

Contents

- [%Problem 1 \(b\)](#)
- [Problem 1\(c\)](#)
- [Problem 1\(d\)](#)
- [Problem 2\(c\)](#)
- [Problem 3\(c\)](#)
- [Problem 4](#)

%Problem 1 (b)

```
V_thresh = 20e-3;
El = -70e-3;
I=input('Input Current Matrix :\n');
T = input('total time of consideration : \n');
V_t = runge_kutta(I, V_thresh, El, T/size(I,2));
disp(V_t);
```

```
Error using input
Cannot call INPUT from EVALC.
```

```
Error in main (line 4)
I=input('Input Current Matrix :\n');
```

Problem 1(c)

```
V_thresh = 20e-3;
El = -70e-3;
alpha = 0.1;
T = 500e-3;
del_t = 0.1e-3;
I_c = 2.7e-9; % calculated in part 1 (a)
m = (T/del_t); % 500/0.1 + 1;
n = 10;
I = ones(n,1);
%I(:,1)=zeros(n,1);
for i = 2:n
    I(i,1) = (1 + i*alpha).*I_c;
end
A = I*ones(1,m);
V_t = runge_kutta(A, V_thresh, El, del_t);
t = 0:0.1:T;
%disp(size(V_t));
hold on
ylim([-0.1 0.05]);
subplot(4,1,1);
plot(V_t(2,:), 'Linewidth', 2)
title('2nd neuron');
xlabel('Time (in 100us)');
ylabel('Spikes');
subplot(4,1,2);
plot(V_t(4,:), 'Linewidth', 2)
title('4th neuron');
xlabel('Time (in 100us)');
ylabel('Spikes');
subplot(4,1,3);
plot(V_t(6,:), 'Linewidth', 2)
title('6th neuron');
xlabel('Time (in 100us)');
ylabel('Spikes');
subplot(4,1,4);
plot(V_t(8,:), 'Linewidth', 2)
```

```

title('8th neuron');
xlabel('Time (in 100us)');
ylabel('Spikes');
%legend('n=2','n=4','n=6','n=8')
%title('LIF Model')
hold off

```

Problem 1(d)

```

Neuron = [1,2,3,4,5,6,7,8,9,10];
spike_time = [240,181.667,149,127.667,111.67 , 101,90.33 , 84, 77, 71 ]; %Manually calculated from above part!
plot(Neuron, spike_time, 'linewidth', 2);
title('Average time interval between spikes');
xlabel('Neuron label');
ylabel('Time (in 100us) ');

```

Problem 2(c)

```

El = -70e-3;
alpha = 0.1;
del_t = 0.1e-3;
I_c = 2.7e-9; % calculated in part 1 (a)
kz = 0;
C = 0;
Er = 0;
Et = 0;
a = 0;
b = 0;
c = 0;
d = 0;
U_steady = 0;
V_steady = 0;
V_thresh = 0;
n = 10;
I_app = 400e-12;
T = 100e-3;
m = (T/del_t); % 500/0.1 + 1;
type1 = 1;
type2 = 2;
type3 = 3;
I = I_app*ones(n,1);
A = I*ones(1,m);
V_t1 = runge_kutta_fourth_order(A, type1, del_t);
V_t2 = runge_kutta_fourth_order(A, type2, del_t);
V_t3 = runge_kutta_fourth_order(A, type3, del_t);
t = 0:0.1:T;
%disp(size(V_t));
hold on
plot(V_t1(4,:), 'Linewidth', 2)
plot(V_t2(4,:), 'Linewidth', 2)
plot(V_t3(4,:), 'Linewidth', 2)
legend('RS','IB','CH')
title('Izhikevich Model')
xlabel('Time (in 100 us)');
ylabel('Spikes');
hold off

```

Problem 3(c)

```

Iapp = 150e-12;
T = 500e-3;
%N = 10;
del_t = 1e-4;
%type = 1 - RS, 2 - IB, 3 - CH

```

```

for j = 1:3
    Iapp = Iapp + 100e-12;
    %hold on;
    figure(j);
    sgtitle(sprintf('Current Applied: %f pA', Iapp*1e12));
for i = 1:3
if (i == 1)
    neuron_type = 'RS';
elseif (i==2)
    neuron_type = 'IB';
else
    neuron_type = 'CH';
end
subplot(3,1,i);
V_t = euler_method(i,Iapp,del_t);
plot(V_t, 'linewidth', 2);
title(sprintf('Neuron: %s',neuron_type));
xlabel('Time (in 100us)');
ylabel('Spikes');
end
%hold off;
end

```

Problem 4

```

%%Problem 4a
T = 50e-3;
del_t = 0.01e-3;
s = 5000;
I_t = zeros(s,1);
Io = 15e-6;
for i = 2000:3000
    I_t(i) = Io;
end
[V_t, I_na, I_k, I_l] = hh_model(I_t, del_t);
figure(1)
subplot(2,1,1)
plot(I_t, 'linewidth', 2);
title('External Current vs time');
xlabel ('Time (in 10us)');
ylabel ('Current (in A/cm2)');
subplot(2,1,2)
plot(V_t, 'linewidth', 2);
title('Membrane Potential vs time');
xlabel ('Time (in 10us)');
ylabel ('Voltage (in V)');
figure(2)
plot(I_t, 'linewidth', 2);
hold on;
plot(I_na, 'linewidth', 2);
hold on;
plot(I_k, 'linewidth', 2);
hold on;
plot(I_l, 'linewidth', 2);
hold on;
legend('External Current', 'Sodium Current', 'Potassium current', 'Leaky current');
title('Currents vs time');
xlabel ('Time (in 10us)');
ylabel ('Current (in A/cm2)');
%%Problem 4(b)
P_na = I_na.*(V_t - 50e-3);
P_k = I_k.*(V_t + 77e-3);
P_l = I_l.*(V_t + 55e-3);
P_mc = V_t.*(I_t - I_na-I_k-I_l);
figure(3)
plot(P_na, 'linewidth', 2);
hold on;

```

```
plot(P_k, 'linewidth', 2);
hold on;
plot(P_l, 'linewidth', 2);
hold on;
plot(P_mc, 'linewidth', 2);
title('Instantaneous Power vs Time');
legend('Sodium Ion', 'Potassium Ion', 'Leaky Ions', 'Membrane Capacitor');
xlabel('Time (in 10us)');
ylabel('Power (in W/cm2)');
%%Problem 4(c)
P_total = (P_na + P_k + P_l + P_mc)*100; %100 factor for Area of 1um2
E_total = zeros(s,1);
for i = 2:s
    E_total(i) = E_total(i-1) + P_total(i-1)*del_t;
end
figure(4)
plot(E_total, 'linewidth', 2);
title(sprintf('Total Energy dissipated: %d J', E_total(5000)))
ylabel('Energy (in J)');
xlabel('Time (in 10us)');
```

Published with MATLAB® R2019a