All right, OK, let's start. 1690, Numerical Matters. Local accuracy and global accuracy, again, this is an distinction between local accuracy and the global accuracy. What is the distinction? What is that we ultimately want in a numerical method?

Do we ultimately want global accuracy or local accuracy? We ultimately want global accuracy, right? Local accuracy, actually, I mean if a scheme only has local accuracy, it's useless. It is useful only [? if it has ?] global accuracy. Global accuracy means the solution using numerical method actually matches the true solution as my delta t goes to 0. So global accuracy is what we want.

But how can we analyze global accuracy? We can only look at global accuracy when we of apply the scheme to solve a differential equation. Right? Is there a way to show global accuracy without actually coding it up and solve a particular differential equation? Is there a way to show global accuracy without applying this definition?

AUDIENCE: [INAUDIBLE]

PROFESSOR: OK, so global accuracy means V at a particular time converges to u at t. So Vt is the numerical solution. ut is some true solution I may or may not know, depending on what equation it is. So this is global accuracy.

I mean, of course I can check global accuracy by applying my scheme to a particular differential equation. But is there a way I can know if a scheme is globally accurate or not without [INAUDIBLE] apply? Yes?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yes. That's the answer I'm looking for. If the scheme is locally accurate, which is also consistent, and it is 0 stable, then I know whatever differential equation I applied it to, I'm going to get global accuracy. This is a good thing because it is independent of what differential equation I look at.

.

If I can show local accuracy, which doesn't require me to implement anything, I just do what? How do I show local accuracy?

AUDIENCE: Parallel theory?

PROFESSOR: Parallel theory is right. You can show local accuracy by applying parallel theory as analysis. And can you also show zero stability without implementing anything?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Exactly. You can plug in the du dt equal to 0. So 0 stability is if a scheme can solve this. Usually, whatever scheme you have, you're plugging this previous differential equation into the scheme. You can analyze this behavior analytically. You can plug in an exponentially growing solution to see if that solution is possible.

If an exponentially growing solution is possible, then the scheme is not zero stable. If an exponentially growing solution is possible, then it's not zero stable. If an exponentially growing solution is not possible, then the scheme is zero stable.

Now you can show local order of accuracy and zero stability all without implementing the scheme on any differential equation. And once you have these tools through the [INAUDIBLE] equivalence theorem, you know the scheme is globally accurate when it is applied to any differential equation. Is it clear, the relationship between these three?

Global accuracy is what we want. This is what we want. What we want. And the zero stability and local accuracy is easy to analyze. And once we have the two easy to analyze things, we check these, then we automatically know global accuracy. So the relationship between these three are this simple. So when we have these two, that leads to global accuracy.

So we discussed zero stability. But zero stability is a pretty weak criterion. So zero stability means it can solve du dt. Yes, if a scheme is locally accurate and zero stable, we are going to have global accuracy. But sometimes, global accuracy may not be enough.

I'm going to show an example of why global accuracy is not enough. So global accuracy only says that [INAUDIBLE] equal to zero. [INAUDIBLE] solution [INAUDIBLE] analytical [INAUDIBLE], right? It doesn't say anything on if my [INAUDIBLE] is finite, how big the error is. It also doesn't say how [INAUDIBLE] my solution is.

And it turns out we can get [INAUDIBLE] solutions for finite delta t even if a scheme is local order of accuracy and zero stable. I'm going to show you such an example.

The example is using a second order accurate scheme and also zero stable. Can you guys think of a zero stable scheme that is second order accurate? What scheme have you learned that is second [? order ?] [? accurate? ?]

AUDIENCE: Midpoint.

PROFESSOR: Midpoint. Exactly. Midpoint is locally accurate. It's zero stable. And in something we implemented last lecture, right? So let me go back to my Dropbox. [INAUDIBLE]. Wait, where is my shared folder? [INAUDIBLE] up there, but, like-- OK, it's over here. OK.

I would ask you, who has completed these implementation we started last class? Of course [INAUDIBLE] here I have finished. Who else did it? Can you raise your--

AUDIENCE: [INAUDIBLE]

PROFESSOR: No, not including [INAUDIBLE]. Great, great. So we have a bunch of people who did this. Anybody willing to try their code here? No?

AUDIENCE: [INAUDIBLE]

PROFESSOR: [INAUDIBLE]

AUDIENCE: The first two.

PROFESSOR: The first two.. All right. So I'm going to copy your code into today's folder. And which one should I run?

AUDIENCE: [INAUDIBLE]

PROFESSOR: [INAUDIBLE], OK. Thank you. Right. So I see you did [INAUDIBLE] and midpoint. OK, you are also doing this. So I'm just going to use-- I see you did this double percentage thing. Does everybody know how to use that?

Yeah, so basically, [INAUDIBLE] according to sections. So I think it's Control-Enter, right? To just around these sections? I think it's already done. And I can run midpoint. [INAUDIBLE]. That's nice.

What if I change the delta t to 0.5? OK, what if I change delta t to 0.5, and I'm going to close all in the beginning of this so that, like, I'm going to close my [INAUDIBLE]. I'm going to run it again. I get this.

And this doesn't actually defeat the purpose of the analysis of the scheme being globally accurate. The scheme is still globally accurate because, as I decrease my delta t, as I make delta t equal to 0.3, for example, the oscillation gets smaller. My solution gets better.

If I decrease to 0.2, the solution gets still better. [INAUDIBLE] it gets better and better. And when I decrease it to 0.1, it's almost invisible. But I'm going to assure you.

If I let these things drop further, the oscillation is going to show up. You want to try? To try, OK? p is equal to 100, let's see. If p is equal to 100, oh, you did the [INAUDIBLE] thing. I'm going to remove the [INAUDIBLE] him and do axis equal. That's too much.

[LAUGHTER]

p is equal to 50. What? OK. Did I change anything? Why do I get this? Do I need to do this again? Or not? Why do I get a--

AUDIENCE: [INAUDIBLE]

PROFESSOR: Variables. The first one, OK? I ran it, and then we're on the next point. Oh, there it is. So let me go back to my p equal to 50. You see what I get over here? I don't get a solution that is smooth. I get a solution that-- [INAUDIBLE].

So what happens-- still, it doesn't defeat the purpose of being globally accurate. If I make delta t equal to 0.05, this oscillation becomes smaller. We can still see it, but it becomes smaller. But when I increase t to be 100 again, this is going to show up again. I mean, I see something is wrong. OK. So I'll do that later.

But the point is clear, right, although the skin is zero stable, it is second order accurate, the midpoint scheme. But it doesn't really give the desired behavior. It's even worse than the forward Euler which is only first order accurate.

If you look at the forward Euler, with a delta t of 0.5, which shows horrible behavior for midpoints, right? I get an error somewhere. Oh, I need to run this first, and run this. So yeah, forward Euler works fine with a delta t equal to 0.5. So in this case, forward Euler actually

works better than midpoint, even though midpoint is a second order accurate scheme.

So why is this the case? Both schemes are zero stable. Midpoint is more accurate, supposedly more accurate than forward Euler. But applied to this specific equation, unlike the equation we saw last week, the pendulum equation-- on the pendulum equation, midpoint actually works much, much better than forward Euler for a large [INAUDIBLE].

Here, a forward Euler works better than midpoint or [INAUDIBLE]. The reason is linked with eigenvalue stability. It turns out that the particular case, we are looking at, for which forward Euler is actually eigenvalue stable, and the midpoint is not eigenvalue stable.

So eigenvalue stability is a requirement that is more strict, more stringent, than 0 stability. Although midpoint is zero stable , it is not eigenvalue stable for that particular equation. OK, any questions before we move on?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Exactly. Good question. So zero stability is per scheme. A scheme is either zero stable or not. It's independent of the differential equation, because it just tells you, can scheme solve this previous differential equation or not.

Eigenvalue stability, on the other hand, depends not only on the scheme, but also on the equation. In the simplest case, eigenvalue stability looks at, are you able to solve this equation? So you can see zero stability is like a standard case of eigenvalue stability, where lambda is equal to zero.

So if a scheme is eigenvalue stable for lambda equal to zero, then it is zero stable. But a scheme being zero stable doesn't mean it is always eigenvalue stable for all differential equations like that. OK, so let's start discussing eigenvalue stability. And at the end of the lecture, I'm also going to start talking about implicit schemes and the Newton Raphson iteration. All right.

So eigenvalue stability analysis, the first thing I want to show you is this link. If you go to this link, which I already opened, I think, you're going to get to the website called mathlets.org. And it hosts a bunch of what they call Mathlets. They are basically Java applets that allows you to visualize these various things.

And if you click on it, it is going to open a window. You need a statue like Java security or something like add exception to mathlets.org. Otherwise, your browser may actually block the thing. But once you have it-- let me make it full screen. OK. Right.

So what it shows you is for what lambda-- remember, we are talking about eigenvalue stability-- is a scheme stable or not for this equation. So we have a lambda here. In the mathlet, a lambda can only be real values, but complex values.

So this is the space of our lambda times delta t. We can think of a point here is 1, 0. That means lambda times delta t equals to 1. So, for example, at this point, this lambda is equal to 10. Delta t is equal to 0.1. Then you're at this point. If lambda is equal to minus 1, [INAUDIBLE].

The blue regions here shows you that for what values of lambda, for what combinations of lambda and delta t-- actually, it only includes the lambda. What lambda times delta t is, for a particular scheme, eigenvalue stable. So here is the forward Euler.

For forward Euler, the scheme is stable only if lambda [INAUDIBLE] times delta t was using this very small circle. OK, what does this mean? This means that if you have a lambda greater than zero, would forward Euler ever be stable?

No. Actually, I would expect, if lambda is greater than zero, even the analytic solution of the PDE is unstable. Why? What is is the analytic solution for TDE?

AUDIENCE: [INAUDIBLE]

PROFESSOR: e to the lambda t, right? Each of the lambda t is the analytic solution of [INAUDIBLE] unstable by itself. So actually, not the only real one, but also the whole [INAUDIBLE] the analytical OD is unstable.

Because if the lambda is here, then the lambda is 1 plus i, how does the solution look like? It's an oscillatory growing solution, right? The imaginary part determines the frequency of the oscillation. But real part determines the rate of growth. As long as the real part is greater than zero, it's a growing oscillatory solution.

So that means, for the whole [INAUDIBLE] of the complex [INAUDIBLE], the analytical solution [INAUDIBLE] is unstable. So you shouldn't really expect a numerical solution to make it stable. I mean, you're going to see some, the numerical scheme actually does make some of them

stable, but you shouldn't expect it.

What is more intriguing is that, even on the [INAUDIBLE] for which the analytical behavior of the ODE is either just a monotonic detail [INAUDIBLE] on the real axis into something negative times t. So is du dt equal to negative lambda times u. The solution is u equal to u to the negative [INAUDIBLE] times t. So you get a monotonic case.

If you are over here, we'll have a negative real time, and a non-zero imaginary [INAUDIBLE], then you have an oscillating and decaying solution. So you would also expect it to be stable. But forward Euler, actually, is only stable within this region [INAUDIBLE] lamba delta [INAUDIBLE] over here, it's unstable.

OK, so that is interesting. I'm just going to do a very brief analysis of showing why that is the case. Forward Euler, V of n plus 1 minus V n over delta t applied to half of V n, which, in this case, is lambda times V n.

So let's move all the knowns to the right hand side. So this term is known, and this term is known. So move this to the right hand side. We have V n plus 1 over delta t is equal to V n over the delta t plus lambda times v n.

And the [INAUDIBLE] is, if we [INAUDIBLE], V n plus 1 would be equal to 1 plus delta t lambda times v n. We can easily see if lambda is greater than 0, we have a growing solution. If lambda is greater than zero, this term is greater than 1, we grow.

What if lambda delta t is less than minus 2? We get an oscillatory growing solution because 1 plus delta t lambda would be less than minus 1. And the next iteration I'm going to get a solution of the opposite sign, but larger than what it is at the n times f. That is why forward Euler is unstable when we have a lambda delta t less than minus 2. Same thing happens when lambda is imaginary and 1 plus delta t lambda has a modulus of greater than 1, that means we are [INAUDIBLE].

So that's how you do eigenvalue stability analysis. You plug in f of u for the lambda u into the differential equation. You collect all the terms that is [INAUDIBLE].

And in general, you plug in the solution that is exponentially growing to see if the script scheme allows an exponentially growing solution. That is how you would analyze eigenvalue stability, similar to how you would analyze zero stability. Any questions?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, the absolute value of that one has to be less than or equal to 1. Yes. Yep?

AUDIENCE: So, when you said that the [INAUDIBLE] has to be less than minus 2, [INAUDIBLE]?

PROFESSOR: [INAUDIBLE]

AUDIENCE: OK. Yeah. [INAUDIBLE]

PROFESSOR: OK, good question. When you are implementing forward Euler on the real thing, how do you know [INAUDIBLE]? It's a good question.

So most of the time, you can analyze the eigenvalues-- let me back up. Let me say, if you look at a scalar differential equation, if you look at, like, just one single differential equation, not a [? coupled ?] system of differential equations, even if the equation is non-linear, you can usually linearize the differential equation and look at what lambda is for linearized differential equation. And the key is not to manipulate lambda, but ensure that delta t is small enough.

So let's look at here. Imagine [INAUDIBLE] lambda delta t. But imagine your lambda is taken over here. If your lambda is, let's say, minus [INAUDIBLE]. Can forward Euler be stable? No?

AUDIENCE: [INAUDIBLE]

PROFESSOR: If my lambda is here, can forward Euler be stable?

AUDIENCE: [INAUDIBLE]

PROFESSOR: We still have delta t, right? If lambda is here, what is to make delta t with a 0.1? Then lambda has delta t which [INAUDIBLE] way over to here, into the stability region. And we are fine, right? OK, what if lambda is minus 500 plus 1,000 times i.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, even if lambda is minus 500 [INAUDIBLE] to make the delta t equal to 0.0001, you're going to be fine. You're going to shrink your eigenvalues times delta t into here. And you're fine.

OK, what pieces are not fine? What are the cases where, no matter how small you make the delta t, you're going to be unstable?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right, if lambda has a positive real cost, if lambda is always here, and no matter how small you shrink it, you get a small delta t-- because delta t has to be real, right? The constant has to be real. You're going to shrink it to over here, but it's still unstable, right? OK, what are the cases where forward Euler is not working?

AUDIENCE: [INAUDIBLE]

PROFESSOR: If it's just an oscillation, if the eigenvalue lambda is just a dual imaginary value, it's a non-decaying oscillation. So if the analytical solution is a non-decaying oscillation, just like what we had on [INAUDIBLE], like the pendulum, if we have a pendulum that has no density, forward Euler would never be eigenvalue stable.

And that's what we saw, right? I mean, even when we made delta t to be small, it's still mildly growing. So yes, forward Euler will work if you could have a positive real eigenvalue, or you could have a zero real eigenvalue. Otherwise, you can always shrink your delta t so that the scheme is stable.

Let's look at another scheme, midpoint. OK, what happens on midpoint? What is the stability region of midpoint? It lies exactly on the imaginary axis from [INAUDIBLE]. So the case we have to analyzed saying two oscillations would never work for forward Euler, would never be stable for forward Euler, it works perfectly fine for midpoint.

That's why, when we did the pendulum case, midpoint worked so well. It was, like, amazingly better than forward Euler. But when we applied to this case, forward Euler works better because-- try to guess where the eigenvalues are for the case we had on [INAUDIBLE].

Anybody wants to take a guess of what lambda is for that problem? Something with a very small imaginary component?

AUDIENCE: [INAUDIBLE]

PROFESSOR: [INAUDIBLE] means you have negative real part, right?

AUDIENCE: Yeah. [INAUDIBLE]

PROFESSOR: If you look midpoint, as long as the eigenvalue lambda has a non-zero [INAUDIBLE], then it wouldn't be stable, no matter how small you make delta t-- I mean, if the eigenvalue is here,

over here, no matter how small you make delta t [INAUDIBLE] was the origin. But no matter how much you shrink it, you will never be exactly on the imaginary axis between [INAUDIBLE], right?

So as long as the solution is not pure oscillation, midpoint wouldn't work. So forward Euler and midpoint are like two schemes that have complimentary values, but they never overlap. A question would either work for forward Euler, or it'll work for midpoint with a small enough time stamp.

If the lambda is exactly on the imaginary axis, with a small enough delta t, you can make midpoint work. If it has a negative real part with a small enough [? timestamp, ?] you're going to make lambda times delta t into this region.

What we're not concerned about is where lambda has a positive real part because the differential equation is unstable, and we don't want to solve it anyway. I mean, we don't expect it to be stable. Yeah, right. So right.

So we don't really expect our scheme to be stable. If it's stable, then it is artificially stable because the real solution here is going to be unstable. Backward Euler, [INAUDIBLE] OK, we can see that, for any differential equation that is actually stable, backward Euler is stable.

The blue region is [INAUDIBLE] no matter how big delta t is-- you can make delta t equal to 1,000 if you like, backward Euler is [INAUDIBLE] stable. So even in this region, where the real differential equation is unstable, artificially makes the solution stable, which actually gives you a wrong answer, but [INAUDIBLE] stable.

[LAUGHTER]

All right. Trapezoidal rule. That is, like, yeah, you said that's what we want because, if you look at the stability ratio of trapezoidal rule, it is exactly the same stability region as the analytical ODE. It doesn't always mean you're going to get the expected behavior because you're going to see is my lambda delta t is always here, the rate of decay of the trapezoidal rule versus the real ODE is going to be very different. So like, at least in terms of stability reading, it is exactly what you have for the analytical ODE.

And the wrong [INAUDIBLE] schemes is something you are going to learn later on for this week's reading. But you're going to see these weird shapes. The [INAUDIBLE] schemes are

explicit schemes. So for non-linear differential equations, you're going to see very soon that explicit schemes are a lot easier to implement than implicit schemes.

And, for example, is great because it combines the advantage of forward Euler and midpoint. Why? Because as long as my lambda has a non-positive real part, as long as I make delta t small enough, I'm going to get into the stability region.

Forward Euler requires me to have a negative real part. Midpoint requires me to have a zero real part. [INAUDIBLE] requires me to have either 0 or negative real part. And I can make delta t small enough. It is going to work. All right. Questions? Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Eigenvalue stability concerns about an equation that is dx dt equal to lambda x. We are looking at, if I apply the scheme, we [? sub ?] delta t to this equation. Am I going to get a growing solution, or I'm going to get a stable solution?

AUDIENCE: Is there a way [INAUDIBLE]?

PROFESSOR: The stability region is different for each [INAUDIBLE].

AUDIENCE: OK.

PROFESSOR: So you have to-- yeah, it's separate from the [INAUDIBLE] solution. For each combination of scheme and lambda delta t, you have an answer of yes or no.

AUDIENCE: [INAUDIBLE] eigenvalue. [INAUDIBLE] eigenvalue. [INAUDIBLE] eigenvalue. [INAUDIBLE]

PROFESSOR: Yes.

AUDIENCE: [INAUDIBLE] and then you linearize it. But because of your error [INAUDIBLE]. This sort of method [INAUDIBLE].

PROFESSOR: I don't have a definite answer to that, but it's possible. And what I want to emphasize is that, if you linearize the equation, and eigenvalue stability shows that it is unstable, then it's very likely, when you apply to your [? normal ?] equation, that it's going to be unstable.

So eigenvalue stability, I would say before you apply the scheme to differential equation, it's a good idea to check eigenvalue stability. And this also gives you a guidance of when you see

something being unstable. Is it because you are using the wrong scheme? Or is it because you have a too large [INAUDIBLE]. Right? You can take a look at these eigenvalues and figure out which case it is. Any other questions?

OK, let me give you a challenge. Now, if I get a couple differential equations [INAUDIBLE] where y is equal to x, where x can range all the way from 0 to t. For what x [INAUDIBLE] delta t is forward Euler table. For what values of [INAUDIBLE] is backward Euler table. And for what value of epsilon is it midpoint stable?

I mean, not for epsilon, but for what combination of epsilon and delta t is each of these [INAUDIBLE]. Form groups of two or three, and figure this out. All right.

A lot of you have got the answer, or almost get the answer, and let me do it here to make sure everybody is on the same page. First of all, the first step is to write the differential equation into a matrix form. The du dt of x and y, let's write it in a similar way as du dt equals lambda u. But here, the lambda is no longer a number. It is a matrix.

And what is the matrix? Just fill in the blanks. dx dt minus y minus epsilon x. So minus epsilon x minus 1y. dy dt equal to x, so 1x plus 0y.

OK, so I get a matrix here. And how do I go from the matrix to eigenvalue stability? Take the eigenvalue. Take the eigenvalue. So the eigenvalue of this matrix, and I may be lazy, so let me do it in Matlab.

[LAUGHTER]

OK? Yeah, so for this 2 by 2 matrix, you can do it by hand. But if you get a large matrix, what do you do? Let me do the bad example.

AUDIENCE: [INAUDIBLE]

PROFESSOR: OK, I'm just going to make epsilon go all the way from 0 to 10. So linked space is putting a bunch of epsilon. And the matrix here is going to be like this. So let me do this. Let me put e to be a symbolic variable. And the matrix is what? Minus e minus 1, 1, 0. So that's what the matrix is. And e means epsilon.

And you can do the eigenvalue of this matrix. It gives you the formula, which you guys have figured out. Minus epsilon over 2 plus or minus the square root of-- I mean, to the 1/2 power is

basically square root, right, of this thing.

So let me just again say, epsilon e is equal to linked space of 0, 10, 100. And I'm going to say lambda 1, the first eigenvalue is-- I'm just going to copy the formula here and paste it. And I need to convert the products into dot and exponenting to dot exponential.

And everything else is going to be fine. I'm going to also look at the second eigenvalue. I'm going to paste it and also do the dot product and dot explanation. OK, I have L1 and L2. Let me plot them.

I'm going to plot a line and dot. So what you get is, in the complex plan, let me do x is equal-- in a complex plan, I'm plotting the trajectory of lambda, of lambda 1, as my epsilon goes from 0 to 10. I'm going to say hold on and plot the evolution of lambda 2.

Actually, I want to plot it in a different color. So you get a [INAUDIBLE] of lambda 1, but this way, [INAUDIBLE]. And remember, this is, in the context [INAUDIBLE], the two eigenvalues, right? OK, anybody remind me what is the stability region of forward Euler?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, so a circle around negative 1. So a circle around negative 1, right? Going to the Matlab plot, it's a circle around here. But remember, the stability region is a stability region of what? Of lambda t. Lambda times t. Well here, I'm just plotting the lambda. So what does that mean?

                   If I have, let's see, epsilon equal to 10, one of my eigenvalues is here. Is there any hope? I mean, the [INAUDIBLE]. Is there any hope of making forward Euler stable?

AUDIENCE: [INAUDIBLE]

PROFESSOR: How?

AUDIENCE: [INAUDIBLE]

PROFESSOR: By a big delta t equal to how much? Or a small delta t? A small delta t equal to how much? Huh?

AUDIENCE: [INAUDIBLE]

PROFESSOR: [INAUDIBLE]

AUDIENCE: [INAUDIBLE]

PROFESSOR: As long as I can make this point with being the [INAUDIBLE], where the [INAUDIBLE] here is minus 2, right? So as long as I can have a delta t that is less than 1, 2, I can make forward Euler stable. Any questions on that?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, right, sure.

AUDIENCE: [INAUDIBLE]

PROFESSOR: OK, you have beat me.

AUDIENCE: [INAUDIBLE] that lambda [INAUDIBLE].

PROFESSOR: Yeah, right. OK, thank you.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right. So we have to make [INAUDIBLE] stability region. So, question? OK. And so we can see, when epsilon is very large, it's difficult to-- you have to have a small delta t to shrink it [INAUDIBLE]. Another case is when epsilon is very small, actually adds epsilon equal to zero.

I get two eigenvalues equal to [INAUDIBLE]. How can I make the case stable for forward Euler?

AUDIENCE: [INAUDIBLE]

PROFESSOR: I can never make forward Euler stable. I can make delta t goes to very, very small, [INAUDIBLE] shrink over here. But they are still on the [INAUDIBLE]. And remember forward Euler, it doesn't include any point on the imaginary axis other than 0.

So when 1 epsilon is very small, it's impossible to make forward Euler stable. And when epsilon is very small, it takes a very, very small delta t [INAUDIBLE]. So when a system is [INAUDIBLE], it's almost pure oscillation. It also takes a lot of [INAUDIBLE] for very small delta t for forward Euler to accurately resolve its behavior. Yes?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Good question. Does both Eigenvalues need to be in the stability region?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yes. Both eigenvalues, or all of the eigenvalues, in this case, with two eigenvalues, [INAUDIBLE] all of the eigenvalues have to be in the [INAUDIBLE] region. And the eigenvalues of a matrix doesn't always conjugate each other. And in this case, [INAUDIBLE] conjugate each other. And as our epsilon [INAUDIBLE]. And if you have a bigger matrix, that's even more [INAUDIBLE].

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right. The instability analysis of the analytical system, the analytical system is stable for the whole [INAUDIBLE]. So you are looking at the most dangerous [INAUDIBLE].

Where we have a numerical scheme, the most dangerous ones are not necessarily the one that lies towards the right. It made be the ones that lies more towards the left. Or most [? low, ?] most dangerous, of the stability region. They are not necessarily the ones that is closest to the imaginary axis.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, for nonlinear problems, sometimes [? they are ?] very extremely important to adapt the timestamp according to-- because they have different [? phase, ?] and different parts of the solution, when you linearize the nonlinear equations, you get different eigenvalues.

So when you get the big eigenvalues, or [INAUDIBLE], it's important to make [? them ?] small. Any other questions? OK. Let me pose another change question. Oh, right. OK. Yes, what about midpoint?

If you look at the eigenvalues, will they work for midpoint? Only when epsilon is equal to 0. Only when epsilon is equal to 0, we get equal eigenvalues. And for what delta t would midpoint be stable? Less or equal to 1, right? Because the stability region for midpoint, when looking at lambda times delta t, that [INAUDIBLE] midpoint to midpoint.

If you make delta t lambda equal to 1, then the [INAUDIBLE] pinpoints was [INAUDIBLE] into the [INAUDIBLE]. If you make delta t greater than 1, you will push this point in the lambda delta t [INAUDIBLE] out of the [INAUDIBLE].

You'll find a scheme where it will be stable for all of the actions-- trapezoidal, right? Because the trapezoidal rule has a stability region that covers the whole left panel. And we are looking at the whole left panel over here. What else?

The [INAUDIBLE] is called a [INAUDIBLE]. OK, that has a stability region somewhere like [INAUDIBLE] or something like that. We can have a small delta t that would bring the whole thing into the [INAUDIBLE].

OK, yes?

AUDIENCE: [INAUDIBLE].

PROFESSOR: How do you find a stability region for various methods? Good point. I have showed you how to find the stability region for forward Euler. I have showed you how to find the stability region for forward Euler by basically plugging du dt for the lambda u into the scheme.

I'm going to show you another example of doing this for midpoint, all right? OK, if you have a midpoint, you have a Vn plus 1 equal to Vn minus 1 plus 2 delta t times half of Vn. I can do this example, but you've got to help me. What's next? When I analyze eigenvalue stability, what do I do next? What is eigenvalue stability?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yes. Eigenvalue basically shows is the skill [INAUDIBLE] for [INAUDIBLE] equal to lambda u. So basically, we are asking, is the scheme stable for this differential equation.

So when we look at these as f of u, we get lambda times Vn. OK, so I'm going to write it down again. This is what we want to analyze [INAUDIBLE] exponentially growing solution or not. How do I analyze if it [INAUDIBLE] the exponentially growing solution or not?

I'm going to try one. I'm going to try to say Vm plus 1 is equal to V0 times Z to the n plus 1. Vn equal to V0 times [INAUDIBLE] minus 1 equal to V0 times Z to the n minus 1. I'm just going to try 1, try an exponential growing solution to see if it is possible for Z to be something greater than 1 or less than minus 1, or having an imaginary having a complex value that has a

modulus greater than 1. But if it's one of these cases, I'm going to get an unstable scheme.

OK, let me plug this in, and I'm going to cancel all of these zeroes. What I'm going to get is Z to the n plus 1 power. Power is equal to z to the n minus 1 power. I mean, these are time steps. These are time steps.

Now what I get is to the powers. I'm going to do parenthesis to denote time steps and the ones without parentheses to denote power. Oh, this is a plus, sorry. Plus 2 delta t times lambda Z to the n. OK, what do I do next?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Factor out Z to the n minus 1. I'm going to get a quadratic equation equal to 1 plus 2 delta t lambda times z. OK, now I can't get the solution for z to see if it can lead to an exponentially growing solution or not. So I believe I get 2 lambda delta t plus minus square root of 4 lambda squared delta t squared plus 4.

So let me go [INAUDIBLE]. OK, so this is the Z I got. I want this Z to have a modulus less or equal to 1. [INAUDIBLE] And this 2, divided by 2, if lambda is-- so let's proceed. If lambda is a real value, if lambda is anything real, and nonzero, then lambda squared is going to be something positive.

And what I get is something greater than 4. And the square root is going to be something greater than 2. So because I have a plus or minus, one of them is going to have a magnitude greater than 2. When divided by 2, I'm going to get something greater than 1.

So for any real lambda, I'm going to be on unstable. And you can also figure out, for complex values of lambda, what values is going to make it stable, what value is not going to make it stable. And in fact, [INAUDIBLE] for all the values of lambda delta t, that will make this Z having a modulus of exactly 31.

So if this has a value exactly 31, my [? schema ?] is going to be exactly on the boundary of stability. So if I have all this level 1, I'm going to be stable. If I have 1 z that is greater than 1, if I have [INAUDIBLE] modulus, [INAUDIBLE], I'm on the boundary.

So by [INAUDIBLE], I'm going to compute the boundary of the stability region, which gives me something like this. So if I take the boundary [INAUDIBLE] modulus 1, I'm going to block its boundary of stability. [INAUDIBLE] Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, Professor Wilcox is saying that, when you look at this equation, and plug in Z is equal to e to the ith theta, you're going to get Z squared, which is e to the 2i theta is equal to 1 plus 2 times delta t lambda times z, which is e to the ith theta.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right. And then--

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right, and you can figure out the stability boundary for delta t lambda using [INAUDIBLE]. All right. OK.

Let me show you one last thing. What if you have a du dt equal to minus u squared? We get a non-linear equation. For what scheme would this be stable? What scheme this would be unstable?

The analysis of this scheme lies in linearization. OK, so let's say, if I have a u of 0 at t equal to 0 equal to 1. OK. What if I linearize this question? What if I say, u, I'm going to put [INAUDIBLE] a u to u plus delta u, or plus a very small v.

What is the result of the linearization? The linearization can be seen by plugging both the u and the u plus V into the differential equation. If you plug just the u, you get this. If you plus u plus v, you get du plus V dt equal to minus u plus V squared. And here is linearization V is very small.

And because v is very small, you can derive du dt plus dv dt. This is just factorizing u plus v out of the derivative, is equal to minus u squared minus 2 uV, minus V squared. But because V is very small, this term is negligible. This term is very, very small, or very square small. OK.

Now, when you look at the answer to the equation has a du dt. The right hand side has a minus u squared. The right hand side has a minus u squared. These two terms cancel out because of the answer to the equation. The only thing you get is the linearized equation dv dt equal or minus 2u times t. No, yes? No, yes?

That's how you would linearize a nonlinear equation-- by perturbing the variable into the

variable something very small and remove all other very, very small terms, and the very, very, very small [INAUDIBLE], if you have cubic terms.

So now we have the linearized equation, which is linear with respect to V. Can somebody tell me, if I use forward Euler, when would this scheme be stable? When would my forward Euler be stable?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Good. In this case, this is my lambda. I'm solving dv dt equal to minus 2u times v. My minus 2u is my lambda. This is where, like, you have a question of adaptive timestamping, right? This is when lambda actually depends on the solution U of the nonlinear equation. As I solve the nonlinear equation, my lambda of the linearized equation actually changes.

So in the beginning, I mean, the solution of this equation looks like this. Actually, we figured out ut, I think, is equal to 1 over t plus 1 or something like that, right? In the first or second lecture we figured that out. So in the beginning, u is large, and minus 2u is kind of on the negative side, and big.

We need to use a smaller timestamp. And later on, as you become smaller and smaller, we are allowed to use larger timestamps for forward Euler to be stable. This is a case where we actually can benefit from adaptive timestamps. The case is minus 2u has to be less than 0, equal to 0, and greater than or equal to minus 2, minus 2u times delta t because I am lambda delta t. This is the stability for forward Euler.

Questions on this? Is it clear how did I linearize this and apply eigenvalue stability? All right. OK.

Now the last thing I want to do, not sure I have time, but, like, I am going to pose these as a question-- is how do I apply backward Euler on this problem. How do I apply backward Euler on this problem.

What is backward Euler? Can somebody help me write down what backward Euler is? OK, what backward Euler is? What is forward Euler? If I use forward Euler, what would it be equal to?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Forward Euler would be e f of u to the n. What would backward Euler be? For f of u n is Forward Euler.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, backward Euler is f of u to the n times 1. So that that is backward Euler. And in this particular case, I'm going to get minus u n plus 1 squared. I'm going to get an equation that looks like this. How do I solve that equation?

If I get any answer that is good, I'm going to end this class. u n is known, right? I'm at the nth concept. So u n is known. Let me put unknown in read. u n plus 1 and the known in blue.

Yeah, that makes it a quadratic equation. We get an unknown square, a linear, and a [INAUDIBLE]. So we can see we need to solve a quadratic equation now in order to go from one step to the next step. We need to solve [INAUDIBLE] nonlinear equation to go from one step to the next step.

And what if this is q? Then I need to solve a [INAUDIBLE]. What is this for? I need to solve [INAUDIBLE].

What if this is final u times exponential u. So this is what we are going to be reading from now to Monday. You are going to be reading about one of the most important techniques in numerical analysis-- that is how to solve a general nonlinear equation like this-- something that is even more complex than a quadratic equation, more complex than cubic equation, something that you may not have analytical solutions.

And it is extremely useful [INAUDIBLE] of nonlinear equations, such as [INAUDIBLE] over here. So I'll see you on Monday.