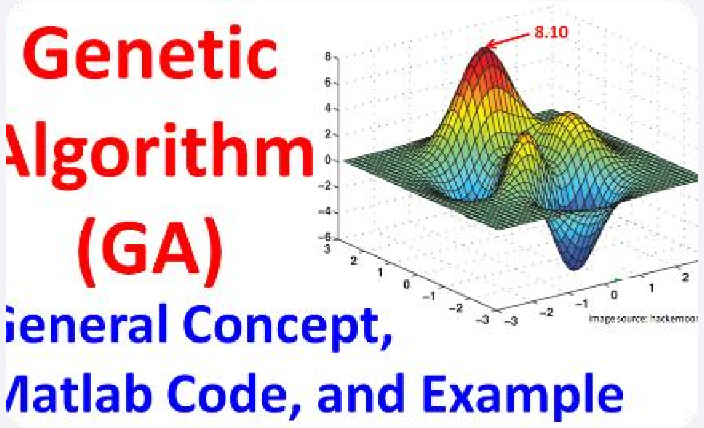
The Art of Numerical Solutions

$$e^{i\pi}+1=0$$



Aesthetics in Mathematics

Explore the profound beauty and elegance found in numerical solutions of differential equations, where art and mathematics intertwine.

Advancing Scientific Research

Learn how numerical methods like Euler's and Runge-Kutta contribute to scientific discoveries and enable simulations in various fields.

The Role of Algorithms

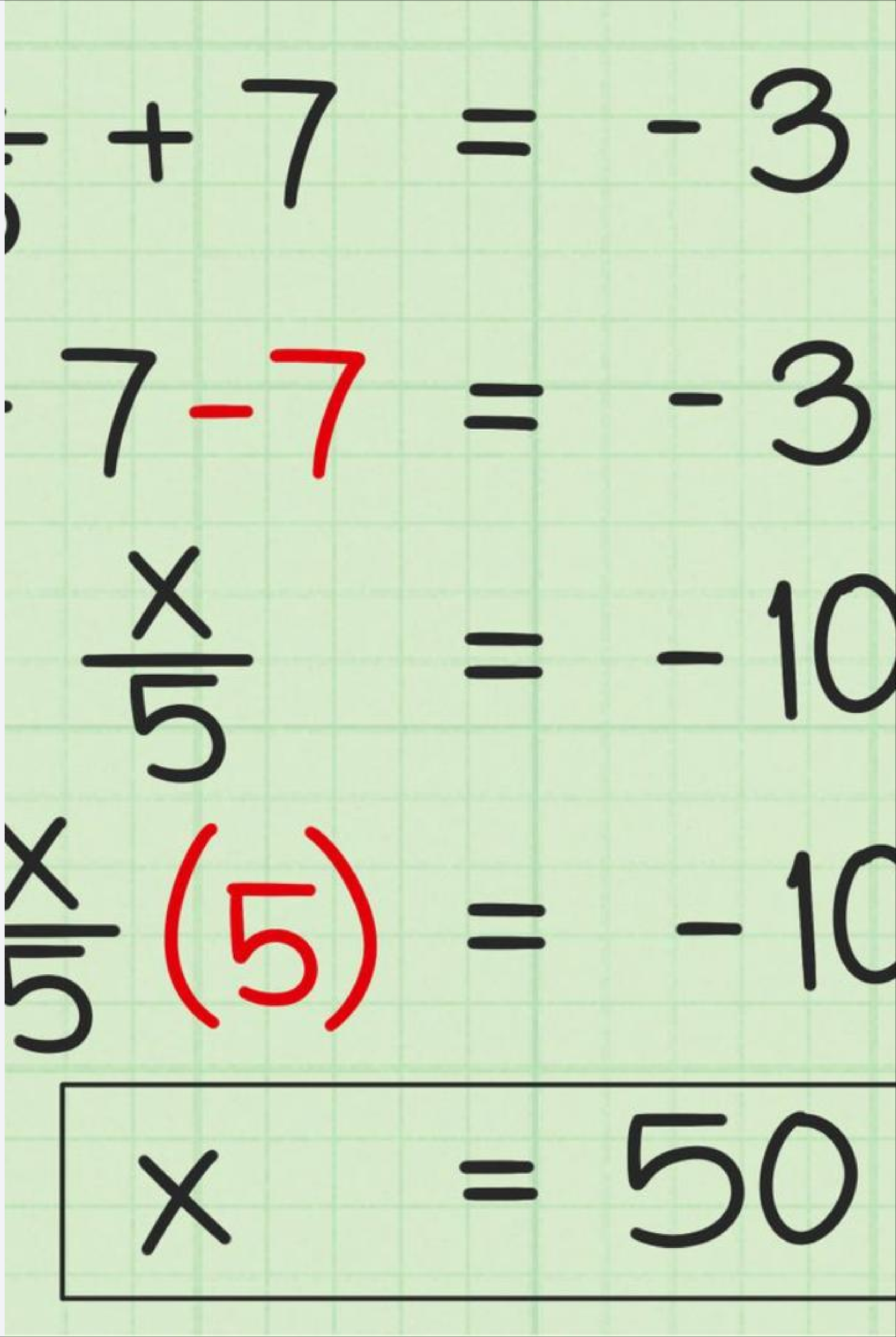
Delve into the algorithms used to implement Euler's and Runge-Kutta methods, from pseudocode to practical implementations.

Introduction to Ordinary and Partial Differential Equations

Alright, imagine you have a box of crayons, and each crayon has a different color. Now, let's say you want to know how much each color is different from the others. That's a bit like what differentiation is. Calculus makes it possible to understand not just how something changes over time, but how it changes in an instant.

Differential equations are mathematical equations that involve one or more derivatives of an unknown function. They are used to model relationships and describe the behavior of systems in various scientific fields.

by Group 5



BASIC CONCEPTS

Independent Variables & Dependent Variable:

- The dependent variable is the quantity that we are trying to slove for or understand. It's the variable whose value is to be obtained.

- Where as the Independent Variable is not influenced by any other variables, It is the variable that we have control over and can manipulate.
  It is the variable to which a value is assigned to.

$$y = x^2 + x$$
$$z = x^2 + y^2$$

$$dy/dt = -2y$$

In this equation, (t) is the dependent variable, representing the amount of a substance at any given time t. The independent variable, t, represents time.
The equation states that the rate of change of y with respect to t is equal to -2 times y. Here, the independent variable (time) influences the dependent variable (amount of substance), causing it to decrease exponentially.

Linear and Non-Linear Differencial Equation

- For Any form of Differential Equation
  $f(x,y,y',y''...y^n)=0$

it is considered linear if it satisfies the following conditions

1. All derivative and dependent variable are of the First order.
2. There does not exist any product form of dependent variable and it's derivative.

Examples of Linear and Non-Linear.

1. $y' + X^2 = 4$
2. $y'' + 3y'^2 + 7y = x$
3. $d^2x/dy^2 + dx/dy + y = x^2$
4. $y'' + \sin y = 0$
5. $yy' + y\sin x = x^2$

ODEs	PDEs
ODEs: Describe how a single variable changes over time	PDEs: Describe how multiple variables change over time and space
ODEs: Involve one independent variable (e.g., time)	PDEs: Involve multiple independent variables (e.g., time and space)
ODEs: Used to model phenomena like temperature change over time	PDEs: Used to model phenomena like heat distribution or fluid mechanics.

dy/dx = f(x, y)

Is an example of an ODE with a single dependent and Independent Variable.

dy/dt = 3x + 2y
dx/dt = x + 2y

In this case we have more that one dependent variable. It is called a system of ODE.

Here is an Example of a PDE with more than one Independent Variable.

2xdz/dx + 5ydz/dy = 1

Ordinary Differential Equations (ODE's)

An ODE involves an unknown function of one variable and its derivatives.
General form: $F(x, y, y', y'', \ldots, y^{(n)}) = 0$, where y' represents the first derivative of y, and n is the order of the highest derivative.

Components of an ODE

- Dependent Variable (y): Represents the unknown function of interest.

- Independent Variable (x): The variable with respect to which differentiation occurs.

Derivatives: Express how the dependent variable changes concerning the independent variable.

Order of ODEs

- First-Order ODEs: Involves only the first derivative of the unknown function.

$$dy/dx = 2x - 3y$$

- Second-Order ODEs: Involves up to the second derivative, and so on.

$$d^2y/dx^2 + 3(dy/dx) + 2y = 0$$

Solving Examples

1. $F = ma$

2. $s(t) = 3t^2 + 2t + 1$

3. Does $x = e^{3t}$ satisfy $d^3x/dt^3 - 9d^2x/dt^2 = 0$

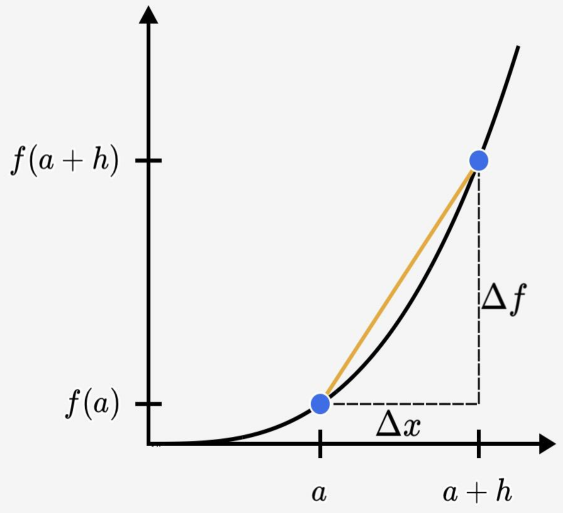
4. $d^2x/dt^2 = 3t + 1$ at init point $x(0) = 2$
   $x'(0) = 3$

Understanding Differential Equations with Euler & Runge-Kutta Methods.

In the world of Differential Equations, Euler and Runge-Kutta methods play a crucial role in making complex problems simpler.

Think of Euler's method like taking small, straightforward steps, offering quick insights. On the other hand, Runge-Kutta adds finesse with its higher accuracy, creating a more precise solution. Together, they form a powerful duo, allowing us to explore a variety of scenarios with ease.

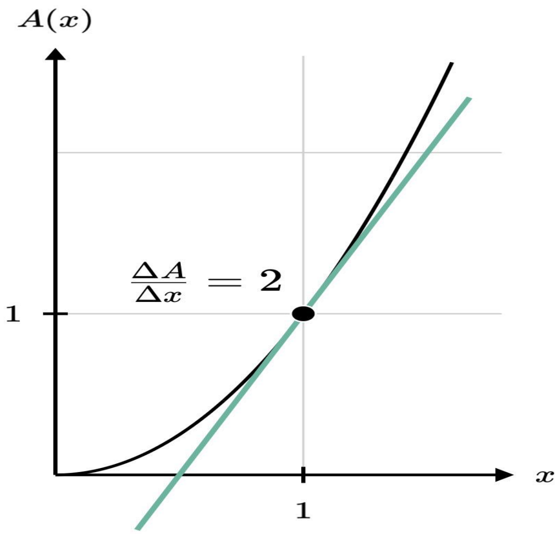
Euler's simplicity gives us quick answers, while Runge-Kutta's accuracy ensures we get the right picture. These methods are like tools that help us uncover the hidden stories in Differential Equations. By working together, they provide a balanced approach, turning complicated problems into solvable puzzles for real-world issues.



Slope is calculated by dividing the vertical change Δf by the horizontal change Δx.

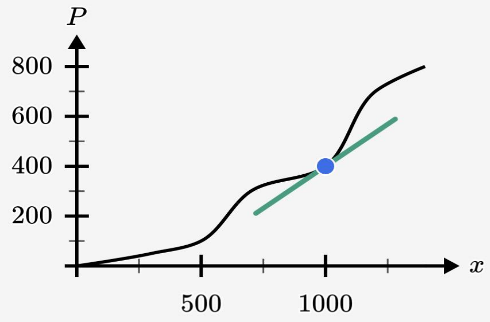
$$\frac{\Delta f}{\Delta x} = \frac{f(a+h) - f(a)}{a+h-a} = \frac{f(a+h) - f(a)}{h}$$

By going from the slope of a secant line to the slope of a tangent line, we move from being able to quantify the average rate of change to being able to quantify the **instantaneous rate of change.**



Discovering that the slope of this tangent line is 2 is equivalent to discovering that at the instant when $x = 1$, the area of the square is growing at precisely $2\,\mathrm{m}^2/\mathrm{s}$.

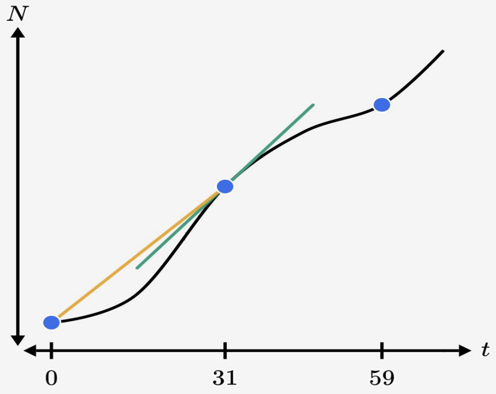
A business models the profits from its new app with P, a function of the number of users, x.

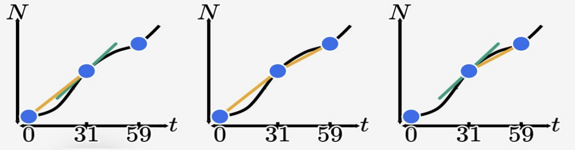


The slope of a line tangent to the graph of $P(x)$ at $x = 1000$ is equal to

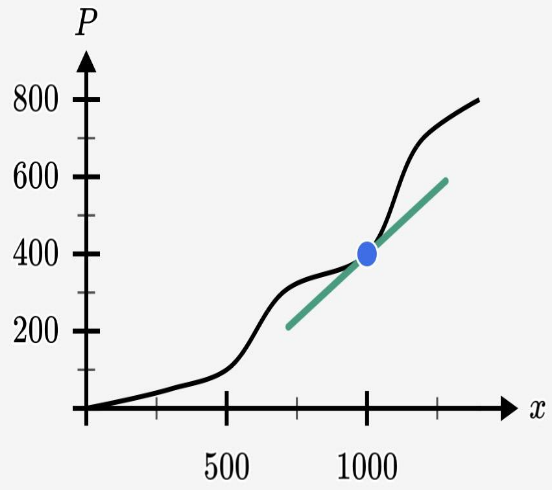
- The average change in profit between $x = 0$ and $x = 1000$
- The average change in profit when the 1000^{th} user joins
- The instantaneous change in profit when the 1000^{th} user joins

The function $N(t)$ gives the growing number of users on a new app over time. The founders of the app wonder whether their average user growth was higher in January or February.



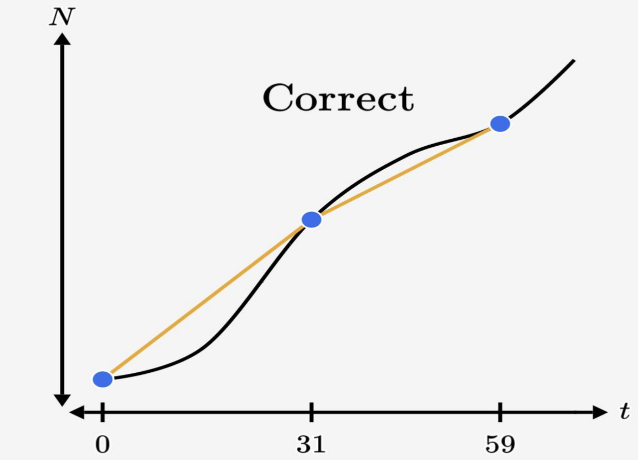


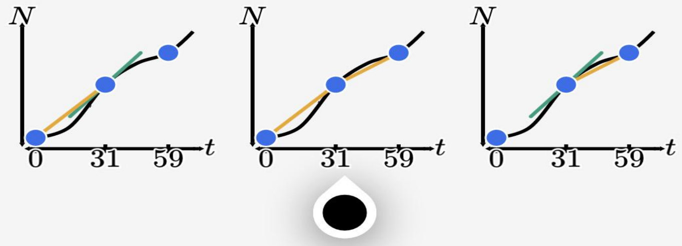
Explanation



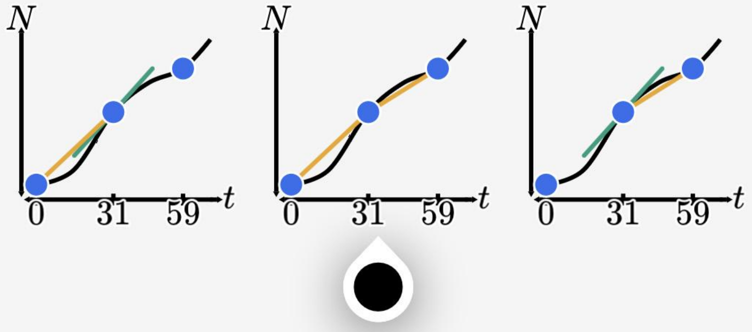
The slope of a line tangent to the graph of a function at a point gives the instantaneous rate of change of the function at that point.

Since P gives the profit of the app, the slope of the line tangent to P at $x = 1000$ gives the instantaneous change in profit when the 1000^{th} user joins.





Since the founders of the app want to know about their average user growth, they will want to know the slope of the secant lines connecting the beginning and end of January and the beginning and end of February. This is represented by the center graph.

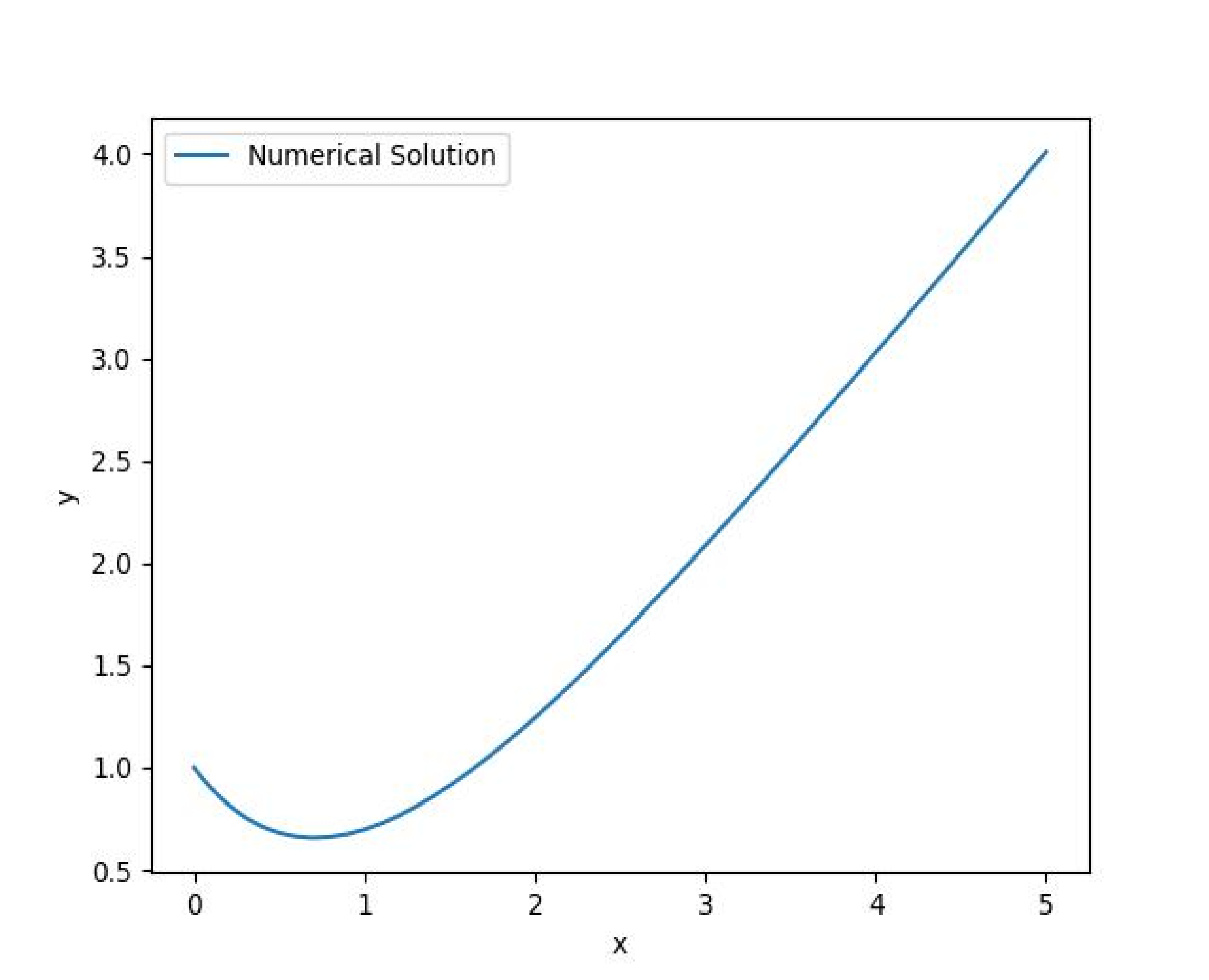


Since the founders of the app want to know about their average user growth, they will want to know the slope of the secant lines connecting the beginning and end of January and the beginning and end of February. This is represented by the center graph.

The other two graphs compare a secant line to a tangent line. This would give the founders information about the average rate of user growth compared to the instantaneous rate of user growth.

Euler's Method for Solving ODE

```
import numpy as np
import matplotlib.pyplot as plt

#  Define ODE and Euler Method
def f(x, y):
    return x - y

def euler_method(func, x0, y0, h, n):
    x_values = [x0 + i * h for i in range(n+1)]
    y_values = [y0]

    for i in range(n):
        y0 += h * func(x_values[i], y0)
        y_values.append(y0)

    return x_values, y_values

# Example Usage
x0, y0 = 0, 1  # Initial conditions
h = 0.1        # Step size
n = 50         # Number of steps

x, y = euler_method(f, x0, y0, h, n)

# Plotting
plt.plot(x, y, label='Numerical Solution')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```



Example 1 For the IVP

$$y' + 2y = 2 - \mathbf{e}^{-4t} \qquad y(0) = 1$$

Use Euler's Method with a step size of $h = 0.1$ to find approximate values of the solution at t = 0.1, 0.2, 0.3, 0.4, and 0.5. Compare them to the exact values of the solution at these points.

Hide Solution ▾

This is a fairly simple linear differential equation so we'll leave it to you to check that the solution is

$$y(t) = 1 + \frac{1}{2}\mathbf{e}^{-4t} - \frac{1}{2}\mathbf{e}^{-2t}$$

In order to use Euler's Method we first need to rewrite the differential equation into the form given in (1).

$$y' = 2 - \mathbf{e}^{-4t} - 2y$$

From this we can see that $f(t, y) = 2 - \mathbf{e}^{-4t} - 2y$. Also note that $t_0 = 0$ and $y_0 = 1$. We can now start doing some computations.

$$\begin{aligned} f_0 &= f(0, 1) = 2 - \mathbf{e}^{-4(0)} - 2(1) = -1 \\ y_1 &= y_0 + h\,f_0 = 1 + (0.1)(-1) = 0.9 \end{aligned}$$

So, the approximation to the solution at $t_1 = 0.1$ is $y_1 = 0.9$.

I'll leave it to you to check the remainder of these computations.

$$\begin{aligned} f_2 &= -0.155264954 & y_3 &= 0.837441500 \\ f_3 &= 0.023922788 & y_4 &= 0.839833779 \\ f_4 &= 0.1184359245 & y_5 &= 0.851677371 \end{aligned}$$

Here's a quick table that gives the approximations as well as the exact value of the solutions at the given points.

Time, t_n	Approximation	Exact	Error
$t_0 = 0$	$y_0 = 1$	$y(0) = 1$	0 %
$t_1 = 0.1$	$y_1 = 0.9$	$y(0.1) = 0.925794646$	2.79 %
$t_2 = 0.2$	$y_2 = 0.852967995$	$y(0.2) = 0.889504459$	4.11 %
$t_3 = 0.3$	$y_3 = 0.837441500$	$y(0.3) = 0.876191288$	4.42 %
$t_4 = 0.4$	$y_4 = 0.839833779$	$y(0.4) = 0.876283777$	4.16 %
$t_5 = 0.5$	$y_5 = 0.851677371$	$y(0.5) = 0.883727921$	3.63 %

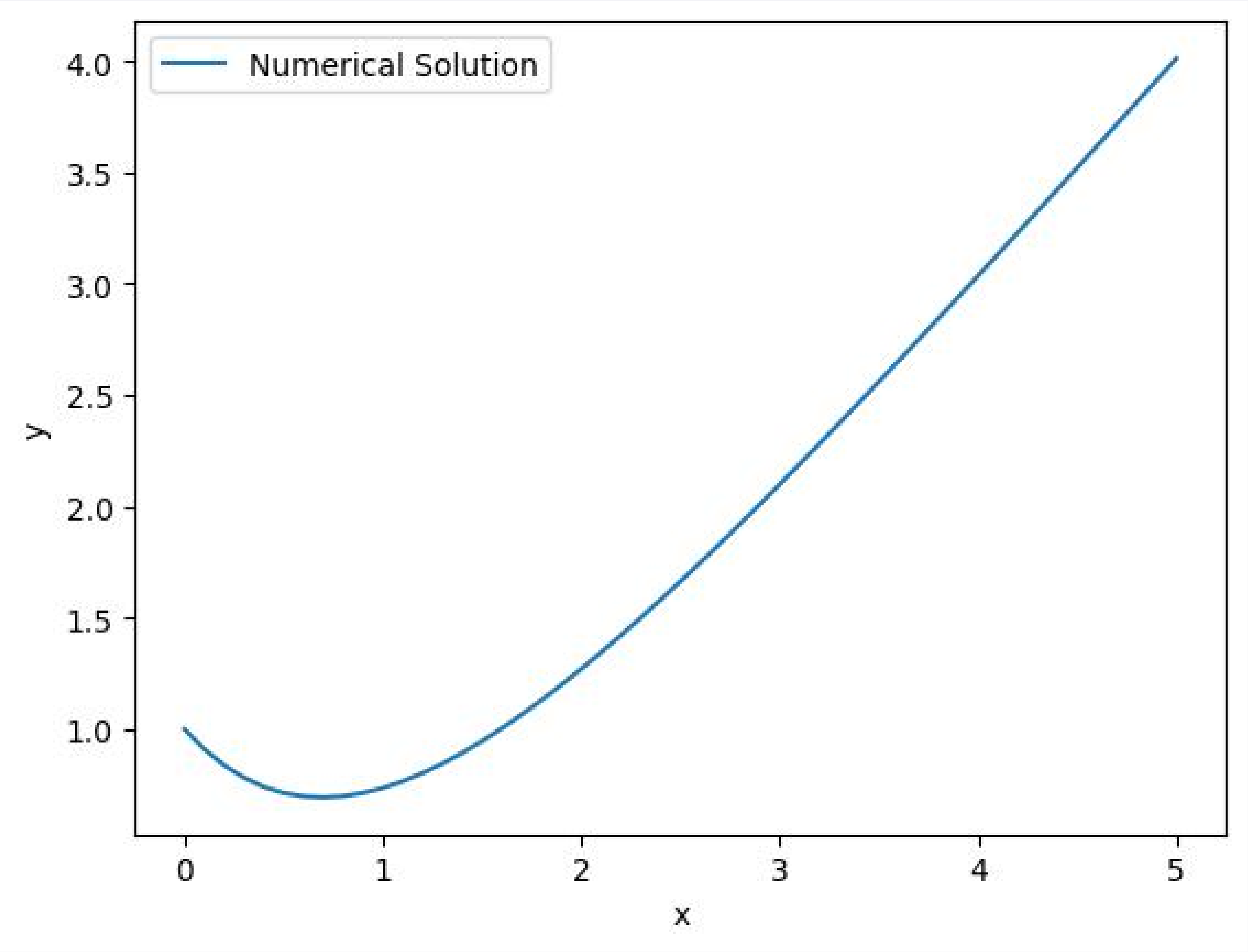
We've also included the error as a percentage. It's often easier to see how well an approximation does if you look at percentages. The formula for this is,

$$\text{percent error} = \frac{|\text{exact} - \text{approximate}|}{\text{exact}} \times 100$$

We used absolute value in the numerator because we really don't care at this point if the approximation is larger or smaller than the exact. We're only interested in how close the two are.

Runge-Kutta Methods for Solving ODE

```
import numpy as np
import matplotlib.pyplot as plt

# Define ODE and Runge-Kutta Method
def f(x, y):
    return x - y

def runge_kutta_method(func, x0, y0, h, n):
    x_values = [x0 + i * h for i in range(n+1)]
    y_values = [y0]

    for i in range(n):
        k1 = h * func(x_values[i], y_values[i])
        k2 = h * func(x_values[i] + h/2, y_values[i] + k1/2)
        k3 = h * func(x_values[i] + h/2, y_values[i] + k2/2)
        k4 = h * func(x_values[i] + h, y_values[i] + k3)
        y0 += (k1 + 2*k2 + 2*k3 + k4) / 6
        y_values.append(y0)

    return x_values, y_values

# Example Usage
x0, y0 = 0, 1 # Initial conditions
h = 0.1        # Step size
n = 50         # Number of steps

x, y = runge_kutta_method(f, x0, y0, h, n)

# Plotting
plt.plot(x, y, label='Numerical Solution')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```



Mastering Runge-Kutta Methods

Variant Selection

Understand the criteria for selecting the most appropriate variant of Runge-Kutta method based on accuracy and computational efficiency.

Comparing Variants

Explore the characteristics, advantages, and limitations of popular Runge-Kutta variants, including RK2, RK3, and RK4.

Optimization Techniques

Discover optimization techniques to enhance the performance of Runge-Kutta methods and achieve more accurate solutions.

Conclusion

As our journey through differential equations and numerical methods comes to an end, we hope you now appreciate the power and versatility of Euler's method and various Runge-Kutta methods. Keep exploring and applying these concepts to solve complex problems in the realm of ODEs. Thank you for joining us!