HW3

November 12, 2024

1 Homework Assignment 3

Fedir Vasyliev s234542

[1]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

2 Section 1

2.1 A) 2

2.2 B) 3

2.3 C) 2

2.4 D) 3

3 Section 2

3.1 1)

$$\frac{dx}{dt} = f(x) \tag{1}$$

The first equation we will need is taylor expansion of a function with 2 arguments.

$$f(t+h, x+k) = f(t,x) + (hf_t(t,x) + kf_x(t,x)) \tag{2}$$

Where x and t subscripts are the partial derivatives with respect to the given argument.

The next step will be expanding x(t + h) expression.

$$x(t+h) = x(t) + hx^{'}(t) + \frac{1}{2}h^2x^{''}(t) \tag{3}$$

$$x^{''}(t) = (x^{'}(t))^{'} = \frac{d}{dt}f(x) = f^{'}(x)x^{'}(t) = f^{'}(x)f(t)x(t+h) = x(t) + hx^{'}(t) + \frac{1}{2}h^2f^{'}(x)f(t) \tag{4}$$

$x(t+h)$ defined in Runge-Kutta method is

$$x(t+h) = x(t) + h(\frac{1}{2}K_1 + \frac{1}{2}K_2) \tag{5}$$

$$K_1 = f(t,x) \tag{6}$$

$$K_2 = f(t + \frac{h}{2}, x + hK_1) \tag{7}$$

K_2 can also be Taylor expanded by the equation given at the beginning.

$$K_2 = f(t,x) + (\frac{h}{2}f_t(t,x) + hK_1 f_x(t,x)) = f(t,x) + \frac{h}{2}f_t(t,x) + hf(t,x)f_x(t,x) \tag{8}$$

given that $f_t(t,x) = 0$ and $f(t,x) = f(x)$ for the same reason, we can rewrite the expression for K_2 as

$$K_2 = f(x) + hf(x)f^{'}(x) \tag{9}$$

Finally, we can substitute K_1 and K_2 into Runge-Kutta and get

$$x(t+h) = x(t) + \frac{1}{2}h(f(x) + f(x) + hf(x)f^{'}(x)) = x(t) + hx^{'}(t) + \frac{1}{2}h^2 f^{'}(x)f(t) \tag{10}$$

We ended up with the Taylor expansion expression.

3.2 2)

$$\frac{dx}{dt} = g(t) \tag{11}$$

We will do the same expansions as in the previous question.

$$x(t+h) = x(t) + hx^{'}(t) + \frac{1}{2}h^2 x''(t) \tag{12}$$

$$x''(t) = g^{'}(t)x(t+h) = x(t) + hg(t) + \frac{1}{2}h^2 g'(t) \tag{13}$$

$$x(t+h) = x(t) + h(\frac{1}{2}K_1 + \frac{1}{2}K_2) \tag{14}$$

$$K_1 = f(t) \tag{15}$$

$$K_2 = f(t + \frac{h}{2}) \tag{16}$$

Taylor expand K_2

$$K_2 = g(t + \frac{h}{2}) = g(t) + \frac{h}{2}g^{'}(t) \tag{17}$$

Now substitute K_1 and K_2 in Runge-Kutta.

$$x(t+h) = x(t) + \frac{1}{2}h(g(t) + g(t) + \frac{1}{2}hg^{'}(t)) = x(t) + hg(t) + \frac{1}{4}h^2 g^{'}(t) \tag{18}$$

As we can see the term with h^2 has an additional $\frac{1}{2}$ coefficient to it which makes the Runge-Kutta and Taylor expansion for $x(t+h)$ agree only up to and including the term with h.

4 Section 3

4.1 1)

$$x''(t) = x'(t) + x(t) - (2t-1)e^t, \quad 1 < t < 2 \tag{19}$$

$$x(1) = 3e, \quad x(2) = 5e^2 \tag{20}$$

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} x \\ x^{'} \end{pmatrix} \tag{21}$$

$$\begin{pmatrix} z_1^{'} \\ z_2^{'} \end{pmatrix} = \begin{pmatrix} z_2 \\ z2 + z1 - (2t-1)e^t \end{pmatrix} \tag{22}$$

[2]:
```
def sys(t,z):
    dz = np.zeros(2)
    dz[0] = z[1]
    dz[1] = z[1] + z[0] - (2*t-1)*np.exp(t)
    return dz
```

4.2 2)

[3]:
```
n = 50
tspan = [1,2]
z0 = np.array([3*np.exp(1), 10])
z1 = np.array([3*np.exp(1), 25])
```

[6]:
```
x0 = MyRK4System(sys, tspan, z0, n)
x1 = MyRK4System(sys, tspan, z1, n)
```

[7]:
```
print("z = 10 -> x(2) =", x0[1][50][0])
print("z = 25 -> x(2) =", x1[1][50][0])
```

```
z = 10 -> x(2) = 29.711023151333443
z = 25 -> x(2) = 59.92586366582473
```

One iteration of secant method

$$\mu_{k+1} = \mu_k - \frac{\mu_{k-1} - \mu_k}{\psi(\mu_{k-1}) - \psi(\mu_k)}\psi(\mu_k) \tag{23}$$

Where $\psi(\mu) = x(2, \mu) - 5e^2$

$\mu_{k-1} = 10$, $\mu_k = 25$, $\psi(\mu_{k-1}) = 29.7 - 5e^2$, $\psi(\mu_k) = 59.93 - 5e^2$

[8]:
```
def secant_update(mu_0, mu_1, x2_mu0, x2_mu1, target):
    psi_mu0 = x2_mu0 - target
    psi_mu1 = x2_mu1 - target
    mu_2 = mu_1 - psi_mu1 * (mu_1 - mu_0) / (psi_mu1 - psi_mu0)
    return mu_2
```

[9]:
```
desired_x = 5*np.exp(2)
z_d = secant_update(10, 25, x0[1][50][0], x1[1][50][0], desired_x)
z_d
```

[9]: 13.591409330714589

$$\hat{z} = 13.6 \tag{24}$$

4.3 3)

[10]:
```
z2 = np.array([3*np.exp(1), z_d])
```

[11]:
```
x2 = MyRK4System(sys, tspan, z2, n)
```

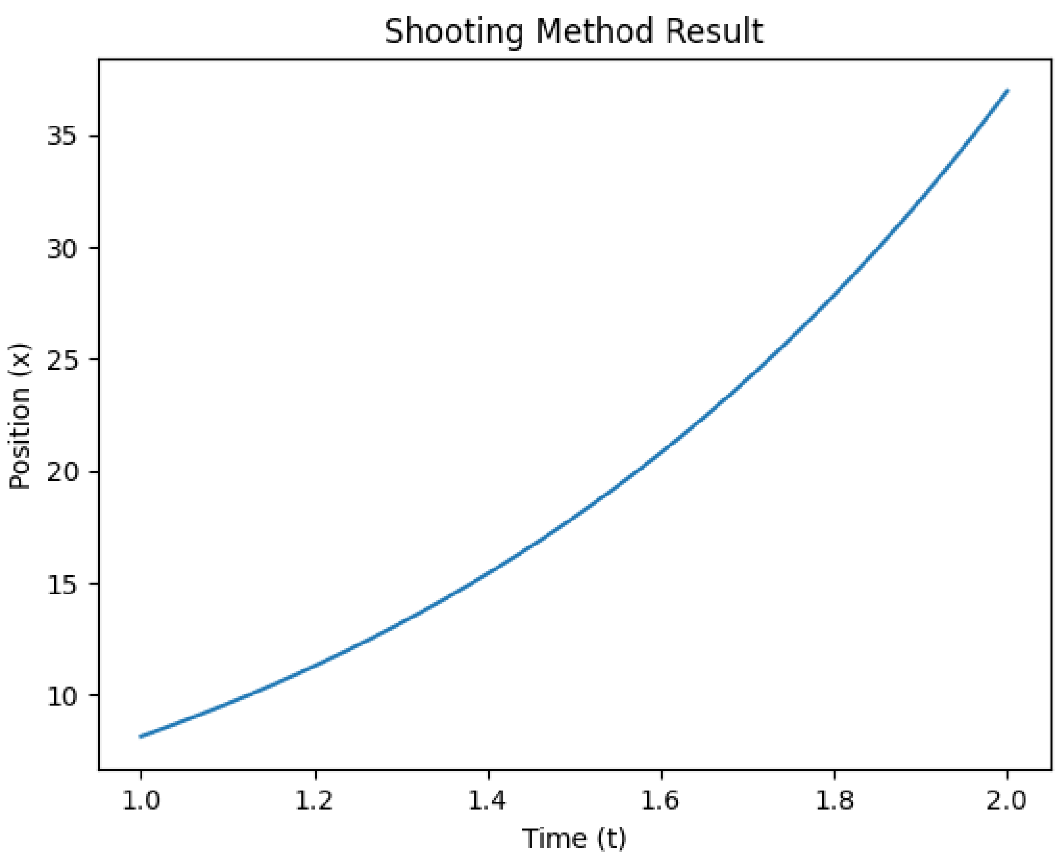
[15]:
```
print("z =", z_d, "-> x(2) =", x2[1][50][0])
print("desired value =", desired_x)
```

```
z = 13.591409330714589 -> x(2) = 36.94528049465326
desired value = 36.945280494653254
```

[16]:
```
data_x = np.array(x2[1])[:,0]
t = np.array(x2[0])
```

[17]:
```
plt.plot(t,data_x)
plt.xlabel('Time (t)')
plt.ylabel('Position (x)')
plt.title('Shooting Method Result')
```

[17]:
```
Text(0.5, 1.0, 'Shooting Method Result')
```



[5]:
```python
def MyRK4System(sys, tspan, x0, n):
    if x0.shape[0] != sys(tspan[0],x0).shape[0]:
        print("Wrong dimensions of x0 and system")
        return
    t = tspan[0]
    h = (tspan[1] - tspan[0])/n
    x = x0
    t_vec = []
    MX = []
    MX.append(x)
    t_vec.append(t)
    for i in range(1,n+1):
        K1 = h*sys(t, x)
        K2 = h*sys(t + 0.5*h, x + 0.5*K1)
        K3 = h*sys(t + 0.5*h, x + 0.5*K2)
        K4 = h*sys(t + h, x + K3)
        x = x + 1/6*(K1+2*K2+2*K3+K4)
        t = tspan[0] + i*h
        MX.append(x)
        t_vec.append(t)
    return t_vec, MX
```