## Note 1

- In this plot, we learn about the Euler method for numerical integration.

- It is the most basic explicit method for numerically solving ordinary differential equations.

- Although many other methods with higher accuracy have been proposed, the simplicity of the Euler method makes it ideal when studying the subject for the first time.

- We will use it to simulate the motions of Brownian particles in the remaining part of this course.

## Note 2

- Let us consider an ordinary differential equation, shown here in Eq.(1).

- We want to know the temporal values of y for any time t, under the initial condition $y=y\_0$ at $t=t\_0$.

- First, let us write down the formal solutions of the differential equation by integrating Eq.(1) in time, from $t\_0$ to $t$.

- Eq. (2) gives the exact value of $y$ at any given time $t$, starting from the initial condition of $y\_0$ at $t=0$.

- However, an analytical evaluation of the time integrals is only possible for very limited cases.

- In most situations, we must solve these differential equations numerically.

- The Euler method is the simplest method for performing this numerical integration.

## Note 3

- To solve differential equations numerically using computers, we must first discretize the time axis.

- We divide the total time span, from $t\_0$ to $t$, into $N$ equally spaced segments, each describing a short time increment $\Delta t$, and define the following discrete variables.

## Note 4

- Now consider how to perform the integration over a small time increment $\Delta t$, from $t\_i$ to $t\_{i+1}$.

- Using the Taylor expansion, we expand the integrand around $t\_i$, in increasing powers of $τ$, where $τ = t’ - t\_i $.

- For simplicity, we only explicitly write the zero-th order term, proportional to $f\_i$.

- The integrals can be performed analytically, with each term one order in $\Delta t$ higher than the previous one.

## Note 5

- In the limit when the time increment $\Delta t$ goes to 0, we can neglect all terms in Eq.(7) of second order or higher.

- This first order approximation is known as the Euler method, shown here as Eq.(8).

- By repeatedly applying Eq.(8) $N$ times, we can obtain an approximation of $y$ at a given time $t=N \Delta t$.

- This process is schematically shown in Eq.(9).

- This is a typical example of how a computer simulation proceeds.

- The Euler method is a first-order method, which means that the local error is proportional to the square of the step size, and the global error is proportional to the step size.

- Although simple and not very accurate itself, the Euler method serves as the basis to construct more accurate methods.

- From Eqs.(1) and (8), it is straightforward to solve for $f$, and to recognize that the Euler method is a forward difference approximation to the derivative of $y$, as defined in Eq. (10).

## Note 6

- Instead of using the forward difference approximation of the Euler method, we can use a centered difference, to design a higher-order integration method.

- The central difference approximation given by Eq. (11) estimates the derivative of $y$ at step $i$, in terms of the $y$ values at steps $i-1$ and $i+1$.

- Substituting Eq. (11) into Eq. (1), we obtain the difference equation referred to as the Leapfrog method, shown in Eq. (12).

- This integration scheme gives a method that is accurate to second order in $\Delta t$.

- The simulation procedure corresponding to the Leapfrog method is schematically represented in Eqs. (13) and (14).

- In contrast to the Euler method, the difference equation for the Leapfrog method (Eq. (12)) involves values of $y$ and $f$ at distinct times.

- In this case $i$ and $i-1$, respectively.

## Note 7

- Here we have written down the difference equations for a 2nd order Runge-Kutta method.

- The first Eq.(15) represents an intermediate operation, to estimate $y\_{i+1/2}$ and $f\_{i+1/2}$ at the midpoint, using the Euler method.

- The second equation represents the main operation, where $y\_{i+1}$ is calculated using a Leapfrog like method, with $f$ evaluated at the midpoint.

- This particular method is second order in $\Delta t$, like the Leapfrog method, but higher order Runge-Kutte methods can be easily derived.

## Note 8

- In fact, the most popular Runge-Kutta method is the 4th order method we present now.

- The appropriate difference equations for this scheme are given in Eqs. From (18) to (21).

- The first three Eqs. From (18) to (20) represent intermediate operations to estimate tentative values for $y$ and $f$ using the Euler method.

- The 4th equation is the main operation, where $y\_{i+1}$ is calculated using the intermediate values of $f$ obtained previously.

- This method is fourth order in $\Delta t$.

## Note 9

- Let us consider a very simple problem here and try to apply the Euler method to numerically solve it.

- We want to numerically solve the following differential equation and determine $y(t)$ for $t$ from 0 to 10, with initial condition $y=1$ at $t=0$.

- Then we compare it with the analytical solution $y=\exp(-t)$.

- As we already learned in the previous lesson, we should first import the libraries needed to draw the plots.

## Note 10

- First, we define the time increment, and the time range, as well as the number of integration steps to take.

- Next, we create an array for the time values and the y values for the analytical and numerical solutions.

- The numerical solution to $y(t)$ is found by repeated application of Eq.(8).

- Finally, to visualize the error of our method, we plot the ratio of the two solutions.

- Although both curves seem identical, it is clear that for large times, the error increases.