Runge-Kutta Method

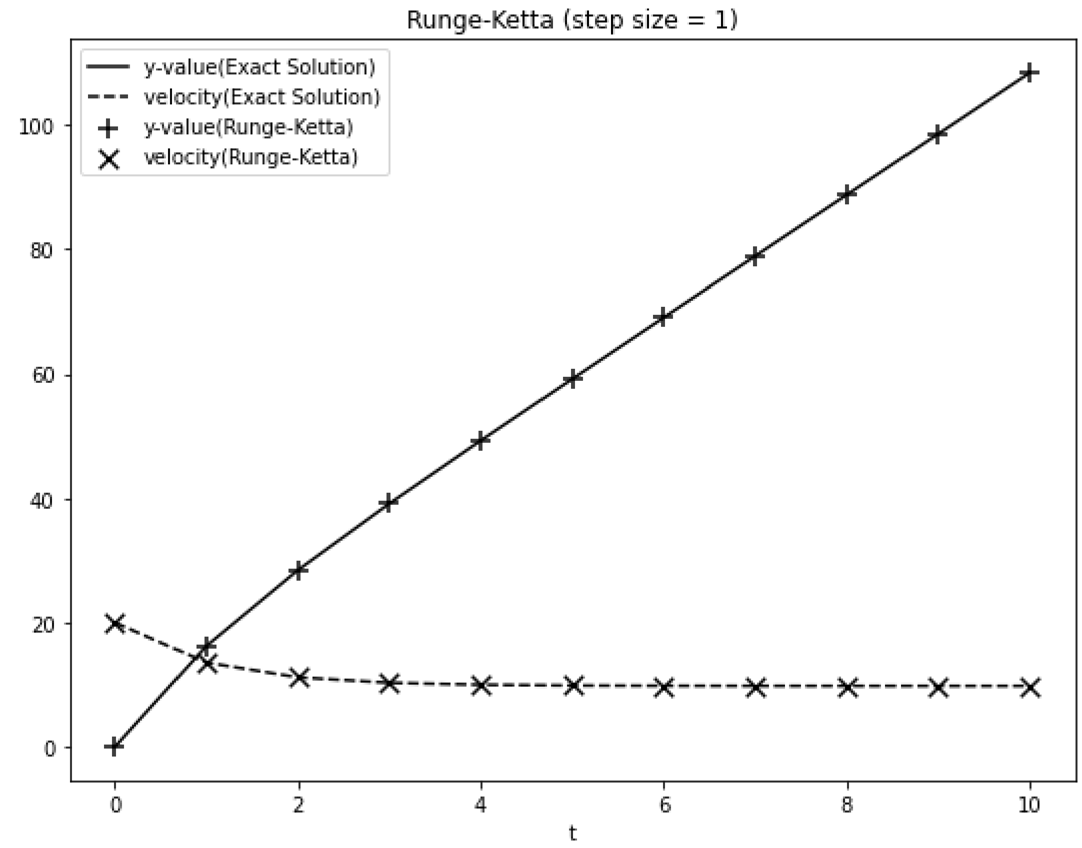
When step size, h=1

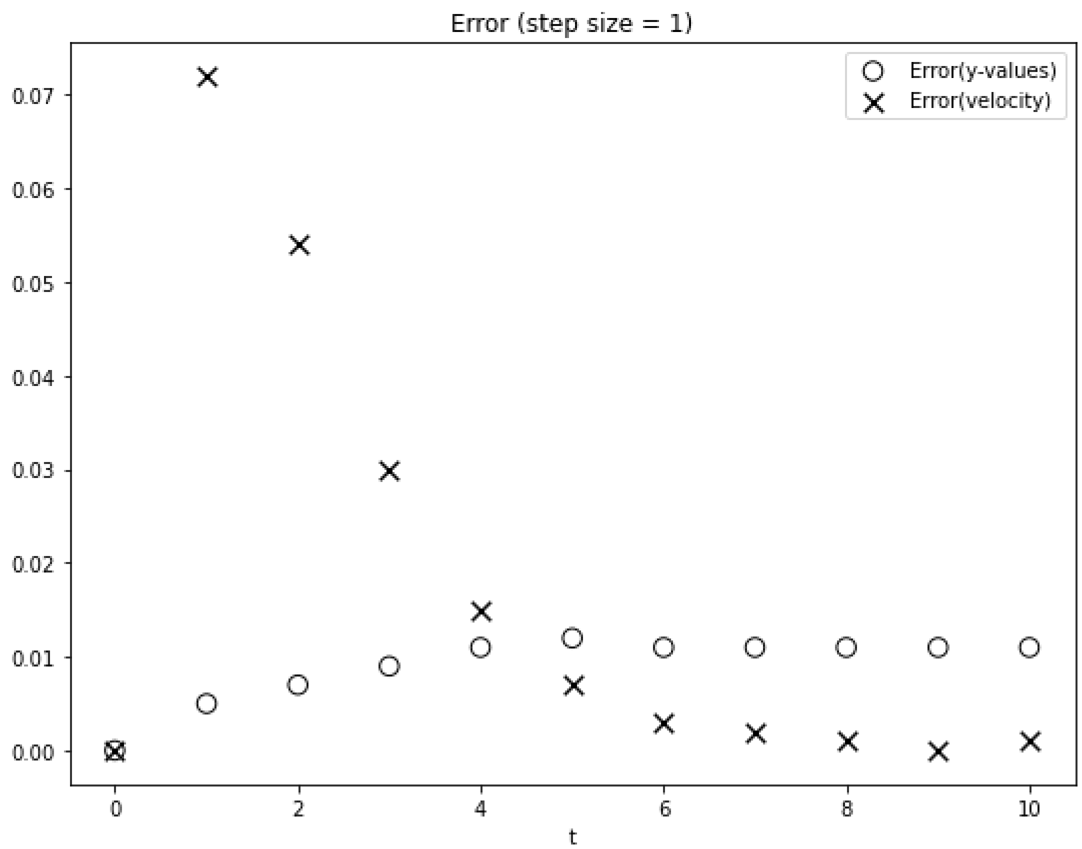
In [1]:

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

g = 9.81
c = 1
m = 1
a = 0
b = 10
n = 10
h1 = (b-a)/n
r = 3

t = [0]
for i in range (0,n):
    t.append(t[i] + h1)
t = list(np.around(np.array(t),r))

#velocity na hinihingi sa problem
v = [20]
for i in range(0,n):
    h = h1
    F1 = h*(g - (c/m)*(v[i]))
    F2 = h*(g - (c/m)*(v[i] + F1/2))
    F3 = h*(g - (c/m)*(v[i] + F2/2))
    F4 = h*(g - (c/m)*(v[i] + F3))
    v.append(v[i] + (F1 + 2*F2 + 2*F3 + F4)/6)
v= list(np.around(np.array(v),r))

#velocity na kelangan sa y-values
v1 = [20]
for i in range(0,2*n+2):
    h = h1/2
    F_1 = h*(g - (c/m)*(v1[i]))
    F_2 = h*(g - (c/m)*(v1[i] + F_1/2))
    F_3 = h*(g - (c/m)*(v1[i] + F_2/2))
    F_4 = h*(g - (c/m)*(v1[i] + F_3))
    v1.append(v1[i] + (F_1 + 2*F_2 + 2*F_3 + F_4)/6)
v1= list(np.around(np.array(v1),r))

# y-values na hinihingi sa problem
y = [0]
for i in range(0,n):
    F = []
    for j in range(0,2*n,2):
        h = h1
        f1 = h*(v1[j])
        f2 = h*(v1[j+1])
        f4 = h*(v1[j+2])
        F.append((f1 + 4*f2 + f4)/6)
    y.append(y[i] + F[i])
y= list(np.around(np.array(y),r))

#exact solution
x=np.linspace(a,b,n+1)
y0=(m*g/c)*x + (20-(m*g/c))*(1-np.exp(-x*c/m))*(m/c)
y0= list(np.around(np.array(y0),r))
v0=20*np.exp(-x*c/m) + (m*g/c)*(1-np.exp(-x*c/m))
```

```python
v0= list(np.around(np.array(v0),r))

E_v = []
for i in range(0,n+1):
    E_v.append(v[i]-v0[i])

E_y = []
for i in range(0,n+1):
    E_y.append(y[i]-y0[i])

print("t: {}".format(t))
print("y_ Runge-Ketta: {}".format(y))
print("y_ exact: {}".format(y0))
print("v_Runge-Ketta: {}".format(v))
print("v_exact: {}".format(v0))

fig, axs=plt.subplots(2,1)
fig.set_size_inches(9,15)
axs[0].scatter(t,y,marker = '+', edgecolors='none', s=100,facecolors = 'black
axs[0].scatter(t,v,marker = 'x', edgecolors='none', s=90,facecolors = 'black'
axs[0].plot(x,y0,c='black', label = 'y-value(Exact Solution)')
axs[0].plot(x,v0,c='black', linestyle = 'dashed', label = 'velocity(Exact Sol
axs[1].scatter(t,E_y,marker = 'o', edgecolors='black', s=90,facecolors = 'nor
axs[1].scatter(t,E_v,marker = 'x', edgecolors='none', s=90,facecolors = 'blac
axs[0].set_title('Runge-Ketta (step size = 1)')
axs[1].set_title('Error (step size = 1)')
axs[0].set(xlabel='t')
axs[1].set(xlabel='t')
axs[0].legend()
axs[1].legend()
plt.show()
```

```
t: [0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0]
y_ Runge-Ketta: [0.0, 16.256, 28.438, 39.122, 49.254, 59.183, 69.036, 78.862,
88.678, 98.49, 108.301]
y_ exact: [0.0, 16.251, 28.431, 39.113, 49.243, 59.171, 69.025, 78.851, 88.66
7, 98.479, 108.29]
v_Runge-Ketta: [20.0, 13.631, 11.243, 10.347, 10.012, 9.886, 9.838, 9.821, 9.
814, 9.811, 9.811]
v_exact: [20.0, 13.559, 11.189, 10.317, 9.997, 9.879, 9.835, 9.819, 9.813, 9.
811, 9.81]
```





When step size, h=0.5</h3

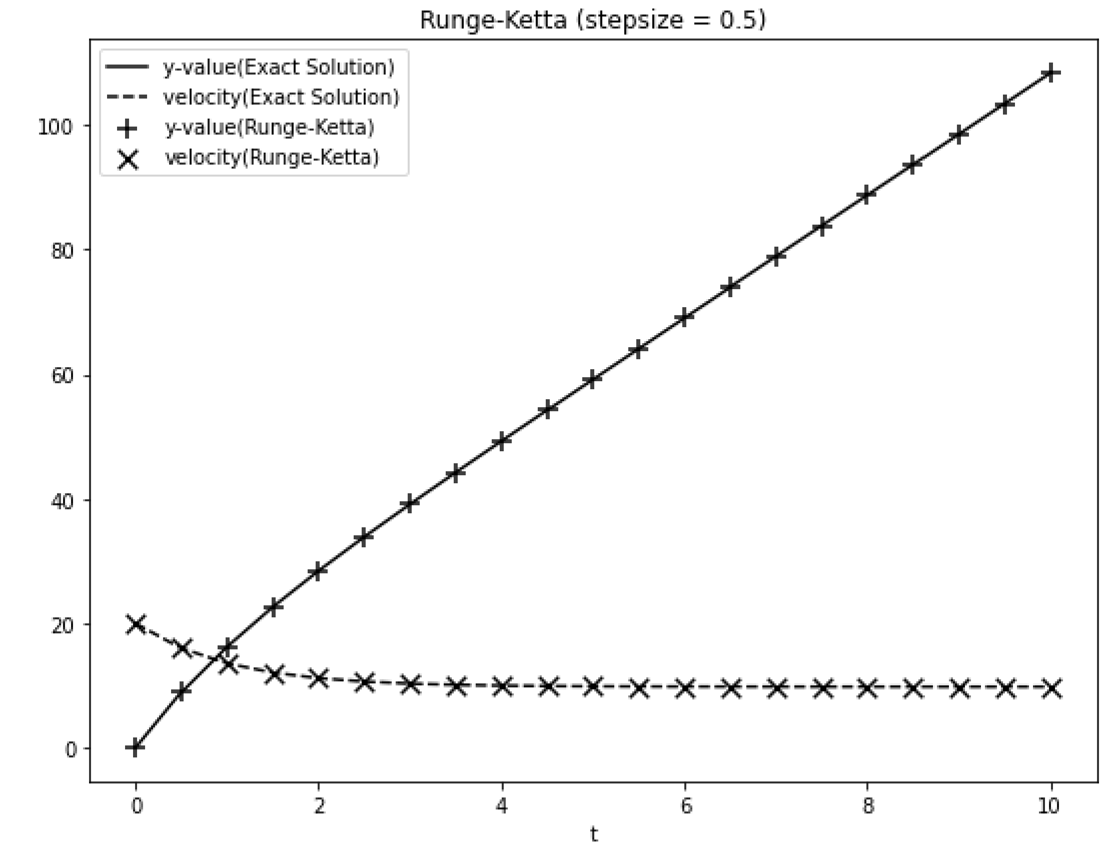
In [2]:

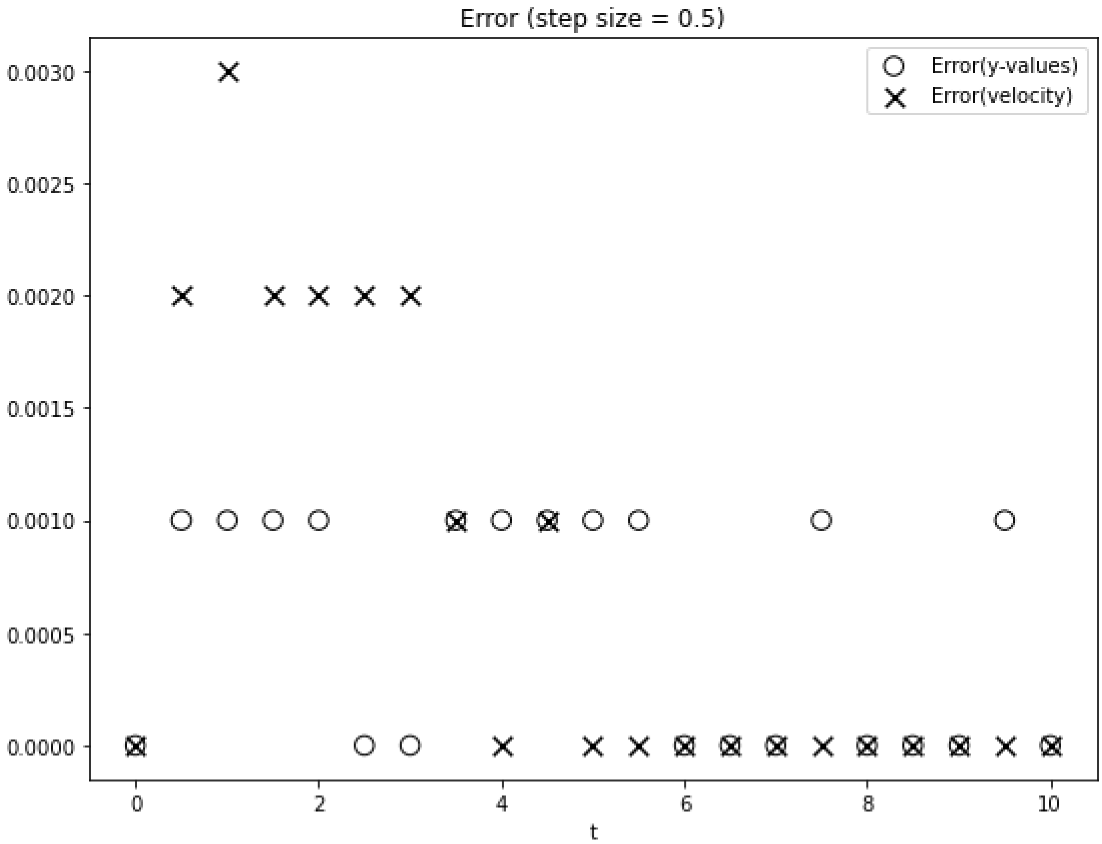
```
import numpy as np
import matplotlib.pyplot as plt

g = 9.81
```

```
c = 1
m = 1
a = 0
b = 10
n = 20
h1 = (b-a)/n
r = 3

t = [0]
for i in range (0,n):
    t.append(t[i] + h1)
t = list(np.around(np.array(t),r))

#velocity na hinihingi ni sir
v = [20]
for i in range(0,n):
    h = h1
    F1 = h*(g - (c/m)*(v[i]))
    F2 = h*(g - (c/m)*(v[i] + F1/2))
    F3 = h*(g - (c/m)*(v[i] + F2/2))
    F4 = h*(g - (c/m)*(v[i] + F3))
    v.append(v[i] + (F1 + 2*F2 + 2*F3 + F4)/6)
v= list(np.around(np.array(v),r))

#velocity na kelangan sa y-values
v1 = [20]
for i in range(0,2*n+2):
    h = h1/2
    F_1 = h*(g - (c/m)*(v1[i]))
    F_2 = h*(g - (c/m)*(v1[i] + F_1/2))
    F_3 = h*(g - (c/m)*(v1[i] + F_2/2))
    F_4 = h*(g - (c/m)*(v1[i] + F_3))
    v1.append(v1[i] + (F_1 + 2*F_2 + 2*F_3 + F_4)/6)
v1= list(np.around(np.array(v1),r))

# y-values na hinihingi ni sir
y = [0]
for i in range(0,n):
    F = []
    for j in range(0,2*n,2):
        h = h1
        f1 = h*(v1[j])
        f2 = h*(v1[j+1])
        f4 = h*(v1[j+2])
        F.append((f1 + 4*f2 + f4)/6)
    y.append(y[i] + F[i])
y= list(np.around(np.array(y),r))

#exact solution
x=np.linspace(a,b,n+1)
y0=(m*g/c)*x + (20-(m*g/c))*(1-np.exp(-x*c/m))*(m/c)
y0= list(np.around(np.array(y0),r))
v0=20*np.exp(-x*c/m) + (m*g/c)*(1-np.exp(-x*c/m))
v0= list(np.around(np.array(v0),r))

E_v = []
for i in range(0,n+1):
    E_v.append(v[i]-v0[i])

E_y = []
for i in range(0,n+1):
    E_y.append(y[i]-y0[i])

print("t: {}".format(t))
print("y_ Runge-Ketta: {}".format(y))
```

```
print("y_ exact: {}".format(y0))
print("v_Runge-Ketta: {}".format(v))
print("v_exact: {}".format(v0))

fig, axs=plt.subplots(2,1)
fig.set_size_inches(9,15)
axs[0].scatter(t,y,marker = '+', edgecolors='none', s=100,facecolors = 'black
axs[0].scatter(t,v,marker = 'x', edgecolors='none', s=90,facecolors = 'black'
axs[0].plot(x,y0,c='black', label = 'y-value(Exact Solution)')
axs[0].plot(x,v0,c='black', linestyle = 'dashed', label = 'velocity(Exact Sol
axs[1].scatter(t,E_y,marker = 'o', edgecolors='black', s=90,facecolors = 'nor
axs[1].scatter(t,E_v,marker = 'x', edgecolors='none', s=90,facecolors = 'blac
axs[0].set_title('Runge-Ketta (stepsize = 0.5)')
axs[1].set_title('Error (step size = 0.5)')
axs[0].set(xlabel='t')
axs[1].set(xlabel='t')
axs[0].legend()
axs[1].legend()
plt.show()
```

```
t: [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.
0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0]
y_ Runge-Ketta: [0.0, 8.915, 16.252, 22.632, 28.432, 33.879, 39.113, 44.218,
49.244, 54.223, 59.172, 64.104, 69.025, 73.94, 78.851, 83.76, 88.667, 93.573,
98.479, 103.385, 108.29]
y_ exact: [0.0, 8.914, 16.251, 22.631, 28.431, 33.879, 39.113, 44.217, 49.24
3, 54.222, 59.171, 64.103, 69.025, 73.94, 78.851, 83.759, 88.667, 93.573, 98.
479, 103.384, 108.29]
v_Runge-Ketta: [20.0, 15.993, 13.562, 12.086, 11.191, 10.648, 10.319, 10.119,
9.997, 9.924, 9.879, 9.852, 9.835, 9.825, 9.819, 9.816, 9.813, 9.812, 9.811,
9.811, 9.81]
v_exact: [20.0, 15.991, 13.559, 12.084, 11.189, 10.646, 10.317, 10.118, 9.99
7, 9.923, 9.879, 9.852, 9.835, 9.825, 9.819, 9.816, 9.813, 9.812, 9.811, 9.81
1, 9.81]
```





When step size, h=0.2</h3

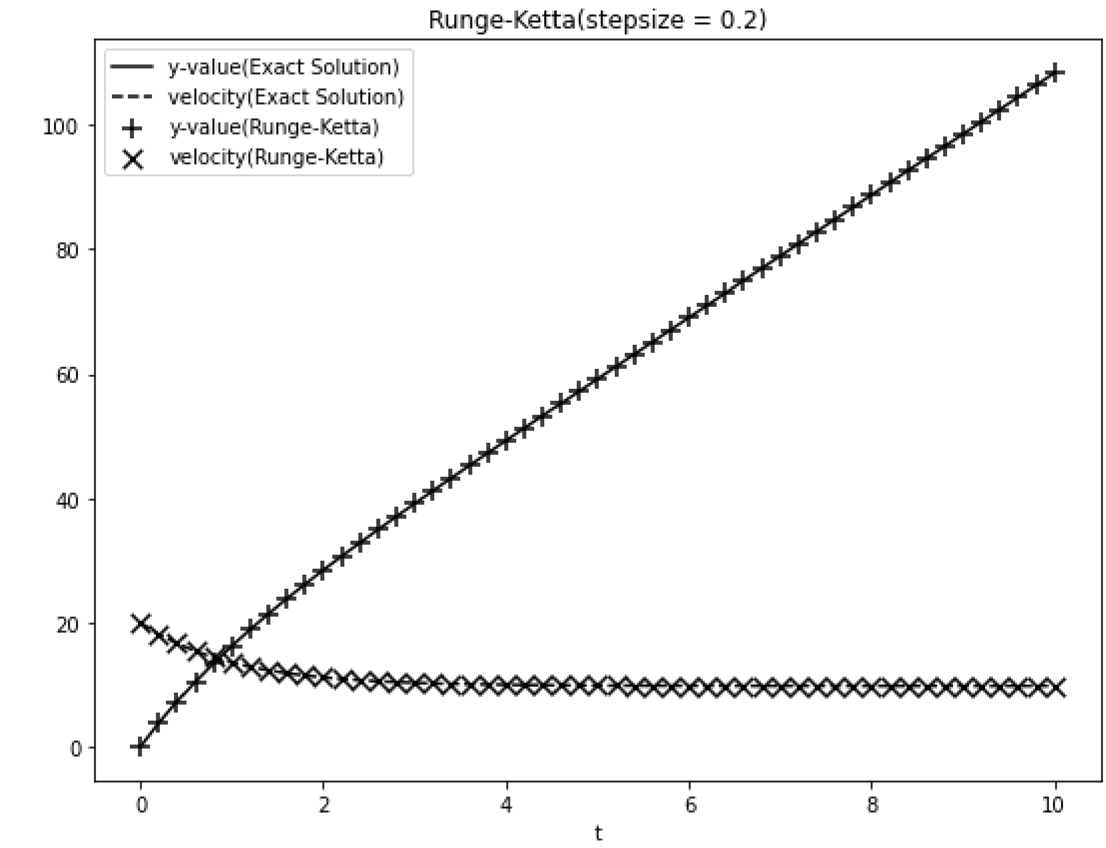
In [3]:

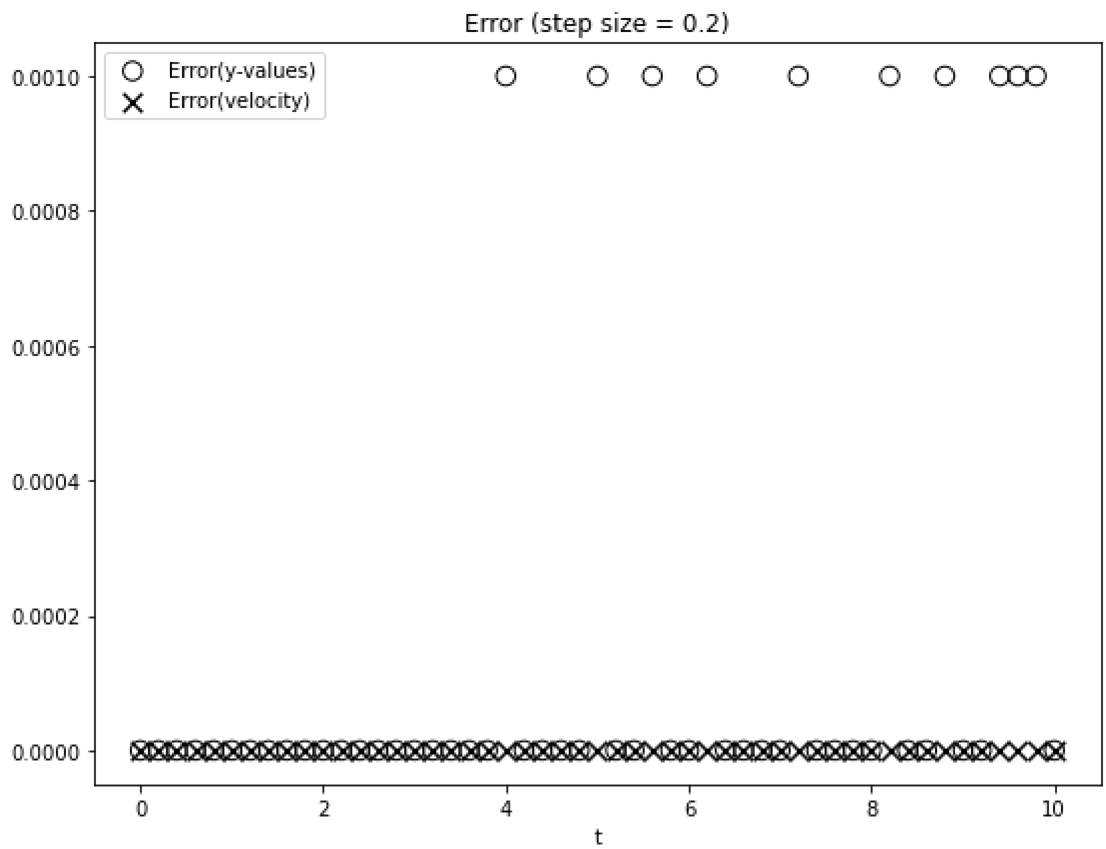
```
import numpy as np
import matplotlib.pyplot as plt

g = 9.81
```

```python
c = 1
m = 1
a = 0
b = 10
n = 50
h1 = (b-a)/n
r = 3

t = [0]
for i in range (0,n):
    t.append(t[i] + h1)
t = list(np.around(np.array(t),r))

#velocity na hinihingi ni sir
v = [20]
for i in range(0,n):
    h = h1
    F1 = h*(g - (c/m)*(v[i]))
    F2 = h*(g - (c/m)*(v[i] + F1/2))
    F3 = h*(g - (c/m)*(v[i] + F2/2))
    F4 = h*(g - (c/m)*(v[i] + F3))
    v.append(v[i] + (F1 + 2*F2 + 2*F3 + F4)/6)
v= list(np.around(np.array(v),r))

#velocity na kelangan sa y-values
v1 = [20]
for i in range(0,2*n+2):
    h = h1/2
    F_1 = h*(g - (c/m)*(v1[i]))
    F_2 = h*(g - (c/m)*(v1[i] + F_1/2))
    F_3 = h*(g - (c/m)*(v1[i] + F_2/2))
    F_4 = h*(g - (c/m)*(v1[i] + F_3))
    v1.append(v1[i] + (F_1 + 2*F_2 + 2*F_3 + F_4)/6)
v1= list(np.around(np.array(v1),r))

# y-values na hinihingi ni sir
y = [0]
for i in range(0,n):
    F = []
    for j in range(0,2*n,2):
        h = h1
        f1 = h*(v1[j])
        f2 = h*(v1[j+1])
        f4 = h*(v1[j+2])
        F.append((f1 + 4*f2 + f4)/6)
    y.append(y[i] + F[i])
y= list(np.around(np.array(y),r))

#exact solution
x=np.linspace(a,b,n+1)
y0=(m*g/c)*x + (20-(m*g/c))*(1-np.exp(-x*c/m))*(m/c)
y0= list(np.around(np.array(y0),r))
v0=20*np.exp(-x*c/m) + (m*g/c)*(1-np.exp(-x*c/m))
v0= list(np.around(np.array(v0),r))

E_v = []
for i in range(0,n+1):
    E_v.append(v[i]-v0[i])

E_y = []
for i in range(0,n+1):
    E_y.append(y[i]-y0[i])

print("t: {}".format(t))
print("y_ Runge-Ketta: {}".format(y))
```

```
print("y_ exact: {}".format(y0))
print("v_Runge-Ketta: {}".format(v))
print("v_exact: {}".format(v0))

fig, axs=plt.subplots(2,1)
fig.set_size_inches(9,15)
axs[0].scatter(t,y,marker = '+', edgecolors='none', s=100,facecolors = 'black
axs[0].scatter(t,v,marker = 'x', edgecolors='none', s=90,facecolors = 'black'
axs[0].plot(x,y0,c='black', label = 'y-value(Exact Solution)')
axs[0].plot(x,v0,c='black', linestyle = 'dashed', label = 'velocity(Exact Sol
axs[1].scatter(t,E_y,marker = 'o', edgecolors='black', s=90,facecolors = 'nor
axs[1].scatter(t,E_v,marker = 'x', edgecolors='none', s=90,facecolors = 'blac
axs[0].set_title('Runge-Ketta(stepsize = 0.2)')
axs[1].set_title('Error (step size = 0.2)')
axs[0].set(xlabel='t')
axs[1].set(xlabel='t')
axs[0].legend()
axs[1].legend()
plt.show()
```

```
t: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.
8, 3.0, 3.2, 3.4, 3.6, 3.8, 4.0, 4.2, 4.4, 4.6, 4.8, 5.0, 5.2, 5.4, 5.6, 5.8,
6.0, 6.2, 6.4, 6.6, 6.8, 7.0, 7.2, 7.4, 7.6, 7.8, 8.0, 8.2, 8.4, 8.6, 8.8, 9.
0, 9.2, 9.4, 9.6, 9.8, 10.0]
y_ Runge-Ketta: [0.0, 3.809, 7.283, 10.484, 13.459, 16.251, 18.893, 21.411, 2
3.829, 26.164, 28.431, 30.643, 32.81, 34.939, 37.038, 39.113, 41.167, 43.204,
45.228, 47.24, 49.244, 51.239, 53.229, 55.214, 57.194, 59.172, 61.146, 63.11
8, 65.089, 67.057, 69.025, 70.992, 72.957, 74.922, 76.887, 78.851, 80.815, 8
2.778, 84.741, 86.704, 88.667, 90.63, 92.592, 94.554, 96.517, 98.479, 100.44
1, 102.404, 104.366, 106.328, 108.29]
y_ exact: [0.0, 3.809, 7.283, 10.484, 13.459, 16.251, 18.893, 21.411, 23.829,
26.164, 28.431, 30.643, 32.81, 34.939, 37.038, 39.113, 41.167, 43.204, 45.22
8, 47.24, 49.243, 51.239, 53.229, 55.214, 57.194, 59.171, 61.146, 63.118, 65.
088, 67.057, 69.025, 70.991, 72.957, 74.922, 76.887, 78.851, 80.814, 82.778,
84.741, 86.704, 88.667, 90.629, 92.592, 94.554, 96.516, 98.479, 100.441, 102.
403, 104.365, 106.327, 108.29]
v_Runge-Ketta: [20.0, 18.153, 16.641, 15.402, 14.389, 13.559, 12.879, 12.323,
11.867, 11.494, 11.189, 10.939, 10.734, 10.567, 10.43, 10.317, 10.225, 10.15,
10.088, 10.038, 9.997, 9.963, 9.935, 9.912, 9.894, 9.879, 9.866, 9.856, 9.84
8, 9.841, 9.835, 9.831, 9.827, 9.824, 9.821, 9.819, 9.818, 9.816, 9.815, 9.81
4, 9.813, 9.813, 9.812, 9.812, 9.812, 9.811, 9.811, 9.811, 9.811, 9.811, 9.8
1]
v_exact: [20.0, 18.153, 16.641, 15.402, 14.389, 13.559, 12.879, 12.323, 11.86
7, 11.494, 11.189, 10.939, 10.734, 10.567, 10.43, 10.317, 10.225, 10.15, 10.0
88, 10.038, 9.997, 9.963, 9.935, 9.912, 9.894, 9.879, 9.866, 9.856, 9.848, 9.
841, 9.835, 9.831, 9.827, 9.824, 9.821, 9.819, 9.818, 9.816, 9.815, 9.814, 9.
813, 9.813, 9.812, 9.812, 9.812, 9.811, 9.811, 9.811, 9.811, 9.811, 9.81]
```


Runge-Ketta(stepsize = 0.2)
y-value(Exact Solution)
velocity(Exact Solution)
y-value(Runge-Ketta)
velocity(Runge-Ketta)
t


Error (step size = 0.2)
Error(y-values)
Error(velocity)
t