Assignment 1 Solutions Neuromorphic Engineering
Bipin Rajendran

Spiking Neuron Models

Problem 1: Leaky Integrate and Fire Model

(a) At steady state,

$$C\frac{dV(t)}{dt} = -g_L(V(t) - E_L) + I_{app}(t) = 0 \tag{1}$$

Hence,

$$V_s = \frac{I_0}{g_L} + E_L \tag{2}$$

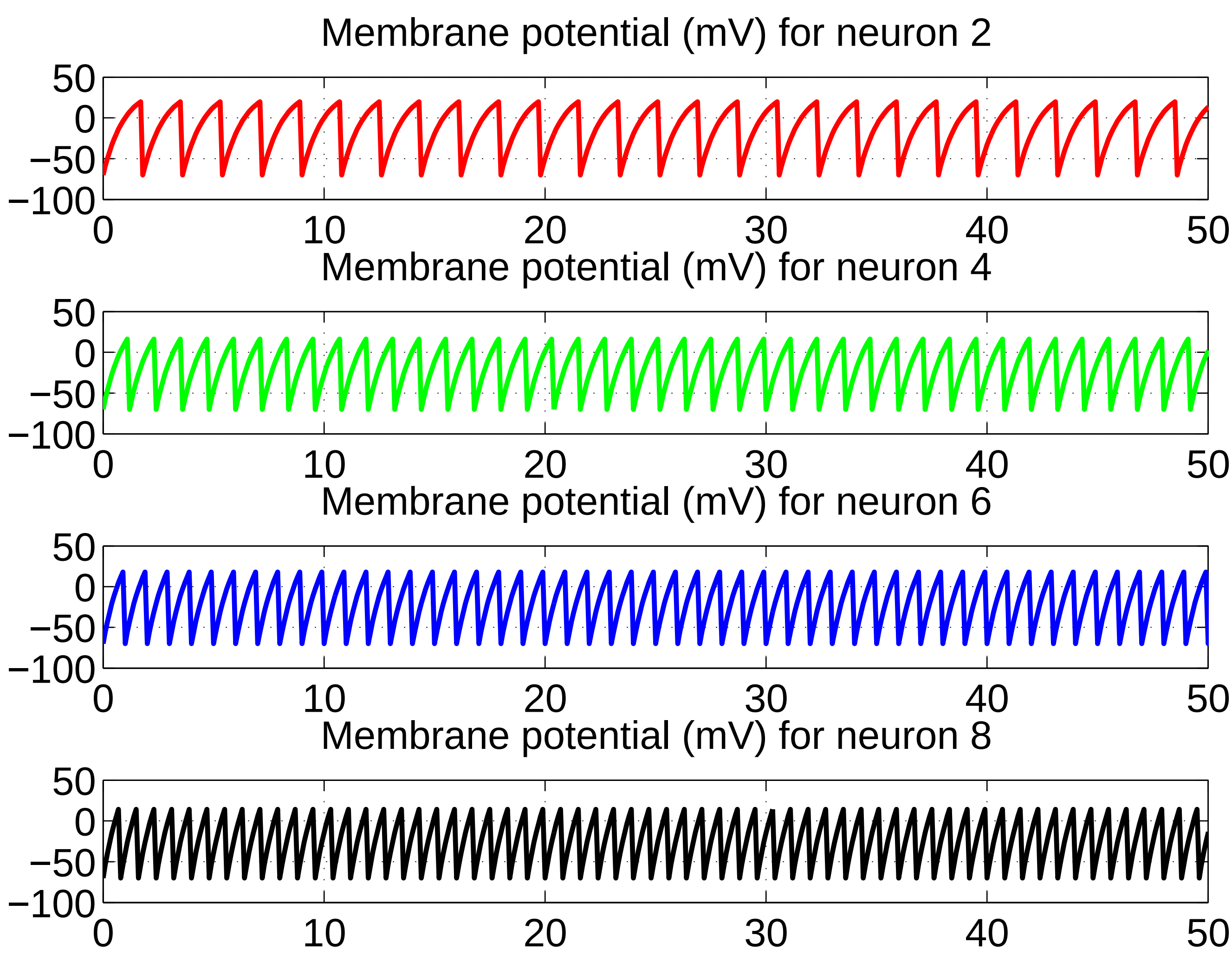
A spike is issued only if $V_s \geq V_T$. Hence, Hence,

$$I_c = g_L(V_T - E_L) \tag{3}$$

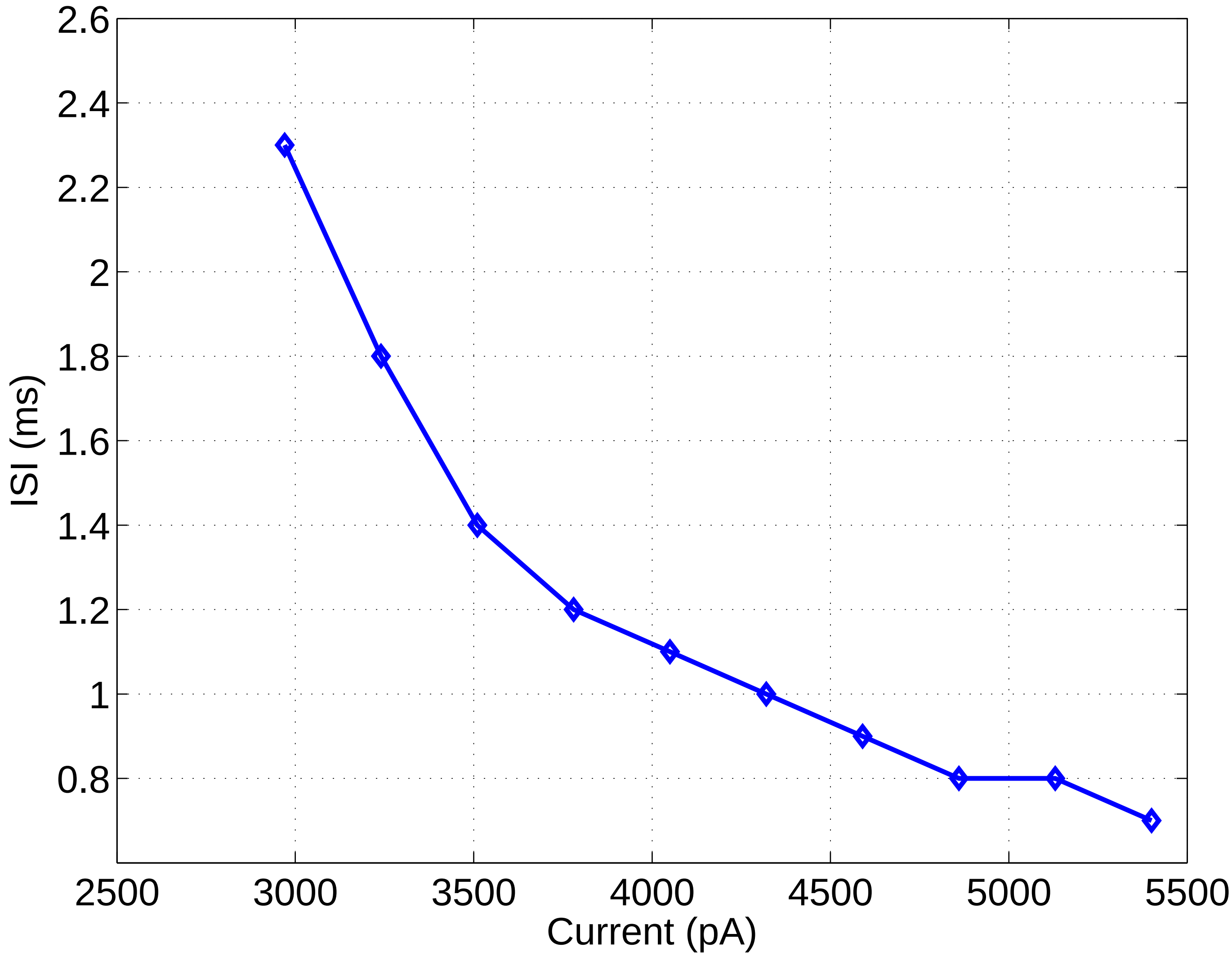
Conductance is given in units of nS and voltages in units of mV, the units of the current will be pA. Plugging in the numbers, we get $I_c = 2700$ pA.

(b) Matlab function is below:

```
[xa,V]=simulate_lif(N,T,dt,Iapp)

iter=ceil(T/dt);
C=300;
gL=30;
EL=-70;
VT=20;

Ic=gL*(VT-EL);
iter=ceil(T/dt);
V=EL*ones(N,iter);

j=1; xa(1)=0;

for j=1:iter-1
    V(:,j+1)=V(:,j)+ (dt/C)* (Iapp(:,j) - (gL*(V(:,j)-EL)));
    spind=sign(V(:,j+1)-VT);
    if max(spind)>0
        ind=find(spind>0); %indices of neuron that just spiked
        V(ind,j+1)=EL;
    end
    xa(j+1)=xa(j)+dt;

end
```



Membrane potential (mV) for neuron 2
50
0
−50
−100
0
10
20
30
40
50
Membrane potential (mV) for neuron 4
50
0
−50
−100
0
10
20
30
40
50
Membrane potential (mV) for neuron 6
50
0
−50
−100
0
10
20
30
40
50
Membrane potential (mV) for neuron 8
50
0
−50
−100
0
10
20
30
40
50



2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
ISI (ms)
2500
3000
3500
4000
4500
5000
5500
Current (pA)

Problem 2: Izhikevich Model

(a) The steady state values are given by

$$\begin{aligned} 0 &= k_z(V_s - E_r)(V_s - E_t) - U_s && (4) \\ 0 &= a\left[b(V_s - E_r) - U_s\right] && (5) \end{aligned}$$

Clearly, $V_s = E_r, U_s = 0$ is a solution. Also,

$$\begin{aligned} V_s &= \frac{b}{k_z} + E_t && (6) \\ U_s &= b\left(\frac{b}{k_z} + E_t - E_r\right) && (7) \end{aligned}$$

is a solution.

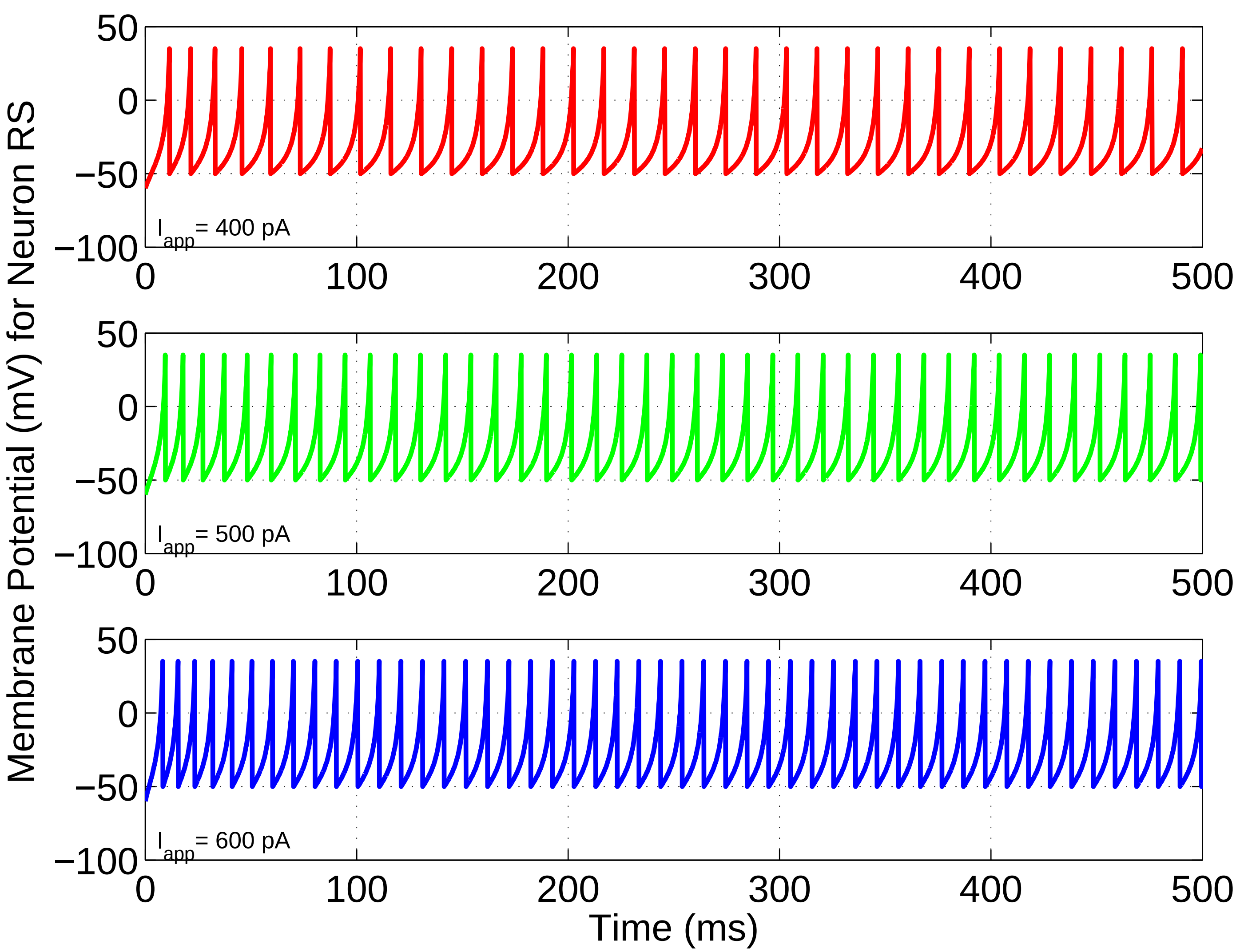
(b) The equivalent difference equations are

$$\begin{aligned} C\frac{V(j+1) - V(j)}{\Delta t} &= k_z(V(j) - E_r)(V(j) - E_t) - U(j) + I_{app}(j) && (8) \\ \frac{U(j+1) - U(j)}{\Delta t} &= a[b(V(j) - E_r) - U(j)] && (9) \end{aligned}$$

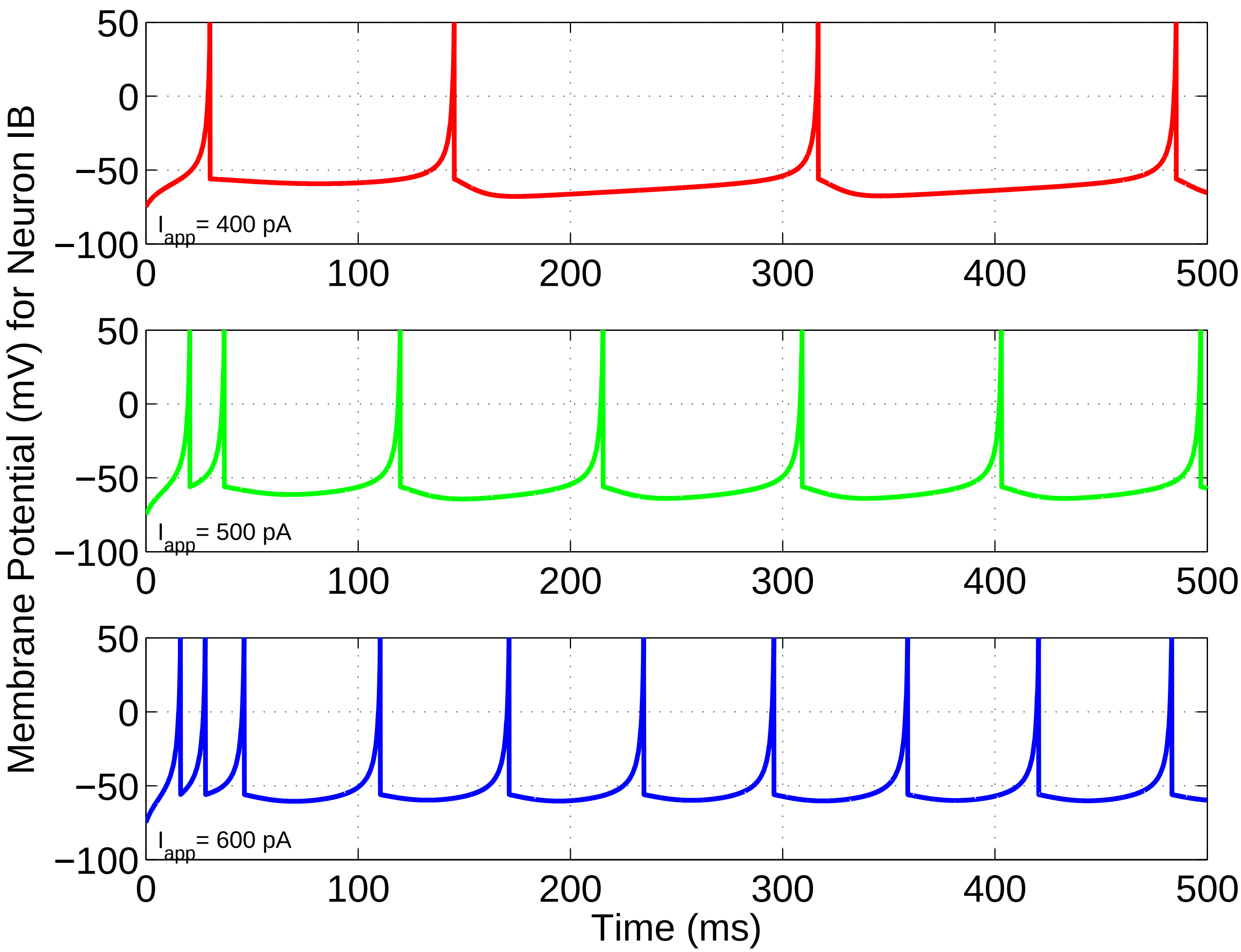
(c) The key in obtaining the correct behavior in these simulations is to have consistent set of units. One way to ensure this is to do a dimensional analysis check. With the numbers as defined below, every term in (8) has the units of pA and every term in (9) has the units of nA/s.

```
function [xa,V]=simulate_iz(N,T,dt,Iapp,type)
iter=ceil(T/dt);


if type==1
    st=['RS'];
    C=100;
    gz=0.7;
    Er=-60;
    Et=-40;
    a=0.03;
    b=-2;
    c=-50;
    d=100;
    vpeak=35;
elseif type==2
    st=['IB'];
    C=150;
    gz=1.2;
    Er=-75;
    Et=-45;
    a=0.01;
```

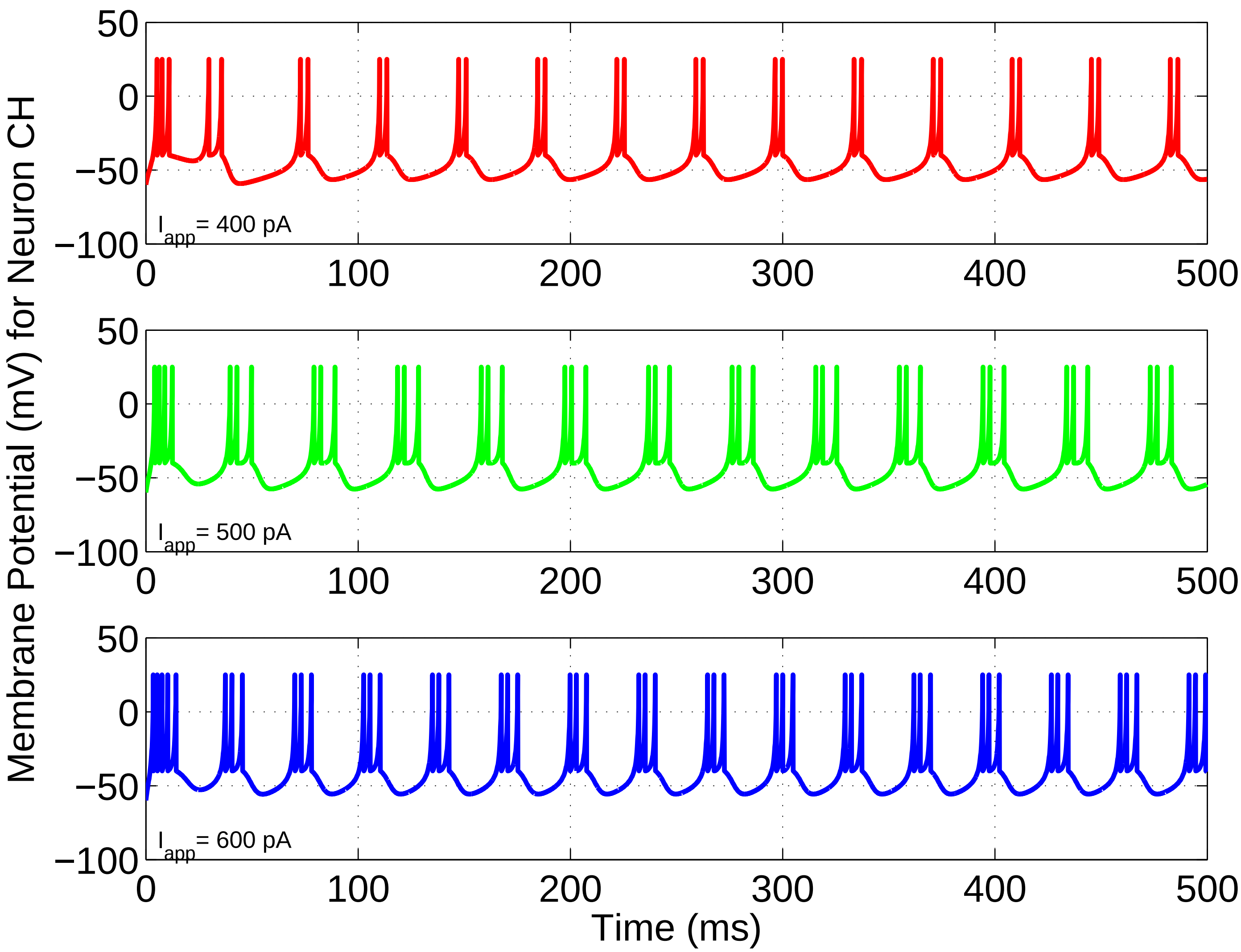
```
    b=5;
    c=-56;
    d=  130;
    vpeak=50;
else
    st=['CH'];
    C=50;
    gz=1.5;
    Er=-60;
    Et=-40;
    a=0.03;
    b=1;
    c=-40;
    d=150;
    vpeak=25;
end

iter=ceil(500/dt);
V=Er*ones(N,iter);
U=0*ones(N,iter);

j=1;xa(1)=0;

for j=1:iter
    V(:,j+1)=V(:,j)+ (dt/C)* (Iapp(:,j) + (gz*(V(:,j)-Er).*(V(:,j)-Et)) - U(:,
    U(:,j+1)=U(:,j)+ (a*dt )* (-U(:,j)  + (b*(V(:,j)-Er)) );

    spind=sign(V(:,j+1)-vpeak);
    if max(spind)>0
        ind=find(spind>0); %indices of neuron that just spiked
        V(ind,j)=vpeak;
        V(ind,j+1)=c;
        U(ind,j+1)=U(ind,j+1)+d;
    end
    xa(j+1)=xa(j)+dt;



end
```



Membrane Potential (mV) for Neuron RS
Time (ms)
I_{app}= 400 pA
I_{app}= 500 pA
I_{app}= 600 pA
50
0
−50
−100
0
100
200
300
400
500



Membrane Potential (mV) for Neuron IB
Time (ms)
I_app = 400 pA
I_app = 500 pA
I_app = 600 pA
50
0
−50
−100
0
100
200
300
400
500



Membrane Potential (mV) for Neuron CH
50
0
−50
−100
0
100
200
300
400
500
I_app = 400 pA
I_app = 500 pA
I_app = 600 pA
Time (ms)

Problem 3: Adaptive Exponential Integrate-and-Fire Model

(a) The equivalent difference equations are

$$C\frac{V(j+1)-V(j)}{\Delta t} = -g_L(V(j)-E_L) + g_L\Delta_T \exp\left(\frac{V(j)-V_T}{\Delta_T}\right) - U(j) + I_{app}(j) \quad (10)$$

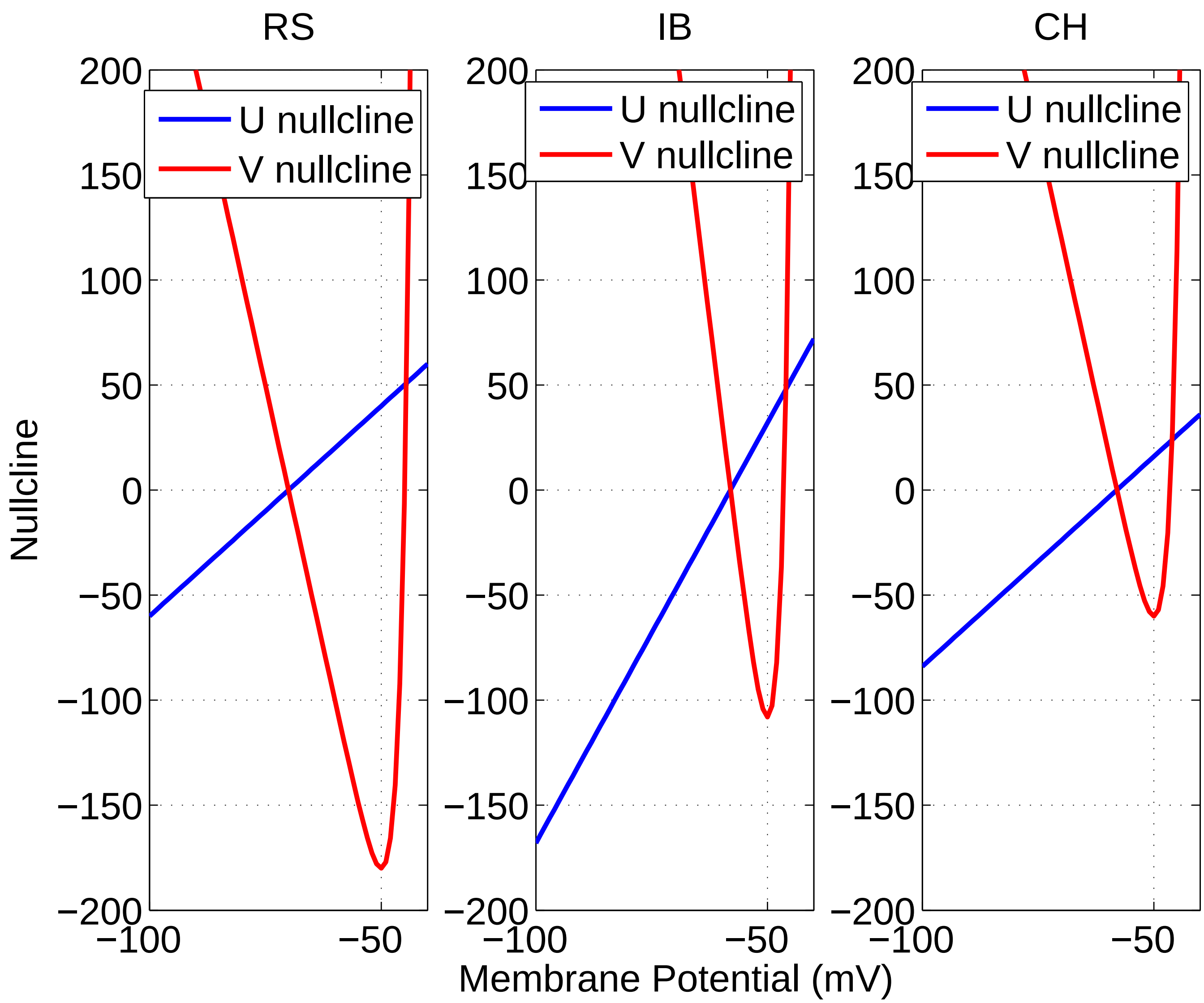
$$\tau_w \frac{U(j+1)-U(j)}{\Delta t} = a\,[V(j)-E_L] - U(j) \quad (11)$$

(b) To determine the steady state values of V and U for $I_{app} = 0$, we have to solve the following equations consistently.

$$U_s = -g_L(V_s - E_L) + g_L\Delta_T \exp\left(\frac{V_s - V_T}{\Delta_T}\right) \quad (12)$$

$$U_s = a\,[V_s - E_L] \quad (13)$$

The plot of U_s vs V_s for the two equations is given below. The points of intersections of the two lines are the critical points to the two differential equations. The resting potential is the intersection point where the two lines have opposite slopes. The other point of intersection is the critical point where an action potential is initiated.



RS
IB
CH
U nullcline
V nullcline
200
150
100
50
0
−50
−100
−150
−200
−100
−50
Nullcline
Membrane Potential (mV)

```
function [Vs,Us]=aef_steady(ntype)

if ntype==1
    gL=10;
    EL=-70;
    Dt=2;
    a=2;
elseif ntype==2
    gL=18;
    EL=-58;
    Dt=2;
    a=4;
else
    gL=10;
    EL=-58;
    Dt=2;
    a=2;
end

v1=-50; v2=-100;

diff=abs(v2-v1); j=1;

while diff>1e-6
diff=abs(v2-v1);

    j=j+1;
    vm(j)=(v1+v2)/2;
    dif1=a*(v1-EL)- ((gL*Dt*exp((v1-Vt)/Dt)) - (gL*(v1-EL))) ;
    dif2=a*(v2-EL)- ((gL*Dt*exp((v2-Vt)/Dt)) - (gL*(v2-EL))) ;
    difm(j)=a*(vm(j)-EL)- ((gL*Dt*exp((vm(j)-Vt)/Dt)) - (gL*(vm(j)-EL))) ;

if difm(j)*dif2>0
    v2=vm(j);
else
    v1=vm(j);
end

end

Vs=(v1+v2)/2;
Us=a*(Vs-EL);
```

From this, we obtain the steady state solutions as

RS: $V_s = -69.9999\text{mV}$, $U_s = 1.5102e - 004\text{mV}$
IB: $V_s = -57.9696\text{mV}$, $U_s = 0.1217\text{mV}$
CH: $V_s = -57.9690\text{mV}$, $U_s = 0.0620\text{mV}$

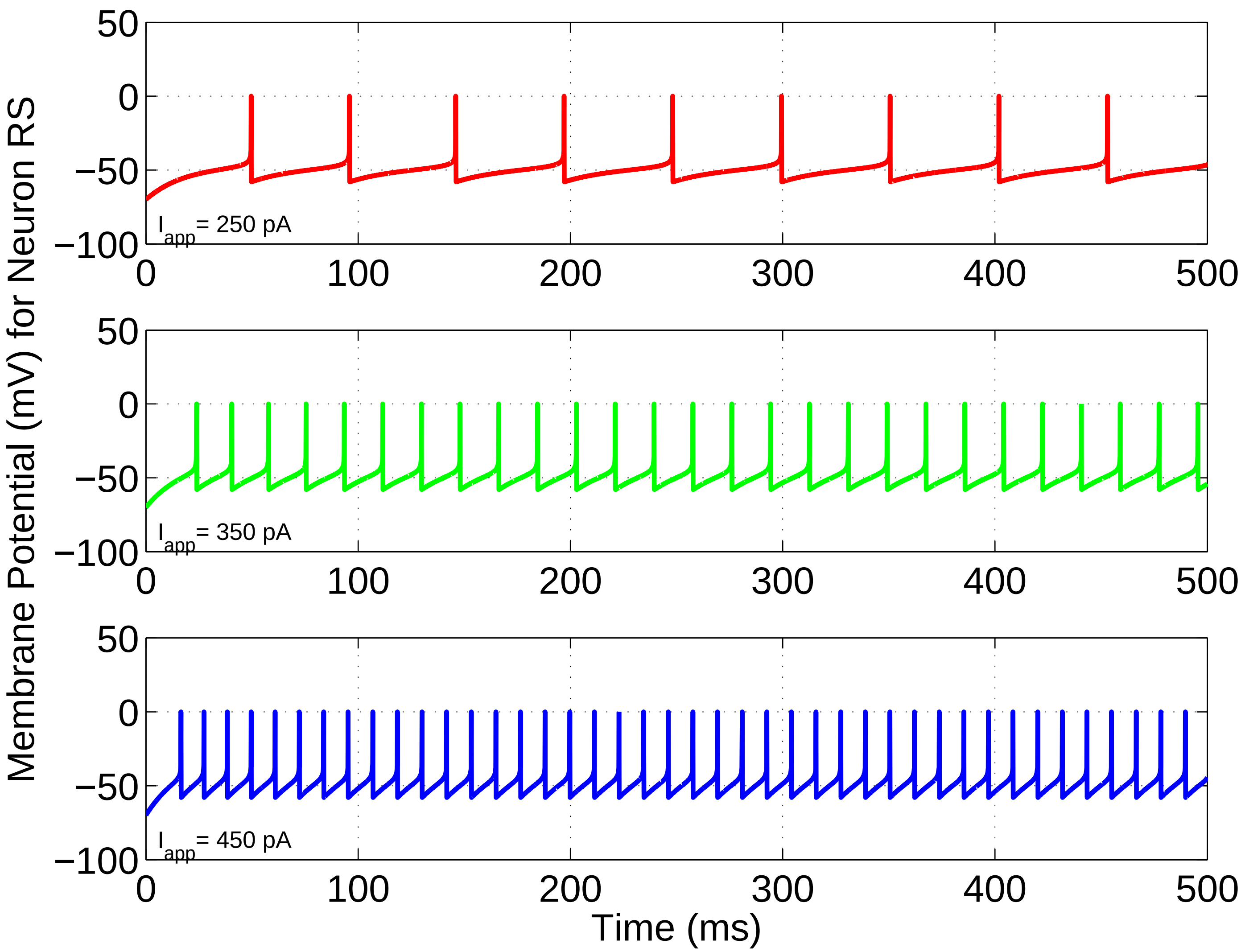
The other critical points are

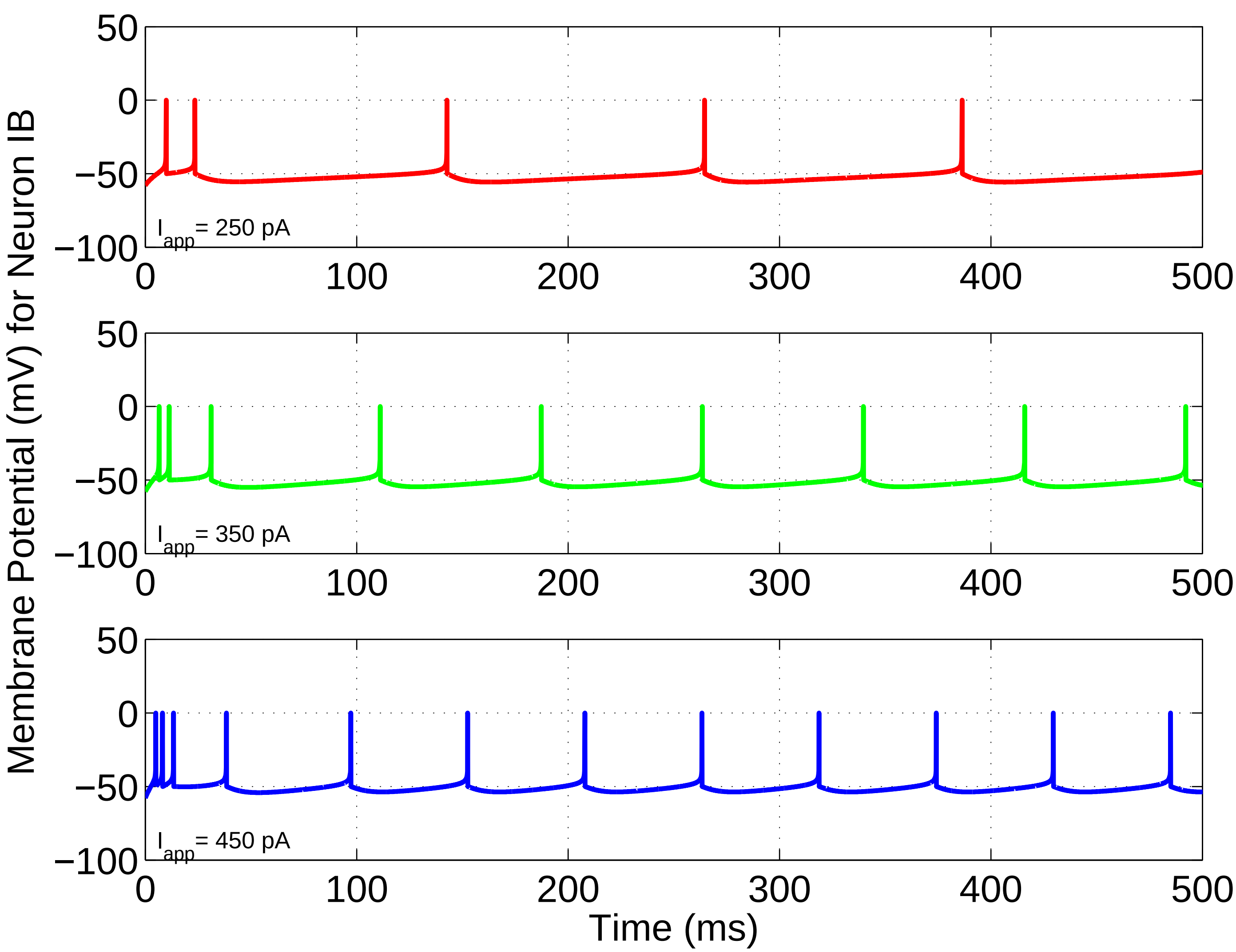
RS: $V_s = -44.5481\text{mV}$, $U_s = 50.9039\text{mV}$
IB: $V_s = -46.0182\text{mV}$, $U_s = 47.9273\text{mV}$
CH: $V_s = -46.0622\text{mV}$, $U_s = 23.8755\text{mV}$

(c)

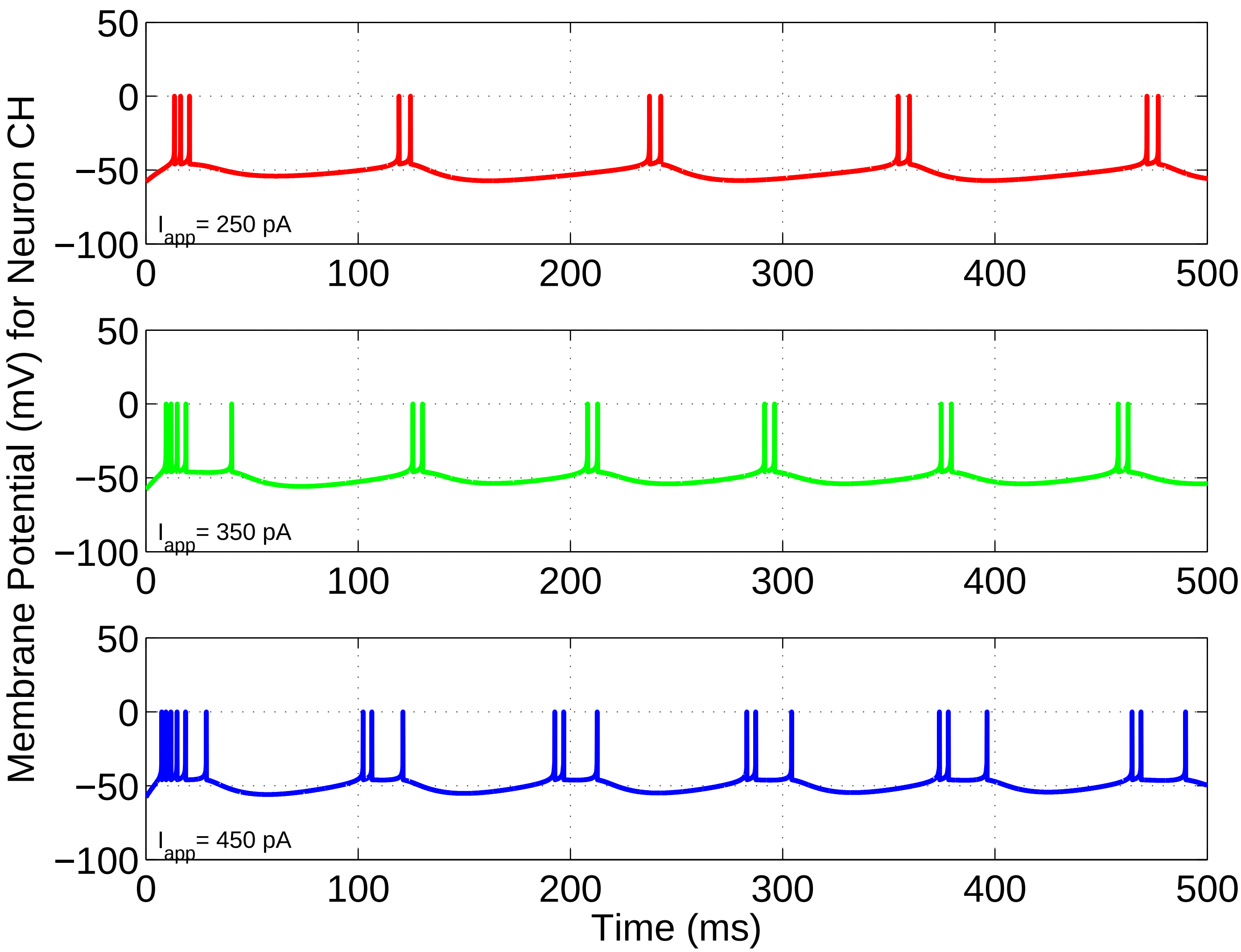
```
function [xa,V]=simulateaef(Iapp,T,dt,type)
N=1;
vpeak=0;
iter=T/dt;
if type==1
    C=200;
    gL=10;
    EL=-70;
    Vt=-50;
    Dt=2;
    a=2;
    tauw=30;
    b= 0 ;
    Vr=-58;
elseif type==2
    C=130;
    gL=18;
    EL=-58;
    Vt=-50;
    Dt=2;
    a=4;
    tauw=150;
    b=  120 ;
    Vr=-50;
else
  C=200;
    gL=10;
    EL=-58;
    Vt=-50;
    Dt=2;
    a=2;
    tauw=120;
    b=  100 ;
    Vr=-46;
end

[Vi,Ui]=aef_steady(type);
V=Vi*ones(N,iter);
U=Ui*ones(N,iter);
```

```
xa(1)=0;j=1;

while max(xa)<  T-dt
    V(:,j+1)=V(:,j)+(dt/C)*(Iapp(:,j)+(gL*Dt*exp((V(:,j)-Vt)/Dt))
                     -(gL*(V(:,j)-EL))- U(:,j));
    U(:,j+1)=U(:,j)+(dt/tauw )*(-U(:,j)+(a*(V(:,j)-EL)));

    spind=sign(V(:,j+1)-vpeak);
    if max(spind)>0
       ind=find(spind>0); %indices of neuron that just spiked
       V(ind,j)=vpeak;
       V(ind,j+1)=Vr;
       U(ind,j+1)=U(ind,j)+b;
    end

    xa(j+1)=xa(j)+dt;
    j=j+1;
end
```


Membrane Potential (mV) for Neuron RS
I_app = 250 pA
I_app = 350 pA
I_app = 450 pA
50
0
−50
−100
0
100
200
300
400
500
Time (ms)



Membrane Potential (mV) for Neuron IB
50
0
−50
−100
0
100
200
300
400
500
I app = 250 pA
I app = 350 pA
I app = 450 pA
Time (ms)


Membrane Potential (mV) for Neuron CH
Time (ms)
I_{app}= 250 pA
I_{app}= 350 pA
I_{app}= 450 pA
50
0
−50
−100
0
100
200
300
400
500

Problem 4: Spike energy based on Hodgkin-Huxley neuron model

(a) The code used to calculate the membrane potential and power is below. We will assume a typical cell area of $10^4\,\mu\text{m}^2$:

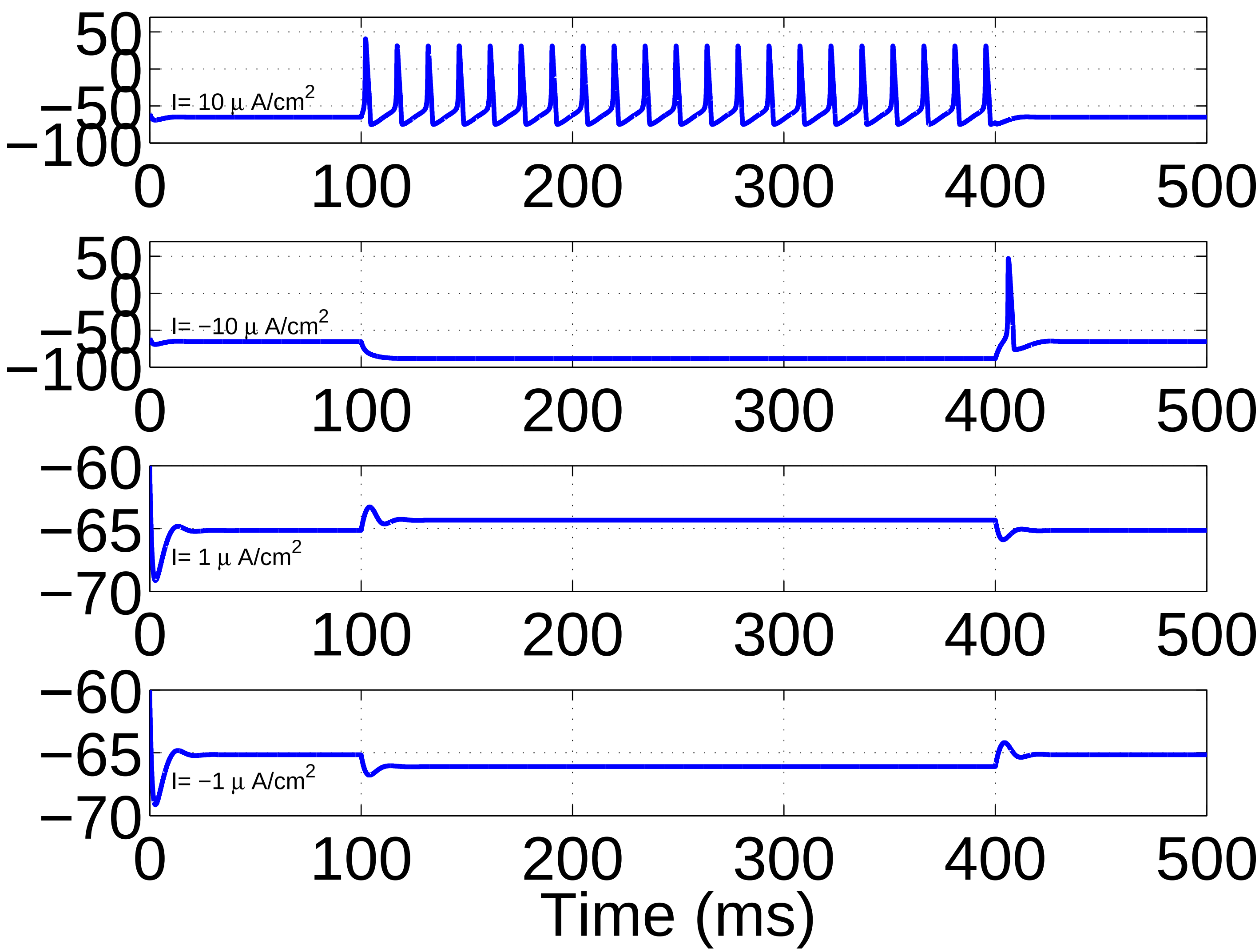
```
clc;clear;
close all

Area=1e4*1e-8 %1e-8; %(in cm^2)

cm = 1*Area; % membrane capacitance
dt = 0.01; % time step
T=100;
I0= 10*Area;
Iext = [zeros(1,ceil(T/dt)),I0*ones(1,3*ceil(T/dt)),zeros(1,ceil(T/dt))]; % ex





ENa = 50; %55.17;
EK = -77;% 72.14;
El =-55;% 49.42;





gNamax = 120*Area; %max conductances
gKmax = 36*Area;
glmax = 0.3*Area;

niter = ceil(T/dt); % max number of iterations

% Intial conditions
vm(1) = -60; % intial membrane voltage
V(1) = vm;

% calculating alphas abd betas for the intial membrane voltage
alpn = 0.01*(vm+55)/(1-exp(-(vm+55)/10));
alpm = 0.1*(vm+40)/(1-exp(-(vm+40)/10));
alph = 0.07*exp(-0.05*(vm+65));

betn = 0.125*exp(-(vm+65)/80);
betm = 4*exp(-0.0556*(vm+65));
beth = 1/(1+exp(-0.1*(vm+35)));

% intial m,n & h
n = alpn/(alpn+betn);
```

```
m = alpm/(alpm+betm);
h = alph/(alph+beth);
M=m;N=n;H=h;

% the loop for calculation of membrane voltage dynamics is here.
for i = 1:length(Iext)-1,

    dm = dt*(alpm*(1-m)-betm*m);
    dn = dt*(alpn*(1-n)-betn*n);
    dh = dt*(alph*(1-h)-beth*h);

    m = m + dm;
    n = n + dn;
    h = h + dh;


    gNa = gNamax*m^3*h;
    gK = gKmax*n^4;
    gl = glmax;


    INa = gNa*(vm-ENa);
    IK = gK*(vm-EK);
    Il = gl*(vm-El);

    dvm = (dt/cm)*(Iext(i)-( INa + IK + Il));

    vm = vm + dvm;
    V(i+1) = vm;
    H(i+1) = h;
    M(i+1) = m;
    N(i+1) = n;

    alpn = 0.01*(vm+55)/(1-exp(-(vm+55)/10));
    alpm = 0.1*(vm+40)/(1-exp(-(vm+40)/10));
    alph = 0.07*exp(-0.05*(vm+65));

    betn = 0.125*exp(-(vm+65)/80);
    betm = 4*exp(-0.0556*(vm+65));
    beth = 1/(1+exp(-0.1*(vm+35)));

    INa(i+1)=gNa*m^3*h*(vm-ENa);
    IK(i+1)=gK*n^4*(vm-EK);
    IL(i+1)=gl*(vm-El);


    Pe(i+1)=Iext(i+1)*V(i+1);
```

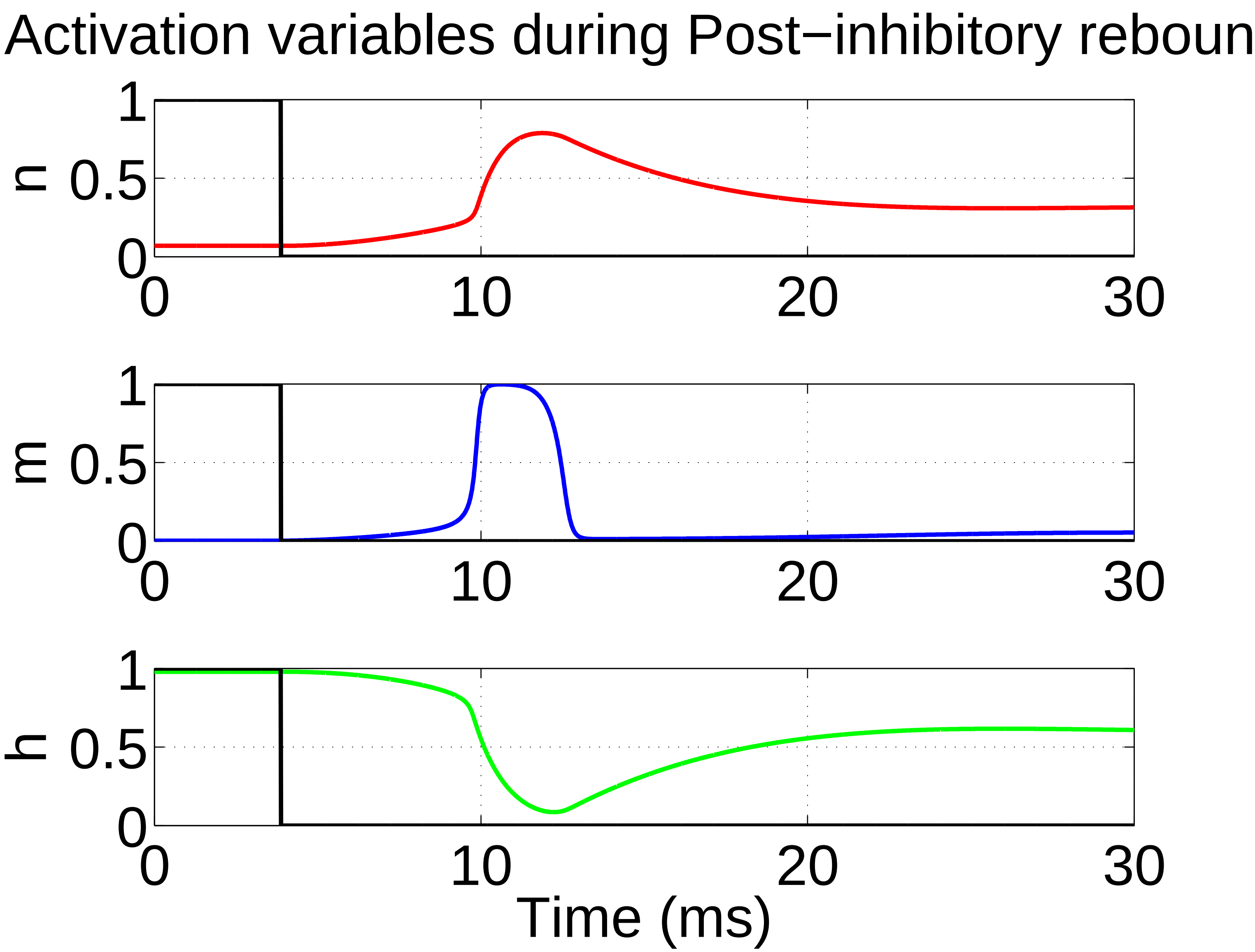
```
    PNa(i+1)=INa(i+1)*(V(i+1)-ENa);
    PK(i+1)=IK(i+1)*(V(i+1)-EK);
    PL(i+1)=IL(i+1)*(V(i+1)-El);
    Pc(i+1)=V(i+1)*(Iext(i+1)-( INa(i+1) + IK(i+1) + IL(i+1)));


end
tim = dt*(1:length(Iext)+1);
h=figure('Position',[100 100   820        700])
subplot(2,1,1)

set(gca,'fontsize',26)
plot(tim(1:end-1),(V)/1,'b','linewidth',2);
ylabel('V (mV)');
axis([0,5*T,-100,100])

subplot(2,1,2)
set(gca,'fontsize',26)
plot(tim(1:end-1),1e6*Iext,'r','linewidth',2);
axis([0,5*T,min(1e6*Iext),max(1e6*Iext)])
ylabel('I (pA)');
xlabel('Time (ms)');



h=figure('Position',[100 100   820        700])
subplot(3,1,1)
set(gca,'fontsize',26)
plot( tim(1:end-1),N,'r','linewidth',2);
ylabel('n')

subplot(3,1,2)
set(gca,'fontsize',26)
plot(tim(1:end-1),M,'b','linewidth',2);
ylabel('m')

subplot(3,1,3)
set(gca,'fontsize',26)
plot(tim(1:end-1),H,'g','linewidth',2);
ylabel('h')
xlabel('Time (ms)');



[pks,locs]=findpeaks(V);
```

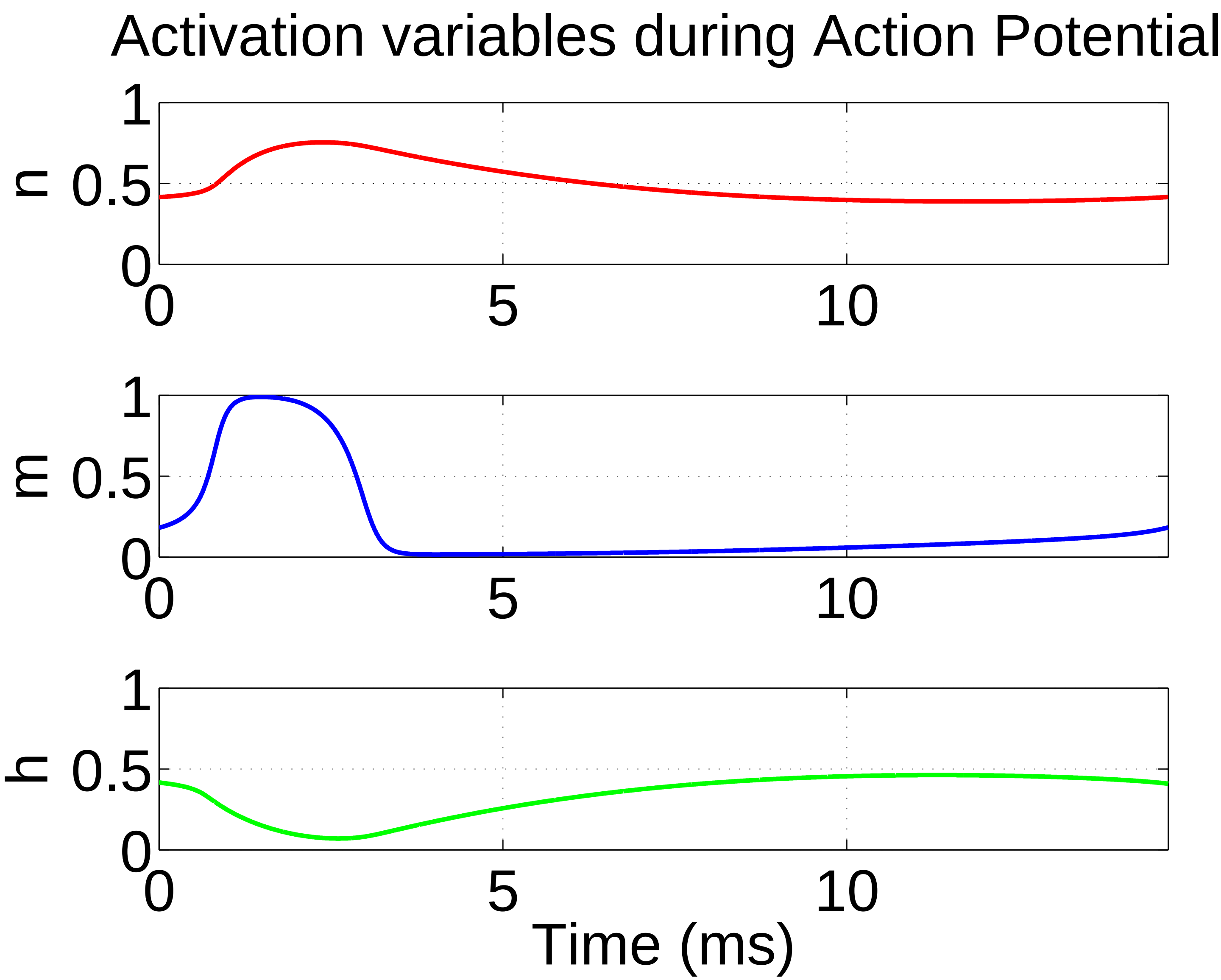
```
ti=(locs(pks>0));
t1=(ti(2)-100)  ;
t2=(ti(3)-100);


%In pJ
ENa=trapz(tim(t1:t2)-tim(t1), PNa(t1:t2))
EK=trapz(tim(t1:t2)-tim(t1), PK(t1:t2))
El=trapz(tim(t1:t2)-tim(t1), PL(t1:t2))

Ec=trapz(tim(t1:t2)-tim(t1), Pc(t1:t2))
Etotal=(ENa+EK+El+Ec)



h=figure('Position',[100 100   820        700]);
set(gca,'fontsize',26)
plot( tim(t1:t2), PNa(t1:t2),'r','linewidth',2);
hold on
plot( tim(t1:t2), PK(t1:t2),'g','linewidth',2);
hold on
plot( tim(t1:t2), PL(t1:t2),'b','linewidth',2);
hold on
plot( tim(t1:t2), Pc(t1:t2),'k','linewidth',2);
grid

ylabel('Power(nW)');
xlabel('Time (ms)');
legend('Na','K','L','C')
axis([tim(t1),tim(t2),-.50,2 ])




h=figure('Position',[100 100   820        700]);
subplot(3,1,1)
set(gca,'fontsize',26)
plot( tim(t1:t2),N(t1:t2),'r','linewidth',2);
ylabel('n')
axis([tim(t1),tim(t2),0,1])
grid

subplot(3,1,2)
set(gca,'fontsize',26)
plot(tim(t1:t2),M(t1:t2),'b','linewidth',2);
ylabel('m')
```

```
axis([tim(t1),tim(t2),0,1])
grid

subplot(3,1,3)
set(gca,'fontsize',26)
plot(tim(t1:t2),H(t1:t2),'g','linewidth',2);
ylabel('h')
xlabel('Time (ms)');
grid
axis([tim(t1),tim(t2),0,1])
```

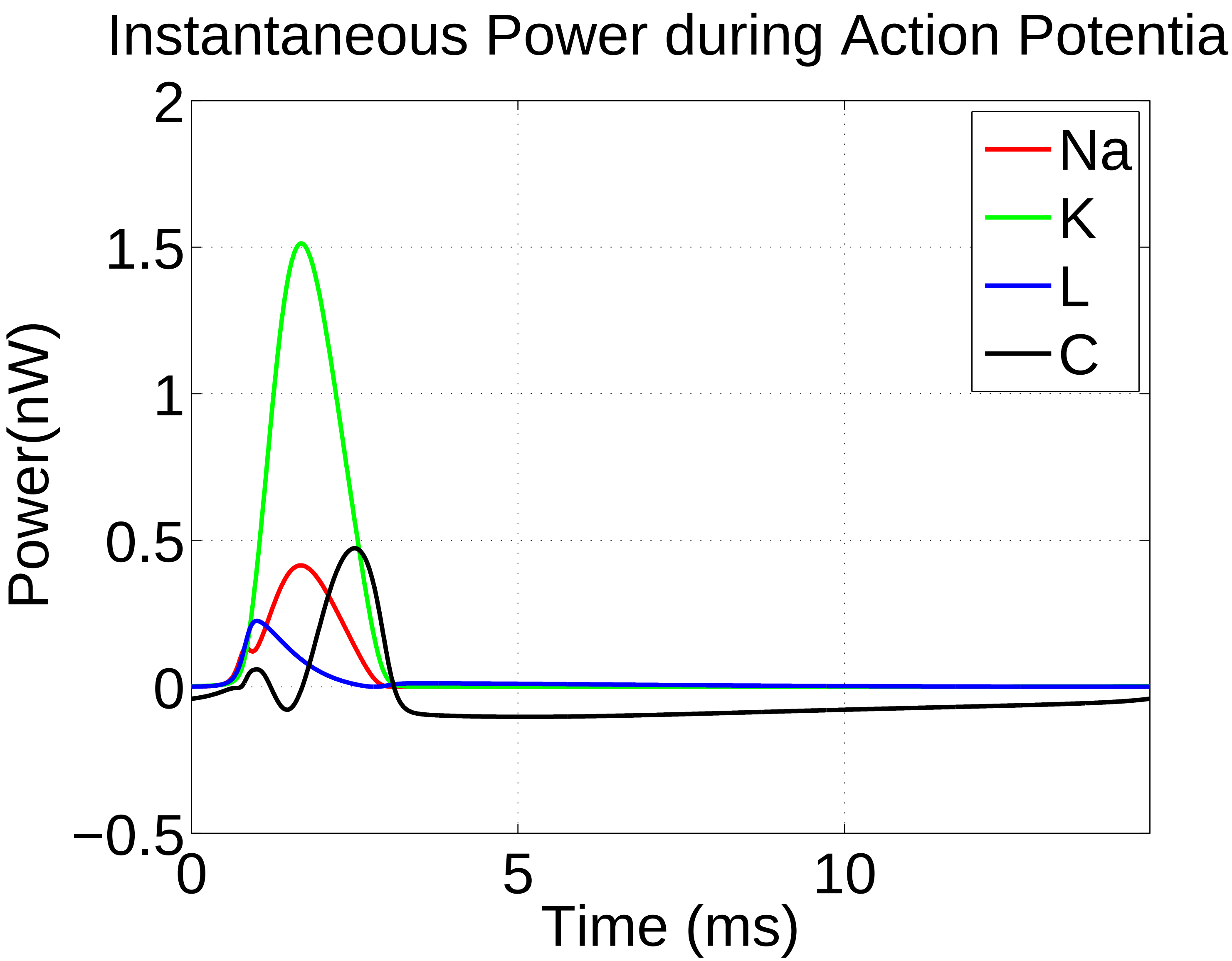


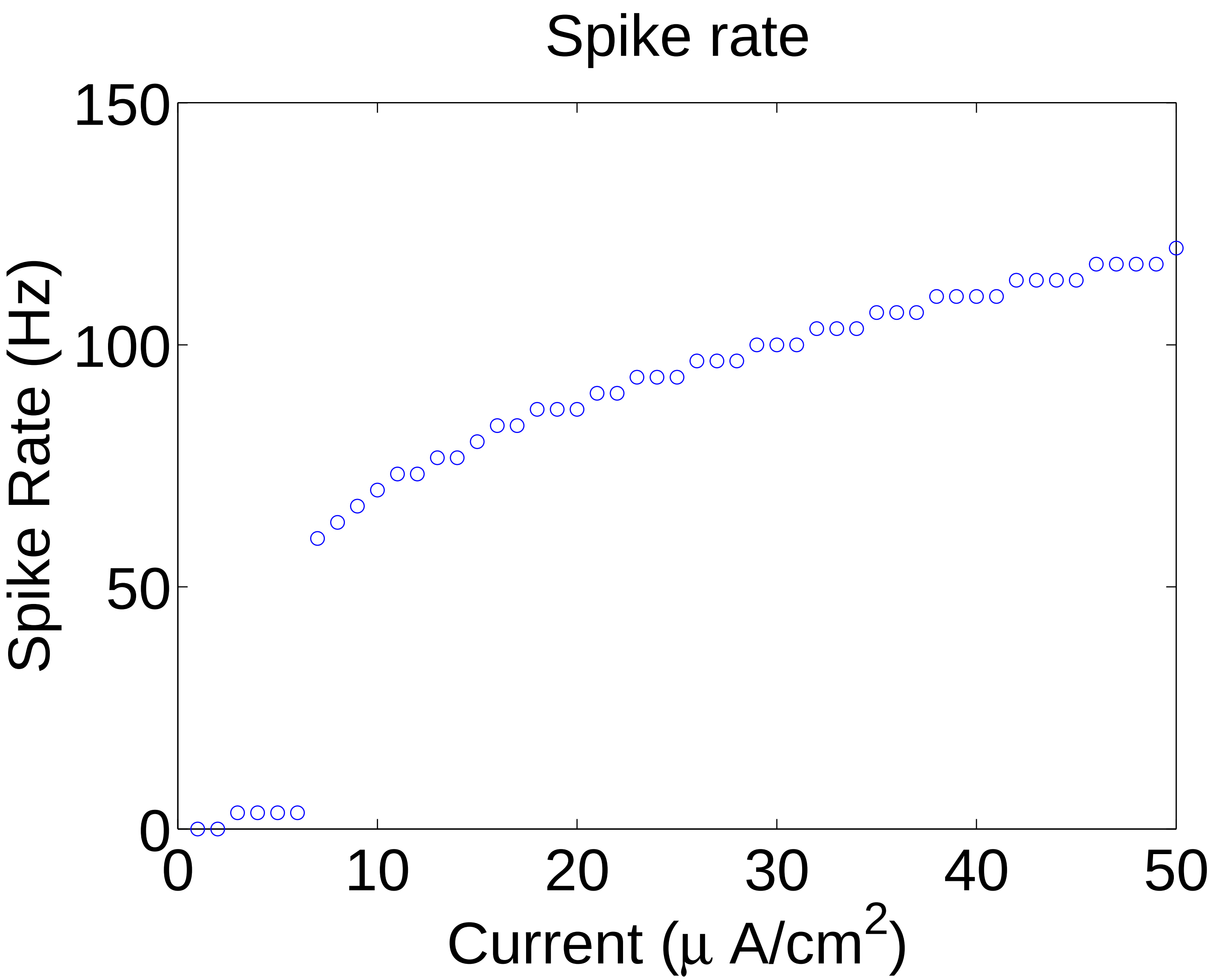
I= 10 μ A/cm2
I= −10 μ A/cm2
I= 1 μ A/cm2
I= −1 μ A/cm2
Time (ms)


Activation variables during Post−inhibitory reboun
n
m
h
0
0.5
1
0
10
20
30
Time (ms)



Activation variables during Action Potential
n
m
h
1
0.5
0
0
5
10
Time (ms)


Instantaneous Power during Action Potential
Power(nW)
Time (ms)
2
1.5
1
0.5
0
−0.5
0
5
10
Na
K
L
C



Spike rate
Spike Rate (Hz)
150
100
50
0
0
10
20
30
40
50
Current (μ A/cm2)

Mechanism for Post-inhibitory rebound: See `http://genesis-sim.org/node/816`

Power calculation: After numerical integration, the total energy dissipated in the channel during one cycle of the action potential is calculated to be about $0.20814e\,$fJ for a patch of area $1\,\mu$m^2. This translates to about $2\,$pJ for an average neuron with diameter of $100\,\mu$m.