

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. ?]

**QIQI WANG:**

All right. So last time I did this using the video recording, there was some complaints about the audio quality, so I'm wearing this today. It feels a little bit weird, but I'll see how much better the audio quality came through. And you can ask your questions [INAUDIBLE].

All right. Can you pass this on, like everybody have one piece? So the purpose of this is for you to just write down anything that is not clear onto this piece of paper, and hand it to me after the lecture. What I'm going to do is, I'm going to draft these questions in the next section so that you're not going to stay confused on what might confuse you. So this important [? Monte ?] cuts. You can write anything that is [? Monte, ?] and I'm going to be lacking them and try to earmark your [? Monte ?] points.

So we are continuing our discussion integration of ordinary differential equations. Let's just do a brief review of what we did in the last lecture. So we are solving differential equations like  $du$  over  $dt$  equal to  $f$  of  $u$ . All right. We discretized the time into  $t_0$  equal to 0,  $t_1$  equal to  $\Delta t$ ,  $t_2$  equal to  $2\Delta t$ , et cetera. We discretized the solution,  $u$ , into  $u_0$  equal to  $u$  at  $t_0$ , et cetera and  $u_i$  denoted as the solution,  $u$  at  $t_i$ .

Forward Euler, solves the equation like this. It is basically taking-- I'm going to change the color. It is essentially taking this differential equation and approximate the time derivative into  $u_{i+1} - u_i$  divided by  $\Delta t$ . And on the right-hand side, it is the same thing. So this is forward Euler.

A midpoint rule, we are going to be doing something very similar, but instead, I'm going to discretize the time derivative term in a different way. I'm going to discretize the time derivative term into  $u_{i+1} - u_{i-1}$  divided by what?

**AUDIENCE:**

[INAUDIBLE].

**QIQI WANG:**

$2\Delta t$ . Exactly. So this is a consistent approximation of the time derivative. So that's the only difference between forward Euler and midpoint rule.

And how to analyze the accuracy of these two methods? Taylor series expansion. In midpoint rule, we are expanding  $u_{i+1}$  equal to  $u_i$  plus-- you should become very familiar with this as you learn more. So  $u_i'$ , which is a shorthand for denoting  $du/dt$  at  $t_i$  plus half of  $u_i''$ , which means  $du^2/dt^2$ , the second derivative of  $u$  times  $dt^2$  plus all the other terms. The third term is  $u_i''' \Delta t^3$  plus et cetera. Is a little bit too wide, maybe.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** You can't see. Let me change this to a little bit darker green. And  $u_{i-1}$  is equal to-- let me scroll this up a little bit-- is  $u_i - u_i' \Delta t$ . Why minus? I'm going backwards. I should test  $\Delta t$  to minus  $\Delta t$ , right? This is why there is a minus here. And this term should have also a minus here because minus  $\Delta t$  is squared. So I have, again, altered it here.

And this term? Should I have a minus here? Yes. Because when I cube it, the negative sign is preserved [INAUDIBLE]. So when I take the subtraction between  $u_{i+1}$  and  $u_{i-1}$ , this got canceled this gets doubled, this gets canceled, and this gets doubled.

So what I ended up having is  $u_i' + 1/6$  of  $u_i''' \Delta t^2$ . I mean, this cube cancels with this  $\Delta t$  and  $\Delta t^2$  equal to  $\Delta t$  of  $u_i$ . And this is  $\Delta t^3$  cubed, so this is the leading term of the truncation error.

Remember my origin original differential equation. My origin or differential equation is  $du/dt = f(u)$ . The midpoint rule gives me  $du/dt$  plus something times  $\Delta t^2$  and plus an even smaller terms equals  $f(u)$ . So these additional term that is append from the discretization is the truncation error of the scheme, is the error made by the scheme, is the approximation. How much accuracy do I lose by approximating the derivative using discretization scheme?

All right. This is the truncation error. And if you do forward Euler, this truncation error, would it be  $\Delta t^2$ ? No. It will be on the order of  $\Delta t$ . So forward Euler is less accurate than midpoint. All right. What we are talking about is this is called local truncation error. The reason why we call it local truncation error is going to be clear in the next section because there is a different way of assessing the error is the global error, and we're going to be talking about that next one. Yes.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Good question. So the question here is, why did I start expanding here. What if I happened to expand all the way here? I can write a hundred terms after this. There is infinitely many terms in the [INAUDIBLE] I stopped over here because I did this, and I know that this term is not attainable. If a term is not unattainable then this is going to be the leading term in your local truncation error.

The way you do your own is if I do an analysis of a scheme I've never seen before, I would expand it somewhere maybe onto here, and then dug it in to see if I get something that is not canceled. What happens if I see everything cancels? What if I see everything cancels, and all I have is exactly the [? C. ?] Why I might have an [? o ?]  $\Delta t$ ?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** I should expand more. Exactly. That means that  $\Delta t^2$  may be the leading term of the truncation error, but would it be the leading term of truncation error if it is given less than  $\Delta t^2$ ? I don't know. I only know that the scheme is at least third order accurate [INAUDIBLE]. Only when I expand it more terms, I can know it happened in what order of accuracy this means.

All right. So when you [? file ?] a scheme, when you're not sure, expand it to whatever you like and see if everything cancels. If everything does cancel, then it means you need to go back and expand more terms. Is it clear?

In the last lecture, I asked you to do this to find out who the best X schemes. Can I have a show of hand of who did this? Who did this? Who attempted? Good. Who attempted the first one?

Let me see, 1, 2, 3, 4, 5. I don't need you to be successful, just tell me what is the best you got.

**AUDIENCE:** The first order [INAUDIBLE].

**QIQI WANG:** No. What is the scheme? So I want the product is, what is the--

**AUDIENCE:** [INAUDIBLE] coefficients?

**QIQI WANG:** Yes.

**AUDIENCE:** I just got  $r_1$ .

**QIQI WANG:** Did you get--

**AUDIENCE:** I got  $u$  of  $t$  sub  $\Delta t$ .

**QIQI WANG:** Good. You get  $u$  of [INAUDIBLE]?

**AUDIENCE:** Yeah.

**QIQI WANG:** -- $t$  plus  $\Delta t$ .

**AUDIENCE:** --is equal to  $u$  prime.

**QIQI WANG:** --is equal to  $u$  prime at--

**AUDIENCE:** -- $t$  plus  $\Delta t$ .

**QIQI WANG:** -- $t$  plus  $\Delta t$ .

**AUDIENCE:** --plus  $u$  of  $t$ .

**QIQI WANG:** --plus  $u$  of  $t$ .

**AUDIENCE:** --plus  $u$  prime of  $t$ .

**QIQI WANG:** --plus  $u$  prime of  $t$ . Anybody get anything different, or everybody agrees with that? Five people got the same thing?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** OK. Let me write this in a more concise manner.  $t$  plus  $\Delta t$  is [? as ?] in  $f$  times  $x$ .  $t$  is at the current time set. So let me denote everything,  $t$  plus [INAUDIBLE]  $i$  plus 1,  $i$  plus 1, and this is  $i$  and  $i$ . So it's just clearer not having all the  $t_n$ ,  $t$  plus  $\Delta t$ . If I know-- come on. Come on.

And you are sure there is no  $\Delta t$  in front of this? I mean, forward Euler, and at midpoint, I have  $\Delta t$ 's. Yeah?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** OK. All the derivative terms should have  $\Delta t$ .

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** You don't have that?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** I was listing delta t's. Sorry for that, but if I do the theta series and I just [ $\Delta t$  carefully  $\Delta t$ ], correctly, I should test that. And if you got something different, let me write it here. You've got something  $u_i + 1 = \frac{1}{2}(u_i + 1) + \Delta t$  plus  $u_i$  plus, you also get  $\frac{1}{2}$  here. Yes? You got something different?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:**  $u'$  prime of delta t. You get negative  $\frac{1}{2}$  here. All right. So we get three answers. Is that all? Anybody get anything different? Good. We have five people got three answers, so at least there are some agreements.

Who did the second one, best explicit, two-step scheme? Who attempted the second one? You? So we have 1, 2, 3, 4, 5, 6, 7. More people did this. What is the best scheme you guys have got?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** The question is, can she-- what's your name?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** OK. Tren's question is, can we express the derivatives as  $f$  as well? The answer is yes. So whenever I see a derivative, I can plug in the derivative into the differential equation. The differential equation is  $u' = fu$ . So whenever I have  $f'_{i+1}$ , I can replace this as  $f_{i+1}$ . Whenever I have  $f'_i$ , I can replace this as  $f_i$ .

So what is your answer of the best explicit two-step scheme?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Negative  $4u_i$ ? OK.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** f u i and--

**AUDIENCE:** --plus 2 f times u [ $\Delta$  sub  $\Delta$ ] i minus 1.

**QIQI WANG:** OK. Good. Anybody who get anything different for best explicit two-step scheme? You all got the same thing?

**AUDIENCE:** [INAUDIBLE]

**QIQI WANG:** Delta t's. And I guess delta t here. You must have that too. All right. Oh, wow. That's amazing. Everybody got this? No disagreements? So the explicit two-step people are more in agreement than implicit one-step people.

Who did best implicit two-step scheme? Who attempted? You also did that? OK, great. And what is the answer?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** OK, you.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** I'm going to append a delta to for you. Plus--

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Wonderful. So can you tell me how you did that?

**AUDIENCE:** I did a bunch of Taylor [INAUDIBLE].

**QIQI WANG:** Nice. I get to have you show me how you did that later on. Who the best explicit three-step scheme? Anybody try that? You also tried that? OK, great. I'm going to have you as an instructor here. What is the answer for that?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Minus what? Sorry?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** 18 and sorry, what?

**AUDIENCE:** 9.

**QIQI WANG:** 9.

**AUDIENCE:** 9, 18, 10, and 3.

**QIQI WANG:** 18, 10, and 3. So what is multiplied on this one?

**AUDIENCE:** [INAUDIBLE]

**QIQI WANG:**  $u_i$  minus 2 and  $u_i$  minus 2 prime. So all the primes have  $\Delta t$ , I guess. Great. So I'm going to do some analysis of what this is. I'm going to go back to this. So I saved the documents. Even if I lose it, I'm going to go back to this. But what I want to show right now is first of all, I'm sure some of you did this, so you know already how to do this. But I also want to see especially resolve the disagreement on the first one.

So that is what I'm going to do first. What is the best implicit, one-step scheme? I'm going to be having a demonstration of that right now, and then I'm going to show you how to implement a scheme once you derive it. You derived a lot of schemes, now how do I put this scheme into a code, into a production, into something actually wrong.

So first of all the best implicit one-step scheme. All right. I'm going to append something after this. The best one-step scheme. So the scheme has to be  $u_{i+1}$  equal to something times  $u_i$ . All right. Did I say implicit, one-step scheme. So one-step scheme means everything has to be written in terms of  $i+1$  and  $i$ . And it can depend on  $u_i$ ,  $u_i'$ , and  $u_{i+1}'$ . This is because it can be implicit. If it's explicit, then it has to always depend on  $i$ .

All right. So let me put in three unknowns,  $x$  plus  $y$  plus  $z$ . And these are unknowns. And what do I do to make the scheme as accurate as possible?

**AUDIENCE:** Minimize the error?

**QIQI WANG:** Minimize the error. Exactly. And the error can only be  $O(\Delta t^4)$  from Taylor series expansion. All right. So Taylor series expansion, again, there are two options. One option is to expand all the  $i+1$ 's at  $i$ . So for example,  $u_{i+1}$  is equal to  $u_i + u_i' \Delta t + \frac{1}{2} u_i'' \Delta t^2 + \dots$  I can do more terms-- triple prime  $\Delta t^3$  plus  $O(\Delta t^4)$ .

But actually, in this case, another equally convenient way is to expand  $u_i$ 's in terms of  $u_i$ 's plus

1's because there are as many terms that is  $i + 1$  as there are many terms of  $u_i$ . So it is equally convenient, so equally cumbersome, whichever way you think about it.  $u$  prime of  $i + 1$ , I can also do a Taylor series expansion also by just taking the derivative.

Now, I'm going to write everything [? correspondence ?] here, but every term has one more derivative than what is above. So when I have third derivative here, I actually have fourth derivative. And this is still to the fourth. So all the delta  $t$ 's to the power is the same, but all the derivatives have one more order because I'm expanding the derivatives.

And when I should take the second load of derivatives, I should be taking the second derivative of the derivative, which is the third derivative. All right. Is it clear why I'm having one more derivative here?

Now, this is all I need to expand. And to figure out the error, we need to plug these extensions into the formula. We also need to plot something else in the formula. Oh, in this case, we don't need to plug anything else. If I were writing the scheme not as  $u$  prime, but as  $X$ , then I also need to plug in the differential equation into the formula. But I'm already writing the scheme in terms of  $u$  prime, then I don't need to plug anything else into the formula. I just need to plug in the Taylor expansions.

When I plug in the Taylor expansions, what I get is, I'm going to write down the truncation error, which is the left-hand side. Well, this case, it is actually the truncation error times delta  $t$  is equal to this. I'm going to let you know why I'm writing this. But the truncation error times delta  $t$  is the left-hand side minus the right-hand side.

And let me just write down term by term. The first term is  $u_i$  plus, which is this. Let me actually start by changing the color. It is  $u_i$  plus  $u_i$  prime times delta  $t$  plus  $u_i$  double prime times delta  $t$  squared over 2. plus  $u_i$  triple prime delta  $t$  cubed over 6 plus our delta  $t$  to the fourth.

The second term is minus  $x$  times  $u_i$ -- let me changes the color to this-- minus  $x$  times  $u_i$ .

Now, what is the third term? It's minus  $y$  times  $u_i$  prime. So it is minus  $y$  times  $u_i$  prime. I'm writing down here so that everything responding to prime is on this column. And the third term is minus  $z$  times  $u_i$  plus 1 prime, and we again plug in the Taylor expansion. But everything starts at the prime, so I have a minus  $z$  times  $u_i$  prime here. I have minus  $u_i$  dot double prime times  $z$  delta  $t$ .



So this is double prime. This is  $z$  times this, and I also have this. I have minus  $ui$  triple prime times  $\frac{1}{2}$  of  $z \delta t$  to the cube. And I don't even need to write this. I'm just going to write to this as  $o \delta t^3$  times  $z$ .

All right.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Up more?

**AUDIENCE:** No, the other way.

**QIQI WANG:** The other way. Scroll up. Right. So what I'm doing is I'm expanding this term as the first line, this term at the second line, this term as the third line, and this is the fourth line. I'm just writing things down, and it's different lines, and I'm also aligning the same derivatives of  $u$  on the same color. Is it clear?

So how do I choose the coefficients? I'm going to choose coefficients to cancel as many terms as possible. I'm going to choose, first of all, the first priority is to cancel anything that is  $ui$ , anything multiplied on  $ui$ . So what  $x$  should I choose?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Exactly.  $x$  has to be equal to 1 because I want these two terms, I want them to be canceled. The only way to cancel this is we choose  $x$  equal to 1. Is it clear? And what should I do to ensure that this term also cancels?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** What? Somebody?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:**  $y$  plus  $z$  equal to 1.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Equal to  $\delta t$ . Exactly.  $y$  plus  $z$  has to equal to  $\delta t$  in order for this, this, and this to cancel.  $y$  plus  $z$  has equal to  $\delta t$ . Do we have a question? No? All right. What can we do to cancel this?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** We need to have minus  $z \Delta t$  to cancel with the last  $\Delta t$  squared over 2. So we have  $z \Delta t$  has to equal to  $\Delta t$  squared over 2, which means  $z$  has to equal to  $1/2$  of  $\Delta t$ . Now, the question is, can I cancel more? Can I make sure I can cancel more? No. I only have three terms to play around with. I only have three unknowns,  $x$ ,  $y$ , and  $z$ .

So I can only satisfy three equations. On [INAUDIBLE] that of the three equations. There is [INAUDIBLE] that satisfy automatically. So unless I'm super lucky, I can only tune the three unknowns to satisfy three linear equations. All right.

So in this case, it is pretty clear that  $z$  has to equal to  $1/2$  of  $t$  and  $y$  because of the second equation also has to be equal to  $1/2$  of  $t$ . Or I can just take these three equations and make it a matrix. I can take these three equations and particularly, I can make, if I can-- well, let me see. If I can divide the second and third equations by  $\Delta t$  so that the unknowns are  $x$  and  $y$  over  $\Delta t$  and  $z$  over  $\Delta t$ . If I make them as the unknowns, then I can write the equation as  $x$ ,  $y$ ,  $z$  equal to 1, 1, and  $1/2$ .

How can I do that? How can I fill in this matrix?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Well, the first line is 1, 0, 0. Thank you. It's just the first equation  $x$  equal to 1. What about the second line? I mean, this is not  $y$ ,  $z$ , but  $y$  over  $\Delta t$  and  $z$  over  $\Delta t$ . What is the second line?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** 0, 1, 1. Exactly. And what is the third line? 0, 0, 1. I mean, this looks obvious, but as you get to more complex schemes, as what's your name again?

**AUDIENCE:** Jacobi.

**QIQI WANG:** Jacobi. Jacobi has discovered in his [? one ?] contact schemes, you cannot just by looking at equations and get all the coefficients. You have to solve a matrix equation like that. And in MATLAB, let me set up this matrix. This matrix is 1, 0, 0, the first line. The second line is 0, 1, 1. The third line 0, 0, 1. This is the matrix. Exactly. The matrix.

And the right answer is 1, 1, and  $1/2$ . Yeah, that's the right-hand side. And by a backslash  $c$ ,

does everybody know what the backslash means? Yes. It's matrix division. It is solving this equation. It is solving  $ax = b$ . It solves for  $x$ , and I push. It give me the coefficients,  $1, 1/2, 1/2$ , which means  $x = 1, y = \frac{1}{2} \Delta t$ , which means  $y = \frac{1}{2} \Delta t$ , and  $z$  also equal to  $\frac{1}{2} \Delta t$ .

All right. So some of the scheme we have is  $u_{i+1} = u_i + \frac{1}{2} \Delta t u_i' + \frac{1}{2} \Delta t u_{i+1}'$ .

All right. So good. I think we resolved the conflict. There is only one scheme that satisfy this matrix equation. And there is only one scheme that can make sure the first term cancels, the second term cancels, the third term cancels. All right. And remember  $z$  is  $\frac{1}{2} \Delta t$ . So what is the truncation error here? What is the truncation error? What is the other of the truncation error of the scheme? What local [? order of accuracy ?] does the scheme have?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Second order  $\Delta t$  square. Because this is square, so sorry. This is square, but  $z$  is equal to  $\frac{1}{2}$  of  $\Delta t$ , so this term is  $\Delta t$  too. So the truncation of the leading term of the  $\Delta t$  times  $\tau$  is  $\Delta t$  too, which means  $\tau$  is  $\Delta t$  square.

All right. And why did I say this is  $\Delta t \tau$ ? This is because if you look at how accurate I am approximating the derivative, I mean, I'm approximating the average of the derivative between  $i$  and  $i+1$  set. I'm basically approximating the average of the  $i$ -th derivative and  $i+1$  derivative. It is equal to what?  $u_{i+1} - u_i$  divided by  $\Delta t$ .

And if you look at the truncation error of this scheme, so if I do this rigorously, this is  $\Delta t$  cube here. And if you transform, go from this equation to this equation, you have to divide everything out by  $\Delta t$ . So you get  $\Delta t$  squared. So the truncation error of this scheme is  $\Delta t$  squared.

All right. I'm basically taking these two terms, dividing by  $\Delta t$ , and getting this. Well, I'm moving this term to here and divide by  $\Delta t$  to get this. And what is that here? Because I'm dividing by  $\Delta t$ , I get  $\Delta t$  squared. So the scheme is second order accurate. Any questions? No? Here?

Let's try to implement the scheme into one of the very simple problems. Let's do a pendulum problem. In a pendulum problem, we have a pendulum that is vibrating on a small angle. So

today we restrict ourselves to small angles so that the equation is not going to be linear. And we're going to be going to [? nulling ?] equations, and we're going to see why [? nulling ?] equations is going to be more complex especially for implicit schemes.

So if it's explicit scheme, actually, I'm going to show you later that it's very easy to do a non-linear scheme, but for implicit method, like the one we just derived, one that actually involves a  $u_{i+1}$ , it is going to be much more difficult to do a non-linear scheme. But here the variable is  $\theta$ . And what determines the acceleration of  $\theta$ ? What is the second derivative of  $\theta$ ? What's that going to be determining?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Force balance, right. What is the acceleration of this ball? Let me just ask, what is the gradual acceleration of this ball?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** [INAUDIBLE]. This is [INAUDIBLE], so the only way to accelerate is along the [? duct. ?] So it is [INAUDIBLE]. is acceleration?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** It can be moving. Let's assume there is no friction. The only force it has is the gravity force and the tension of its spring, which is [? non-elastic. ?]

**AUDIENCE:**  $g \sin \theta$ .

**QIQI WANG:**  $g \sin \theta$ . Why is that  $g \sin \theta$ ?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Yeah.  $a$  is equal to  $F$  over  $m$ . It's force over mass, and what is the force? The force is actually here. This is the force in interaction. So it is  $mg \sin \theta$  over  $m$ , which is exactly  $g \sin \theta$ . So you are right. Sorry. What's your name again?

**AUDIENCE:** Libby.

**QIQI WANG:** Libby. OK. Libby is right. So this is acceleration. And what is the acceleration in  $\theta$ ? What is the second derivative of  $\theta$ ? Just a geometric relation between this and deceleration.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Divide by  $L$ . Exactly. So this is  $a$  over  $L$ . So it is going to be  $g \sin \theta$  over  $L$ . All right. And we are going to be linearizing this equation because  $\sin \theta$ , when  $\theta$  is very small, is what? Again, Taylor series expansion  $\sin \theta$  is equal to  $\sin \theta$  at 0 plus  $\theta$  times the derivative of  $\sin \theta$  at 0, which is equal to 1, which is  $\cos \theta$  at 0, et cetera. So this term is 0. This term is  $\theta$ , so it is approximately equal to  $\theta$ .

I mean, this is probably  $\sin \theta$  can be approximated by  $\theta$  a lot of times, but in order to know this is, again, a result of Taylor's rule expansion. So  $\sin \theta$  can be an approximate of  $\theta$ , and now I'm going to write the question  $\theta \ddot{\theta}$  is equal to  $g \theta$  over  $L$  in a pretty strange way. I'm going to write it as  $\theta \ddot{\theta} =$  no, I'm not going to write it like this. I'm going to be writing this as  $d\theta/dt = \dot{\theta}$ , which is just a definition of  $\dot{\theta}$ . It's equivalent.

$d\dot{\theta}/dt$  is actually equal to, again, the second derivative, which is equal to  $g \theta$  provided by  $L$ . You may be asking, why am I doing this? Again, I'm sorry. So there is a negative sign on this. I have forgotten because-- so if  $\theta$  is positive, the acceleration is in the negative direction, so I'm using a negative sign. Yeah, if you discover something like this, please shout.

I'm writing it this way because remember the schemes we have been deriving are also first-order ODEs. What are first-order ODE mean? It means I can only have first-order derivatives through time. All right. So if I have a second-order ODE here, I need to convert it into a first-order ODE. Actually, I cannot convert it into first-order ODE. I have to convert it into two coupled first-order ODEs.

Whenever you have a second-order ODE, you can always convert it into two first-order ODEs. Whenever you have a third-order ODE, you can always convert it into three first-order ODEs by introducing basically a separate variable,  $\dot{\theta}$ . So here I am having  $\theta$  and  $\dot{\theta}$  as two variables. The derivative of both  $\theta$  and  $\dot{\theta}$  is a function of  $\theta$  and  $\dot{\theta}$ .

Is it clear? Let me do this. Let me go to MATLAB and start to write this. Let me do this. I'm going to create an empty file here. I'm going to find out why MATLAB doesn't listen to me. It's called `pendulum.m`. Open `Pendulum`. That's good. So when Windows doesn't work, you can always rely on MATLAB.

So what do we do? We want to code up this differential equation first. Whatever scheme we use, we need to have a function that computes  $f$  of  $u$ . The function should do like this. You give me  $u$ , the function returns  $f$  of  $u$  or  $du/dt$ . I just call this function  $du/dt$  equal to  $f$  of  $u$ .

And what is  $u$ ?  $u$  actually contains  $\theta$  and  $\dot{\theta}$ . Let me do this.  $\theta$  equal to this first element of  $u$ .  $\dot{\theta}$  is equal to the second element of  $u$ . I'm going to be computing  $\ddot{\theta}$ -- I'm just going to call this  $d^2\theta/dt^2$ -- is equal to  $-g \sin \theta / L$ .  $L$  is equal to  $5$ .

So now what I need to return is  $du/dt$ . And what is  $du/dt$  now? Remember  $u$  is  $\theta$  and  $\dot{\theta}$ . So  $du/dt$  is actually  $\dot{\theta}$  and  $\ddot{\theta}$ . That's all. That function encodes the differential equation. If it will give me a  $u$ , I'm going to return a  $du/dt$ . Here  $u$  is a  $2 \times 1$  vector.  $du/dt$  is going to be a  $2 \times 1$  vector. Clear what this function does?

Now, let's see what forward Euler does. Let's define  $dt = 0.1$ , and we want to simulate this thing for how long?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** 60 seconds. All right. I have to initialize a  $u_0$ . So what  $u_0$  you want? I mean, it has to be a  $\theta$  and  $d\theta/dt$ . Anybody?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** If you look at this, when I am approximating  $\sin \theta$  as  $\theta$ , is it in degrees or radians? It can be either, right? It can be either, so it doesn't matter. Oh, yes, it doesn't matter because it's a linear equation.

It's a linear equation, which means if I change a unit, I'm just scaling everything by the unit conversion factor. And the linear equation still should be satisfied. That's like the definition of the linear equation. If you scale everything by some factor, it's still satisfied.

So it can be either degrees or radians.

**AUDIENCE:** 0.

**QIQI WANG:** 0. And 1?

**AUDIENCE:** [? It's ?] really boring.

**QIQI WANG:** It's really boring. OK.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** So this is my initial condition. And here's the forward Euler. Did you see this? Actually, I think I should do this. I should make another file so that you don't have to look at the bottom. `forwardeuler.m`. I have to open `forwardeuler.m`. So I'm going to be setting  $u_0 = 1, 0$ . I'm going to set  $dt$  equal to  $0.1$ ,  $t$  equal to  $60$ , and I'm going to forward  $t$  equal to  $0, 0, dt$ . So this is a loop.

I'm going to compute  $du/dt$  is equal to actually, I think pendulum of  $u_0$ . And what is forward Euler? Forward Euler is my  $u$  at the next step is equal  $u_0 + u' dt$  times  $dt$ . And when I go to the next step,  $u_0$  should be set to  $u$ .

All right. And now what I want to do is I want to record the history. All right. I want to record the history. What I want to do is `history` is equal to an empty array, and the `history` is equal to `history u_0`. I'm going to run this. It doesn't run. I'm going to forward Euler. It's done, and I have an array `history`, and I can plot `history`. It is done to give me a history of the numerical solutions. Does it look good? No.

So this is forward Euler, and I'm using a pretty big time step. Let's see what can I do better if I use a smaller time step. You see, this is better. And a difficult solution should be an oscillating pendulum with no increase in the magnitude of the oscillation. And now can I do even better by having an even smaller  $\Delta t$ ?

All right.

**AUDIENCE:** It's still working.

**QIQI WANG:** Still working? Am I having a reasonable  $\Delta t$ ? Yeah. It should be-- It actually takes a pretty absurd value of  $\Delta t$  to make this like not even super accurate. And you can visually see the empty array still growing. It is visually different from the analytical solution of the ODE.

So let's go back and try to implement something that is better. Just name a scheme that is better than forward Euler.

**AUDIENCE:** Midpoint.

**QIQI WANG:** Midpoint. Sure, let's do midpoint. To avoid repetitive work, let me just rename the Forward Euler using Midpoint. And just copying the Forward Euler and to rename it as Midpoint. I'm

going to load this guy. I'm opening midpoint.m. So this is Forward Euler, but I'm going to change it to Midpoint.

How should midpoint be different. What is the difference between midpoint and forward Euler?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** If you look at the formula, midpoint, instead of  $u_{i+1} = u_i + \Delta t f(u_i)$ , I have  $u_{i+1} = u_i + \Delta t f(u_{i+1/2})$ . So two things, instead of  $u_i$ , I need  $u_{i+1/2}$ . Instead of just  $f(u_i)$ , I need  $2f(u_{i+1/2})$ . The second one is easy the change. I can just multiply by 2.

Now, how do I do about this? How can I get  $u_{i+1/2}$ ?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Good point. I need to do one step of forward Euler or something to get me one actually at time step first. So let me do this. Let me do one step of forward Euler.  $u_{i+1} = u_i + \Delta t f(u_i)$ . I'll just put a comment here, one step of forward Euler. And the  $u_{i+1/2}$  is equal to  $u_i + \frac{\Delta t}{2} f(u_i)$ .  $u_0$  is equal to  $u$ . Is that enough?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Exactly. I also need to store something else. I need to do this. See, I need to store an extra step. That is why I think now we get to why it's called a two-step scheme because you need to store two steps in your computer's memory. While for the forward Euler scheme, you only need to store  $u_0$ . You only need to store one step in your memory.

So implement of midpoint scheme, the two-step scheme, you need to store two time steps in your memory. You have to store  $u_0$  and  $u_{0.5}$ . So here I'm just going to use  $u_{0.5}$ , and  $u_{0.5} = u_0 + \frac{\Delta t}{2} f(u_0)$ , and  $u_0 = u$ . You keep two previous time steps in memory. So this is going to work, and let me go back to 0.1 L of  $t$ , for which forward Euler did a pretty crazy job. It went over to something like 1,000 or something.

Let's try midpoint. Let me Close All. Did I Close All? Midpoint, but I didn't plot it. Plot History. How about this? Is this much better? It's pretty. So let me actually plot it not just the history. Let me actually make the primes to be accurate. Let me make  $\Delta t = 0.1$ . Let me actually plot the x-axis as the real time instead of time steps. If I plot x-axis as  $t$ , it would be like 1, 2, 3, 4,



et cetera. [? It's ?] the number of time steps.

Now, this is plotting [INAUDIBLE] solution of the ODE. So here we get visibly the same answer as an original solution, which is sinusoidal oscillations. We got half an hour left, but I want to take a short break now to get our minds back. And next what we're going to be doing is [INAUDIBLE]. I'll see how it goes.

Let's continue. Is everybody back? Before I try your schemes, let me just show how easy it is to change what we did with the linear differential equation, change it to a non-linear one. To change it to a non-linear one, we just keep the sine theta here. Just replace this with sine of theta. As soon as we change this to sine of theta, we lose the analytical solution.

I don't know how to solve this anymore. Do you? But once we have a computer, what I need to change is let me open my pendulum.m. What do I need to do? I just need to replace theta with sine of theta.

And let me do this again. Midpoint. I get a solution as easy as I'm solving the linear equations. And when I plot it, I want to plot this with history. Here I get the solution to non-linear equation. I mean, it may look similar, but let me zoom in a little bit to see. Oh, man, MATLAB doesn't listen to me.

I'm going to the x lim, which is kind of a scale to x limit to 0 and 5 here. And well, it doesn't really show a lot of difference, but let me change the initial condition. Open midpoint to m. Let me change the initial condition to how much? 1 is like 60 degrees.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** [? Reads ?] very close to pi, so you get like-- so for a let's try it. Let me plot it again. Oh, yeah. So we get [INAUDIBLE] So it will stay close to the top for a while and then drop down and stay at [? 0 ?] for a while and then [? conditions ?] the other part of the chart. To get over the middle of the area [INAUDIBLE].

So you get some very nonlinear behaviors where you have a big initial [? areas ?] as easy as you're solving a linear equation. And that is why we use numerical methods. This is when we don't have an easy way to get analytical solutions, and it is certainly true for most of the engineering problems we are dealing with. And it's quite rare you can find any of the solutions. And most of the time you have to use numerical solutions.

So let's go back and start to try-- we did forward Euler. We did midpoint rule. Let's try best implicit one-step scheme. And I think I remember what it is. Please correct me if I'm wrong.  $u_i^{n+1} = u_i^n + \frac{1}{2} \Delta t u_i^{\prime n} + \frac{1}{2} \Delta t u_i^{\prime n+1}$ .

This is the one we just derived from writing down all the rows, having all the columns aligned and canceled three of the most important terms. And we get this. We get  $x$  being 1,  $y$  being  $\frac{1}{2} L$  of  $t$ ,  $z$  being  $\frac{1}{2} L$  of  $t$ .

Now, let's try this, and for this, I have to say for now I am going to go back cowardly to a linear equation. And we are going to be talking about how to do the same with nonlinear equations. I think three lectures from now or something like that, we are going to be using something called the Newton-Raphson. But for now, let's stick to linear equations.

What we have is we have  $u'$  is equal to something times  $u$ .  $u$  is 2 by 1 vector.  $u'$  is going to be 1, 0, 0, minus  $g$  over  $L$  times  $u$ . Why is that true? You just need to look at the code.  $du/dt$  equal to  $\theta \cdot$ , which is  $u^2$ . And  $\theta \cdot$ , which is  $u_1$  times minus  $g$  over  $L$ .

This is the same as saying  $du/dt$  equal to  $u^2$ ,  $u_1$  times minus  $g$  over  $L$ . I can basically rephrase all of this by this. Or all the stuff I just commented out, made green is just this. Agreed?

And this is just saying that  $du/dt$  is equal to this matrix times  $u$ . And this is going back and forth between the formula and the code, the formula and the code. Doing this is the same as a matrix multiplication.

Now, the same thing happens for  $u_i^{n+1}$ . Agreed? So now if you plot this into the equation, we removed all the primes and replaced all the primes with  $u_i$  and  $u_i^{n+1}$ . We can actually collect the terms. We can say  $u_i^{n+1} = u_i^n + \Delta t$  times  $u_i^{\prime n}$ , which I am going to write like this plus-- and switching to red, and you're going to see why I am doing this--  $u_i^{n+1}$ . This is really important. It is  $u_i^{n+1}$ .

Why am I using different colors? What is the difference between the blue terms and the red terms?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Yes.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** The current step versus next step. The known versus unknown. When I have the  $i$ -th step when I want to go to the next step,  $u_{i+1}$  is unknown.  $u_i$  is known.

All right. So what I can do is I can rearrange and say that something times  $u_{i+1}$  is equal to something times  $u_i$ . Can somebody tell me what these two matrices are? If I move all the red terms to the left and let me actually write it as blue because [? there-- ?] and all the blue terms to the right, what do I get? What is the red matrix? What is the blue matrix?

The red matrix.

**AUDIENCE:** [INAUDIBLE]

**QIQI WANG:** 1.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Great. The two 1's are because  $u_i$  can be represented as identity matrix times  $u_i$ . Identity matrix is 1, 0, 0, 1. Now, these two terms are basically this matrix times  $\Delta t$  over 2. Any questions about this matrix? I mean, this is just combining these two terms.

How about the red one?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** 1.

**AUDIENCE:** [INAUDIBLE]

**QIQI WANG:** Negative  $\Delta t$  over 2, yeah. Exactly. It is basically taking this matrix takes the negative sign. And here?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** 1. Great. So to implement the scheme, we need this. We need to basically multiply this matrix with  $u_i$  and invert this matrix. All right. Clear? I'm going to make another file. I'm going to call it - this scheme by the way, the best one step implicit schemes you guys invented is called the trapezoidal rule. Not  $\Delta t$ ,  $t$ . Sorry-- dot  $m$ . Yes.

Oh, I should be copying. this. Trapezoidal rule is a one-step scheme, is a single-step scheme, so I don't need to do any forward Euler or anything like that. All right. And I'm going to be

doing  $t$  equal to 0 to  $dt$  equal to  $t$ . [? I'm ?] going to type  $m$  first.

What I'm going to do is I'm going to talk to construct two matrices. One matrix, I call it  $A$ . Let me just call it red matrix and the blue matrix. The red matrix is  $1 - \frac{\Delta t}{2L}$ ,  $\frac{\Delta t}{2L}$ . I'm just going to type it.  $1 - \frac{\Delta t}{2L}$ ,  $\frac{\Delta t}{2L}$ . I think that I get a minus here. That will be over 2 times  $g$  over  $L$ . and 1.

The blue matrix is roughly the same, but there is a minus sign here, and there is a plus sign here. All right. Then what do I do next?  $u$  is equal to what?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Inverse of the red matrix times blue matrix times  $u_0$ . So actually here I need to do  $u_0$  transpose because  $u_0$  is a row matrix. I need to make it a column matrix, and the whole thing has to be transposed again. All right. And I'm going to be doing history is equal to history is 0 and  $u_0$  equal to [?  $u$ . ?]

Is it clear what I did? Anybody have any comments on this? Is it a good way of doing matrix inverse?

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Yes.  $u$  is  $u_i$  plus 1.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Yes.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Good point.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Inverse red matrix is equal to inverse red matrix.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:** Sure. I can also combine blue matrix.

**AUDIENCE:** [INAUDIBLE].

**QIQI WANG:**

Sure. A good suggestion. I'm just going to be  $u$  equal to  $a$  times-- all right. And let me actually make  $u_0$  a column matrix by replacing this with this. All right. So I don't need to do all of this. Here, instead of when I recall the history, I transpose it. Great. Let's see how it works.

Trapezoidal. Yeah. Let's plot this again. It works perfectly. All right. So good job, guys. The scheme you came up with works.

That's it for today. And so next class, what I want to do is bring your own computer because we're going to be doing something-- next class, not I'm going to code up things like this. Instead I'm going to ask you to code up something like this, and you are going to be working class and hopefully, show things on this screen.