QIQI WANG: Hello, everybody. So I guess yesterday was the a really serious reading due. How is that going? How do you like the reading?

AUDIENCE: What's consistency?

QIQI WANG: Hm?

AUDIENCE: What's consistency?

QIQI WANG: What's consistency? Consistency is the scheme being at least first order accurate. OK? So the it's local-- consistently is related to local order of accuracy. So if the scheme is at least first order local accuracy, then it is consistent.

AUDIENCE: [INAUDIBLE].

QIQI WANG: It means the truncating error. If you do the truncating error analysis, the data series gives you a truncating error of at least delta t squared if you don't divide by delta t. At least delta t if you divide by delta t. So that's what it means by being consistent. I mean the word consistent basically means if you look at the scheme, it is somewhat consistent to the differential equation. So the scheme on one hand, the differential equation on the other hand. They are consistent, right?

AUDIENCE: So that would mean that the scheme is self-satisfied with the [INAUDIBLE]?

QIQI WANG: The scheme here-- so the question is does that mean the scheme satisfies the differential equation? The answer is the scheme is never going to satisfy the differential equation exactly. But as you make delta t smaller and smaller, it is going to satisfy the differential equation better and better. OK. That's what it means by being consistent.

AUDIENCE: So if it wasn't consistent, it would make the [INAUDIBLE].

QIQI WANG: Yes. So, right. If a scheme is not consistent, what does that mean? That means if you make

the delta t smaller and smaller, the difference between the discrete scheme and the continual differential equation is not going to become smaller. It may become larger. It may remain constant. It may go around but it doesn't go to zero.

AUDIENCE: How does that differ from the definition of stability?

QIQI WANG: How does that differ from the definition of stability? We're going to see there are-- when you talk about stability, there are two types of stability. They are different in the sense that a consistency concerns the Taylor series analysis. You can figure out consistency by doing data series analysis. Stability has nothing to do with data series. Right?

We are going to see when you check stability, you're plugging in a half solution that grows exponentially and seeing can the exponential be a growing exponential? Or the exponential has to be a decaying exponential? If all the exponential solutions have to be decaying exponentials, then the scheme is stable. If there is one mode that can be a growing exponential, then the scheme is unstable.

OK. I clarified a question about consistency. That is basically the scheme is at least first order accurate. What is the difference between consistency and stability? Consistency is by data series analysis. Stability is by plugging exponentially growing solutions. So basically asking does the scheme allow exponentially growing solutions? Any other questions that arise in the reading? If one person has a concern, probably your classmate also has it. So please raise it.

AUDIENCE: Maybe one other way to think about the definition of stability is analysis. [INAUDIBLE] the solution behaves mostly [INAUDIBLE]. When we think about stability, meaning that we're taking the relationship. We're looking at how [INAUDIBLE] analysis. Other things [INAUDIBLE]. The other is [INAUDIBLE]. Because as w goes to 0, [INAUDIBLE]. Because I think more and more [INAUDIBLE] really [INAUDIBLE]. You could probably think there's a few different [INAUDIBLE]. You're going to need both of those in order to analyze [INAUDIBLE].

QIQI WANG: Yes. Thank you. Thank you for the [INAUDIBLE]. Any other questions? No? Then I basically want to very quickly review and recap what we did in the previous lectures and also what we read before today. So we did local order of accuracy. That is basically by plugging Taylor series into the numerical scheme and figuring out how, which is the truncating error. Is it the left hand side [INAUDIBLE] right hand side of the numerical scheme? When you plug in both the differential equation and the data series, does it go-- what order of accuracy does it go?

So as delta t goes to 0, how fast does my inconsistency between the differential equation and my numerical scheme go to 0? OK. And the local order of accuracy is closely related to consistency. Actually the definition of consistency is by looking at is the scheme at least a first order accurate or not.

OK. Global accuracy. You should have read about global accuracy. Could somebody answer me the question of what is the difference between global accuracy and local accuracy? Or are they the same? Yes?

AUDIENCE: [INAUDIBLE]

QIQI WANG: So the global accuracy is the long term accuracy while the local accuracy is not the long term?

AUDIENCE: Local accuracy.

QIQI WANG: The local order of accuracy you can figure out by looking at only one time step, right? The global order of accuracy, you have to look at really by taking the number of time steps to infinity. And as Professor Willcox said, if you look at the solution U at a particular time-- let's say U at t equal to 1-- and compare the numerical solution-- let me say V at t equal to 1-- right? If you look at the difference between them, let's just take the absolute value of the difference. That difference is really looking at how much error has the solution accumulated over one time step all the way from t equals 0 to t equals 1? Right?

And if I look at that area as I take delta t to infinity, I'm actually accumulating more time steps, right? So if delta t is to 1 I'm accumulating 10 time steps. If delta t is [INAUDIBLE], then I'm looking at the total area that has been accumulated over 100 time steps. Now we can for sure see that this puts a more stringent requirement on how my scheme behaves. Because when I only look at local order of accuracy and just look at one time step, now I'm looking at not only how much area have I produced in one time step but also how much has the area grown? I mean, does the area made in one time step actually only matter locally? Or is this area actually amplified by the later time steps?

So it puts additional requirements on a scheme. A scheme is globally accurate only if it is both locally accurate and what? I hear two answers. One says stable. Another is consistent. Who votes for stable? And who votes for consistent? Stable? One, two, three, four. Consistent? One, two, three, four, five. OK.

I said a scheme is globally accurate only if it's locally accurate and what? So if we the answer

is consistent, then it doesn't add anything, right? Being consistent means being locally accurate. OK? A scheme is locally accurate and consistent means it's just locally accurate. It doesn't say anything additional to being locally accurate. That's the definition of being consistent. It is just to be locally accurate.

A scheme is globally accurate only if it's locally accurate and is stable. Actually zero stable. And what does zero stable mean? You should have just read about it. I'm just going to say I mean you have read about it. I'm just going to say another way of understanding zero accuracy is that when you solve the differential equation du/dt equal to zero, the scheme is going to be stable. OK? That's what it means by a scheme being zero stable.

In this sense, it's a very weak requirement. Right? You only need to be stable when solving this trivial differential equation. Right? What is the solution for a differential equation?

AUDIENCE: Confidence.

QIQI WANG: Confidence, right. I mean, if you're so very [INAUDIBLE], even your scheme can't even solve that differential equation, what good is the scheme? Right? OK.

So this is, yes. It is an additional requirement on top of local accuracy. But it is a pretty weak requirement. All right? That is why I think it is such a great result that we can say being locally accurate [INAUDIBLE]. And being zero stable, which is such a weak requirement, you can immediately have the conclusion of global accuracy. Right? So what theorem or result gives you that conclusion?

AUDIENCE: [INAUDIBLE].

QIQI WANG: What is it called? Dahlquist Equivalence Theorem. Yes. Right. So that theorem basically says that these are locally accurate, which only looks at one time step. And if you have are zero stable-- which means you can solve this trivial differential equation-- then your scheme is globally accurate which means you can solve the differential equation for a fixed amount of time as it takes delta t to go to zero, which also means the amount of time steps goes to infinity. Your global solution becomes more and more accurate. OK.

And in addition to this, zero stability gives you the result that the local order of accuracy is the same as the global order of accuracy, right? So if you can do Taylor series analysis, you can not only know the local order of accuracy. You also know the global order of accuracy. OK.

Eigenvalue stability is a step beyond zero stability. This tells you that your scheme has to be not only solving this equation but also du/dt equal to lambda U. Right? And then this is also a pretty easy differential equation. You have analytical solutions to that. The zero stability, you can think of as a special case. It's a much weaker requirement than Eigenvalue stability. Eigenvalue stability means you need to solve now nontrivial differential equations.

OK. So today what we're going to do is we are going to exercise this local order of accuracy, global order of accuracy, and the zero stability, and Eigenvalue stability by looking at a pretty simple problem. It's a ballistic trajectory prediction problem. I hope most of you brought your computer. Does everybody have your computer with you? And do you have MATLAB installed? Yes? OK.

So what I want you to do is, I want you to do mathematical modeling which is deriving some ODEs. I'm not going to give you any ODEs. You need to derive it. And you're going to solve the ODE using forward Euler. Solve the ODE using midpoint. And solve the ODE with one of the homework, the first homework question you did, which is the most accurate two-step explicit scheme. Anybody who can tell me what that scheme is?

V of m plus 1 equal to minus 4 V m plus 5 V m minus 1.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Plus 4 delta t F of V m plus 2 delta t F of Vm minus 1. So I mean, yeah. It is the most accurate two-step scheme. But why doesn't it have a name? Something like this should have a name, right? Let's try to see why it doesn't have a name by applying this onto the ballistic trajectory prediction problem.

All right. And let's look at accuracy, stability, convergence and how they relate to each other through the Dahlquist Equivalence Theorem using this example. OK. Ballistic trajectory predictions. Here is some history. It is really the motivating factor of developing the first real computer, the first real electrical computer is what they call it. They are really designed to help engineers solve this kind of problem. And here we are going to be programming a MATLAB to solve such a prediction problem.

So we are thinking if the motor fires a small 3 kilogram and 10 centimeters in diameter spherical cannonball, so it is fired over here. At sea level in standard atmosphere, you can look for the air density or whatever from the internet. And so I want to derive the differential

equation. That is going to allow me to predict the impact if I give you the initial velocity, the x velocity and y velocity of this thing.

Here is some additional data. Aerodynamic drag has to be considered. The cannon ball, the force on the cannonball is gravity and drag. Right? Those are the only two forces. And let's assume it's fired at a subsonic speed. Subsonic Cd, that coefficient. Let's use 0.5. And yeah, actually we give you the air density. So you don't have to find it out. So what's the differential equation? The differential equation such that you can [INAUDIBLE] in MATLAB.

For deriving the equation, you can work in groups of two or three. Writing the code has to be done individually. All right? So if you're sitting alone, please tend to gather. Form groups. And continue working out the differential equations together. Any questions before we dive into the lab?

AUDIENCE: [INAUDIBLE].

QIQI WANG: Uh huh.

AUDIENCE: [INAUDIBLE]

QIQI WANG: What is the angle? This is [INAUDIBLE]. And I'm going to tell you that. Let's leave it as the parameter [INAUDIBLE]. And I'm going to tell you where the [INAUDIBLE] what happens [INAUDIBLE]. OK. Lot of you have figured out the ODE. So I'm just going to write it down here so that everybody is on the same page when you go into implementation. OK.

What happens is that the force-- if you look at the cannon ball, if the velocity is Ux and Uy, it has a force of gravity, m times g. It has a force of drag, right? And the magnitude of the drag is equal to Cv times half of rho U squared. U squared is Ux squared plus Uy squared. And the drag also has an angle. The angle is proportional to the angle of Ux and Uy but in the opposite direction.

So if you write down the differential equations, it's dUx/dt. The force on the x component only has a component of drag on it. Right? So it is minus D times Ux divided by square root of Ux squared plus Uy squared. This is the angle. And as you plug it in, it is-- oh, and times the area. When you plug it in, it is equal to minus 1/2 of Cd times row S times square root of Ux squared plus Uy squared times Ux. And dUy dt is equal to first of all--

Wait. I think I'm missing an m here. m times d Uy/dt. That is the acceleration on the y direction,

which is also equal to the force in the y direction. It is equal to minus m times g minus a similar drag component which has the same Cd Row S over two. It has the same square root of Ux squared plus Uy squared. But the direction is in the y. So it is proportional to the Uy. All right. So these are the two differential equations.

And when you want to compute the trajectory, you also need dx/dt equals what? x and y is the location of the cannon ball. Ux have and dy/dt equal to Uy. So these are the four differential equations you need to implement and solve.

So let's try fourth order first. And if you checked your email over the last half an hour, you're going to see I shared a Dropbox folder with you. How many people got that? OK. So if you are happy with it and if you have Dropbox on your computer, you can make the subdirectory in the shared folder and you can pull in that directory so I can take a look from here after you put it up.

So if you look at your email, you are going to see a folder called 1690shared2014 being shared with you. If you accept it, everybody is going to see the same content here. And I see people have already uploaded code over here. Great. Thank you very much. I'm going to open your model.

AUDIENCE: [INAUDIBLE].

QIQI WANG: That's a what?

AUDIENCE: Let me upload my [INAUDIBLE].

QIQI WANG: Oh OK. Sure. Please.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Are you going to also--

AUDIENCE: OK.

QIQI WANG: Oops. OK. I'm going to wait. Oh OK. So somebody must have not had Dropbox before. Because if I shared with you and you actually--

AUDIENCE: [INAUDIBLE].

QIQI WANG: Yeah.

[LAUGHTER]

AUDIENCE: [INAUDIBLE].

QIQI WANG: I get more space if I shared. And as a result of my sharing, somebody who didn't use Dropbox before now uses Dropbox.

AUDIENCE: Oh.

QIQI WANG: Yes.

AUDIENCE: So you don't get access to anything?

QIQI WANG: All right. OK. So I have codes here. And let me first run it to see if it also runs in my computer. Mmm.

AUDIENCE: What is [INAUDIBLE]?

QIQI WANG: Oh. I see, I see. OK. So let's go through the [INAUDIBLE] here and explain to us what is your code doing.

AUDIENCE: Right. So first, I put in some constants that we need-- the diameter, the coefficient drag. I apologize for any errors. I'm definitely not immune to errors. I don't think there are any. But we calculate drag. And then I calculate the cross-sectional area, rho, gravity. And then I figure out the initial conditions. So I didn't actually hear if you gave us an output. So I just put in pi over 4.

QIQI WANG: OK great.

AUDIENCE: But I could have put anything in there. So I calculated the initial x and y velocities here. And then I create vectors with plenty of space to hold all my [INAUDIBLE] and set the initial conditions in the velocity. And then I create a time step. And then this is the integration.

So what I say is the position at 1 times 7, the future x position is the current x position times et times the x velocity. And same thing for y. And I have to calculate the changes to the velocity of both x and y. And then I add those, again multiplying by the time step. And then here it hits the ground again, I think. And then I plot.

QIQI WANG: Great. Thank you. Any questions about this [INAUDIBLE]? Is it perfectly clear, everything?

AUDIENCE: [INAUDIBLE].

QIQI WANG: Here. Let me run it again to see if it actually still works. I am going to click Run. It works.

AUDIENCE: And there's some residuals here [INAUDIBLE].

QIQI WANG: So thank you. And let me take a look at another code that is by same thing. Did you update?

AUDIENCE: Yeah.

QIQI WANG: Let me close it. So--

AUDIENCE: Probably ballistic.

QIQI WANG: Ballistic.

AUDIENCE: Yeah. That's been upgraded.

QIQI WANG: OK.

AUDIENCE: Try again.

QIQI WANG: No. OK. I got another code from-- all right. Thanks. And which one should I look at first?

AUDIENCE: [INAUDIBLE].

QIQI WANG: [INAUDIBLE]. Do you mind coming here and explaining what you did? I think it's a little bit different from what-- so you organized the code in a different way. I think this is more consistent with what I did in the last section.

AUDIENCE: Yeah.

QIQI WANG: Of the other one. Yeah, the other one.

AUDIENCE: So what I did was to find all my initial values, I chose a state of 45 degrees. And this is all [INAUDIBLE], actually.

[LAUGHTER]

I defined my rate on each side as 1,000 elements. And then I chose a dt of 0.01. And then I made an initial vector which has my x0, y0, and the dx0 and dy0. And then we go into the while loop to do our forward Euler method. And then to calculate our next element in our

forward Euler method, I go to this function called trajectory that I made.

QIQI WANG: So trajectory t is a function you made. So let me open this also. All right.

AUDIENCE: Yeah. And then here, you can see the equation [INAUDIBLE]. And so what this trajectory does is it takes in the position and velocity. It applies them to the [INAUDIBLE] we derived, and then outputs two velocities and two [INAUDIBLE], which we then multiply by a change in our time step to solve for the new position

QIQI WANG: We have any questions on this? And in the other file, I believe I need to change this to this. Because I think you changed the name of this which you also have to change this too. So that way, [INAUDIBLE]. So [INAUDIBLE], you can do the derivative that you did.

AUDIENCE: After we did the derivative, we have to multiply it by our time step to go up an order to change it from velocity to speed and position. We add that to our old position and get our new position. And then we store that in our position vectors. And then we [INAUDIBLE] the position.

QIQI WANG: OK. Let's try to see if it works. OK. Let me close everything. Let's click Run. Yeah. We get a [INAUDIBLE].

AUDIENCE: [INAUDIBLE]

QIQI WANG: Yeah.

[LAUGHTER]

And we do see that [INAUDIBLE], right? Because the angle here is much deeper than the other angles. [INAUDIBLE] slower than when it was shut off.

OK. Now my question to you is, what if I want to challenge you with a question? What if I want to change this to midpoint? From last lecture, we saw that at the same time step, midpoint was much more accurate than forward Euler. Right? So what should we do to change this to midpoint? I'm asking the whole class. Anybody have any idea? Yes?

AUDIENCE: [INAUDIBLE].

QIQI WANG: Good.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Thank you.

AUDIENCE: [INAUDIBLE].

QIQI WANG: [INAUDIBLE]. OK. OK. So, right. Now the question is what if we want to change this to midpoint? If we change this to midpoint, the first thing is that midpoint is a how many step scheme? It's a two-step scheme. Forward Euler is a one step scheme. So in terms of implementation, what do we need to change? What do we need to ignore when we switch from a one step scheme to a two-step scheme? Why do we call it a two step scheme?

AUDIENCE: [INAUDIBLE].

QIQI WANG: Because we need to store two time steps. Here we only need to have that. That is one previous time step. Now we have to have two previous time steps. Right? So we need this. And we need one step of forward Euler to actually get to the next time step. So I'm just going to say a one step of forward Euler.

OK. So in doing one step of forward Euler, I'm just going to copy this and copy this. Right? Here, I'm just going to pass my vect 0. And the vect equal to vect 0 plus Vt times d vect dt. Any questions on this one step of forward Euler?

OK. Now we are using midpoint. OK. In mid-point, now what we want to change is these two lines. Right? Let me scroll it down. We need to change these two lines. Can somebody tell me how do I change lines when [INAUDIBLE] to make this midpoint instead of forward Euler? Yeah?

AUDIENCE: You have the vect. And you add it to 2 times U vect dt and your time step.

QIQI WANG: OK. So first of all, in midpoint, we have 2 times delta t times F of v. Right? So we have a 2 times. What else do we need to change?

AUDIENCE: [INAUDIBLE].

QIQI WANG: For which one?

AUDIENCE: In the previous scheme?

QIQI WANG: Yeah. You are right. When I do this, I need to also somehow store the old vect. So I need to do vect of 0. Let me actually do this. Let me actually call this vect 00 as the previous time step.

And the vect 0 as-- so, at any point, that 00? Let me call it k minus 1. OK. vect 0. Let me call it vect k.

All right. So this is one step of forward Euler going from k minus 1 to k. All right. Now in midpoint, where do I compute the derivative? x time step k? Or k minus 1? K, right. So that's k. So this d vect/dt is the time derivative at step k. Now my vect is equal to vect k minus 1 plus 2 times delta t times F of v. Right?

This is because in midpoint-- so forward Euler is V at k plus 1 equal to Vk plus f of Vk. Midpoint, we have Vk plus 1 equal to be Vk minus 1 plus 2. There is a delta t here. 2 times f Vk times delta t. Right? This is what we are implementing right now.

All right. So we have this. And everything else I think is same.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Right. Then I need to also update. Because I need to set vect of k equal to vect. Right? And before I do that, I need to have vect k minus 1 is equal to vect k. Right? So I'm done. Should we run this?

Is it clear like if when you guys have fourth order implemented on your computer how to change your implementation into midpoint? Good? Let's run it. Let me close out. Now I'm going to run it. Is it the same as before? No.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Oh.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Oh. Is this something to--

AUDIENCE: Yeah. [INAUDIBLE].

QIQI WANG: You mean remotely?

AUDIENCE: Yeah. [INAUDIBLE].

[LAUGHTER]

QIQI WANG: OK. OK. What did you do?

AUDIENCE: I didn't [INAUDIBLE].

QIQI WANG: Oh. OK. OK. So let me just make it 10 more time steps.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Oh, yeah. Right.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Yes. OK. Let me run it.

AUDIENCE: [INAUDIBLE].

QIQI WANG: Let me-- I think this is too much. Let me change it to-- I think I just do this much. It may be enough. Let me see it. And when you plot it, let me make sure to do x is equal. Do you know what this means? This just means that the x-axis and y-axis are going to be on scale. Right? OK. That's good. So midpoint also works.

So first of all, I want to use the remaining few minutes, let's try to implement the most accurate scheme we got to see-- we already have a two-step scheme. Right? And it should be pretty easy to change to this scheme we have over here.

So instead of here, so we have a d vect dt at step k, right? We also need a d vect dt at step k minus 1. That is just called in the same function at step k minus 1. Right? You see what I'm doing? I'm computing two time derivatives.

Now in this update, I have a minus 4 times Vk and the plus 5 times Vk minus 1 plus 5 times this, right? OK. And then the second line is plus 4 delta t f of Vk plus 2 delta t f of Vk minus 1. So this is 4 plus 2 times dt times d vect dt k minus one. Right?

So that completes our change from midpoint to an even more accurate two-step scheme. And by more accurate, I mean global or local order of accuracy?

AUDIENCE: Local.

QIQI WANG: Local, right? Because we did this using Taylor series analysis. So let's see if it translates into global order of accuracy. Let me click, let me close this first. And let's run it.

You see this? You see this? What does this mean, 10 to the 162? That's where my cannonball bounced. What happens? What happens? Let's look at our scheme solution not for this many time steps. But let's just go 19 time steps. And let's put a circle at every time step to see what actually happens. OK.

After just 19 time steps, we've got 10 to the 22. That's amazing. And we basically just shoot out and diverge, right? So why do you think that is the case? Do we have any global order of accuracy? I mean, we can do that by changing the time steps to be even smaller.

And we would change-- don't change it to this. Let me change it to twice smaller. And the time steps to also twice as much. That is a good way of assessing the global order of accuracy, right? It will decrease the time step by a factor of two, increase the number of time steps by a factor of two, right? To see if the solution gets more accurate.

OK. It gets wilder. Instead of 10 to the 100 or something, it's 10 to the 200 and something. All right. So something is wrong. What is wrong? Can somebody just shout out? What we reviewed in the beginning? Our scheme is not zero stable. It can't even solve this differential equation. Let me repeat. The most accurate two-step scheme can't even solve du/dt equal to 0.

You want me to prove that? I'm going to prove that analytically. OK. We have the following scheme. I'll see if I can do it in five minutes. It's equal to minus 4 Vk plus 5 Vk minus 1. And the rest of them on the right hand side are f of V, right? If I solve the differential equation du/dt equal to f of U equal to 0, then f of U is equal to 0. I have nothing after this. And this is my scheme. I want this scheme to reproduce a constant solution. Right?

The question is why can this scheme not reproduce a constant solution? The answer is like this. The answer is because this scheme is going to allow an exponentially diverging solution. OK. If I have a solution Vk is equal to V0 times a zeta-- let me just write out z because I can't write zeta-- times Z to the k. I'm going to see-- if I plug into this equation and the equation is satisfied and my V is something greater than 1-- then what does it mean?

It means I have a very, very small V0, even if I have an extremely small V0, I can end up having a huge solution after sufficiently many time steps. Right? And when I plot this thing, I'm going to have a V of k plus 1 is V0 times V to the k plus 1 is equal to minus 4 times Vk V0 times V to the k. And sorry, this is k minus 1. Right? This is my scheme. Times V0 V to the k

minus 1.

I cancel this. Cancel this. Cancel the V0. Because they are the same on both sides. And though all of these have V to the k minus 1 at least-- some of them are k, some of them are k plus 1. So I remove that vector. I have V squared is equal to minus 4 Z plus 5. We get a quadratic equation. Right?

This quadratic equation can have real or complex solutions. But let's see what it has. So a is equal to 1. b is equal to 4. c is equal to minus 5. So we get a minus b plus minus b squared, which is 16. Minus 4ac, which is plus 20. All right? So that's my two solutions.

And I can see it is-- right? I'm basically factoring the two into these. And I minus 2, plus or minus 3. So that's equal to what? It's equal to either minus 5 or 1. All right. OK. If I only look at the one, that means good. It allows a constant solution, right? Z is equal to 1 basically means Vk is equal to V0 period, or whatever k. So in this sense, it does allow a numerical solution of this differential equation. du/dt is equal to zero.

But it has another solution of Z equal to minus 5. And it is that solution that makes the scheme diverge. You only need one bad solution Z to make the scheme diverge. And that's what we saw over here. The scheme is not zero stable. Therefore, although it is locally accurate, it is not globally accurate. But because it's not zero stable, it can't solve this equation.

All right? I'll see you tomorrow and continue discussing this and also look into Eigenvalue stability.