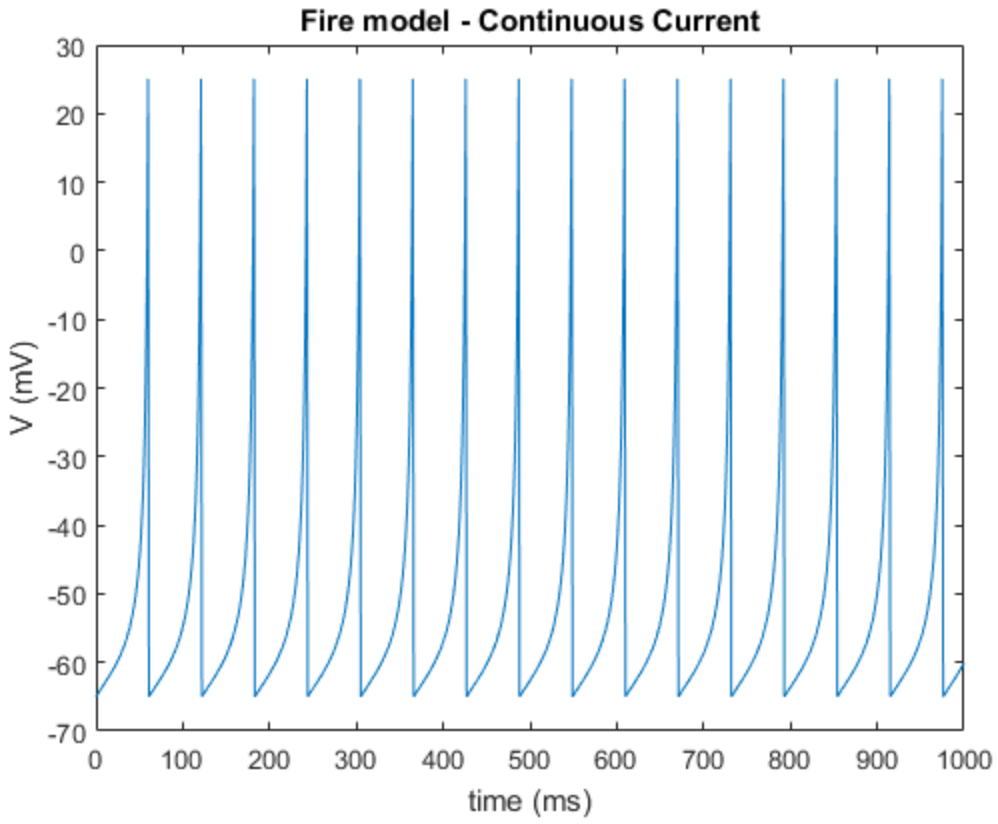
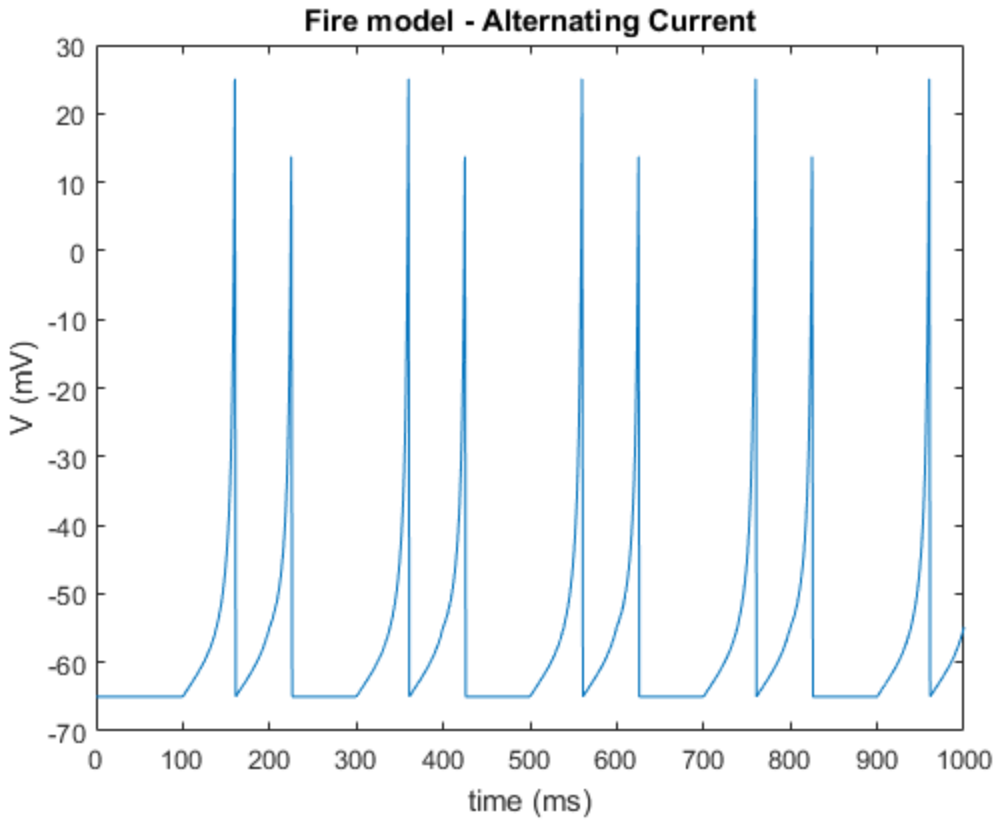
Diego Alba HW#1

```
% Problem #1

% ODE: Tau_m*dV/dt=E-V+R_m*I_e
% reorder terms: dV/dt + V/Tau_m = (E+R_m*I_e)/Tau_m
% int. factor: u(t) = exp(integral(1/Tau_m,t,0), du/dt = u(t)/Tau_m
% multiply through: u(t)*dV/dt + u(t)*V/Tau_m = u(t)*(E+R_m*I_e)/Tau_m
% apply product rule: d(u(t)*V(t))/dt =  u(t)*(E+R_m*I_e)/Tau_m
% solution: V(t) = 1/u(t)*(integral(u(t)*(E+R_m*I_e)/Tau_m)
%
%  V(t)= (exp(int(1/Tau_m))^-1*(int(exp(int(1/Tau_m)*(E+R_m*I_e)/
Tau_m)
%      = exp(t/Tau_m)^-1*(exp(t/Tau_m)*(E+R_m*I_e)/Tau_m)*Tau_m +C1)
%      = E+R_m*I_e+C1*exp(t/Tau_m)^-1, let C1=V(0)

%  V(t) =E+R_m*I_e+V(0)*exp(t/Tau_m)^-1


% Problem #3

E = -65;
Vmax = 30;
Tau = 10;
Rm = 10;
Ie = 2;
delQ = 2.5;
tf = 1000; % ms
t0 = 0;

% constant current
dvdt = @(t,v,p) (E-v+Rm*Ie + (v-E)^2/delQ)/Tau;

[V,t] = neuronRK(dvdt, E, [t0 tf], [E,Vmax]);
figure
plot(t,V)
title('Fire model - Continuous Current')
xlabel('time (ms)')
ylabel('V (mV)')

% pulses every 100ms
plength = 100;
pulse = [0 2];
Ie = repmat([repmat(pulse(1),1,plength)
 repmat(pulse(2),1,plength)],1,1+(tf-t0)/(plength*length(pulse)));

dvdt2 = @(t,v,p) (E-v+Rm*Ie(p) + (v-E)^2/delQ)/Tau;
[V2,t2] = neuronRK(dvdt2, E, [t0 tf], [E,Vmax]);
figure
plot(t2,V2)
title('Fire model - Alternating Current')
xlabel('time (ms)')
```

```
ylabel('V (mV)')

% Problem #2 - Runge-Kutta Method (modified for question #3)

function [y,t] = neuronRK(dfun, y0, tspan,options)
   step = 0.1;
   E = options(1);
   Vmax = options(2);
   n = (tspan(2)-tspan(1));
   t = linspace(tspan(1), tspan(2), n+1);
   y = [y0 zeros(1,n)];
   for i = 1:n
      k1 = dfun(t(i),y(i),i);
      k2 = dfun(t(i)+step/2,y(i)+k1*step/2,i);
      k3 = dfun(t(i)+step/2,y(i)+k2*step/2,i);
      k4 = dfun(t(i)+step,y(i)+k3*step,i);

      y(i+1)=y(i)+step/6*(k1+2*k2+2*k3+k4);

      if y(i+1) >= Vmax
          y(i+1) = E;
      end

   end
end
```





Published with MATLAB® R2017a