Amath481 HW1 presentation skills

HAOLOU SUN

October 2022

1 Problem 1

1.1 Graph

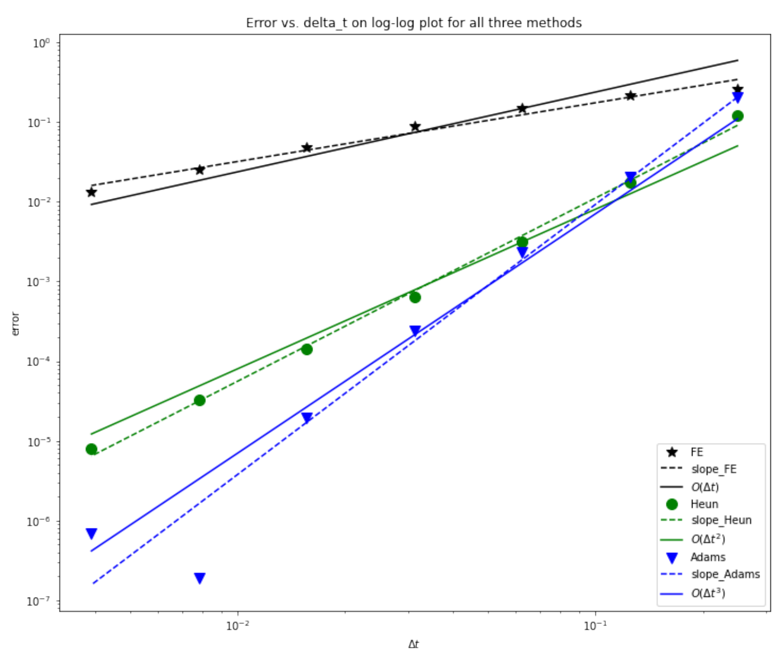


Figure 1: This is the error vs. Δt log-log plot for the Forward-Euler method(black points), Heun's method(green points) and Adams predictor-corrector method(blue points) with their corresponding slope(FE slope = 0.736, Heun slope = 2.296, Adams slope = 3.385). The other three lines are the $O(\Delta t)$, $O(\Delta t^2)$, $O(\Delta t^3)$ lines.

1.2 Code

```
import numpy as np
import math
import matplotlib.pyplot as plt
import scipy.integrate
f = lambda t, y: -3*y*np.sin(t) # dydt = f(t,y)
f_true = lambda t: math.pi*math.exp(3*(np.cos(t)-1))/math.sqrt(2)
delta_t = np.array([2**(-2), 2**(-3), 2**(-4), 2**(-5), 2**(-6), 2**(-7), 2**(-8)])

def FE(tn, yn, delta_t):
    yn_plus_1 = yn + delta_t*f(tn, yn)
    return yn_plus_1

def y5_FE(delta_t):
    result = []
    t=0
    y = math.pi/math.sqrt(2)
    result.append(y)
    while t < 5:
            yn_plus_1 = FE(t, y, delta_t)
            result.append(yn_plus_1)
            y = yn_plus_1
            t = t + delta_t
    return result

y_N_FE = np.zeros(7)
for i in range(7):
    y_N_FE[i] = y5_FE(delta_t[i])[-1]

y_true = f_true(5)
error_FE = np.zeros(7)
for i in range(7):
    error_FE[i] = abs(y_true - y_N_FE[i])

slope_FE, intercept_FE = np.polyfit(np.log(delta_t), np.log(error_FE), 1)
print(slope_FE)
def Heun_method(tn, yn, delta_t):
    yn_plus_1 = yn + delta_t/2*(f(tn, yn) + f(tn+delta_t, yn+delta_t*f(tn,yn)))
    return yn_plus_1
def y5_Heun(delta_t):
    result = []
    t=0
    y = math.pi/math.sqrt(2)
    result.append(y)
    while t < 5:
```

```
            yn_plus_1 = Heun_method(t, y, delta_t)
            result.append(yn_plus_1)
            y = yn_plus_1
            t = t + delta_t

    return result
y_N_Heun = np.zeros(7)

for i in range(7):
    y_N_Heun[i] = y5_Heun(delta_t[i])[-1]

y_true = f_true(5)
error_Heun = np.zeros(7)
for i in range(7):
    error_Heun[i] = abs(y_true - y_N_Heun[i])

slope_Heun, intercept_Heun = np.polyfit(np.log(delta_t), np.log(error_Heun), 1)
print(slope_Heun)

def Adams(tn, yn_1, yn, delta_t):
    y_p = yn + delta_t/2*(3*f(tn, yn) - f(tn-delta_t, yn_1))
    yn_plus_1 = yn + delta_t/2*(f(tn+delta_t, y_p) + f(tn, yn))
    return yn_plus_1

def y5_Adams(delta_t):
    result = []
    t = delta_t
    yn_1 = math.pi/math.sqrt(2)
    y = yn_1 + delta_t*f(0+delta_t/2, yn_1+delta_t/2*f(0, yn_1))
    result.append(yn_1)
    result.append(y)
    while t < 5:
        yn_plus_1 = Adams(t, yn_1, y, delta_t)
        result.append(yn_plus_1)
        yn_1 = y
        y = yn_plus_1
        t = t + delta_t
    return result

y_N_Adams = np.zeros(7)
for i in range(7):
    y_N_Adams[i] = y5_Adams(delta_t[i])[-1]

y_true = f_true(5)
error_Adams = np.zeros(7)
for i in range(7):
```

```
    error_Adams[i] = abs(y_true - y_N_Adams[i])

slope_Adams, intercept_Adams = np.polyfit(np.log(delta_t), np.log(error_Adams), 1)
print(slope_Adams)

fig, ax = plt.subplots(1,1)
fig.set_size_inches(12, 10, True)
ax.loglog(delta_t, error_FE, 'k*', markersize=10, label='FE')
ax.loglog(delta_t, (delta_t**slope_FE)*np.exp(intercept_FE), 'k--', label='slope_FE')
ax.loglog(delta_t, (2.5*delta_t**1)*np.exp(intercept_FE),'k-', label='$O(\Delta t)$')
ax.loglog(delta_t, error_Heun, 'g.', markersize=20, label='Heun')
ax.loglog(delta_t, (delta_t**slope_Heun)*np.exp(intercept_Heun),
    'g--', label='slope_Heun')
ax.loglog(delta_t, 0.8*(delta_t**2), 'g-', label='$O(\Delta t^2)$')
ax.loglog(delta_t, error_Adams, 'bv', markersize=10, label='Adams')
ax.loglog(delta_t, (delta_t**slope_Adams)*np.exp(intercept_Adams),
    'b--', label='slope_Adams')
ax.loglog(delta_t, 7*(delta_t**3), 'b-', label='$O(\Delta t^3)$')

plt.xlabel('$\Delta t$')
plt.ylabel('error')
plt.title('Error vs. delta_t on log-log plot for all three methods')
plt.legend()
plt.savefig('error_loglog_plot.png')
plt.show()
```

2 Problem 3

2.1 Model equations

The two neurons Fitzhugh model is given by the system of ODEs:

$$\frac{dv_1}{dt} = -v_1^3 + (1 + a_1)v_1^2 - a_1v_1 - w_1 + I + d_{12}v_2 \tag{1}$$

$$\frac{dw_1}{dt} = bv_1 - cw_1 \tag{2}$$

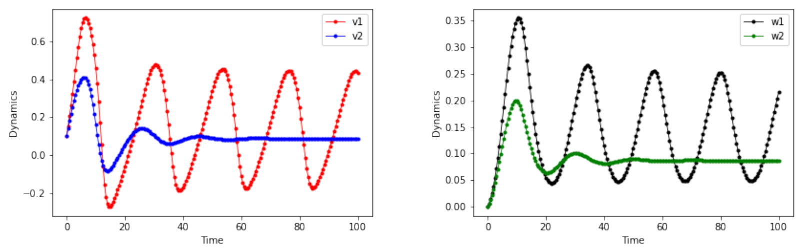
$$\frac{dv_2}{dt} = -v_2^3 + (1 + a_2)v_2^2 - a_2v_2 - w_2 + I + d_{21}v_1 \tag{3}$$

$$\frac{dw_2}{dt} = bv_2 - cw_2 \tag{4}$$

where a_1, a_2, b, c and I are parameters. v_1, v_2, w_1 and w_2 are the dependent variables representing voltage in a neuron and the relaxation potential respectively. d_{12} and d_{21} are constants representing the interaction between two neurons.

For this experiment we solved the Fitzhugh model with $a_1 = 0.05$, $a_2 = 0.25$, b = c = 0.1, and I = 0.1. We used the initial condition ($v_1(0)$, $v_2(0)$) = (0.1, 0.1), ($w_1(0)$, $w_2(0)$ = (0, 0)). To solve the IVP we used python's **scipy.integrate.solve_ivp(..., method = 'BDF')**. Then we plot the solutions for each of the five different sets of interaction parameters.((d_{12}, d_{21}) = (0, 0), (0, 0.2), (-0.1, 0.2), (-0.3, 0.2), and (-0.5, 0.2) on time span [0, 100])

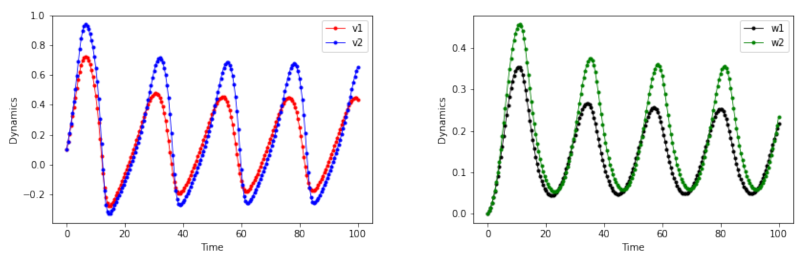
2.2 Graphs



(a) v_1, v_2 plot

(b) w_1, w_2 plot

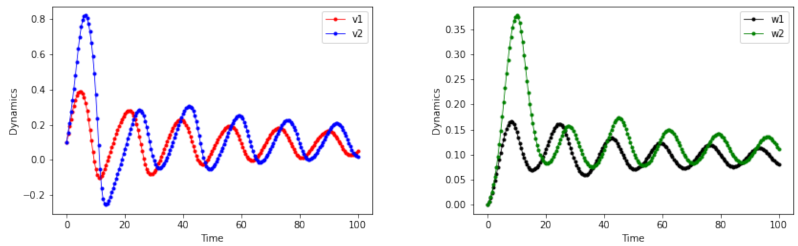
Figure 2: These are the v_1, v_2 vs. time plot and w_1, w_2 vs. time plot when (d_{12}, d_{21}) = (0, 0)



(a) v_1, v_2 plot

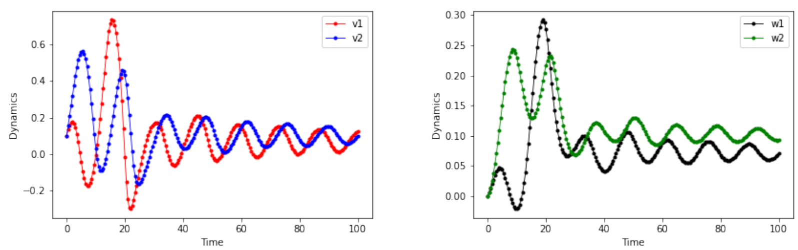
(b) w_1, w_2 plot

Figure 3: These are the v_1, v_2 vs. time plot and w_1, w_2 vs. time plot when (d_{12}, d_{21}) = (0, 0.2)



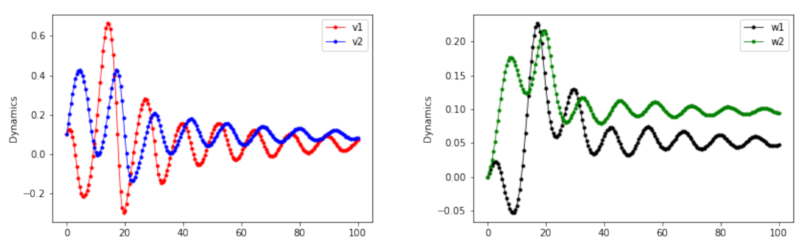
(a) v_1, v_2 plot (b) w_1, w_2 plot

Figure 4: These are the v_1, v_2 vs. time plot and w_1, w_2 vs. time plot when (d_{12}, d_{21}) = (-0.1, 0.2)



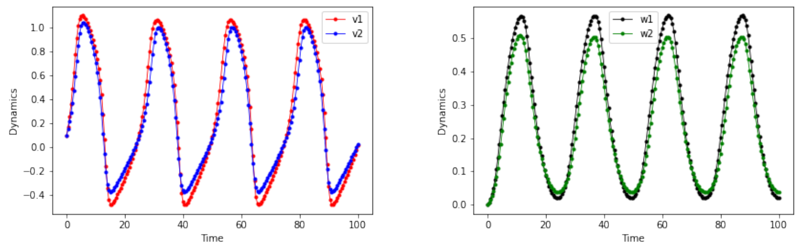
(a) v_1, v_2 plot (b) w_1, w_2 plot

Figure 5: These are the v_1, v_2 vs. time plot and w_1, w_2 vs. time plot when (d_{12}, d_{21}) = (-0.3, 0.2)



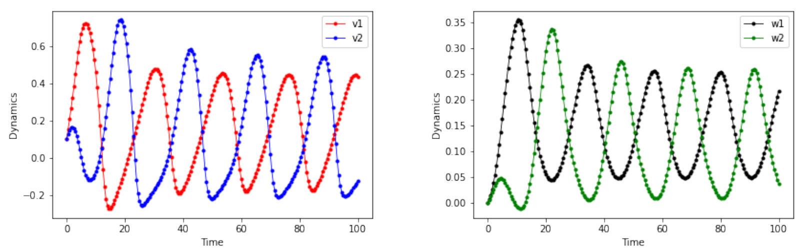
(a) v_1, v_2 plot (b) w_1, w_2 plot

Figure 6: These are the v_1, v_2 vs. time plot and w_1, w_2 vs. time plot when (d_{12}, d_{21}) = (-0.5, 0.2)



(a) v_1, v_2 plot (b) w_1, w_2 plot

Figure 7: These are the v_1, v_2 vs. time plot and w_1, w_2 vs. time plot when $(d_{12}, d_{21}) = (0.3, 0.2)$



(a) v_1, v_2 plot (b) w_1, w_2 plot

Figure 8: These are the v_1, v_2 vs. time plot and w_1, w_2 vs. time plot when $(d_{12}, d_{21}) = (0, \text{-}0.2)$

2.3 Discussion

From the seven figures above, we can see that the v_1, v_2 plot has quite similar dynamics tendency with the w_1, w_2 figures. Thus, for the following discussion, we will just focus on the v_1, v_2 plot.

For Figure 2, Figure 3 and Figure 8, we can see that one of the two constants is the same and the other one is the same magnitudes with different sign. Then from the graph, we can see that if the constant has a positive value, then v_1 and v_2 will move closer to each other. If the constant has a negative value, then v_1 and v_2 will move further away from each other. We can also see this rule by comparing Figure 5 and Figure 7.

For Figure 3, Figure 4, Figure 5, and Figure 6, we can see that they have the same v_2 value and have smaller and smaller v_1 value. From the graph, we can see that the first trough of v_1(red curve) is moving to the left. In Figure 3, the peaks and troughs of the v_1 and v_2 occur at nearly the same time point.

While In Figure 6, the first trough of v_1 occurs nearly at the same time point of the first peak of v_2.

Therefore, from the above analysis, we can conclude that the sign of the interaction constant affects the similarity between the two neurons. If the sign is positive, the two neurons will have more similar voltage. If the sign is negative, the two neurons will have less similar voltage. The magnitude of the interaction constant affects the phase of the two neurons. If the magnitude is small, then the time of occurrence of peaks or troughs for the two neurons will be close. If the magnitude is large, then the time difference of occurrence of peaks or troughs will be great until the peak occurs for one neuron while at the same time, the trough occurs for the other one.

In the future, we may try more pairs of d_{12} and d_{21}. There are three cases of values for d_{12} and d_{21} - negative value, zero, and positive value. In the above discussion, we have seen the following cases: both d_{12} and d_{21} are zeros, d_{12} zero and d_{21} positive, d_{12} zero and d_{21} negative, both d_{12} and d_{21} are positive, d_{12} negative and d_{21} positive. There are four more cases we have not shown. If we draw more graphs of these cases with different amplitudes, we can get a more precise conclusion.