predict when these differences are going to occur, how to name them. So the stirred tank example 1, it carried a local truncation error, one time step order delta-t squared. Stirred tank example 2 carried a local truncation error. It's order delta-t. This third DAE example had a local truncation error that's order 1. No change in delta-t will improve my solution to that problem. It's independent of how I choose to do my time setting, which, of course, is ridiculous.

So here's another problem. And I'd like you to try to do this. We'll see how far you can get through it. You don't have to get all the way through it, but see how far you can get through this problem. So can you apply the backward Euler method to this system of DAEs and try to predict how the air propagates.

[INTERPOSING VOICES]

OK, I don't want to break up the conversation too much, but tell me what you're finding as you try to do this. Yes.

AUDIENCE: We found that c3 has to be 0.

PROFESSOR: OK. How did you find that c3 has to be 0?

AUDIENCE: Stick in this [INAUDIBLE] that's what the difference is of the [INAUDIBLE]

PROFESSOR: Yes.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right.

AUDIENCE: [INAUDIBLE]

PROFESSOR: OK, so this is very clever, and this has to do with how the model itself is formulated. So you did manipulations with the fundamental equations that I wrote down to determine that c3 had to be 0. That's good. We can do that. We can look at the equations. We can figure how to eliminate variables and find out-- c3 in this case always has to be 0. That may or may not be obvious, but one way to do it is take this constraint equation and take its derivative. This has to hold at every point in time, so its derivative must hold at every point in time. Substitute these other two equations in, c1 dot plus c2 dot eliminates c1 and c2, and all that will be left is c3 has to be 0.

So if I manipulate these equations, I know c3 has to be 0. Let's suppose I don't manipulate the equations, though. Let's suppose I just put in the backward Euler approximation over here for the derivative, and I evaluate c1, c2 and c3 at the current time and ask what is c1, c2, and c3 at the current time in terms of what it was at the previous time And you get an answer that looks like this. So there's a simple system of equations that one has to solve. It's a three-by-three system, and it's easy to eliminate. I didn't really expect you-- actually expected this answer to come up because you guys are all very clever, so you try to make things easy beforehand.

But it's-- there's a problem here. If I try to just do the backwards difference formula with the equations as they're written, I'll find out that when I solve this thing that c1 will be c1 minus c2 at the previous time step over 2 plus an order delta t-squared error. C2 will take on the opposite sign plus an order delta t-squared error. C3 will be minus c1 plus c2 over 2 delta t plus an order delta t error. And actually if I were to apply successive approximations, different-- if I step-- take many steps in a row with this backward Euler method, there's nothing to guarantee that these constraints are satisfied exactly.

I can have numerical error in my solution of this algebraic equation. So it isn't necessarily true that in my numerical solution c1 plus c2 is equal to 0. I can show you right here if I add c1 to c2. Well, these two are the same. They'll cancel, but am I guaranteed that these errors will cancel, too? Or will there be a small numerical error that propagates?

So model formulation is key. This problem is like stirred tank problem 2. One of our solutions lost in order of accuracy in the local truncation error. It's not delta t-squared. It's order delta t. And what you want it to do, which is not necessarily a bad thing, was to try to use this constraint to somehow eliminate equations and simplify things. That can be helpful, but it can also be true that if I use this constraint to simplify equations and eliminate things, my numerical solutions may no longer satisfy the constraint at all. They may drift away from that algebraic equation. Does that makes sense? I'm not controlling the error with respect to that equation anymore.

So how do you give a name to this kind of behavior? And the thing one talks about is the differential index of a DAE system. So let's look at the stirred tank example 1, and let's ask this question. How many time derivatives are needed to convert to a system of independent ODEs having differentials of all the unknowns. So there are two unknowns in this system. One is c1, and the other c2. Actually have one equation, which contains a differential of c2. How many derivatives of these equations with time do I need to take in order to get an equivalent ODE system?

And it's just one. If I take the derivative of equation 2, I'll get a differential equation for c1. Now, I have a differential equation for c1 and a differential equation for c2. DAEs of this type, where I only have to take the time derivative once of the algebraic constraints, are called index 1 DAEs. We saw that when we applied the backwards difference formula to stirred tank example 1, the local truncation error was the same as what we would get in an ODE IVP system.

So index 1-- DAEs are easy to solve. They behave like ODE IVPs. You determine whether it's index 1 or not by asking how many derivatives do I have to take in order to get a system of independent ODEs. So I put this together with this, I have two independent ODEs. I could solve these, subject to some set of initial conditions, and the solution might be the same as the solution to the DAE.

Let's do stirred tank example 2 now. So here we go. They look very similar, but now c2 appears in the algebraic equation instead of c1. So let's take a derivative of the algebraic equation, and I get a differential equation for c2. But I already had a differential equation for c2. I want a differential equation for c1. So what do I do? While I know dc2 dt from up here. So let's substitute that in and solve for c1. So a substitute equation 1 in here. I solve for c1. Now, let's take another derivative. Call this equation 3. Derivative of equation 3 now gives me a differential equation for c1. It's also in terms of dc2 dt. If I want-- I don't have to-- but if I want to, I can substitute for that again to just get dc1 dt in terms of c1 and c2. But it took two derivatives to generate a system of independent ODEs from the DAEs. And this is called index 2.

So it's a different character from index 1. We saw what happens. We tend to lose an order of accuracy at index 2. Somehow index 2 problems are more sensitive than index 1 problems.

Here's another example. So DAE sample 3 is going to proceed the same way. So first we need a differential equation. Which one's missing here? We don't have differential equation for c1, and that's what we're hunting for. So let's take a derivative of the third equation, the algebraic equation. We get c3 dot is gamma dot. And we already have an equation for c3 dot. That was 2. So substitute 2 in. So now we got c2 is gamma dot, but we're really looking for something in terms of c1, so let's take the derivative again. We've got c2 [? dot ?] is gamma dot dot. Doesn't help us much, but c2 dot we know is equal to c1. So we substitute one more time and take a third derivative. Now, we have a differential equation for c1. And we have differential equations for c2 and c3.

We can take this is the differential equation for c3. This is the differential equation for c2. This is the differential equation for c1. Some subset of these can be chosen, and we can replace this algebraic equation with a differential equation. This is an index 3 DAE. It's not always a good idea to replace the DAE system with these derivatives.

I can write down-- I can have a particular function that's equal to 0 and higher-order derivatives of that function are also equal to 0. But the solution to those differential equations are not necessarily equal to that function. There are lots of functions that might have this same property that its derivative-- certain number of derivatives of it are equal to 0. So there have to be extra initial conditions or constraints on the solution that confine me to the manifold of solutions that reflects the DAEs. We'll talk about consistent initialization later on. And that's going to fix this problem. So it's not always a good idea to do this, but it's one way of understanding, given this model, what sort of sensitivity is it can exhibit. We try to calculate its differential index. Yes?

AUDIENCE: So [INAUDIBLE]

PROFESSOR: Yes. Yes.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Well, we have to be careful. We need to choose a set that are independent of the others. OK. So like these three are clearly going to be independent, and it's going to be OK. It's going to be a problem if I choose this one and these two. They're not going to be independent of each other. So we have to select independent ones from the set. It may not be obvious what independent means in general.

AUDIENCE: [INAUDIBLE]

PROFESSOR: 1, 2, 6 could also work, yes. Yes. So in general-- not in general. Let's talk about the differential index of a semi-explicit DAE system. Remember semi-explicit meant that the differential variables can be written as x dot or dx dt is equal to some function of x and y. Y are the algebraic states, and they satisfy a separate equation, g of x and y and t is equal to 0. Semi-explicit form.

So with a semi-explicit DAE, the differential index is defined as the minimum number of differentiations required to convert the DAE to a system of independent ODEs. What does that mean? Means let's take the algebraic equations and let's take a time derivative of them. That will give us a new function. Call it g1. It's going to be a function of the differential state, the algebraic state, and the time derivative of the algebraic state. It could also be a function of the time derivative of the differential state, but we know that in terms of f, which is a function of x and y.

So let's not try putting x dot in there for convenience. Let's just write out like this. In principle, we can do this. Let's take two derivatives of it. It will be the same way. It'll give us some function g2, which is a function of the differential state, the algebraic state, and the time derivative of the algebraic state. Let's take as many of these as we need. It will give us a system of equations that we can eventually solve for the time derivatives of all the algebraic states and convert this DAE to a system of ODEs. And the question is how many of these derivatives do I have to take? Index 1, I need to take one. Index 2, I need both sets of equations in order to get something that I can actually solve for dy dt. Index 3, I get to take three derivatives and higher.

OK, so here's an example. Let's see if we can work through this. So we have a DAE system. Can you calculate the differential index of this system? Go ahead.

Not 0. Not 0. ODE IVP is-- are DAEs of index 0. They require no derivatives to generate a system of ordinary differential equations. What do you think?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Index 2 you say. How do you come to index 2? Why is it index 2? How'd you do it?

AUDIENCE: [INAUDIBLE]

PROFESSOR: OK, differentiate the algebraic equation.

AUDIENCE: Stick it in [INAUDIBLE] 4. You get c3 for 0. Then you have to differentiate [INAUDIBLE]

PROFESSOR: Good so differentiate this equation. Substitute for c1 dot and c2 dot. C1 minus c1 will be 0 c2 minus c2 will be 0. So you have 2c3 equals 0. That's still an algebraic equation. We need one more derivative to get a differential equation for the only algebraic state we have, c3. So one more derivative will tell us dc3 dt is 0. So two derivatives to get differential equations. It's an 2 to DAE. Sam?

SAM: What if you can get c3 equals 0 and just eliminate it then wouldn't you write an equivalent some simple equations, but that would be [INAUDIBLE]?

PROFESSOR: You could. So we could write an equivalent system of equations that says instead of this equation, report c3 equals 0. That's model formulation. Here we've given-- we're given a model, and we're asked what index it is. We could formulate a different model, and the model will have a different index. If I were to replace this equation with c3 equals 0, what would the index of the DAE system be instead? Index 1 instead. I already told you index 2 is harder to solve than index 1, so if you can formulate an index 1 DAE, you should probably do it. But it may not be obvious whether you have or haven't. So it's an issue of model formulation. It's a great question.

Does that distinction-- is that clear? Or is it a little-- am I not making it clear how this works? Unresponsive. OK. It's OK. Let's-- there's a generic index 1 example that we can talk about actually.

So let's take a look at a semi-explicit DAE system. So we have x dot is f of xy and t and 0 is g of xy and t. Let's take the time derivative of the algebraic equation. So the total derivative of g with respect to t is the Jacobian of g with respect to x times dx dt plus the Jacobian of g with respect to y times dy dt plus the derivative of g-- partial derivative of g with respect to t. And now let's solve. Let's push the dy dt term to the other side of the equation, and let's substitute for dx dt. We know dx dt is f, so you get this equation here. If the Jacobian is full rank-- if dg dy is full rank, then I can invert this matrix, and I automatically get my system of ODEs for the algebraic states.

What's problematic about this equation when dg dy is not full rank? That should be a partial. That's sloppy. I was doing this at 1:00 last night, so that's why that's there. What's the-- what's problematic about this root finding problem here if dg dy or the Jacobian of g with respect to y is not full rank? You recall? If--

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yes, you won't be able to compute the inverse to get your dy dt's. That's true. Remember this Jacobian-- if it's not full rank, it's determinant is 0. There is no inverse of this thing. We talked once about the systems of nonlinear equations and locally unique solutions, the so-called inverse function theorem. We can only guarantee there is a locally unique solution when I can invert this thing. That if I got in really close to the solution, it looked like a linear system of equations. That linear system of equations had a unique solution associated within. And everything was good. We were happy.

So index 1 DAEs have that property. If I wanted to solve this equation for y, there will be a locally unique solution for y. Higher index DAEs don't have that property. We know that's a sort of unhappy generic circumstance to be in if we have to solve this equation. It can be hard to find those roots using, say, Newton-Raphson because the Jacobian [? e ?] would need to become singular during the root finding procedure. This is connected intimately to what we did for systems of nonlinear equations. So for an index 1 DAE, you can show its index 1 because this Jacobian is invertible.

Let's do some more examples on differential index. This seems to be the most important thing. If I have a model, what's its index because its index is going to tell me how sensitive it is to small perturbations. So here's a model. It's the same as the model you saw before, but now it's two mixing tanks. So here comes an inlet flow q1 carrying concentration c1. Out comes the same flow q1 carrying concentration c2. The mixer has volume v1. And then I blend this with some more of whatever the solute is at a flow rate of q2 and a concentration c3. And those both go into this tank, and they come out concentration c4, flow rate q1 plus q2, and this has volume v2.

Does that look good? Well-posed model? OK, so here's your material balances on the mixers. And I'm going to give you different algebraic constraints. So this is a problem now where we say measure c1 and c3. And we want to try to predict c2 and c4. Can you figure out the differential index of this DAE system? Feel free to talk to each other. It's OK.

OK, the differential index is 1. This one's the easy one right? Just take one derivative of the algebraic equations. Now, I've got dc2 dt, dc4 dt, dc1 dt, dc3 dt. Differential index of 1 is easy to solve as an ODE IVP. This is the natural problem, too. It's useful to think about the inputs c1, c3 and asking about what the outputs are. Physically, this problem seems easier in nature.

OK let's change it now. Same problem. But let's change the algebraic constraints. Now, the algebraic constraint is I measure c3 and c4, And I want to predict c1 and c2. What's the differential index here?

Yes.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Sure.

AUDIENCE: Why was this a differential index of 1 if you have to take a derivative for c1 and this equation?

PROFESSOR: Oh, good question. I'm going to push the slide back, and I apologize for this. And I'll go back for it. So the question was why was this differential index one if I had to take a derivative of both this equation and that equation. So the way to think about this conceptually is I took a derivative of the algebraic equations, the set of algebraic equations. It was one time derivative of a vector valued function. I can see where the ambiguity is there. So it's important to be clear. I took one time derivative over the equations that prescribe the algebraic states.

That's the distinction. So this is index 1 because I needed one time derivative of this type, dg dt of the entire set. Here we go. We're on this one. Prescribe c3. Prescribe c4. What is the differential index now? What do you think? We got an answer?

[INTERPOSING VOICES]

I hear 2. I hear 3. I saw 3 like this. Almost mistook it for a 2, but that's 3. Yes. It's 3. Let's see if we can work through why it's 3.

So I take a derivative of the algebraic equations, one derivative. And after taking that derivative, I'll get a differential equation for one of the algebraic states. So c3 is taking care of. Now, I'm in the hunt for c1. So I have-- after that first derivative, I have an equation dc4 dt is gamma 2. And I know dc4 dt, so I drop that in, and I get an algebraic equation relating the derivative of gamma 2 to c2, c3, and c4.

This new algebraic equation. This is like the g1 that I prescribed in the generic scene. I take a derivative of it again because I'm in the hunt for c1. So I take the derivative of this new equation, and I'll get a derivative of c2, a derivative of c3, and a derivative of c4. And I know all those. I know the derivative of c2. I know the derivative of c4. And I know the derivative of c3 from the previous level in the hierarchy. I figured that out already. So it's two derivatives. Still hasn't gotten me a time derivative of c1. But when I make those substitutions, I'll get an algebraic equation in terms of c1. And I can take one more derivative, and that will give me a dc1 dt. And I'll have an od for c1, c2, c3, and c4. So its differential index 3.

It makes sense sort of. I'm taking a measurement of an output way down the line here. And I'm trying to predict what the input was in the first place. It's easy to imagine that there's a huge amount of sensitivity in that calculation. Here's another one. What's the differential index here? I'm now prescribing c1 and c2, and I want you to tell me c3 and c4.

[INTERPOSING VOICES]

Oh, good. Somebody noticed early on. So you say it's not possible. Right. Why isn't it possible?

AUDIENCE: You take derivatives of both of those--

PROFESSOR: Yes.

AUDIENCE: [INAUDIBLE] you can't isolate c3--

PROFESSOR: Right. We'll never be able to isolate a derivative of c3 here. There's actually there's something wrong physically with this problem. Yes, somehow I'm supposed to measure c1 and c2 and use that to predict c3 and c4. I don't know c4, and c3 is an input. I don't know it either. How do I figure out an input when I don't know the output? It's impossible. So this model is flawed. We can formulate any model we want, pick any set of these variables to prescribe algebraically, but not all of them are going to admit solutions or make sense. There's no number of derivatives that's going to give us a closed system of ODEs.

This again comes back to the point that really DAEs is all about model formulation. There are lots of good numerical methods. They work like the numerical methods for ODE IVPs. What's important is getting consistent models down, models that, for example, have solutions. You might feed this to a DAE solver and get nonsense because I can almost freely prescribe what c3 is, and I'll get some answer for c4. Any c3 can give me an answer here. So who knows whether this numerical solver is sensitive or not to this particular pathology in the problem we've written down.

We got to speed up just a little bit, and I'm sorry for that. So we looked at index 1. That was stirred tank 1. Index 2, that was stirred tank 2. Index 3, that was stirred tank 3. And one thing we saw when we looked at them was this index 1 solution. It had pieces that were proportional to this forcing function gamma, this prescribed function in our system of equations and proportional to its derivative. So if gamma is jumping around, well, c1 will jump around. C2 is going to be smoothed out version of gamma because it depends on its integral. It's not very sensitive, kind of like an ODE IVP. Doesn't really show any more sensitivity than an ODE IVP does.

The dynamics are different for index 2, though. When we look at the solution of the index 2 DAE, we found c2 goes like gamma, but c1 has to go like gamma dot. So that's pretty sensitive if you're making a measurement, and there's noise in the measurement. How do you even know what this derivative is? So c1 is wildly bouncing around. Our prediction of what c1 is is-- it's not going to be very good.

Our solution of DAE example 3 when it told us c1 was related to the second derivative of the forcing function, c2 to the first and c3 was proportional to the forcing function. So this is hugely sensitive to changes in the forcing function. So the higher the index goes, the greater the sensitivity to perturbations in the system.

Here's another simple example. You have all that data, actually, in your paper, so I won't ask you to do this one given our time constraints. But here's-- mechanical systems that have constraints are often indexed 3 DAEs it turns out. You can show this one is an index 3 DAE. This is the case of a pendulum swinging back and forth. So it's a-- change in position is its velocity. Its acceleration balances gravity, and there's some arm that holds the pendulum in place. We can imagine that that arm has some tension in it. It acts like a spring with a spring constant that changes in time in order to hold the pendulum at a fixed distance from its center.

So two differential equations, one algebraic equation. This is a differential algebraic equation. You can imagine lots of mechanical systems work in exactly this way. They can only move along prescribed paths. They're constrained in how far they can stretch or go. I don't know. If you're trying to design a robot or something, DAEs are pretty important, and they're all of index 3 type it turns out.

So differential variables here are x and v, and the algebraic variable is the spring constant, k, which has got to adjust dynamically in time in order to-- got to get away from the speaker-- adjust dynamically in time in order to provide just the right stiffness to keep this going around on a circular orbit. So when-- let's suppose I start with my pendulum down, and it's not moving, then this has to be just stiff enough to balance gravity. The pendulum is 90 degrees, and it's not moving. I don't need any stiffness. K and just be 0. There's no forces to balance. The pendulum is swinging around, and I have to be stiff enough to balance gravity where I am and also counteract any sort of centripetal [? acceleration. ?] So this k has got to be wildly fluctuating. If I was trying to control that k somehow to give this system these particular dynamics, you might imagine it's very difficult to do.

So there's your differential variables. Here's your algebraic variable. You take a derivative of this equation, you'll get a constraint that tells you the velocity is orthogonal to the position. Of course it is. I'm on a circular trajectory. You take a derivative of this equation now, this algebraic equation. You'll get another constraint that gives you some relationship between k, x, and v but not a differential equation for k. Take another derivative of this algebraic equation, and you'll get a differential equation for k. So in principle, I could formulate a system of ODEs with dk dt, dx dt, dv dt will be equivalent to this. It took three derivatives to do this, though. So it's an incredibly sensitive system. It's index 3 in nature.

So if I transform to this equivalent set of ODEs, the problem-- we discussed this earlier-- is that the solutions may drift away from the initial set of constraints. The solutions also need the right sort of initial conditions. Here I know that at time 0, c3 better be gamma 0. I know that in the actual solution to this problem, c3 dot was gamma dot. So at time 0, c2 better be gamma dot of 0. And at time 0, c1 better be gamma dot dot of 0. And if it's not, then there's going to be some step jump or some strange behavior in the solution to this ODE system. So I have to choose the right sort of initial conditions. If those initial conditions aren't chosen correctly, then I'm not going to get the right solution. I won't be constrained to the manifold of solutions that's given by the original DAE.

So here's some things you need to know. Index 1, semi-explicit DAEs can be safely handled in MATLAB. So ode15s, ode23t, they have the ability to take an input of mass matrix. We discussed mass matrix last time. And a right hand side to the system of equations in f of x and t and solve it just like all the other solvers, all the other ODE IVP solvers in MATLAB. If it's not index 1, MATLAB can catch it. So it'll try to look at the Jacobian of f with respect to the variables and the mass matrix and to infer from that what the differential index is. And it will tell you often times-- I think depends which package you're using. But if you're using certain packages, it'll tell you. It's not index 1oe-- DAE. We can't handle this. The methods built into it aren't suitable.

For generic DAEs, there are specific DAE solvers. So something like SUNDIALS or DAEPACK-- this is Professor Barton's software actually-- that are designed to handle DAEs up to some certain index instead. They're built to reliably solve these problems. So we have robots with constraints on them. We solve all sorts of problems for chemical process systems that are of higher index. And the way we do it is using specific numerical methods. They're based on the same sorts of schemes, backwards difference formulas, but careful analysis of the equations, which are in general unstructured, the software doesn't know beforehand what those equations look like, but a careful analysis of them in order to figure out how to minimize numerical errors and stay on the correct solution manifold.

As input to these, though, we have to give initial conditions. So that's going to be the last thing that we talk about here. And those initial conditions have to be prescribed what we call consistently, or we can get numerical errors, or the software can just quit and throw an error, or it can run off on some other solution manifold that doesn't look anything like the solution we're after. The pendulum is an interesting example to think about. So you might say, well, what do I want to know about this pendulum. Maybe I want to know where it started initially and what its velocity initially was. Can I prescribe the initial position of the pendulum arbitrarily? What do you think yes or no?

AUDIENCE: No.

PROFESSOR: No. Why not? Why no?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Good, yes. We know it's got a fixed arm length. Can't pick any position. That's not going to work. It's got to satisfy that algebraic constraint. Can I specify its velocity arbitrarily? What you think? Can I give it any initial velocity that mass at the end of the pendulum?

AUDIENCE: No.

PROFESSOR: No. Why not?

AUDIENCE: [INAUDIBLE]

PROFESSOR: What's that?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Well, explain that to me, though, because I had equations for the velocity and for that acceleration and then an equation that told me that the length of this moment arm had a certain length associated with it. So why does that constrain the velocity? You're right. It does. I'm not saying you're wrong. You're right, but why? Those initial three equations don't tell me anything about the initial velocity was supposed to be.

We know something about the trajectory this is supposed to sweep out, though. It's supposed to sweep out a circular trajectory. And the velocity in that circular trajectory is always going to be orthogonal to the moment time. And that equation actually-- it appeared. It appeared when we took a derivative of the constraint. So this was a hidden constraint. It's secret. You didn't know it was there until you took a derivative, and you tried to go and figure out what index this was. Then you discovered velocity and position or orthogonal to each other. My initial condition should respect that. How can they not? The initial conditions have to be a solution to the equations as well.

So now can the initial stiffness be specified arbitrarily? No. There was also an algebraic equation that popped up at some point right here when I took the second derivative, which tells me how the stiffness has to be related to position and velocities. It's also hidden inside the structure of this DAE. So this-- again goes to model formulation, what are the initial conditions that I have to prescribe? If I prescribe the right ones, I'll get a solution that matches the dynamics of the problem I was actually interested in. If I prescribe them incorrectly, who knows what's going to result? We can't really predict. Depends on the solver we're working with.

Here's a formal way to think of it. So usually in an initial value problem, usually we want to know what's-- for these first order difference-- differential equations, what's the initial value of the state, and what's the initial value of its derivative. That completely specifies what's going on at the beginning. One way to think about these problems is I've got this equation that I want to solve. I could think in some sense that x and x dot are independent of each other. I could always write this is something like an equation for f of x and z. And then I could add some extra information that tells me actually x dot is the same as z.

So initially, I would like to know the values of x and z, that z is x dot. So I'd like to know initially the values of x dot and x. And tell me where I start, and these need to be consistent with the initial value problem that I specified. If I'm doing an index 0, an ODE IVP, at least be consistent with this. So if you tell me x naught, that I should put x into this equation, then that should tell me what x dot is. If you tell me x dot, then I should solve my governing equation for x naught. You could tell me some equation that relates x naught and x dot initially, and now I got to solve this equation in conjunction with my governing equation to determine these-- this set of values. I'd like to know both of these things initially.

The fully-implicit DAE, I really have two n unknowns here. I don't know x, and I don't know x dot. And I've only got n equations for those. So apparently there's n degrees of freedom to specify here. I need n more equations to say what those initial conditions are. And these hidden constraints I point out actually reduce the number of degrees of freedom. The fact that these are not index 0 problems but have a finite index will introduce extra constraints that reduce the number of degrees of freedom or the number of other equations I can use to specify those initial conditions. Does that make sense? Let me give you some examples, and then I'll let you go.

So if I have a problem that has separate differential states and algebraic states, this is like f of x dot x and y and t equal 0, the set of things that I need to specify-- I need the initial derivative of the differential state, the initial value of the differential state, and the initial value of the algebraic state. I'd actually need to know the initial value of the derivative of the algebraic state here because I don't need that to satisfy this initial equation.

Let's look at a couple of examples. So here's stirred tank example 1. I converted to a system of ODEs by taking the derivative of equation 2 here. So here's my two ODEs. I got to look back at my initial equations and see how do they constrain the initial conditions. So equation 2 tells me that c1 initially better be gamma initially. Otherwise, my initial condition doesn't satisfy the governing equations of the model that I wrote down. Equation 1 tells me that the derivative of c2 initially better be related to the derivative of c1 and the derivative of n-- I'm sorry to the value of c1 and the value of c2 initially.

This third equation that I came up with, it doesn't constrain. It's a derivative of one of the original algebraic variables, so it doesn't constrain that original set of-- I need to know the initial condition on the differential variables, the derivative of the differential variables, and the algebraic variables. This only tells me something about the derivative of this algebraic variables. This equation doesn't give me any other constraints. So I just got a free variable that I can specify. One more equation. It can be whatever I want.

So maybe I say c2 of 0c0, or maybe I say there's some relationship between c1 of 0 and c2 0. Or maybe I say that c2 of 0 and c2 prime of 0 are related in some way. Whatever I specify, I have a system of three equations for three unknowns, c1 of 0, c2 dot of 0, and c2 of 0. I need to have a unique solution to that. Otherwise, there's going to be a problem solving for it in time, the system of equations. So I have to choose this one consistent with these other ones, but beyond that I'm OK.

Here's stirred tank example 2. So I take a derivative of equation 2, and that gives me a differential equation for dc2 dt. Actually, I already had one of those, so I substitute for dc2 dt, and I take another derivative to get a differential equation for c1. And let's ask about the initial conditions. So the initial algebraic equation, it constrained dc2-- it constrained c2. It said c2 of 0 has to be gamma 0. The initial differential equation told me that c1 of 0 had to be equal to c2 of 0 plus v over q, c2 dot of 0. Remember, I need three equations to specify c2 of 0, c1 of 0, c2 dot of 0.

I actually can't specify c2 dot of 0 freely because in the process of deriving this differential equation here so that I had dc1 dt and dc2 dt, I introduced a constraint on the derivatives of c2. And this tells me that c2 dot of 0 has to be gamma dot of 0. There are no free variables to specify. There's three equations. Two of them come from the initial system of DAEs, and the third one is related to this extra equation that popped up as we tried to define the index. It's an implicit constraint on the problem. Does that make sense? I can't specify these things freely. They're specified by the signal that I measured gamma.

There are two examples here that you should be able to work through. One is an index 2 example, and one is an index 1 example. We're not going to have time to go over them. Sorry for that. But they work the same way. So convert to a system of ODEs, ask what initial conditions will satisfy the initial constraints and were there any other-- were there any other constraining equations that popped up? So if you work through this example, you'll find out there was a hidden strength that either the initial derivatives of c1 and c2 sum together have to be 0 or equivalently the initial value of c3 had to be equal to 0. You can take this constraint in either place. Maybe you want to take both of them to make sure that it's always satisfied, that there is no numerical error that drives you off of this constraint. It's OK to have these things over constrained. It's just not OK to have them under constrained.

And you'll find out that while there were five things I could pick, the derivatives, and the values of c1 and c2, and then the algebraic variable c3 initially. So there are only four constraints-- for equations here for the initial conditions. I get to pick one more. I can get it however I want. Just has to be consistent with these.

There's one more example. Again, should work through these on your own. But you do the same thing. Convert to a system of ODEs. You'll find out that when you do this, you'll get a differential equation for c3. It didn't introduce any new constraints on c1 dot, c2 dot, or c1, c2, and c3. So you just need to satisfy-- your initial conditions have to satisfy the initial equations, and you can put in two more conditions, whatever you want them to be. Add five unknowns and only three equations for the initial conditions. And that's it. So thank you very much.