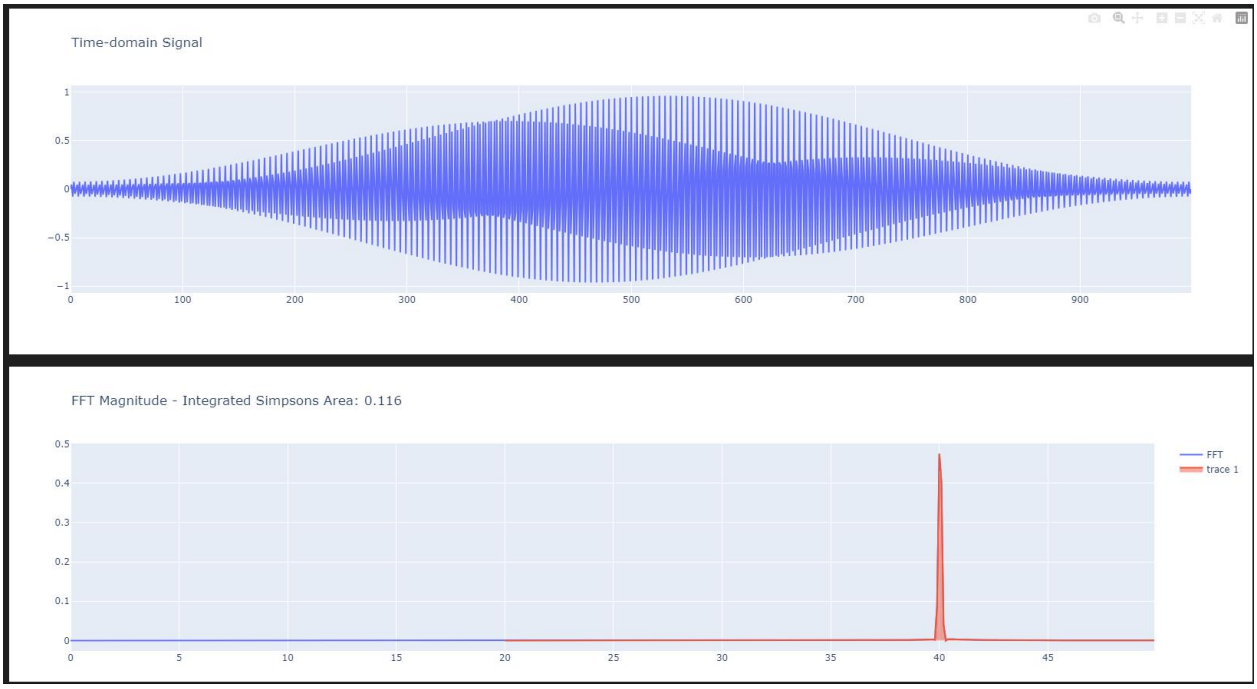
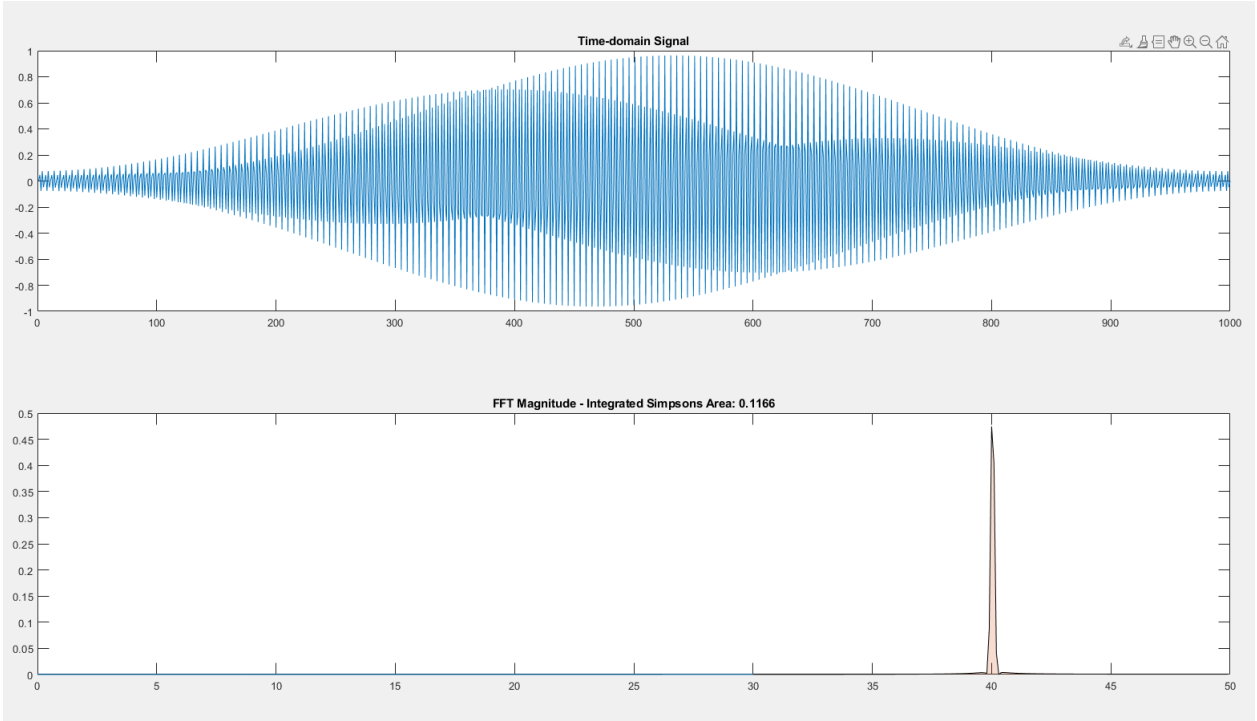
1a - python



1a - matlab



1b - python

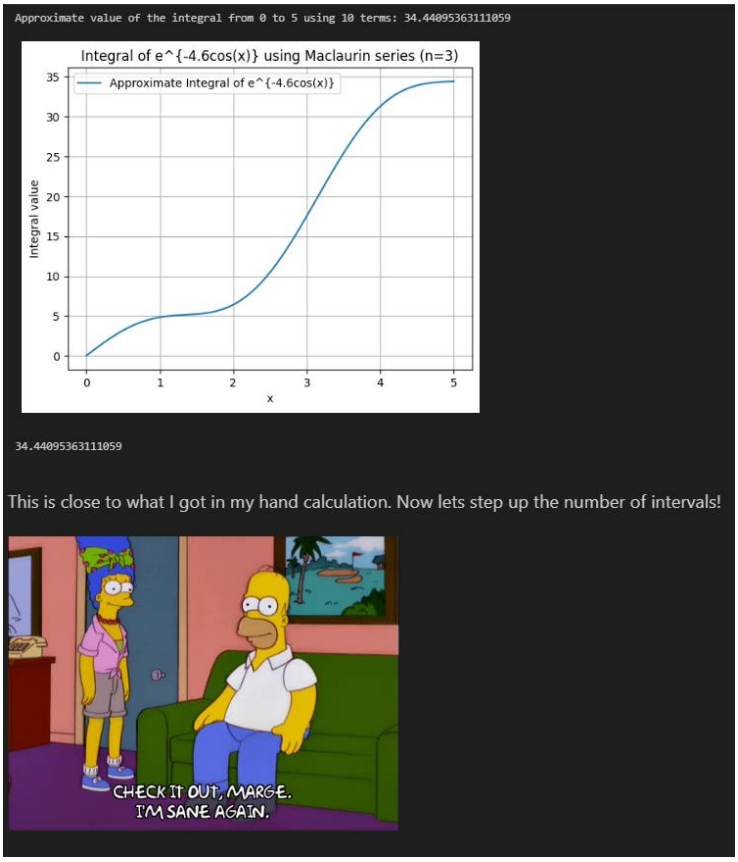


Figure 1: Maclaurin Series - checking validity of low N value hand calc

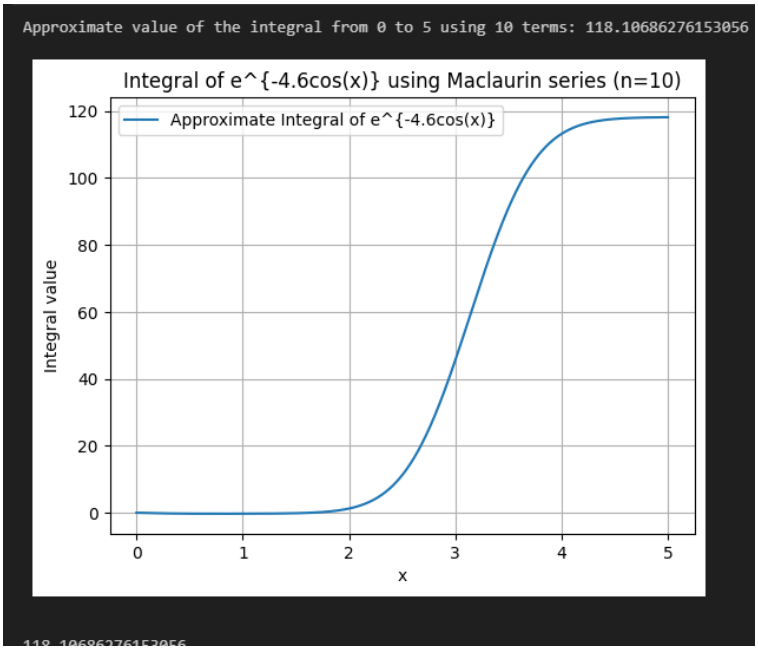
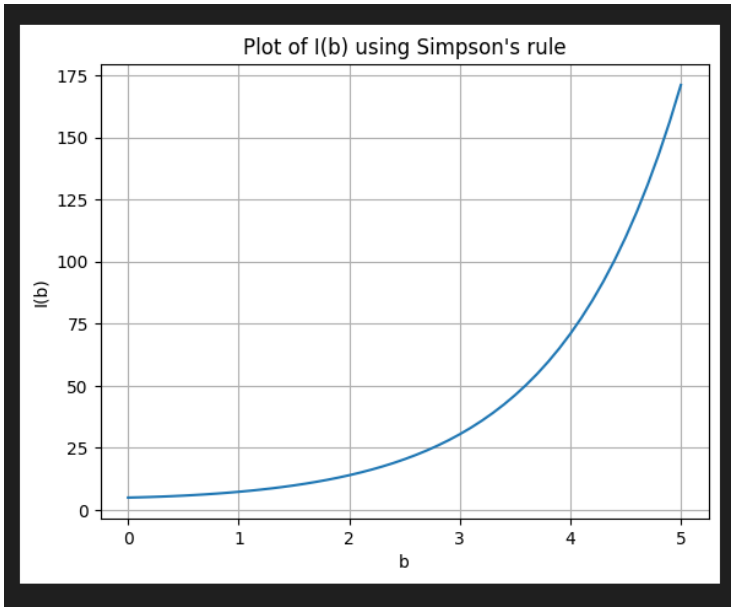
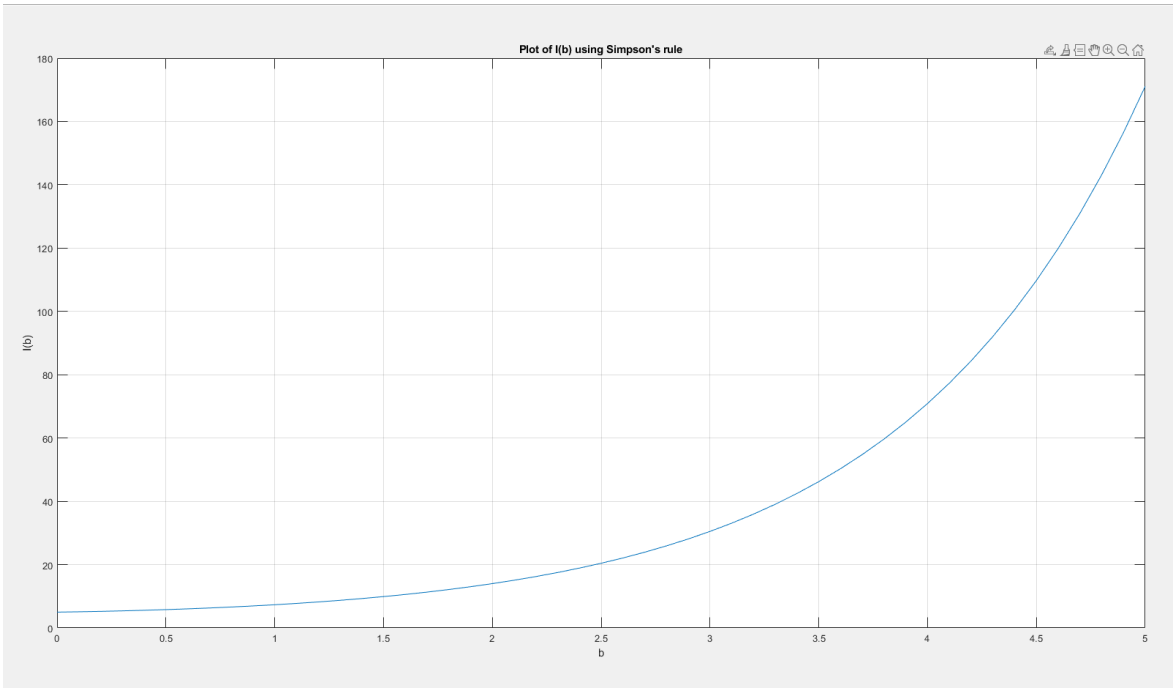


Figure 2: increasing N to compare to Simpson Rule


Plot of I(b) using Simpson's rule
I(b)
b
0
25
50
75
100
125
150
175
0
1
2
3
4
5

1b – matlab


Plot of I(b) using Simpson's rule
I(b)
b
0
20
40
60
80
100
120
140
160
180
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

1c – python

Method 1

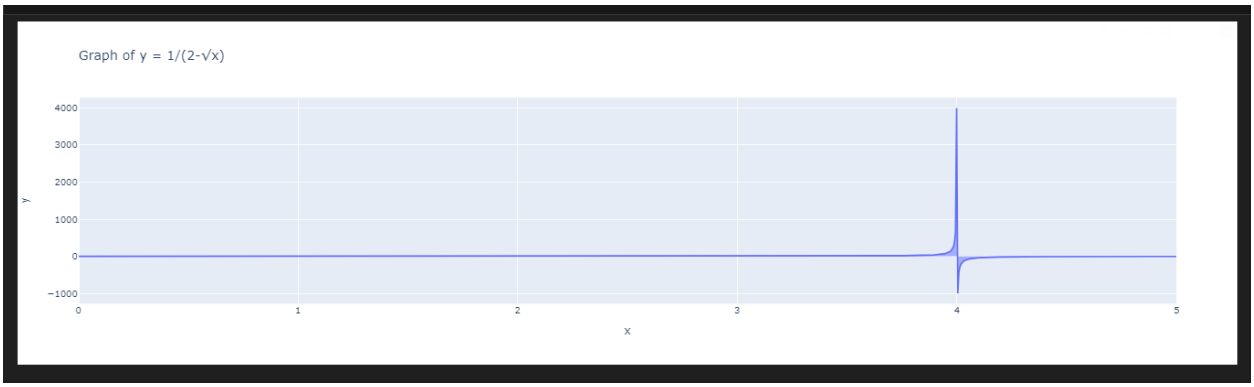
```
# Define the function to be integrated
def f(x):
    return 1 / (2 - np.sqrt(x))

# Integrate the function from 0 to 5
integral_value, _ = quad(f, 0, 5)

# Generate x values for plotting
x = np.linspace(0, 5, 1000)
y = f(x)
```

```
C:\Users\Aaron\AppData\Local\Temp\ipykernel_19532\3069320982.py:6: IntegrationWarning: The integral is probably divergent, or slowly convergent.
  integral_value, _ = quad(f, 0, 5)
```

what my computer really means....

good thing Gaussian Quadrature is robust

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i \cdot f(x_i)$$

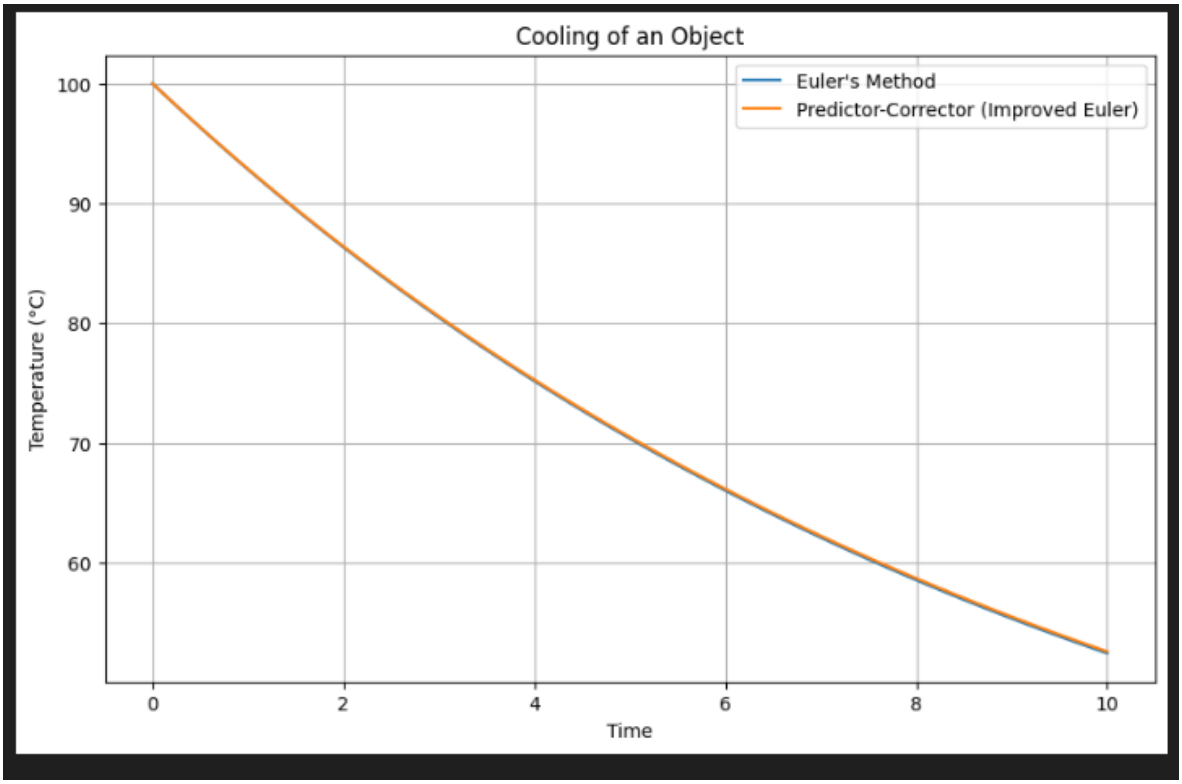



```
Value of I from 0 to 5: 4.07499
```

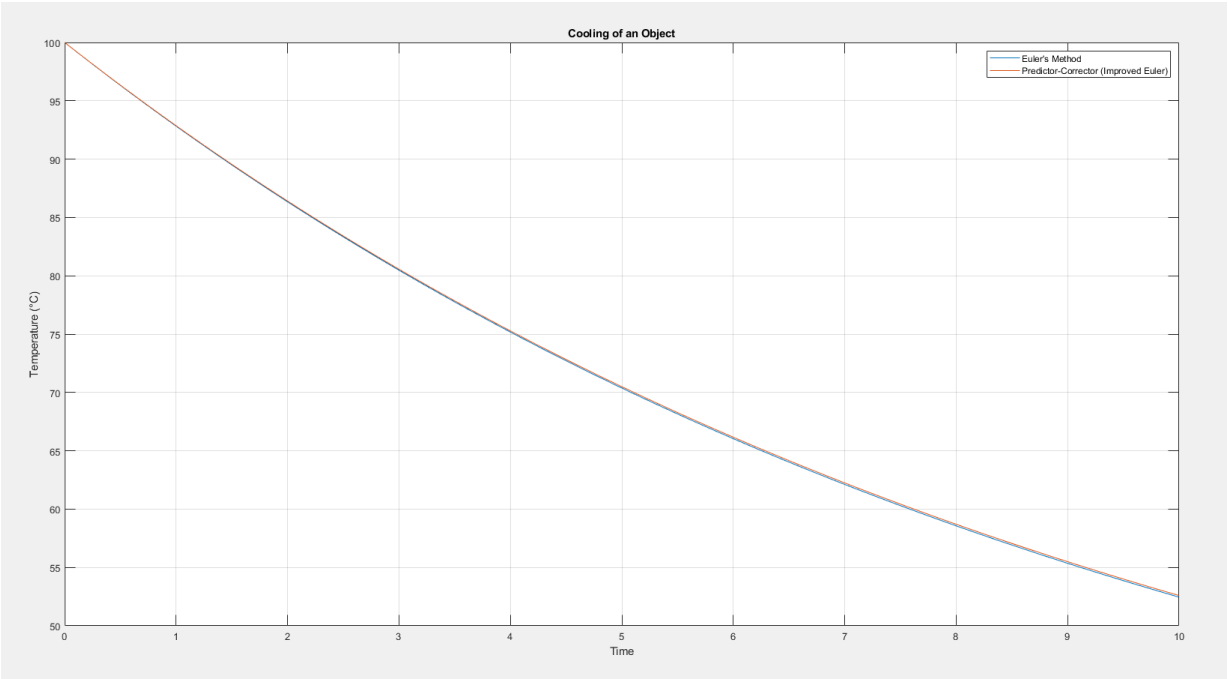
1c – matlab

Output - Value of I from 0 to 5: 4.07500

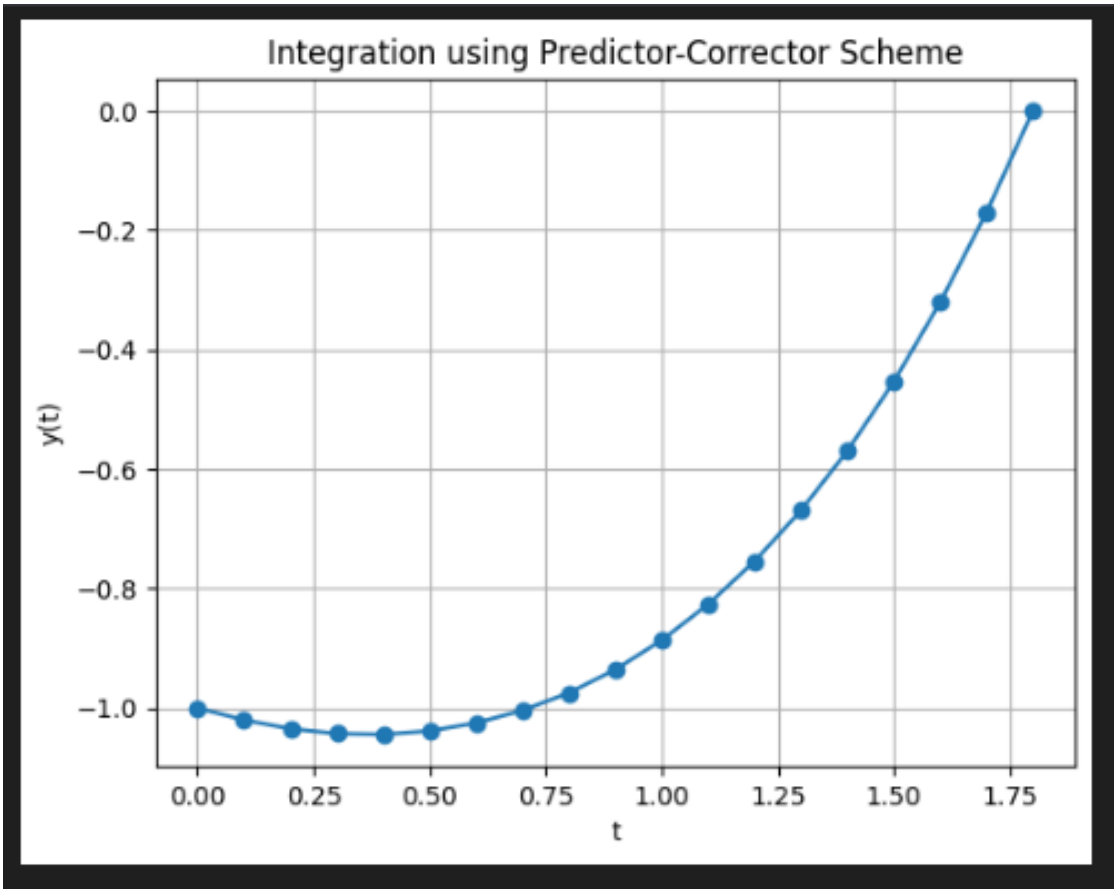
2a – python



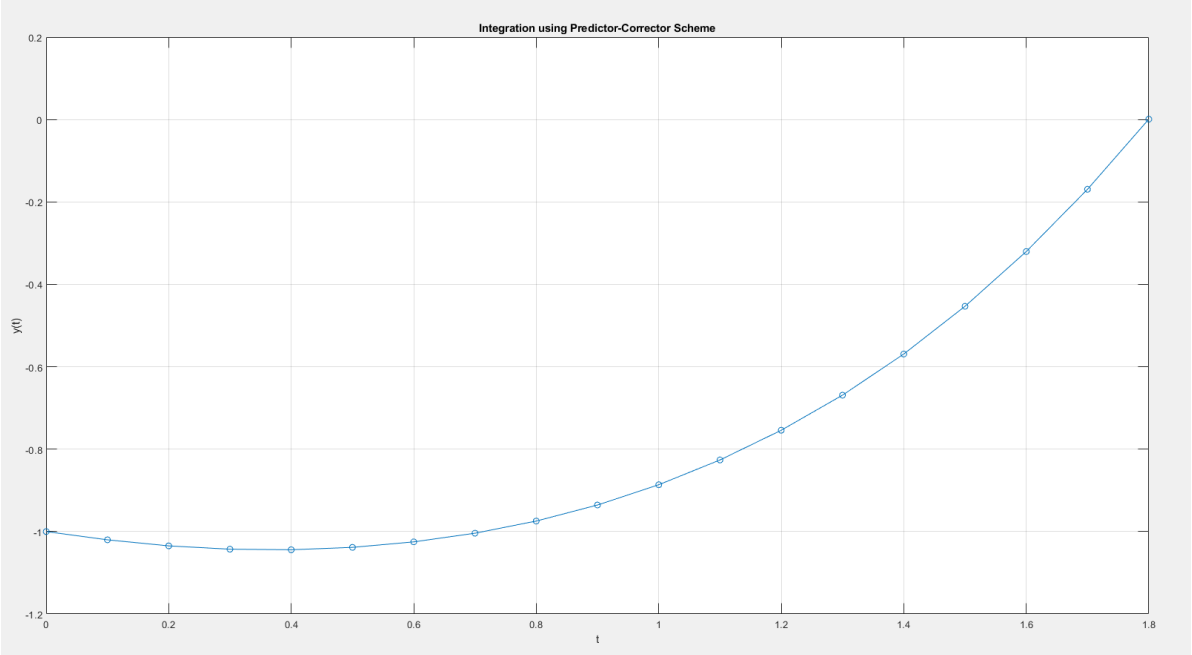
2a – matlab



2b – python



2b – matlab



3a – python

Logistic Growth Differential Equation: $\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right)$

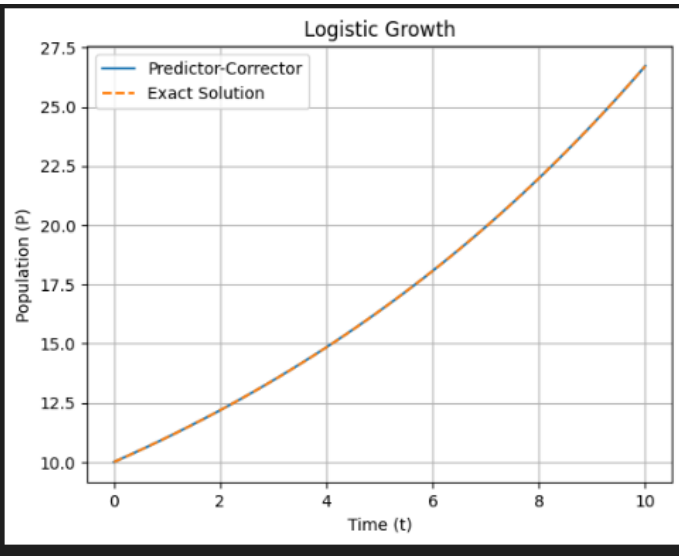
Where:

$P(t)$ is the population at time t.
r is the growth rate.
K is the carrying capacity of the environment.

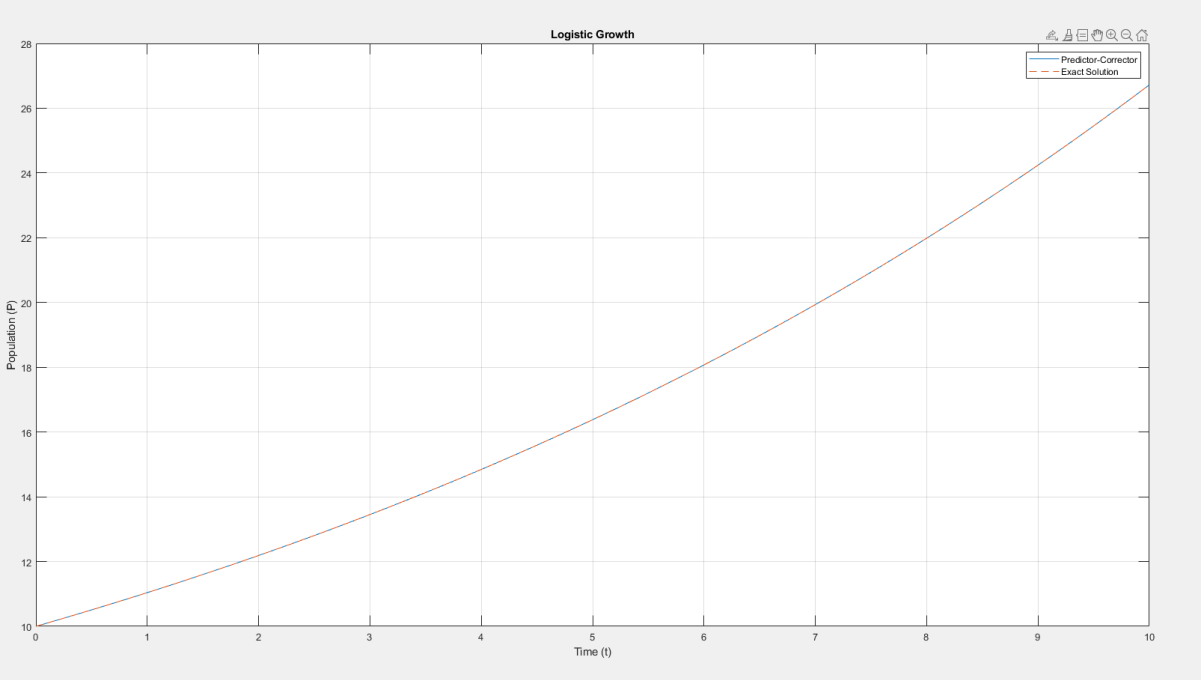
Exact Solution for Logistic Growth: $P(t) = \frac{KP_0e^{rt}}{K+P_0(e^{rt}-1)}$

Where:

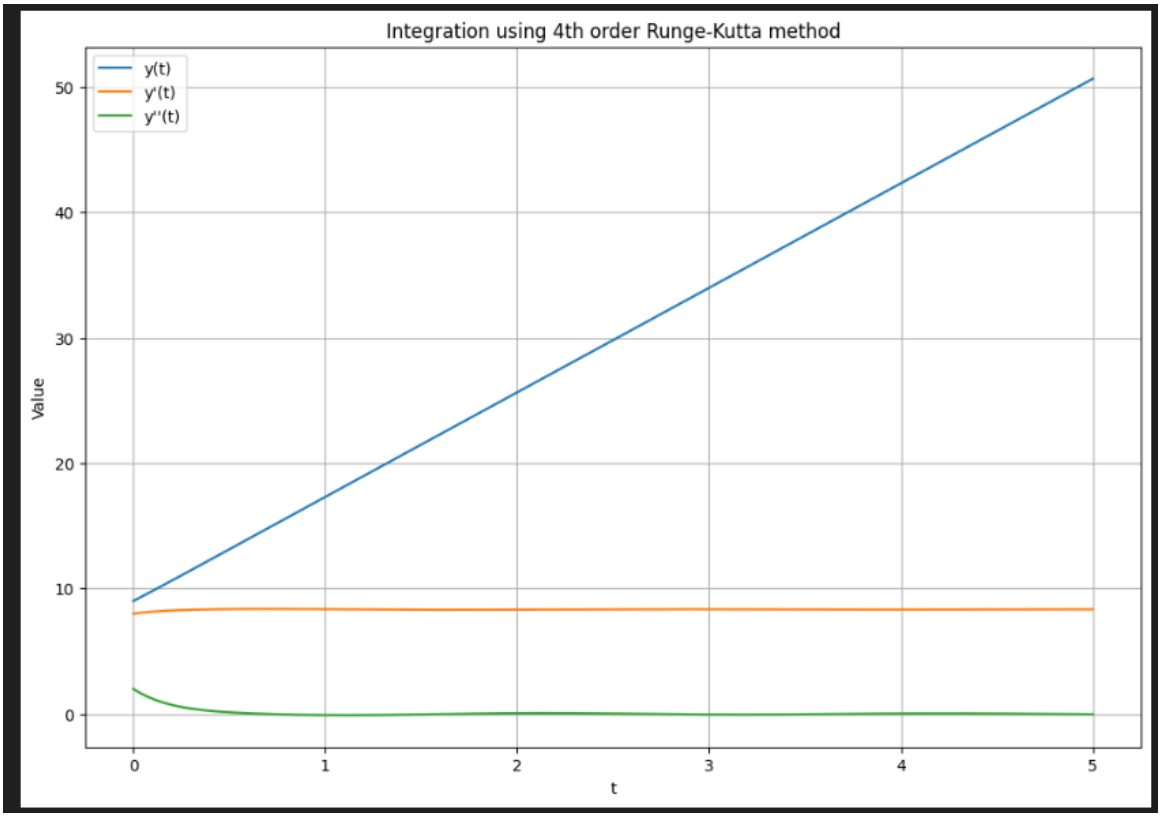
P_0 is the initial population.



3a – matlab



3b – python



3b – python