

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](http://ocw.mit.edu).

**WILLIAM GREEN,** Shall we get started? Yes. Apologies for the notes being posted so late. I had trouble with the new two factor authentication system, which I need to get into Stellar and put things up there. **JR.:** My phone wasn't working so I had to find my iPad someplace and use that to get credentials to log in.

So if you don't have them don't worry about it. The full notes are posted online, no missing pieces. So you can utilize those afterwards if something is unclear. All right?

So your quizzes are graded. The TAs have them. I think overall, we were really pleased with the outcome of this quiz. It is very conceptual in nature and we thought people really understood a lot of the material quite well. And you can see that from the distribution of scores.

So the mean on the quiz was about 77, a standard deviation of 12. I think everyone did really quite well. And on a number of the problems, perfect solutions were posted.

So I'll take a second here. If there are any questions about the material that was on the quiz, or the format of it that you'd like to have addressed, I'm happy to answer them. No?

Does it seem like it took too much time to complete all three questions in the two hours allotted? Or was it about the right length of material for the time slot? So we vetted it with the two TAs. And it took each of them less than an hour to complete it. So that was a good sign that we thought you guys should be able to get through most of the material in two hours.

So let's recap. We're going to move on today to a topic called differential algebraic equations. But before doing that, there are a couple of concepts I want to review from numerical integration, or actually concepts that weren't covered when we talked about numerical integration on Friday that I think are important. And then I want to review briefly implicit methods for ODE-IVPs because those are going to be important for differential algebraic equations.

So for numerical integration we talked about quadrature of various integrals, how to develop quadrature formulas. But there's actually one type of integral we didn't talk about there, which is sort of improper integrals, where the limits of integration are unbounded. And these come up all the time in engineering problems.

We'd like to know what the, say, integrated response of some function is over all possible times. And we may have to evaluate that numerically. So how do you do those sorts of integrals? And it turns out the strategy that's most often used is to divide this domain up into two pieces.

One piece, which is proper integral over a finite domain, and another piece, which is an improper integral over an infinite domain. This first integral, you can handle it with an ODE-IVP method or some higher order polynomial interpolation. But the second one we have to do differently. And there are couple of ways to do this.

One is to transform variables. So you map this domain onto a finite domain. And the other option is to substitute an asymptotic approximation. So we say, when, say,  $\tau$  is large,  $f$  of  $\tau$  has a characteristic asymptotic approximation. Maybe it's exponentially decaying. And we take the integral of that asymptotic approximation. And the more accurate that approximation is the better our approximation for this whole integral will be.

There's another type of interval that we have to do, where the same idea can be applied. That's the integral over integrable singularities. So here's an example. So say we want to integrate the function cosine of  $\tau$  over square root of  $\tau$  from 0 to some finite positive time point.

As  $\tau$  goes to 0, cosine goes to 1 and square root of  $\tau$  diverges. But this integral actually has a finite value. This 1 over square root of  $\tau$ , that's an integrable singularity. So how do you do this?

Well, you want to split the domain again into two parts. One part that contains the integrable singularity and the other part, which excludes it. And in the part that contains the singularity we might do an asymptotic expansion of the function, and integrate each of those asymptotically accurate terms separately.

So the integral of 1 over square root of  $\tau$  is going to give us  $2\tau^{1/2}$ . The integral of this ratio here is going to give us  $-\frac{1}{2}\tau^{5/2}$ . And then we have an integral over

a finite domain, which doesn't include the singularity.

And the accuracy of this method can be dictated by two things now. One is how accurately can we do this integral. And the other is how accurate is our asymptotic expansion here.

So if we want higher accuracy we need more terms. We might need to be selective about how we choose the end point for this domain in order to minimize the error. There are lots of methods to do this. Sam.

**AUDIENCE:** Will you talk a little more about what typical integral [INAUDIBLE] is?

**WILLIAM GREEN,** Yeah. So you know these things. If I try to integrate  $1$  over  $x$ , as  $x$  goes to  $0$  this thing will always diverge. You know the antiderivative of  $1$  over  $x$  is going to be the log. And the log diverges. So power is weaker than  $1$  over  $x$ , like  $1$  over  $x$  to the  $1/2$ .

**JR.:**

Don't diverge like the log. They actually go to a finite value. The log is like the most weakly singular function there is. And integrals over weaker powers of  $1$  over  $x$  actually don't produce a singularity anymore.

They're what we call integrable singularities. The function itself diverges but its integral does not. So you usually have this asymptotic power law type behavior. OK.

So they come up all the time. We see these things in places like, say, squeezing a thin film of fluid between two plates. How long does it take for these two plates to come together? You might say, well, as the plates get closer and closer together the pressure starts to diverge in the gap and they'll never come together. But maybe depending on the geometry of the plates there can be a sort of finite time at which they'll come together. Depends on their geometry.

So that's one topic. I think that's useful because these things come up all the time. And you can't actually handle those cases with quadrature by itself.

The function diverges. There's no polynomial interpolation that's suitable to match it. So the quadrature can't handle it. And if you're trying to integrate over an infinite domain, well, good luck. You're never going to be able to fit enough points in to know that you've accurately integrated out to infinity.

The other thing I want to recap here, and this is a problem for you to try. So here we have a simple first order ordinary differential equation. We want to use the implicit Euler method to



And if I plot the real part of  $\Delta t \lambda$  versus the imaginary part of  $\Delta t \lambda$ , this inequality is represented by this pink zone on the outside of the circle. So any values of  $\Delta t \lambda$  that live in this pink area will give stable integration. And the solution  $x_k$ , well, the  $k$  to zero as  $k$  goes to infinity. So it makes sense?

This is a little funny though. What does the solution to this equation do when  $\lambda$  is bigger than 1? When it's bigger than 0, excuse me. When  $\lambda$  is bigger than 0, what does the solution to this ODE-IVP do? Does it decay?

**AUDIENCE:** [INAUDIBLE]

**WILLIAM GREEN,** It blows up. It grows exponentially. But the numerical method here, when I have  $\lambda$  is bigger than zero, can be stable and produce decaying solutions. So this is sort of peculiar.

**JR.:**

So we're often concerned when we solve ODE-IVPs and we try to use these sorts of backwards difference methods, these implicit methods, to solve them. We're often concerned with getting stability. We want to get solutions that decay when they're supposed to decay.

Sometimes we get solutions that decay even when the solution is supposed to blow up. So that's sort of funny. So the exact solution should look like this.

But there are going to be circumstances where  $\lambda$  is bigger than zero. In fact,  $\lambda$  is bigger than--  $\lambda \Delta t$  is bigger than 2 along the real axis, where the solution will actually blow up rather than decay away. So stability and accuracy don't always correlate with each other.

We really have to understand the underlying dynamics of the problem we're trying to solve before we choose a numerical method to apply to it. And this is going to be true of differential algebraic equations as well. Are there any questions about this?

There's a discussion of these things in your textbook, these sorts of stability regions associated with different backwards difference methods. The implicit Euler method is one backwards difference formula. It's the simplest one. It's the least accurate one that you can derive but it's an easy one to use.

But now we want to move on to a different sort of problem. And it's one that we come across in engineering all the time. These problems are called differential algebraic equations. And they're problems of the sort a function of some state variables, and the derivatives of those

state variables and time is equal to zero. And there are some initial conditions. The state variables at that time 0 is equal to some  $x_0$ .

So this is a vector valued function of a vector valued state and it's time derivatives. And we want to understand the dynamics of this system. How does  $x$  vary with time in a way that's consistent with this governing equation?

Usually we call these well-posed problems when the dimension of the state variable and the function are the same. They both live in the vector space  $\mathbb{R}^n$ . So we have  $n$  different states. I have  $n$  different functions that I have to satisfy. And I want to understand how  $x$  varies with time.

Here's an example. A stirred tank. I call this stirred tank, example one. So into the stirred tank comes some concentration of solute. Out of the stirred tank comes some different concentration of the same solute. And we want to model the dynamics as a function of time.

So we know that a material balance of the solute on the stirred tank, if it's well mixed, will tell us that  $\frac{dc_2}{dt}$  is related to the volumetric flow rate into the tank divided by the volume of the tank multiplied by the difference between  $c_1$  and  $c_2$ , the amount carried in and the amount carried out of the tank.

And let's put a constraint in here too. Let's suppose we're trying to control this tank in some way. So we say that  $c_1$  of time has got to be equal to some control signal,  $\gamma$  of  $t$ .

And then we want to understand the dynamics of this system. At time 0,  $c_2$  is some  $c_0$ , the initial concentration of the tank. And  $c_1$  is whatever the initial value of this control signal is.

And the solution is  $c_1$  equals  $\gamma$  of  $t$ . And  $c_2$  is, well, the initial concentration's got to decay out exponentially. And then I've got to integrate over how the input is changing in time. Those also produce exponential decay.

So if the input is a little delta function spike then I'll get an exponential decay leaving the tank. If the input is changing dynamically in time, I need this convolution of the input with this integral. So you just to solve this system, this ordinary differential equation by substituting  $c_1$  in there.

But this system of equations here is not just a differential equation. It's a differential equation and an algebraic equation. There's something peculiar about that that's different than just

solving systems of ordinary differential equations.

Oftentimes in models, we write down all the governing equations associated with the model. Some of them may be differential, some of them may be algebraic. There are times when we can look at those equations and see clever substitutions that we can make.

But if you have a hundred equations, or a thousand equations, or a million equations, there's no way to do that reliably. We just kind of wind up with a system of ordinary differential equations and algebraic equations mixed together. And we have to solve these reliably.

Here's another example. Oh, excuse me. So let me write this in this form. So we said there should be a vector valued function of the states and the derivatives of the states. The states are the concentration  $c_1$  and  $c_2$ . And the derivative here is  $dc_2/dt$ .

So here's my vector valued function. The first element is this, and it's equal to 0. The second element is this, and it's equal to 0. Here's my initial condition over the states. So I can just transform that list of equations to the sort of canonical form for a differential algebraic equation.

Here's a separate example. Suppose instead, I'm trying to either control or make a measurement of the outlet concentrations  $c_2$ . And we call that  $\gamma$  of  $t$ . And I want to know now what is  $c_2$  and what is  $c_1$ ?

So I've got to solve this system of equations for  $c_2$  and  $c_1$ . The solution for  $c_2$  is easy. I know that.  $c_2$  is  $\gamma$ , by definition. I got to substitute this into the first equation and solve for  $c_1$ .

And I see,  $c_1$  is  $\gamma + v$  over  $q \dot{\gamma}$ . And I have to have initial conditions to go along with this. There's something funny here. We can see the initial condition for  $c_2$  has to be  $\gamma_0$ .

The initial condition for  $c_1$ , what does that have to be? Well, it needs to be consistent with the solution here. There were no free variables when I solved this equation for  $c_1$ . It isn't like solving a differential equation, having a free coefficient to specify.

So it better be the case that this  $c_0$  here is the same as  $\gamma_0 + v$  over  $q \dot{\gamma}_0$ . Somehow I have to know this input signal to prescribe the initial conditions for  $c_1$ . So that's peculiar and different from just ODE-IVPs.

You see this picture? You see how this goes together? The solution is funny too. Suppose we are trying to use this system of equations to do the following. We are trying to measure  $c_2$  and use the measurement of  $c_2$  to predict  $c_1$ .

So  $\gamma$  is the measurement, and we're trying to solve this system of differential algebraic equations to predict what  $c_1$  is. All measurements incur numerical error. So even though our signal for  $\gamma$  that we measure may be continuous, it's bouncing around wildly. It's not a constant signal. It's going to move around a great deal.

And  $c_1$ , our predicted value for  $c_1$ , depends on the derivatives of  $\gamma$ , which means  $c_1$  is going to be incredibly sensitive to how that measurement is made. So it's peculiar. It means that there's going to be a lot of problems with stably solving these equations because the solutions can admit cases where they're not integrals of the input but they're actually related to derivatives of the input.

Look at the solution back here again. Oop, sorry. The solution here was related to an integral of the input. Suppose I was making a measurement of  $c_1$  in trying to predict  $c_2$  instead?

My prediction for  $c_2$  would be related to the integral of that measured signal. And we know integrals smooth signals out. So it's not going to be so sensitive to the measurement.

So one way of doing this seems really sensible. And the other way of doing it seems a little bizarre. Well, we can construct these equations however we want. So there's something about how we formulate models that are going to make them either easily solvable, as DAEs, or quite difficult to solve.

So these pop up all the time in engineering problems. They pop up in dynamic simulations, where we have some sort of underlying conservation principle, or constraints, or equilibria. So if we have to conserve something, like total energy, total mass, total momentum, total particle number, total atomic species, total charge, there will always be an algebraic constraint that tells us the total mass has to stay this, or the total number of atoms of a certain type has to stay this. Yeah, Ian.

**AUDIENCE:** Just to be clear, were you saying, on the stirred tank example, that those were the same problem with different solution approaches?

**WILLIAM GREEN,** So good question. Let me go back. So they are fundamentally different problems. So in one case I specified the value of  $c_2$  to be this  $\gamma$ . I tried to make a measurement of  $c_2$  and

**JR.:**

then predict  $c_1$ .

In the other case, I specified  $c_1$ . I tried to measure  $c_1$  and predict  $c_2$ . So one case I measured the input and tried to predict the output to the tank. And in the other case I measured the output and tried to use it to predict the input. Yeah?

**AUDIENCE:** So physically the same system with a choice by the operator.

**WILLIAM GREEN, JR.:** Yes. That's right. That's right. So I formulated the model differently. The same underlying system but I formulated the model differently. And it led to completely different behavior in the underlying solution to the differential algebraic equations.

So you've already solved problems that involve conservation. You've done things like ODE-IVPs on batch reactors, where the total amount of material in the reactor had to remain constant. Instead, you probably solved for the dynamics of each of the components undergoing the reaction. You might have checked to see if at each point in time the total mass in the reactor stayed the same.

Because you solved all these differential equations, they were interconnected with each other but they all incurred some numerical error. It's possible that numerical error accrues in a way that actually has you losing mass from your system. So there may be benefits to actually trying to solve these sorts of systems of equations with, say, one less differential equation for one of the components and instead an algebraic constraint governing the total mass or total number of moles in the system.

So these pop up all the time. You see them in models of reaction networks, where we try to use a pseudo steady state approximation. We say some reactions go so fast that they equilibrate. So they're not governed by differential equations but by algebraic equations for equilibria.

They can pop up in models of control, where we neglect the controller dynamics. So you say I'm going to try to control this process and I get to instantaneously turn on or off this valve in response to a measurement. Actually, controllers have inherent dynamics in them. But if I don't put those dynamics in then I may get an algebraic equation instead of a differential equation for how the control process occurs.

There are different types of DAEs that we talk about. So there's one type that's called semi-

explicit DAEs. And in these types of differential algebraic equations we can write them in the form  $M \frac{dx}{dt}$  is equal to  $f$  of  $x$  and  $t$ , some initial condition.

You say that this looks like an ODE-IVP.  $M$  is called the mass matrix. And it may or may not be the case that  $M$  is invertible.  $M$  may or may not be a full rank matrix.

So let's look at that stirred tank example one. This was the underlying equation,  $f$  of  $x$  and  $\frac{dx}{dt}$  and  $t$ . Can write it in semi-explicit form. If  $c$  has two components,  $c_1$  and  $c_2$ , then  $\frac{dc_2}{dt}$  -- oops, this is a typo here.

I really apologize. This should be a 0, and this element should be a 1. I'll correct that online. I apologize for that.

So this should be this matrix multiplied with this differential should give me  $\frac{dc_2}{dt}$ , this term here. And a 0 for the second equation. And then I've got  $q$  over  $v$  multiplying  $c_1$  and minus  $c_2$ . So that gives me the first line here.

I've got minus  $c_1$  coming on the second equation here. And those balance with a 0 and a gamma. So the lower line reads like this equation here and the upper line -- fix this typo -- replace 0 with one, reads like the upper equation here. This is the semi-explicit form of the differential equation.

You can see this mass matrix is singular and has a row that's all zeros. There's no way to invert this matrix and convert it into a system of ODE-IVPs, which you already know how to solve. If the mass matrix is full rank, which can happen, you can formulate your model in such a way that it takes on this semi-explicit form.

But if the mass matrix is full rank and invertible, then you just invert the mass matrix. And now you can apply all your ODE-IVP techniques to solve this thing.

The mass matrix doesn't need to be constant. Could depend on the concentration. It can depend on the states as well in this form.

I just chose an example where that isn't the case. But in general, it can depend on the states. If it depends on the states and we invert it, then we need to be sure that the mass matrix is invertible for all values of the states, which may or may not be easy to ensure.

So here's what I'd like you to try to do. So here's that stirred tank example two. The only

difference between this and the previous one is that  $c_2$  now is set equal to  $\gamma$  instead of  $c_1$ . I want you to try to write it in semi-explicit form. A sort of test of understanding.

Can you define the mass matrix? Can you write out the right-hand side of the equation in semi-explicit form? Take a second, work with your neighbor. See if you can figure out how to do that.

**AUDIENCE:** Is this clear? OK to do this?

**WILLIAM GREEN,** You can look back on the previous slide. Actually, the semi-exclusive form is going to be the same. This is a 0, that's a 1.

The only difference is going to be  $c_2$  is now balanced with  $\gamma$ , the input. So this minus 1 is got to shift over. So I switched the 0 for the minus 1, you got the second, the semi-explicit form.

There's another way that these equations can pop up. So we have this mass matrix. And it might be the case that  $M$  is diagonal over many of the states and then 0 for all the other states.

So we might be able to write the equation in this form. This comes up all the time. So we might be able to write  $\frac{dx}{dt}$  is a function of  $x$ ,  $y$ , and  $t$ . And 0 is equal to another function of  $x$ ,  $y$ , and  $t$ .

And we have some initial conditions prescribed for the  $x$ 's, of which we're taking the differential in one of these equations. And we've also got to satisfy those initial conditions for these variables  $y$  associated with the second nonlinear equation. The  $x$ 's are called the differential states because they're involved in equations where they're difference with respect to time, or the differential with respect to time is taken.  $y$  are called the algebraic states because they only appear as themselves and not as their time differential in any of these equations.

So we might want to solve for both the differential and the algebraic states simultaneously. We have to know these algebraic states to determine the differential states. We have to know the differential states to determine the algebraic states.

**AUDIENCE:** Professor?

**WILLIAM GREEN,** Yes.

**JR.:**

**AUDIENCE:** [INAUDIBLE]

**WILLIAM GREEN,** That's right. So here the  $x$  that I have is not all of the states. It's not all of the unknowns. It's

**JR.:** only the set of the unknowns of which I have to take a differential. That's right, that's right.

So we've sort of tooled down the kinds of equations we might want to look at. We've gone from the generic differential algebraic equation to a semi-explicit form to the splitting, this differential algebraic splitting. Many problems can be formulated in this fashion. This one's the easiest one to think about so that's the one that we'll discuss for most of the lecture.

So let's look at some examples. I think examples are interesting to think about. So here's one. Think about a mass spring system. There's an algebraic equation that governs a conserved quantity, the total energy of this mass spring system.

So the kinetic energy,  $\frac{1}{2} m$  times the velocity of the mass squared. Plus the energy stored in the spring,  $\frac{1}{2} kx$  squared has got to be equal to the total energy. And the total energy can change in time.

So we have a differential algebraic equation. We have a differential state because we're taking the time derivative of  $x$ . And we'd like to be able to determine  $x$  as a function of time. So  $f$  of  $x$ ,  $\dot{x}$  and  $t$ . That's this equation,  $\frac{1}{2} m \dot{x}^2$  plus one half  $kx^2$  minus  $E$  is equal to 0.

We'd like to solve this differential algebraic equation. It's well-posed. We have one equation for one unknown, one state. The equation has a solution, which is  $A \cos(\omega t)$ .

It's not the only possible solution. It's a solution to this equation. We can substitute that solution into the equation and determine, for example what this oscillation frequency  $\omega$  has to be. It's square root of  $k$  over  $m$ . Or what the amplitude has to be. It's the square root of  $E$  over  $k$ .

So you've seen this from physics before. We've already solved differential algebraic equations. The thing you saw before though, was actually conservation of momentum. You converted this entirely to a second order differential equation in  $x$  and solved it. But in principle, we could have tried to find various solutions to this equation instead.

We would have said, well, we've observed masses and springs and we know they oscillate in time. So let's propose some oscillatory solutions and see what values of the amplitude and omega we need to satisfy this equation. It would have been a perfectly acceptable way of solving this problem.

The thing is, this mass spring problem, the differential algebraic equation contains nonlinearities with respect to the states. So it's nonlinear in the velocity and it's nonlinear in the position. Nonlinear equations in general are hard to solve.

When you converted that to a momentum balance you got linear equations. You got the second derivative of position was equal to stiffness over mass times position. It was a linear differential equation to solve, which is easy to solve. So you've got higher order differentials, but you linearize the equation so it made it easier to solve, in a certain sense.

But we'd like to sometimes understand what we can do with these equations. There are certain limiting cases where we can do interesting things with these equations. I'm going to show you one now.

So in the case where the Jacobian of this function  $f$  with respect to the differential of the state variable. So take the derivative of  $f$  with respect to  $\dot{x}$  and hold time in the state constant. So this is a matrix, one that's full rank. Then the DAE can be represented as an equivalent to ODE.

Here's how you can show that. The total differential of  $f$  is this Jacobian with respect to  $\dot{x}$  times the change  $\dot{x}$  plus the Jacobian with respect to  $x$  times the change in  $x$ . Plus the partial derivative of  $f$  with respect to time, times the change in time. And this total differential has to be equal to 0 because  $f$  is equal to 0 for all states in time.

So let's divide by  $dt$ . We'd like to know how  $f$  changes with time. So total differential of  $f$  with time. We divide each of these little differentials by time. And then we solve for  $\dot{x}$ .

That's going to involve moving this term to the other side of the equation and inverting this Jacobian. So if I can invert the Jacobian then I can convert my DAE system into a system of ODEs. This equals 0 shouldn't be here. I apologize. That's another typo.

I move this term to the other side of the equation to where the 0 is and I solve. So I went from a first order equation, a first order equation in  $\dot{x}$  to a second order equation in  $x$ . Two differentials of  $x$  with time instead of one differential of  $x$  with time. But I could convert it to an equivalent system of ordinary differential equations.

Does that make sense? Do you see how that works? Any questions? Let's go back and look at an example. The mass spring system.

So here's our ODE that we're trying to solve. I'm sorry, our DAE that we're trying to solve. Can you convert this to an equivalent ordinary differential equation using the approach I just described? Think you guys can figure that out? Try it. You can work together.

[SIDE CONVERSATIONS]

**WILLIAM GREEN,** Guys are either tired or friendships have been destroyed over the mid-term season. Maybe  
**JR.:** both.

So let's compute the things we need to know. We needed to know the partials of this  $f$  with respect to the states and with respect to time. So partial  $f$  respect to  $\dot{x}$ , holding all the other variables constant, is  $m\dot{x}$ .

Partial  $f$  with respect to  $x$  is  $kx$ . Partial  $f$  with respect to  $t$  is  $t$ . And I told you before that you could write this says  $\frac{dx}{dt}$  is equal to  $\frac{df}{dx} \frac{dx}{dt}$  inverse multiplied by-- is equal to minus  $\frac{df}{dx} \frac{dx}{dt}$  inverse multiplied by  $df/dx$ . This was on the previous slide, which just gives you conservation of momentum.

This gives you the acceleration is equal to the force minus  $kx$  divided by the mass  $m$ . So we can solve either, the DAE or the ODE.

Here's a more complicated example, but it works, well, it's more complicated. So a lot of times we do molecular dynamics simulations. We try to model molecules moving around. There we also want to conserve energy. There's usually a conserved quantity, something we're trying to hold constant. So suppose we try to do that.

The total energy in this system of molecules, maybe it's  $1/2$  mass times the velocity squared. Now  $x$  is the position of all these molecules.  $\dot{x}$  is the velocity of all these molecules plus the potential energy, which is a function of  $f$ , a function of their positions.

So the DAE representing this constraint is this. It's the same as for the mass spring system, where we've replaced the potential energy with a generic one. This actually isn't a well-posed DEA.

We have one equation for, let's see, if we have  $n$  molecules in here we have three  $n$  states.

We can move around in three dimensions. So this is a really ill-posed equation.

If we try to apply the same procedure, then we say, well, this quantity still should be conserved. Over time it needs to be equal to zero. Then we can try to compute all these differentials. But we'll find out that partial  $f$ , partial  $\dot{x}$ , the Jacobian of  $f$  with respect to  $\dot{x}$ , that's just the momentum.

That's not a matrix. That's not invertible. So we can't convert this equation to an equivalent system of ordinary differential equations. So it's just something pathological for a one-dimensional system.

Instead, what we find is zero is equal to the velocity dotted with the mass times the acceleration plus the gradient in the potential. So minus the forces acting on the molecules. We know that if we were to integrate the equations of motion for the molecules exactly, the momentum balances for those molecules exactly, then this term here would always be equal to zero. And we'd satisfy conservation of energy.

Of course, all solutions of ordinary differential equations require approximations. So as you move the molecules around, their velocities and positions will tend to drift away from the solution where this is exactly in balance and equal to zero. If you desire to do that in the molecular dynamic simulation then you better have a method that somehow respects this geometric property instead. That the velocities are orthogonal to  $m \dot{x}$  plus  $\text{grad } v$ .

And there is a set of ODE-IVP methods that do that. They're called symplectic integrators. And they integrate the equations of motion while exerting some control over the error in the total energy. So if you were just to take a ODE45 and try to solve this system of equations, over long times you would find that the energy drifts away from the place where it started.

Even though your positions are being integrated with reasonable accuracy, over time the energy will drift away from where you started. The system will heat up effectively or cool down. That's undesirable if you're trying to do modeling. So instead, one uses the symplectic integrators, which are designed to control the error and the energy in addition to integrate the equations of motion.

So you can look those up and read about them. I think they're quite interesting. If you're going to do molecular modeling you'd like to understand better how they work. But actually, conservation of energy here doesn't give us a well-posed DAE. We actually need the

momentum equations to formulate or to determine the trajectories of the molecules.

Let's talk about their numerical simulation. So let's think about these semi-explicit DAEs, the ones that can be split between differential and algebraic states. It's kind of useful to look at this problem in particular.

And you might say, well, let's do the simplest possible thing. Let's do the forward Euler approximation. So let's take our derivative here and represent it as the state evaluated at  $x$  plus  $\Delta t$  minus the state at  $t$  divided by  $\Delta t$ . Let's make the right-hand side of our equation be evaluated at time  $t$ .

And let's try to step in time, do time marching forward to determine the trajectory of  $x$  and  $t$ . How do the states change? We see the first equation is an easy one to solve because presumably I know  $x$  of  $t$  and  $y$  of  $t$ . Well, do I know  $y$  of  $t$ ?  $y$  of  $t$  has to satisfy this second equation.

So first, if I know  $x$  of  $t$  I need to solve this nonlinear equation determine  $y$  of  $t$ . Then I know  $x$  of  $t$  and  $y$  of  $t$  and I can step forward in time to determine  $x$  of  $t$  plus  $\Delta t$ . So every iteration here with the simplest possible approximation for the differential requires solution of a system of nonlinear equations.

We already saw before that a forward Euler approximation is not a great one to apply to solutions of IDE-IVPs because it has very limited stability properties. You need small time steps in order to stably integrate in time. So even though this seemed like it was easy to get a solution for the differential equation, we've always got to satisfy this nonlinear equation here.

So inherently, simulation of DAEs are implicit. They require solution of nonlinear equations in order to advance in time. There's actually no point in using this weakly stable method for any very small time steps. It makes more sense to choose a naturally implicit method for the differential equation, where I can take big time steps and still have reasonable stability of the integration. Does that make sense?

Consider now the fully implicit DAE. So  $f$  of  $x$ ,  $\dot{x}$ , and  $t$  is equal to 0 and have some set of initial conditions.  $x$  of 0 is equal to  $x_0$ . Here there is no way in general to avoid solving systems of nonlinear equations.

And so one substitutes often backwards difference approximations for  $\dot{x}$ . And then you

solve for the next point in time. So here's an example with the backward Euler, implicit Euler approximations. So I replaced the time derivative with the difference between the state at point  $t_k$  and the state of point  $t_k - 1$  divided by the difference between  $t_k$  and  $t_k - 1$ .

The other state variables are evaluated at  $t_k$ . So  $x$  at  $t_k$ . And I evaluate time at  $t_k$ . And I try to satisfy this equation and use it to determine  $x$  of  $t_k$ .

So I solve this system of nonlinear equations for  $x$  at  $t_k$ . And then I go to the next point in time,  $t_k + 1$ . And I solve the same equation, but replacing  $t_k$  with  $t_k + 1$  and  $t_k - 1$  with  $t_k$ .

So at each iteration I solve a system of nonlinear equations. No more painful than doing the forward Euler approximation I showed you before. So that's good. But still a lot of work. We might wonder how do we control the error in these sorts of approximations. And that's what we're going to talk about next.

So here's the equivalent time marching formula applied to a semi-explicit DAE. So I've got to satisfy the equation  $0$  is equal to the derivative of  $x$  with respect to time, which I approximate with the backward Euler approximation, minus  $f$  evaluated at  $t_k$ . And the equation  $g$  of  $x$  at  $t_k$ ,  $y$  at  $t_k$  for this  $x$  of  $t_k$ . And actually  $y$  of  $t_k$  as well. I'm sorry, I left that off.  $y$  of  $t_k$  as well.

Got to solve for the differential and the algebraic states simultaneously. And I use a backwards difference formula to try to do this in a stable way. One way to think about in particular these sorts of DAEs is as exceptionally stiff ordinary differential equations. Did you guys discuss stiffness?

So imagine if I replace the  $0$  with, say,  $dy/dt$ . And I introduced a constant on the right-hand side, a multiplicative constant, say,  $1/\lambda$  here. Let's say  $\lambda$ , actually. Say I introduce a multiplicative constant  $\lambda$  on the right-hand side here.

So as  $\lambda$  becomes very large, the system becomes increasingly stiff. The dynamics of this equation take place over very, very short time scales. So short in fact, that eventually, if I let  $\lambda$  diverge to infinity, I've just got to satisfy this algebraic equation at each point in time. So these are exceptionally stiff equations. And the methods for solving these equations share a lot in common with stiff ODE-IVP solvers.

So I've got a couple of minutes left. I'm going to work through some examples. So consider the stirred tank example one.  $dc_2/dt$ . It's related to  $q/v$ , flow rate over volume. The difference in the concentrations coming in and out of the tank. And  $c_1$  is equal to this input signal

gamma. Maybe that's a measurement and I'm trying to predict  $c_2$ .

So can you apply the backward Euler method to these equations to determine  $c_1$  and  $c_2$ ?  
What would one step of the backward Euler method look like, say, going from time 0 to time  $\Delta t$ , or time  $t$  to  $t + \Delta t$ , or  $t_k$  to  $t_{k+1}$ , however you want to write it?

So do you get something like this? So this equation is easy.  $c_1$  to  $t_k$  is gamma of  $t_k$ . I got a substitute up here.  $c_2$  of  $t_k$  minus  $c_2$  of  $t_{k-1}$  over  $\Delta t$ .  $T$  Just  $t_k$  minus  $t_{k-1}$ . And then solve for  $c_2$  of  $t_k$ . And you get a formula like this.

This formula was based on an approximation for the derivative, which had a leading order error that was proportional to  $t_k - t_{k-1}$ . When I go through and I solve for  $c_2$  of  $t_k$ , the leading order error in  $c_2$ , the local truncation error is going to be order  $t_k - t_{k-1}$  squared.

So as I shrink the spacing between each of my two time points, the local accuracy, the error induced in one advance of this time stepping algorithm is going to scale like the step size squared. So I have the error-- I have the spacing. The error will go down by a factor of four.

I reduce the spacing by an order of magnitude. The error will go down by two orders of magnitude. This is a really desirable sort of error control and a method. And we can do better. You've seen us do better with other methods.

Let's look at this equation now. This is stirred tank example two. I just replace  $c_1$  with  $c_2$ . You don't have to write this one down. I'll just show it to you. The notes are online so you're not missing anything. All these things will pop up in the notes online.

So  $c_2$  at  $t_k$  is equal to gamma of  $t_k$ . I got to come up to my first equation now and I have to substitute an approximation for the derivative. It's this, the backwards difference formula.

So if find  $c_1$  of  $t_k$  is  $c_2$  of  $t_k$  plus  $v$  over  $q$ .  $c_2$  of  $t_k$  minus  $c_2$  of  $t_{k-1}$  divided by the time step. Remember, this approximation for the derivative has a leading order error that goes like  $t_k - t_{k-1}$ .

So my approximation for  $c_1$ , the local approximation for  $c_1$ , is not second order accurate. It's only first order accurate. I applied the same approximation method to both equations. One equation I got something that was second order accurate. The next equation I got something that was first order accurate.

They look almost identical. It might be hard to see from the start why it should work out that way. That's how it works out.

Let's do one more example. Here's a system of DAEs.  $\dot{c}_2$  is equal to  $c_1$ .  $\dot{c}_3$  is equal to  $c_2$ . And  $0$  is equal to  $c_3$  minus  $\gamma$ . So  $\gamma$ , again, is some measurement.

We solve this equation to determine  $c_3$ . We substitute  $c_3$  up here to determine  $c_2$ . We substitute  $c_2$  up here to determine  $c_1$ . But we want to find the solution with the backward Euler method.

And the solution there is going to be  $c_3$  of  $t_k$  is  $\gamma$  of  $t_k$ .  $c_2$  is related to the derivative of  $c_3$ . So I use my backwards difference formula for the derivative of  $c_3$ .  $c_1$  is related to the derivative of  $c_2$ . So I use my backward difference formula for  $c_2$ . But how accurate are each of these approximations?

So here I have to determine  $c_2$  from an approximation for the derivative of  $c_3$ . We know this backwards difference approximation has a leading order error proportional to  $\Delta t$ , the time step. Now I got to approximate  $c_1$  with the finite difference backwards difference approximation in  $c_2$ .

We know this should incur an error proportional to the time step  $\Delta t$ , except I don't know  $c_2$  at  $t_k$  exactly. Actually,  $c_2$  at  $t_k$ , the exact value of  $c_2$  at  $t_k$  is this plus something proportional to the time step. Which means the exact value of  $c_1$  is this plus something that's order one because I carry over the error from my solution to the previous equation. And I divide it by the time step.

So I went from having something-- well, let's see. The first equation had order  $\Delta t$  squared local truncation error. As I shrink the spacing, my solution gets more and more locally accurate. And it seems to do it in a fairly robust fashion.

The next system of equations we solved had a local truncation error proportional to  $\Delta t$ . Again, I can shrink  $\Delta t$  and I can make my solution more and more accurate. That seems like what you want in a numerical algorithm.

This one, there's no hope. Change  $\Delta t$ , who knows what you're going to get? And this looks fairly innocuous. You just look at it and you say, well, OK. Let's solve it and see what happens.

Well actually, you can't solve this problem accurately with the backward Euler method. There's something fundamentally different about this problem from the previous one. There's something fundamentally different about stirred tank example two from stirred tank example one. Right you might have already perceived that.

You can see what's going on here.  $c_3$  is equal to  $\gamma$ . That's the exact solution.  $c_2$  is equal to  $\dot{\gamma}$ . And  $c_1$  is equal to two derivatives of  $\gamma$ , the exact solution to this equation.

So  $c_1$  is incredibly sensitive to changes in  $\gamma$ . And that's peculiar. And it has important numerical consequences.

So I'm going to show you, our next lecture on Wednesday, how to formalize an understanding of the differences between these problems. And we'll talk about some more subtleties associated with differential algebraic equations.

You guys can pick up your quizzes. We should do it outside rather than inside because the next class has to start.