Final Project

Sarah Eid	S22107757
Judy Abuquta	S22107883
Abeer Hussain	S22107714

Title of the Project:

Numerical Solution of Initial Value Problems Using Euler, Taylor, and Runge-Kutta Methods

Abstract

This report presents the numerical solution of an initial value problem (IVP) for the first-order differential equation y' = $3e^{2x}$y using three methods: Euler's Method, the Taylor Method (2nd order), and the Runge-Kutta Method (6th order). Each method's theoretical background is discussed, and Python implementations are provided. A comparison of the numerical results and the exact solution demonstrates the trade-off between accuracy and computational complexity.

Theoretical Background

1. Euler Method

Euler's method is the simplest numerical technique to solve an IVP of the form y' = f(x, y), $y(x_0) = y_0$. It approximates the solution at discrete points:

$$y_{n+1} = y_n + h\,f(x_n, y_n)$$

Euler's method is explicit and first-order accurate. It is simple but suffers from large truncation errors, especially for stiff or nonlinear equations.

2. Taylor Method (2nd Order)

The Taylor Method is a numerical approach to solving differential equations by expanding the solution as a Taylor series around a known point. For a first-order ODE, it approximates y(x) using:

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2} f'(x_n, y_n)$$

Here, $f'(x, y)$ is the derivative of f with respect to both x and y, and h is the step size. The more terms included, the more accurate the result, but it becomes harder to compute. It works best when derivatives are easy to find.

3. Runge-Kutta 6th order

The Runge-Kutta Method of 6th Order is a high-accuracy method that uses six weighted slope evaluations to estimate the next value.

After calculating k1 through k6 the formula simplifies to:

$$y_n + 1 = y_n + h/90\,(7k_1 + 32\,k_3 + 12\,k_4 + 34\,k_5 + 7k_6)$$

note: repeat using using loop for step size h=0.05, and compute from x=0 to x=1 (20 times)

Python Code

Equation and Exact Value:

```
# Differential equation
def f(x, y):
    return 3 * math.exp(2 * x) * y

# Exact solution
def exact_solution(x, y0, x0):
    return y0 * math.exp(1.5 * (math.exp(2*x) - math.exp(2*x0)))
```

Euler Method:

```
# Euler Method
def euler_method(x0, y0, h, n):
    xs, ys = [x0], [y0]
    for _ in range(n):
        y0 += h * f(x0, y0)
        x0 += h
        xs.append(x0)
        ys.append(y0)
    return xs, ys
```

Taylor Method (2nd order):

```
# Taylor Method (2nd order)
def taylor_method(x0, y0, h, n):
    xs, ys = [x0], [y0]
    for _ in range(n):
        f1 = f(x0, y0)
        df_dx = 6 * math.exp(2 * x0) * y0 + 9 * math.exp(4 * x0) * y0
        y0 += h * f1 + (h**2 / 2) * df_dx
        x0 += h
        xs.append(x0)
        ys.append(y0)
    return xs, ys
```

Runge-Kutta 6th Order:

```
# Runge-Kutta 6th Order
def rk6_method(x0, y0, h, n):
    xs, ys = [x0], [y0]
    for _ in range(n):
        k1 = f(x0, y0)
        k2 = f(x0 + h/4, y0 + h*k1/4)
        k3 = f(x0 + h/4, y0 + h*(k1 + k2)/8)
        k4 = f(x0 + h/2, y0 + h*(-k2 + 2*k3)/2)
        k5 = f(x0 + 3*h/4, y0 + h*(3*k1 + 9*k4)/16)
        k6 = f(x0 + h, y0 + h*(-3*k1 + 2*k2 + 12*k3 - 12*k4 + 8*k5)/7)
        y0 += h * (7*k1 + 32*k3 + 12*k4 + 32*k5 + 7*k6) / 90
        x0 += h
        xs.append(x0)
        ys.append(y0)
    return xs, ys
```

Numerical Example

Question:

$$y' = 3e^{2x}y \qquad y(0) = 1, \qquad x\in[0, 0.5], \qquad h = 0.05$$

Its exact solution is:

$$y(x) = e^{\frac{3}{2}(e^{2x}-1)}$$

Euler's Method

Euler's Method is the simplest way to estimate the solution to a differential equation. It works by using the slope at the current point to predict the value at the next point.

Formula:

$$y_{n+1} = y_n + hf(x_n, y_n)$$

where $f(x_n, y_n) = 3e^{2x}y$

Steps:

1. Initialize:

$$x_0 = 0,\ y_0 = 0.5,\ h = 0.05$$

2. for $n = 0\ to\ 9$, compute:

$$y_{n+1} = y_n + 0.05 \bullet 3e^{2x_n}y_n, \qquad x_{n+1} = x_n + 0.05$$

Taylor Method

2nd Order Taylor's Method estimates the next value by using both the slope (first derivative) and the curvature (second derivative) at the current point. This gives a better approximation than Euler's method by accounting for how quickly the slope is changing.

Formula:

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2} f'(x_n, y_n)$$

We must compute:

- $f(x_n, y_n) = 3e^{2x}y$
- First derivative f' using the chain rule:

$$f' = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot y'$$

Calculating:

- $\frac{\partial f}{\partial x} = 6e^{2x}y$
- $\frac{\partial f}{\partial y} = 3e^{2x}$
- $y' = f = 3e^{2x}y$

So the final formula simplifies to:

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2}(6e^{2x_n} + 9e^{4x_n})y$$

Runge-Kutta Method (RK6)

The Runge-Kutta Method of 6th Order is a high-accuracy method that uses seven weighted slope evaluations to estimate the next value.

Let k_1 through k_7 be intermediate slopes:

$$k_1 = f(x_n, y_n)$$
$$k_2 = f(x_n + \frac{h}{4}, y_n + \frac{h}{4}k_1)$$
$$k_3 = f(x_n + \frac{h}{4}, y_n + \frac{h}{8}(k_1 + k_2))$$
$$k_4 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}(-k_2 + 2k_3))$$
$$k_5 = f(x_n + \frac{3h}{4}, y_n + \frac{h}{16}(3k_1 + 9))$$
$$k_6 = f(x_n + h, y_n + \frac{h}{7}(-3k_1 + 2k_2 - 12k_3 = 12k_4 + 8k_5))$$

Final Formula for the IVP =

$$y_{n+1} = y_n + \frac{h}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)$$

Results & Discussion

This study aimed to compare the performance of Euler's Method, Taylor's 2nd Order Method, and the 6th Order Runge-Kutta Method (RK6) in approximating the solution to the differential equation

$$\frac{dy}{dx} = 3e^{2x}y,\ y(0) = 1$$

over the interval [0,0.5].

The results presented in the table and visualized in the graph show a clear hierarchy of accuracy among the methods. Initially, all methods started at the same point with y(0)=1. However, as x increased, their divergence from the exact solution became evident.

Table:

x	Euler	Taylor	RK6	Exact
0.00	0.500000	0.500000	0.500000	0.500000
0.05	0.575000	0.584375	0.585440	0.585440
0.10	0.670321	0.694124	0.696949	0.696949
0.15	0.793131	0.839302	0.845050	0.845050
0.20	0.953723	1.034945	1.045599	1.045599
0.25	1.167141	1.304031	1.323043	1.323043
0.30	1.455785	1.682531	1.716060	1.716060
0.35	1.853677	2.228235	2.287550	2.287551
0.40	2.413603	3.036609	3.142956	3.142957
0.45	3.219339	4.270215	4.464944	4.464948
0.50	4.407084	6.215067	6.581575	6.581585

Table 1 - Comparison of Euler, Taylor, RK6, and Exact Solutions for y′=3e2x over the Interval [0, 0.5] with Step Size h=0.05

- **Euler's Method** underestimated exact values due to its first-order nature, resulting in a significant error of 4.407084, far below the exact solution of 6.581585.
- **Taylor's 2nd Order Method**improved Euler's performance by incorporating second derivative information, resulting in a final value closer to the exact solution but showing considerable deviation.
- **Runge-Kutta 6th Order (RK6)** offers the most accurate approximation, closely matching the exact value at x=1, demonstrating the power of higher-order Runge-Kutta methods in handling steep or non-linear differential equations.

Graph Overview:

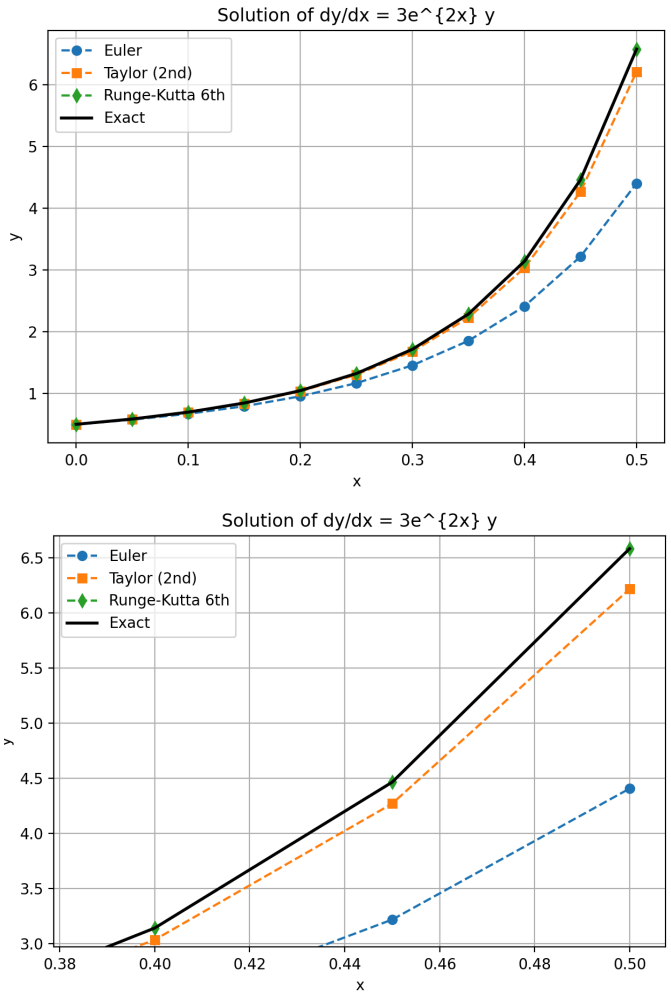

Figure 1 - Graphical Comparison of Numerical Methods vs. Exact Solution over [0, 0.5], h=0.05

In summary, while Euler is useful for quick approximations and Taylor improves with more derivative information, RK6 stands out as the most reliable and accurate for problems like this with fast-growing solutions.

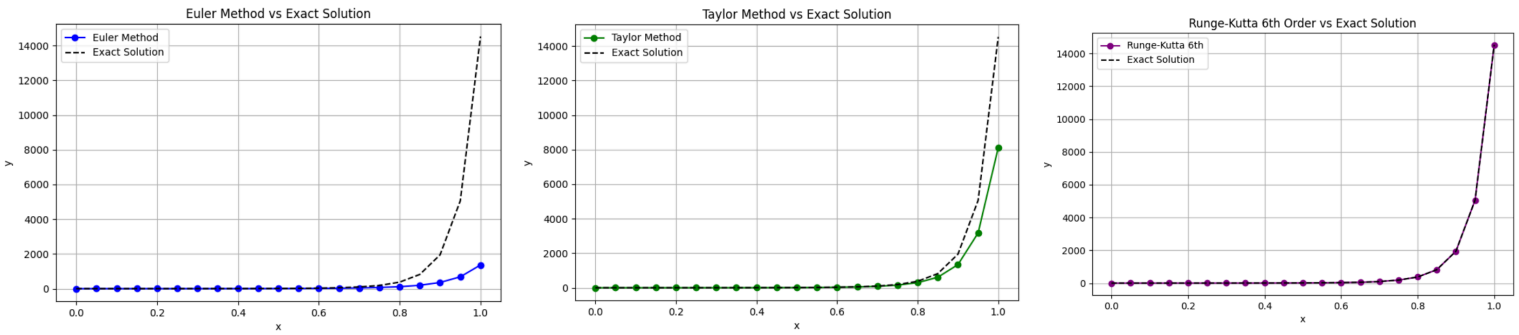

Figure 2 - Graphical Comparison of Numerical Methods vs. Exact Solution over [0, 1], h=0.05

Conclusion

The project analyzed three classical numerical methods for solving first-order initial value problems. Euler's method is easy but has high error accumulation. Taylor's method offers improved accuracy but requires careful symbolic differentiation. The 6th-order Runge-Kutta method is the most reliable and accurate, especially for exponential or rapidly growing solutions. The choice of numerical method should balance accuracy requirements and differential equation nature. Higher-order methods like RK6 are recommended for high-precision applications or rapidly changing systems.