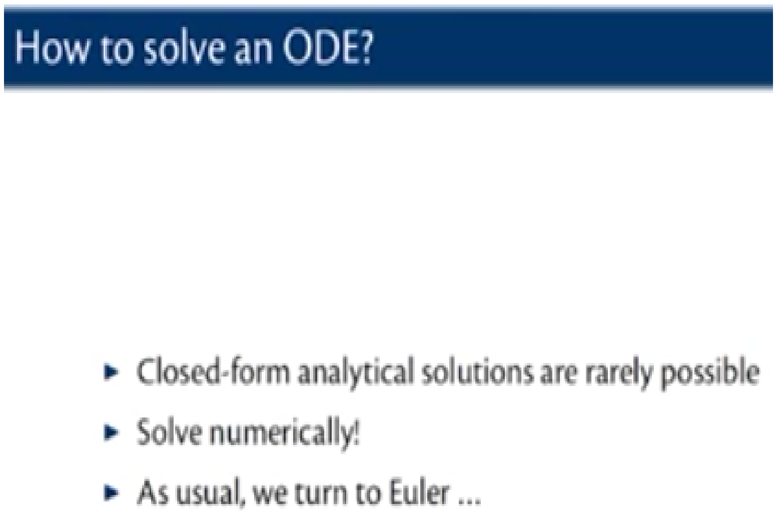
Computational Systems Biology

Karthik Raman
Department of Biotechnology
Indian Institute of Technology – Madras

Lecture 37
Introduction to Dynamic modelling

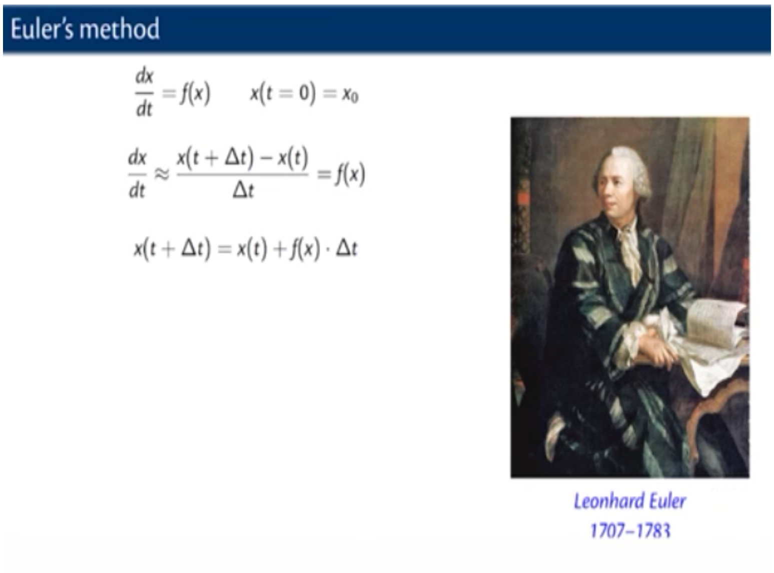
In this video, let us study how to solve these ordinary differential equations generated through any of the models, such as Michaelis-Menten or Hill and so on and typically you will see that you cannot solve these analytically, but you have to go in for numerical solutions based on different kinds of methods.

(Refer Slide Time: 00:31)



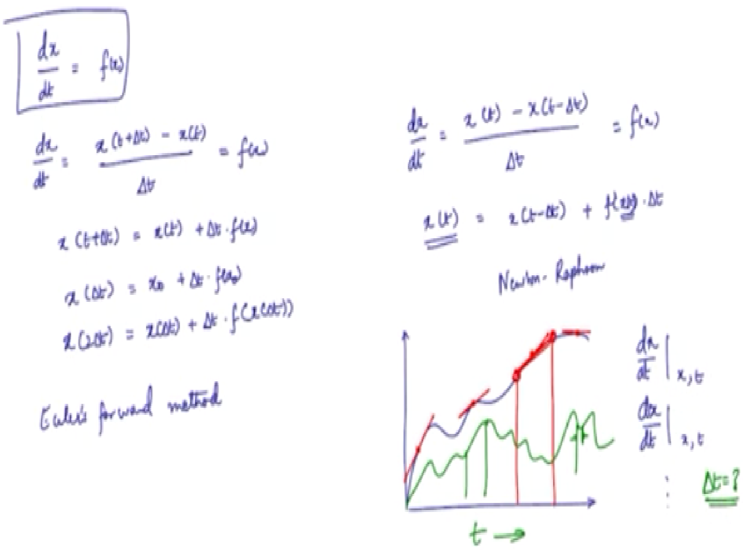
So how do you solve differential equations numerically, as usual we will go back to Euler, so in every module, we will have to pay our respects to Euler.

(Refer Slide Time: 00:41)



So what did Euler do, he came up with a method for integrating equations. So let us say dx/dt is f(x) and x at t=0 is x0, you can also write essentially dx/dt you can write as x(t+delta t)-xt/delta t. So on simplification, this will give you x(t+delta t) is x(t)+f(x)*delta t.

(Refer Slide Time: 01:05)

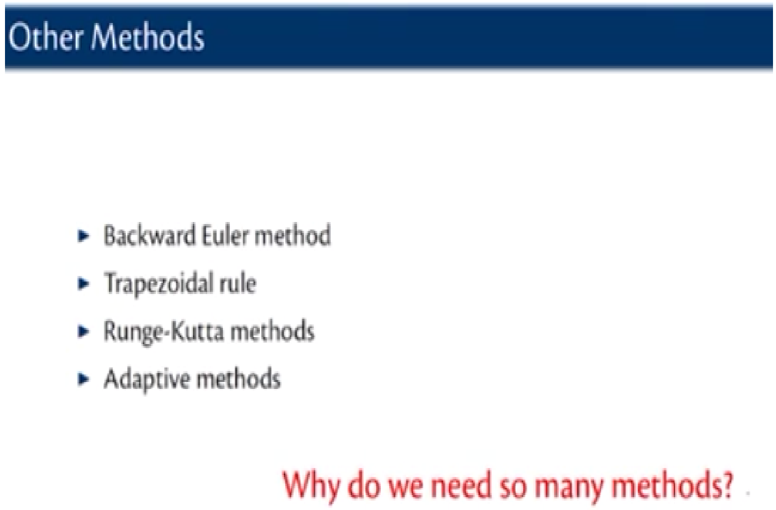


Let us just write it out, we have our rate equation which is basically dx/dt is some f(x), or you can say that dx/dt is x(t+delta t)-x(t)/delta t = f(x), which means x(t+delta t)=x(t)+delta t*f(x). If t=0, x(delta t) is x0+delta t*f(x), x(2delta t) will be x(delta t)+delta t*f(x), and you can just extend this. So this is called, Euler's forward method, there is also Euler's backward method, what do you think that is, it just have a small variation here.

So instead you say dx/dt is x(t)-x(t-delta t)/delta t, so now the interesting part is x(t) is x(t-delta t)+f(x)*delta t. You can actually write it in the same fashion, here is a very analogue equation, except that this now becomes an implicit equation. Why is it an implicit equation?

So you have to now solve an equation in x(t), you have to iteratively solve it, maybe you will use something like Newton–Raphson to solve this equation even.

(Refer Slide Time: 03:56)



So that is the backward Euler method, there are trapezoidal rule, there are Runge-Kutta methods and there are adaptive methods. There are many, many methods to solve ODEs, but essentially all of these methods use a particular logic, you have dx/dt is f(x), and what you want to find out? You want to find out x(t). In other words, what do you have and what is that you want to reconstruct.

You have the slope for the derivative at different points and you want to construct the original function, is that right. So you know the value of dx/dt at x0, dx/dt at some x1, and so on. Based on this you want to construct, some t or you can basically write as, see dx/dt is going to be f(x, t) essentially, so it is some x, t. Here this is the slope, here this is the slope, here this is the slope, here this is the slope.

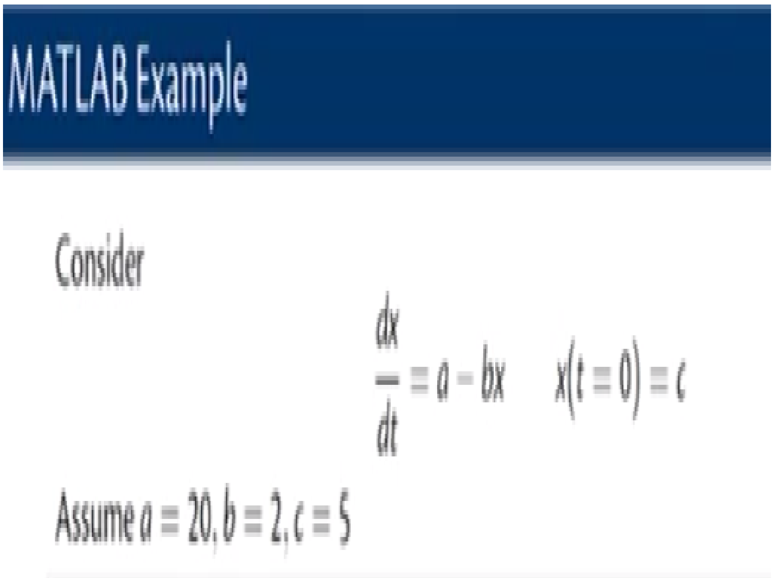
Based on these slopes, we have to reconstruct the original curve. So what do you do, you basically assume that, the slope is constant over a small range and then you use some value of this slope for this entire range. So trapezoidal will basically take half of these 2. Runge-Kutta does something fancier, it will take twice of this and 4 times of this and divide by 6 or something like that.

So each, Runge-Kutta methods themselves come from different orders and so on. Why this is useful is that when you have a function that can behave little crazy, you have to have an adaptive method, meaning what is your delta t, this is your t. What is the time step that you take? So at this point you can afford taking a nice time step, at this point you can only take this much of a time step, and a good adaptive solver will choose your delta t.

You will look at more of this during the lab session, because it will give you a better picture of why we need to go adaptive and things like that. So we need so many methods because your function can be very complex, and you have to adopt to it, that you cannot have 1 single delta t across the entire length of the function and so on or you could have an aggressively small delta t to be safe, but that is again going to increase you competition.

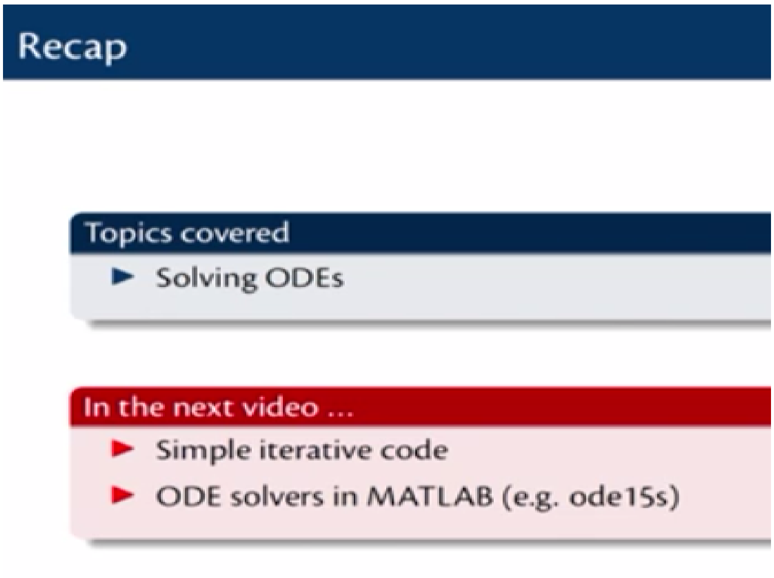
So you need to use a good solver, if you want to solve any of these differential equation, it becomes quite a trick as we will see.

(Refer Slide Time: 07:46)



So this we will reserve for the afternoon session, we will try to do a small, we will have a lab where in we will look at how do we solve these kinds of ODEs using MATLAB. You have a function dx/dt is some f(x, t), what do you mean, no, no you will be able to, you will compute the slope at different places right, and then you use that essentially trace out this entire function. If it were a line, one (x, t) is enough to, but when you have like a more complex function you need to have many more values of (x and t).

(Refer Slide Time: 08:24)



So in this video, you had a brief overview of how we go about solving ODEs and how you can use Euler's forward method or backward method to solve ODEs and so on and in the next video we will go on with the lab, wherein we write some simple iterative code to solve these ODEs, and also use ODE solvers in MATLAB, something like ODE 15s.