Chapter 2

Numerical Methods for ODEs

Euler method

We used JSim to simulate the models in the previous chapter. Does JSim solve the ODEs by integration, the way we do with pen and paper? Not really. You must have noticed that the result of a simulation is a set of numerical values rather than a function written symbolically.

We do not need the function either. In our second model, we wanted to know how the infection spreads with time, and the simulation by JSim gave us a plot showing that.

We could have got the same result analytically, too. However, for many models, we will not be able to solve the differential equations analytically. For some other models, the analytical method is complicated and time-consuming. For these models, we need a method that would give us an approximate but workable solution.

Numerical methods for ODEs do the job. JSim uses numerical methods to simulate a model. There are several numerical methods for solving ODEs. Software for simulations often allow the user to choose a particular algorithm. You will also find library functions or modules for the numerical solution of ODEs in most modern programming languages.

We will learn the Euler method to understand the key concepts for the numerical solution to ODEs. This method is simple, and for easy problems, you can implement it by paper and pen.

Suppose we have to integrate the following ODE:

$$\frac{dx}{dt} = f\left(x,t\right) \tag{1}$$

Consider at time t the value of x is $x(t)$. For a tiny change in time, Δt, x changes by a very small value Δx. By approximation,

$$\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t} = \frac{x(t+\Delta t) - x(t)}{\Delta t} \tag{2}$$

Rearranging the terms, we get

$$x(t+\Delta t) = x(t) + \frac{dx}{dt}\Delta t = x(t) + f(x,t)\Delta t \tag{3}$$

Now we take a specific ODE to understand the implication of Equation 3,

$$\frac{dx}{dt} = rx(1-x) \tag{4}$$

For this ODE, we can write Equation 3 as

$$x(t+\Delta t) = x(t) + rx(1-x)\Delta t \tag{5}$$

We can use Equation 5 to calculate x at $(t+\Delta t)$ using the value of x at t. Numerical integration starts at $t = 0$ with $x = x_0$. Taking a very small Δt, we calculate $rx(1\text{-}x)\Delta t$ at this position and increase x by that value. The time is also incremented by Δt. This process is repeated until we reach the final time point. Δt is called the timestep or step size.

Let us use this method to solve the ODE in Equation 4. To do so, we need to specify the numerical values of r and Δt. Let r = 0.5 and Δt = 0.1. We also have to define the initial value of x. Let, at t = 0, x = 0.02. We will do the calculations using a table:

t	x	Δx = rx(1-x)Δt	x + Δx
0	0.02	0.00098	0.02098
0.1	0.02098	0.001026992	0.022007
0.2	0.022007	0.001076134	0.023083
0.3	0.023083	0.001127515	0.024211
...	...	...	...
...	...	...	...
...	...	...	...
19.8	0.997677	0.000115898	0.997793
19.9	0.997793	0.00011013	0.997903
20	0.997903		

x(t = 0.3)
= x(t = 0.2) + Δx(t = 0.2)

The blue line in Figure 1 shows the data from this table. The analytical solution of an ODE is the exact or correct result. The pink line is the function $x = \frac{1}{1+\left(\frac{1}{x_0}-1\right)e^{-rt}}$ that we obtained by integrating the differential equation in Equation 4. The pink and blue lines are almost overlapping in the plot. That shows that our estimation using the Euler method is correct.

The grey line is obtained by implementing the Euler method but with $\Delta t = 1$. This line has a substantial deviation from the analytical result. That means, for large timestep, the result of the Euler method deviates from the exact result.

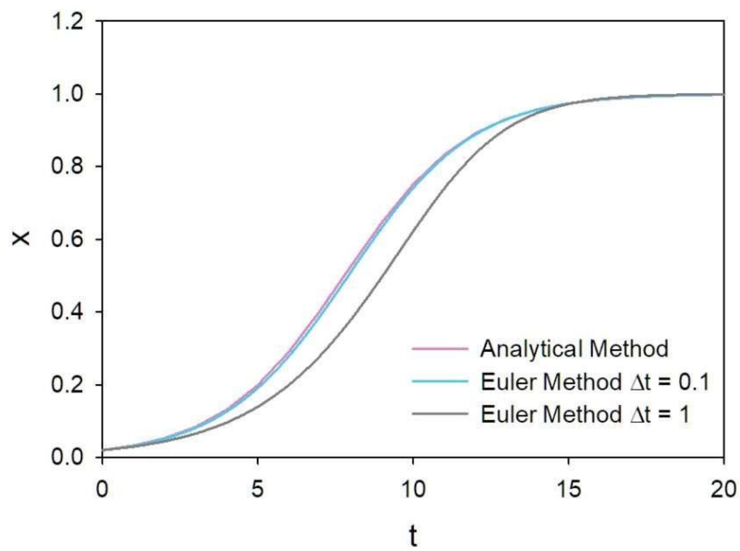

Figure 1: Comparison of the data obtained by the Euler method for different Δt with the result of the analytical method. Numerical solutions for Equation 4, with $r = 0.5$, are shown here. The initial condition is at $t = 0$, $x = 0.02$.

In an ODE, both the dependent and independent variables are continuous. However, a numerical method, like the Euler method, calculates the value of the dependent variable at discrete time points with an interval of Δt. Note that Δt is tiny but not infinitesimal like dt. That leads to the error in the numerical solution.

This problem is explained in Figure 2. The blue curve in the figure shows the exact solution of a differential equation. At time t, the value of x is $x(t)$. We want to calculate the value of x after an interval Δt. As shown in the figure, the exact value of x at $t + \Delta t$ is $x(t + \Delta t)$.

For numerical estimation, we draw a tangent to the point $(t, x(t))$. The pink line is the tangent, and it has a slope of dx/dt at that point. According to Equation 3, we should follow this tangent to reach the value of x at $t + \Delta t$. As shown in the figure, following the tangent, we reach x', which is away from the correct value $x(t + \Delta t)$.

The difference between x' and $x(t + \Delta t)$ will decrease with smaller Δt. Therefore, we have to make the timestep as small as possible for our calculation. However, a smaller step size increases the time of computation. Therefore, one has to choose the step size judiciously.

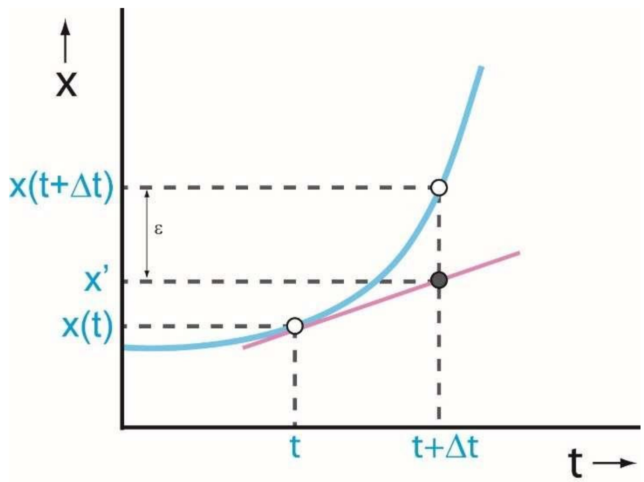

Figure 2: Graphical representation of the Euler method. The grey colored circle represents the numerically estimated x at $t+\Delta t$. ε is the error in our estimation.

Runge-Kutta Method

You may skip Runge-Kutta method for BT612. But I will suggest you to read it.

In the Euler method, we followed the tangent at $x(t)$ to reach x at $t+\Delta t$. However, we reached a point away from the correct one. In Figure 2, we notice that if we had moved along another line with a different slope, we could have reached close to the correct result. However, how can we decide the slope of that line?

To solve this problem, we take the help of the Taylor series.

Consider a generalized ODE,

$$\frac{dx}{dt} = \dot{x} = f(x,t)$$

Using the Taylor series, we can write

$$x(t+\Delta t) = x(t) + \dot{x}\Delta t + \ddot{x}\frac{\Delta t^2}{2} + O(\Delta t^3) \tag{6}$$

$\dot{x}$, and $\ddot{x}$ are the first, and second derivatives of x, respectively. The first derivative $\dot{x}$ is the slope of the pink line in Figure 2.

In Equation 4, the expansion is up to the second order term, and we neglected the higher-order terms. If we consider up to the first order term, Equation 6 will be the same as Equation 3.

Let us rewrite Equation 6,

$$x(t+\Delta t) = x(t) + \Delta t\left(\dot{x} + \ddot{x}\frac{\Delta t}{2}\right) \tag{7}$$

Following Equation 3, we can say that $\left(\dot{x} + \ddot{x}\frac{\Delta t}{2}\right)$ represents some sort of slope of a line. Imagine that this slope is equal to $\bar{m}$. Here we define,

$$\bar{m} = (w_1 m_1 + w_2 m_2) \tag{8}$$

m_1, and m_2 are slopes at two points in the interval $[t, t+\Delta t]$. w_1 and w_2 are corresponding weights.

Following Equation 7, we write,

$$x(t+\Delta t) = x(t) + \Delta t(w_1 m_1 + w_2 m_2) \tag{9}$$

We have to calculate these slopes and weights. However, this system is underdetermined, and we can have a family of solutions. Here, we will discuss a particular solution:

$$w_1 = w_2 = \frac{1}{2}$$
$$m_1 = f\left(x(t), t\right)$$
$$m_2 = f\left(x(t+\Delta t), t+\Delta t\right) \qquad \text{here, } x(t+\Delta t) = x(t) + m_1 \Delta t$$

This solution is explained graphically in Figure 3. At time *t*, we calculate $m_1 = \frac{dx}{dt} = f\left(x(t), t\right)$. That is the slope of the tangent at that point, shown by the green line in Figure 3. Following that line, we reach the point at time $t+\Delta t$. The value of *x* at this point is $x(t+\Delta t) = x(t) + m_1 \Delta t$. We calculate the slope m_2 at this position using the same ODE; $m_2 = \frac{dx}{dt} = f\left(x(t+\Delta t), t+\Delta t\right)$. The purple line in Figure 3 has a slope of m_2. Now take the average of m_1 and m_2 and draw the pink line at the initial position ($x(t)$, t) with this average slope. Following this line, we reach *x*' the estimated value of *x* at time $t+\Delta t$. Note that this estimate is better than the one we made using the Euler method (Compare Figure 2 and Figure 3). This method of numerical solution of an ODE is called the Heun's method.

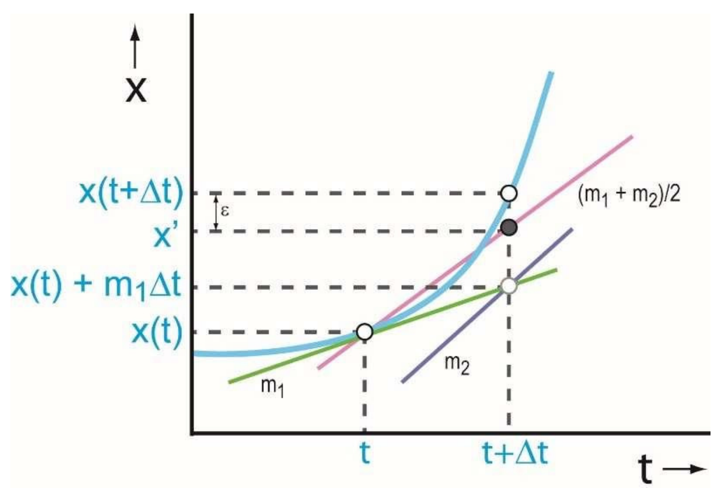

Figure 3: Numerical solution of an ODE using Heun's method. The grey colored filled circle represents the numerically estimated *x* at $t+\Delta t$. ε is the error in our estimation.

We will now include the third and fourth order terms of the Taylor series in Equation 6. That should increase the accuracy of our numerical solution. Using the Taylor series, we write

$$x(t+\Delta t)=x(t)+\dot{x}\Delta t+\ddot{x}\frac{\Delta t^2}{2}+\dddot{x}\frac{\Delta t^3}{6}+\ddddot{x}\frac{\Delta t^4}{24}+O\left(\Delta t^5\right) \tag{10}$$

$\dot{x},\ddot{x},\dddot{x}$, and $\ddddot{x}$ are first, second, third, and fourth derivatives of *x*, respectively.

We rewrite Equation 10 as,

$$x(t+\Delta t)=x(t)+\Delta t\left(\dot{x}+\ddot{x}\frac{\Delta t}{2}+\dddot{x}\frac{\Delta t^2}{6}+\ddddot{x}\frac{\Delta t^3}{24}\right) \tag{11}$$

Similar to Equation 9, we can assume that there is an average slope $\bar{m}=\left(w_1m_1+w_2m_2+w_3m_3+w_4m_4\right)$ such that

$$x(t+\Delta t)=x(t)+\Delta t\left(w_1m_1+w_2m_2+w_3m_3+w_4m_4\right) \tag{12}$$

m_1, m_2, m_3, and m_4 are slopes at four points in the interval $[t, t+\Delta t]$. w_1, w_2, w_3, and w_4 are weights. This system is also underdetermined, and one solution is,

$$\begin{aligned}
&w_1=\frac{1}{6},\ w_2=\frac{1}{3},\ w_3=\frac{1}{3},\ w_4=\frac{1}{6}\\
&m_1=f\left(x(t),t\right)\\
&m_2=f\left(x(t)+m_1\frac{\Delta t}{2},t+\frac{\Delta t}{2}\right)\\
&m_3=f\left(x(t)+m_2\frac{\Delta t}{2},t+\frac{\Delta t}{2}\right)\\
&m_4=f\left(x(t)+m_3\Delta t,t+\Delta t\right)
\end{aligned}$$

Therefore, *x* at $t+\Delta t$ is given by,

$$\begin{aligned}
x(t+\Delta t)&=x(t)+\left(\frac{1}{6}m_1+\frac{1}{3}m_2+\frac{1}{3}m_3+\frac{1}{6}m_4\right)\Delta t\\
&=x(t)+\frac{1}{6}\left(m_1+2m_2+2m_3+m_4\right)\Delta t
\end{aligned}$$

This method is called the fourth order Runge-Kutta (RK4) method. It was formulated more than a hundred years back, and this method and its variant are still widely used for the numerical solution of ODEs.

We solved the ODE in Equation 4 using the RK4 method. The result is shown in Figure 4. Notice that the numerical solutions are almost the same as the analytical solution, even when $\Delta t = 1$. That shows that the RK4 method has better accuracy than the Euler method.

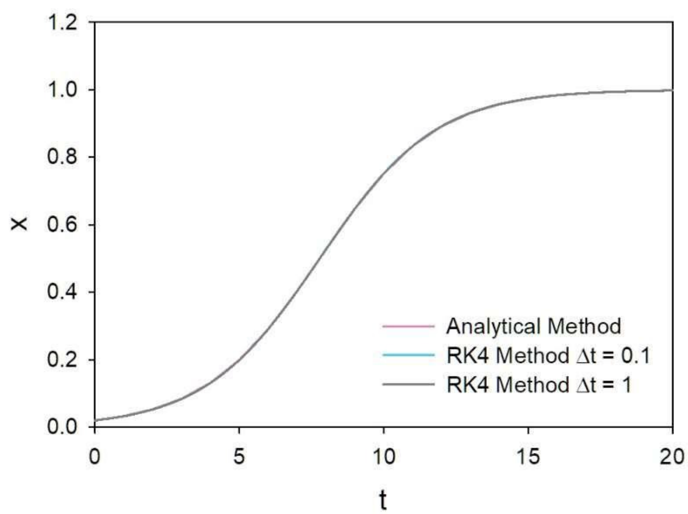

Figure 4: Comparison of the data obtained by the fourth order Runge-Kutta method for different Δt with the result of the analytical method. Numerical solutions for Equation 4, with $r = 0.5$, are shown here. The initial condition is at $t = 0$, $x = 0.02$.

Adaptive timesteps

The Runge-Kutta method produced an accurate result for Equation 4, even for $\Delta t = 1$. That does not mean that we will achieve accuracy for any arbitrary large timestep. The Runge-Kutta method replaced one slope with a weighted average of multiple slopes. It has not resolved the issue of step size that we explained using Figure 2. The question is still open – how we decide the optimum step size (Δt) for a given ODE?

The answer is in using a timestep that changes as we progress recursively in our numerical algorithm. For a nonlinear system, a fixed timestep is not useful. That is explained in Figure 5.

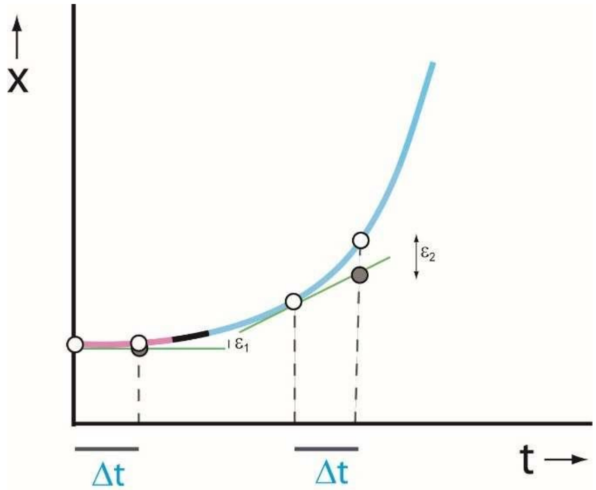

Figure 5: Effect of step size on the error in numerical solution at different regions of a nonlinear function. Grey filled circles are estimated values of x at different positions. ε_1, and ε_2 are errors. $\varepsilon_1 << \varepsilon_2$.

The nonlinear function shown in Figure 5 has different slopes at different positions. In the pink region, the slope is very shallow. While in the blue region, x changes very fast with t. In the pink region, we can use a large Δt. However, in the blue region, the same step size will lead to high error in our estimate. It will be better to use a very small step size in the blue region. So, we have to keep on changing the size of the timestep as we move along the function.

However, while numerically solving an ODE, we do not know the underlying function. Then how can we vary the step size as we perform the iterations of our numerical method? An approximate but ingenious method helps us to circumvent this problem. Rather than using one method, we will use two methods for solving the ODE; one more accurate than the other. If our choice of the step size is correct, then the results from both the methods will be similar.

For example, the well known Runge-Kutta-Felhberg (RKF45) method uses the fourth order and fifth order Runge-Kutta methods. In principle, the fifth order method is more accurate than the other. Suppose at a particular iteration, the estimated x at t+Δt is x_4 and x_5 for the fourth and fifth order method, respectively. If the absolute difference between x_4 and x_5 is zero or very small, we can assume that our choice of the step size (Δt) at this iteration is correct. Otherwise, we have to make Δt smaller.

Here we explain this approach stepwise:

1. Set the values of upper tolerance (H_{max}) and lower tolerance (H_{min}).
2. Set the value of Δt.
3. At time t,
    a) Use Method 1 (less accurate) to calculate x at $t+\Delta t$. Call this x_1.
    b) Use Method 2 (more accurate) to calculate x at $t+\Delta t$. Call this x_2.
4. Calculate the local error, $\varepsilon = |x_2 - x_1|$
5. If $\varepsilon > H_{max}$, reject the calculation, reduce Δt to $\Delta t/2$, and go back to step 3 to do the calculations with the reduced Δt.
6. If $\varepsilon < H_{min}$, accept the result of Method 2, change Δt to $2\Delta t$, and iterate the algorithm for the next time point with the updated Δt.
7. Otherwise, accept the result of Method 2, and iterate the algorithm for the next time point using the same Δt.

All modern ODE solvers use some sort of adaptive step size algorithm. JSim allows the use RKF45. While using programming languages like C to solve an ODE-based model, you can use library functions for RKF45 and other algorithms.

Exercises

1. For the ODE $\frac{dy}{dx} = y\left(1 - \frac{y}{10}\right)$, use the Euler method to calculate the value of y at x = 0.5. At x = 0, y = 0.1. Consider Δx = 0.1.

2. The following ODE represents the temporal dynamics of a protein that autoregulates its expression.

$$\frac{dP}{dt} = 0.1 + 0.1\frac{P^2}{4 + P^2} - 0.1P$$

   Solve this ODE using the Euler method and plot the P vs. t graph. The initial condition is P = 0 at t = 0. The final time point is 60. Consider Δt = 0.1.
   You can write a program to solve this problem. Otherwise, use any spreadsheet software like Microsoft Excel and implement the Euler method using the tabular approach shown in this chapter.

3. Solve the ODE in the previous question again using the Euler method, but this time with a larger timestep Δt = 0.1. Compare the data of this solution with the previous one graphically.

4. Suppose we are using an algorithm for the numerical solution of an ODE with an adaptive step size. The maximum value of the tolerance is 0.001. How the accuracy and time for computation would change if we reduce this tolerance to 0.000001 to solve the same ODE?

5. The second order Runge-Kutta method is derived using the Taylor series of a function. For the following ODE, write the first three terms of the Taylor series that will be used for the Runga-Kutta method.

$$\frac{dx}{dt} = \frac{vx}{K + (1 - x)}$$