```
% Tutorial 4.2
% Email: jlindbloom@smu.edu

clear all
close all

% Define parameters.
global g_leak g_na_max g_k_max g_t_max e_na e_k e_ca e_leak c_membrane

g_leak = 10e-9;
g_na_max = 3.6e-6;
g_k_max = 1.6e-6;
g_t_max = 0.22e-6;
e_na = 55e-3;
e_k = -90e-3;
e_ca = 120e-3;
e_leak = -70e-3;
c_membrane = 100e-12;


% Setup time vector.
global dt
dt = 0.01e-3;
t = 0:dt:0.75;


% Create matrices for storing data.
numspikes = zeros(21,21);
interspikes = zeros(21,21);
for j=1:21
    for n=1:21
        [I_app] = applied_current(-200e-12 + (n-1)*20e-12,
 (j-1)*5e-12, t);
        [v_sim, h_sim, n_sim, h_t_sim, spikes] = PIR(t, I_app);
        nspikes = sum(spikes);
        results = find(spikes);
        min_spike_time = 10.0;
        for m=1:length(results)-1
            if (results(m+1)-results(m))*dt < min_spike_time
                min_spike_time = (results(m+1)-results(m))*dt;
            end
        end
        if min_spike_time == 10.0
            min_spike_time = 0;
        end

        numspikes(22-j,n) = nspikes;
        interspikes(22-j,n) = min_spike_time;
    end
end

numspikes
```

```
interspikes

f1 = figure;
figure(f1);
imagesc(numspikes);
c = colorbar;
c.Label.String = 'Total Number of Spikes';
xlabel("Baseline Current (pA)");
ylabel("Current Step (pA)");
xticks([1:2:21]);
xticklabels({'-200','-160','-120','-80','-40','0','40','80','120','160','200'});
yticks([1:2:21]);
yticklabels({'100','90','80','70','60','50','40','30','20','10','0'});
title("Total Number of Spikes");
saveas(f1,"numspikes.png");

f2 = figure;
figure(f2);
imagesc(interspikes);
c = colorbar;
c.Label.String = 'Minimum ISI (seconds)';
xlabel("Baseline Current (pA)");
ylabel("Current Step (pA)");
xticks([1:2:21]);
xticklabels({'-200','-160','-120','-80','-40','0','40','80','120','160','200'});
yticks([1:2:21]);
yticklabels({'100','90','80','70','60','50','40','30','20','10','0'});
title("Minimum ISIs");
saveas(f2,"interspikes.png");


% Plot qualitatively distinct behaviors.

% Plot A: baseline=-150, step=15.
[I_app] = applied_current(-150e-12, 15e-12, t);
[v_sim, h_sim, n_sim, h_t_sim, spikes] = PIR(t, I_app);
nspikes = sum(spikes);
results = find(spikes);
min_spike_time = 10.0;
for m=1:length(results)-1
    if (results(m+1)-results(m))*dt < min_spike_time
        min_spike_time = (results(m+1)-results(m))*dt;
    end
end
if min_spike_time == 10.0
    min_spike_time = 0;
end

fA = figure;
figure(fA);
subplot(2,1,1);
plot(t, I_app);
xlabel("Time (seconds)");
ylabel("Applied Current");
```

```
title("Applied Current vs. Time");
subplot(2,1,2);
plot(t, v_sim);
xlabel("Time (seconds)")
ylabel("Membrane Potential");
title(sprintf("Plot A. Spikes = %d, Min ISI =
 %f",nspikes,min_spike_time));
saveas(fA,"PlotA.png");

% Plot B: baseline=-80, step=90.
[I_app] = applied_current(-80e-12, 90e-12, t);
[v_sim, h_sim, n_sim, h_t_sim, spikes] = PIR(t, I_app);
nspikes = sum(spikes);
results = find(spikes);
min_spike_time = 10.0;
for m=1:length(results)-1
    if (results(m+1)-results(m))*dt < min_spike_time
        min_spike_time = (results(m+1)-results(m))*dt;
    end
end
if min_spike_time == 10.0
    min_spike_time = 0;
end

fB = figure;
figure(fB);
subplot(2,1,1);
plot(t, I_app);
xlabel("Time (seconds)");
ylabel("Applied Current");
title("Applied Current vs. Time");
subplot(2,1,2);
plot(t, v_sim);
xlabel("Time (seconds)")
ylabel("Membrane Potential");
title(sprintf("Plot B. Spikes = %d, Min ISI =
 %f",nspikes,min_spike_time));
saveas(fB,"PlotB.png");

% Plot C: baseline=-20, step=45.
[I_app] = applied_current(-20e-12, 45e-12, t);
[v_sim, h_sim, n_sim, h_t_sim, spikes] = PIR(t, I_app);
nspikes = sum(spikes);
results = find(spikes);
min_spike_time = 10.0;
for m=1:length(results)-1
    if (results(m+1)-results(m))*dt < min_spike_time
        min_spike_time = (results(m+1)-results(m))*dt;
    end
end
if min_spike_time == 10.0
    min_spike_time = 0;
end
```

```
fC = figure;
figure(fC);
subplot(2,1,1);
plot(t, I_app);
xlabel("Time (seconds)");
ylabel("Applied Current");
title("Applied Current vs. Time");
subplot(2,1,2);
plot(t, v_sim);
xlabel("Time (seconds)")
ylabel("Membrane Potential");
title(sprintf("Plot C. Spikes = %d, Min ISI =
 %f",nspikes,min_spike_time));
saveas(fC,"PlotC.png");

% Plot D: baseline=100, step=45.
[I_app] = applied_current(100e-12, 45e-12, t);
[v_sim, h_sim, n_sim, h_t_sim, spikes] = PIR(t, I_app);
nspikes = sum(spikes);
results = find(spikes);
min_spike_time = 10.0;
for m=1:length(results)-1
    if (results(m+1)-results(m))*dt < min_spike_time
        min_spike_time = (results(m+1)-results(m))*dt;
    end
end
if min_spike_time == 10.0
    min_spike_time = 0;
end

fD = figure;
figure(fD);
subplot(2,1,1);
plot(t, I_app);
xlabel("Time (seconds)");
ylabel("Applied Current");
title("Applied Current vs. Time");
subplot(2,1,2);
plot(t, v_sim);
xlabel("Time (seconds)")
ylabel("Membrane Potential");
title(sprintf("Plot D. Spikes = %d, Min ISI =
 %f",nspikes,min_spike_time));
saveas(fD,"PlotD.png");


% Plot E: baseline=180, step=45.
[I_app] = applied_current(180e-12, 45e-12, t);
[v_sim, h_sim, n_sim, h_t_sim, spikes] = PIR(t, I_app);
nspikes = sum(spikes);
results = find(spikes);
min_spike_time = 10.0;
for m=1:length(results)-1
    if (results(m+1)-results(m))*dt < min_spike_time
```

```
        min_spike_time = (results(m+1)-results(m))*dt;
    end
end
if min_spike_time == 10.0
    min_spike_time = 0;
end

fE = figure;
figure(fE);
subplot(2,1,1);
plot(t, I_app);
xlabel("Time (seconds)");
ylabel("Applied Current");
title("Applied Current vs. Time");
subplot(2,1,2);
plot(t, v_sim);
xlabel("Time (seconds)")
ylabel("Membrane Potential");
title(sprintf("Plot E. Spikes = %d, Min ISI =
 %f",nspikes,min_spike_time));
saveas(fE,"PlotE.png");



%%%%%%%%%%%%%%%%%%%%%%%%
% Function Definitions:

function [v_sim, h_sim, n_sim, h_t_sim, spikes] = PIR(t, i_applied,
 v_init, h_init, n_init, h_t_init)
% Simulates the thalamocortical neuron model with a T-type calcium
 current given the input time vector and
% applied current.

global dt g_leak g_na_max g_k_max g_t_max e_na e_k e_ca e_leak
 c_membrane

% Default parameters if not inputted.
if (~exist('v_init'))
    v_init = e_leak;
end
if (~exist('h_init'))
    h_init = 0;
end
if (~exist('n_init'))
    n_init = 0;
end
if (~exist('h_t_init'))
    h_t_init = 0;
end

% Setup vectors.
v_sim = zeros(1, length(t));
h_sim = zeros(1, length(t));
```

```
n_sim = zeros(1, length(t));
h_t_sim = zeros(1, length(t));
spikes = zeros(1, length(t));

v_sim(1) = v_init;
h_sim(1) = h_init;
n_sim(1) = n_init;
h_t_sim(1) = h_t_init;


% To count the number of spikes, a spike will be recorded when the
 membrane
% potential exceeds v_exceeds. The variable blocking will be set to 1,
% which will prevent more spikes from being recorded until the
 membrane
% potential falls below v_unblock, upon which the variable blocking
 will
% be set back to 0.

blocking = 0;
v_exceeds = 0.0;
v_unblock = -0.06;

% March forward in time.
for n = 1:(length(t)-1)
    if v_sim(n) > v_exceeds
        if blocking == 0
            spikes(n) = 1;
            blocking = 1;
        end
    end

    if v_sim(n) < v_unblock
        blocking = 0;
    end

    % Update v_sim.
    alpha = ((10^5)*(v_sim(n) + 0.035))/(1 -
 exp(-100*(v_sim(n)+0.035)));
    beta = 4000*exp((-(v_sim(n)+0.06))/(0.018));
    m = alpha/(alpha+beta);
    m_t = 1/(1 + exp((-(v_sim(n)+0.052))/(0.0074)));
    term1 = g_leak*(e_leak-v_sim(n));
    term2 = g_na_max*(m^3)*h_sim(n)*(e_na-v_sim(n));
    term3 = g_k_max*(n_sim(n)^4)*(e_k-v_sim(n));
    term4 = g_t_max*(m_t^2)*h_t_sim(n)*(e_ca-v_sim(n));
    v_sim(n+1) = v_sim(n) + (dt/
c_membrane)*(term1+term2+term3+term4+i_applied(n));

    % Update h_sim.
    alpha = 350*exp(-50*(v_sim(n)+0.058));
    beta = 5000/(1 + exp(-100*(v_sim(n)+0.028)));
    term1 = alpha*(1-h_sim(n));
    term2 = -beta*h_sim(n);
```

```
    h_sim(n+1) = h_sim(n) + dt*(term1+term2);

    % Update n_sim.
    alpha = (( 5*(10^4) )*(v_sim(n)+0.034))/(1 -
 exp(-100*(v_sim(n)+0.034)));
    beta = 625*exp(-12.5*(v_sim(n)+0.044));
    term1 = alpha*(1-n_sim(n));
    term2 = -beta*n_sim(n);
    n_sim(n+1) = n_sim(n) + dt*(term1+term2);

    % Update h_t_sim.
    h_t_inf = 1/(1 + exp(500*(v_sim(n)+0.076)));
    if v_sim(n) < -0.080
        tau_h_t = 0.001*exp(15*(v_sim(n)+0.467));
    else
        tau_h_t = 0.028 + 0.001*exp((-(v_sim(n)+0.022))/(0.0105));
    end
    h_t_sim(n+1) = h_t_sim(n) + dt*((h_t_inf-h_t_sim(n))/tau_h_t);
end
end

function [I_applied] = applied_current(baseline, step, t)
% Returns a vector for applied current given input baseline and step
% currents.
global dt
    I_applied = zeros(1, length(t));
    third = floor(length(t)/3);
    twothird = 2*third;
    I_applied(1:third) = baseline;
    I_applied(third+1:twothird) = baseline+step;
    I_applied(twothird+1:length(I_applied)) = baseline;
end


numspikes =

  Columns 1 through 13

     0     2     5     7    10    12    13    12    11    10    10
 12    13
     0     1     4     7     9    12    12    11    10    10    10
 11    13
     0     1     3     6     8    11    11    10    10     9     9
 10    12
     0     0     2     5     8    10    11    10     9     8     9
 10    12
     0     0     2     4     7    10    10     9     8     8     8
 9    11
     0     0     1     4     6     9    10     9     8     7     7
 9    11
     0     0     1     3     5     8     9     8     7     6     7
 8    10
     0     0     0     2     5     8     8     8     6     5     6
 7     9
```

```
    0     0     0     1     4     7     8     7     6     4     5
6     9
    0     0     0     1     3     6     7     6     5     3     4
6     8
    0     0     0     0     2     5     6     5     4     3     2
5     7
    0     0     0     0     2     5     6     5     3     2     1
3     6
    0     0     0     0     1     4     5     4     2     1     0
1     6
    0     0     0     0     1     3     4     3     2     1     0
0     5
    0     0     0     0     0     2     3     2     1     0     0
0     3
    0     0     0     0     0     2     3     2     1     0     0
0     0
    0     0     0     0     0     1     2     1     0     0     0
0     0
    0     0     0     0     0     0     1     1     0     0     0
0     0
    0     0     0     0     0     0     1     0     0     0     0
0     0
    0     0     0     0     0     0     0     0     0     0     0
0     0
    0     0     0     0     0     0     0     0     0     0     0
0     0

  Columns 14 through 21

   17    29    37    43    49    54    59    64
   18    29    36    43    48    54    59    64
   16    28    36    42    48    53    58    63
   17    28    35    42    48    53    58    63
   15    27    35    41    47    52    58    63
   16    27    34    41    47    52    57    62
   14    26    34    40    46    52    57    62
   14    26    33    40    46    51    57    62
   13    25    33    40    45    51    56    61
   13    25    33    39    45    51    56    61
   12    24    32    39    45    50    55    60
   11    24    32    38    44    50    55    60
   11    23    31    38    44    49    55    60
   10    23    31    37    43    49    54    59
    8    22    30    37    43    49    54    59
    7    21    30    36    42    48    53    59
    6    21    29    36    42    48    53    58
    4    20    28    35    42    47    53    58
    2    19    28    35    41    47    52    57
    0    18    27    34    41    46    52    57
    0    18    27    34    40    46    51    57


interspikes =
```

Columns 1 through 7

0	0.0436	0.0189	0.0149	0.0136	0.0114	0.0133
0	0	0.0219	0.0162	0.0143	0.0119	0.0138
0	0	0.0265	0.0177	0.0152	0.0125	0.0144
0	0	0.0351	0.0196	0.0163	0.0131	0.0151
0	0	0.0622	0.0223	0.0176	0.0138	0.0158
0	0	0	0.0262	0.0192	0.0146	0.0167
0	0	0	0.0326	0.0212	0.0155	0.0176
0	0	0	0.0458	0.0239	0.0166	0.0188
0	0	0	0	0.0277	0.0180	0.0201
0	0	0	0	0.0334	0.0197	0.0216
0	0	0	0	0.0438	0.0219	0.0235
0	0	0	0	0.0717	0.0248	0.0259
0	0	0	0	0	0.0289	0.0290
0	0	0	0	0	0.0353	0.0331
0	0	0	0	0	0.0473	0.0391
0	0	0	0	0	0.0866	0.0490
0	0	0	0	0	0	0.0708
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

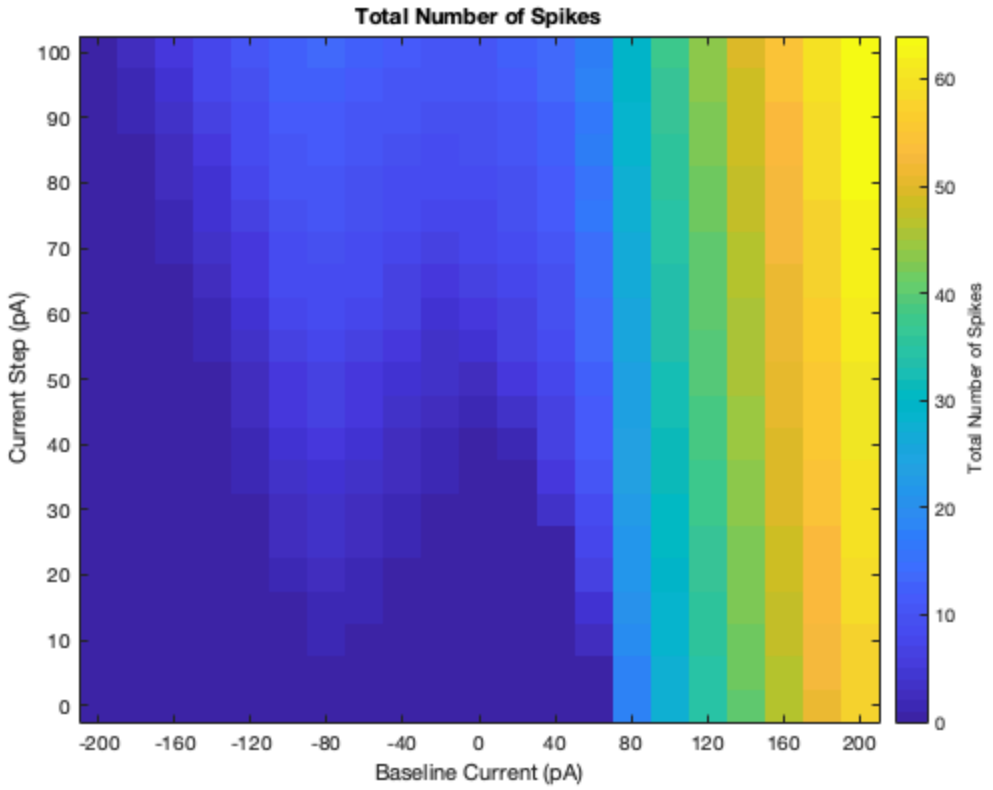
Columns 8 through 14

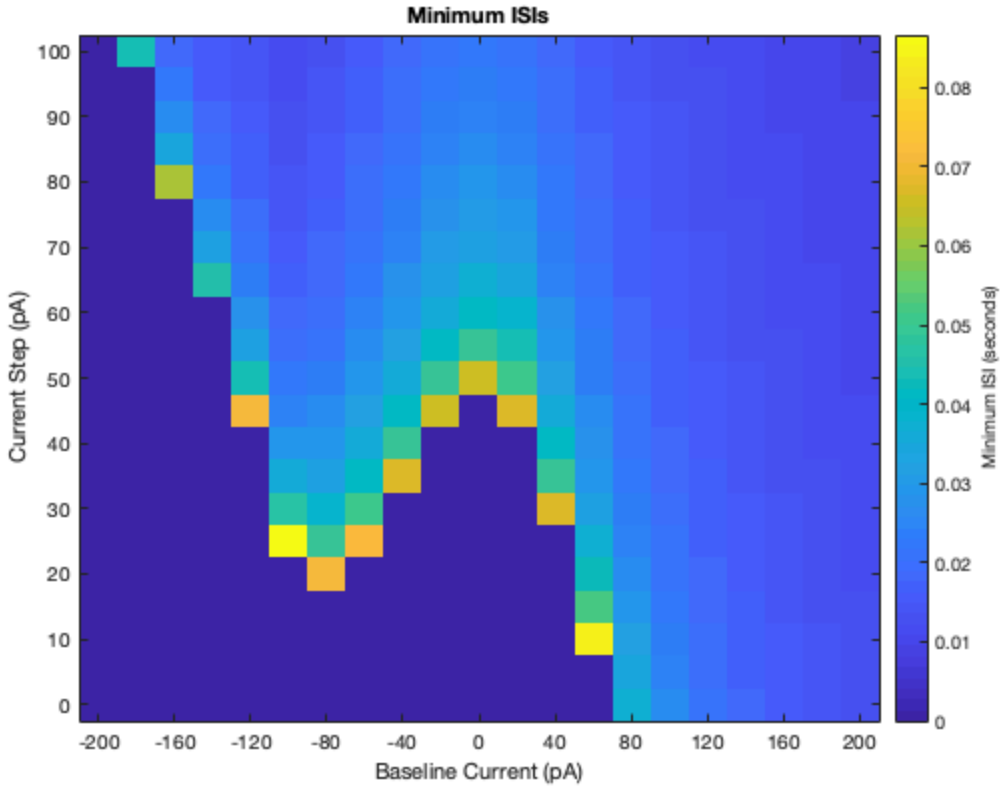
0.0160	0.0185	0.0210	0.0227	0.0214	0.0184	0.0161
0.0166	0.0194	0.0221	0.0238	0.0224	0.0191	0.0167
0.0174	0.0203	0.0232	0.0252	0.0236	0.0199	0.0172
0.0182	0.0213	0.0246	0.0267	0.0249	0.0208	0.0178
0.0191	0.0225	0.0261	0.0286	0.0266	0.0217	0.0184
0.0202	0.0239	0.0279	0.0308	0.0286	0.0229	0.0192
0.0214	0.0255	0.0301	0.0335	0.0309	0.0241	0.0199
0.0228	0.0273	0.0329	0.0370	0.0339	0.0256	0.0209
0.0245	0.0296	0.0364	0.0419	0.0380	0.0275	0.0218
0.0265	0.0324	0.0413	0.0496	0.0435	0.0295	0.0230
0.0289	0.0361	0.0489	0.0660	0.0512	0.0323	0.0242
0.0320	0.0412	0.0662	0	0.0664	0.0360	0.0258
0.0361	0.0492	0	0	0	0.0412	0.0278
0.0419	0.0665	0	0	0	0.0496	0.0298
0.0512	0	0	0	0	0.0672	0.0327
0.0723	0	0	0	0	0	0.0366
0	0	0	0	0	0	0.0422
0	0	0	0	0	0	0.0515
0	0	0	0	0	0	0.0848
0	0	0	0	0	0	0
0	0	0	0	0	0	0

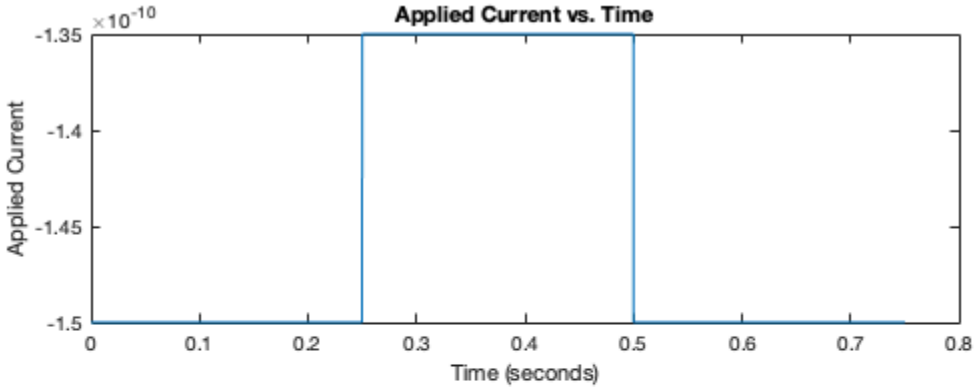
Columns 15 through 21

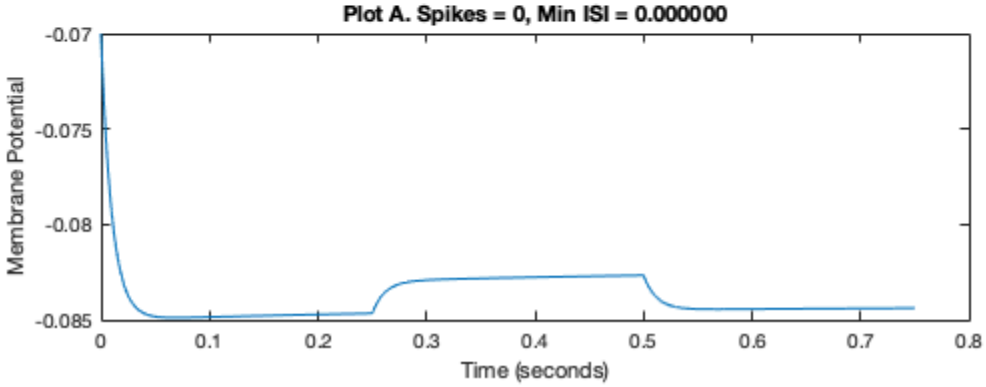
0.0143	0.0130	0.0120	0.0112	0.0105	0.0099	0.0093
0.0147	0.0133	0.0122	0.0114	0.0106	0.0100	0.0095
0.0151	0.0136	0.0125	0.0116	0.0108	0.0101	0.0096
0.0155	0.0139	0.0127	0.0118	0.0110	0.0103	0.0097

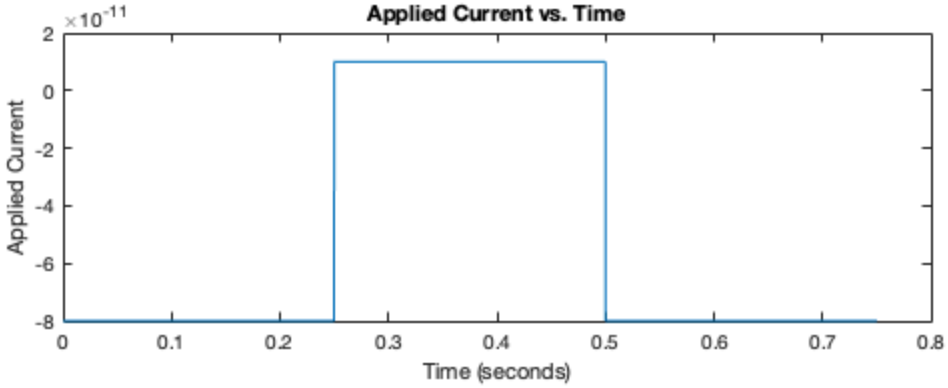
0.0160	0.0143	0.0130	0.0120	0.0112	0.0105	0.0099
0.0165	0.0147	0.0133	0.0122	0.0114	0.0106	0.0100
0.0170	0.0151	0.0136	0.0125	0.0116	0.0108	0.0101
0.0176	0.0155	0.0139	0.0127	0.0118	0.0110	0.0103
0.0182	0.0159	0.0143	0.0130	0.0120	0.0112	0.0104
0.0189	0.0164	0.0146	0.0133	0.0122	0.0113	0.0106
0.0197	0.0169	0.0150	0.0136	0.0125	0.0115	0.0108
0.0205	0.0175	0.0155	0.0139	0.0127	0.0118	0.0110
0.0215	0.0181	0.0159	0.0143	0.0130	0.0120	0.0111
0.0225	0.0188	0.0164	0.0146	0.0133	0.0122	0.0113
0.0237	0.0195	0.0169	0.0150	0.0136	0.0125	0.0115
0.0252	0.0204	0.0175	0.0154	0.0139	0.0127	0.0118
0.0268	0.0213	0.0181	0.0159	0.0143	0.0130	0.0120
0.0288	0.0223	0.0188	0.0163	0.0146	0.0133	0.0122
0.0312	0.0235	0.0195	0.0169	0.0150	0.0136	0.0125
0.0343	0.0249	0.0203	0.0174	0.0154	0.0139	0.0127
0.0379	0.0263	0.0211	0.0180	0.0158	0.0142	0.0130

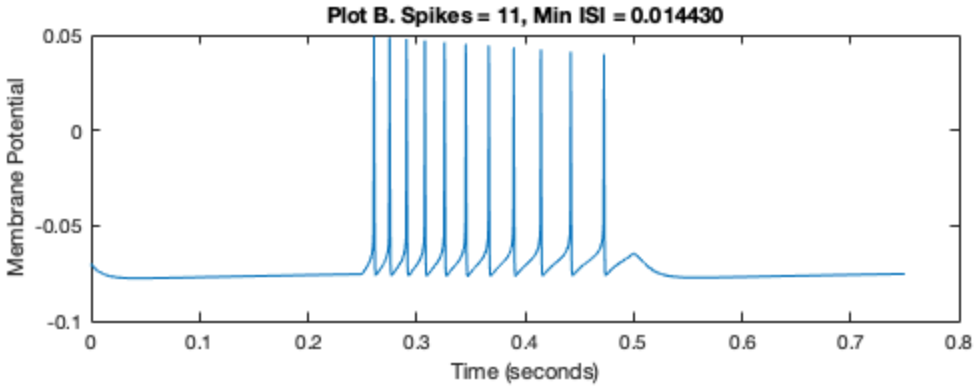


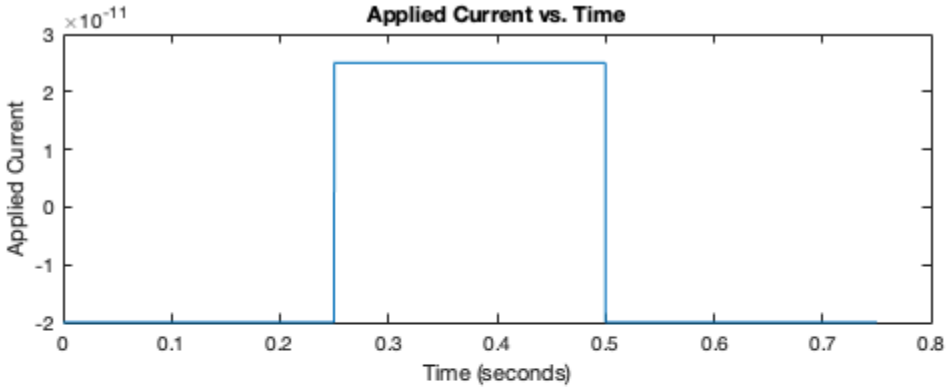

Minimum ISIs
Current Step (pA)
Baseline Current (pA)
Minimum ISI (seconds)
100
90
80
70
60
50
40
30
20
10
0
-200
-160
-120
-80
-40
0
40
80
120
160
200
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0

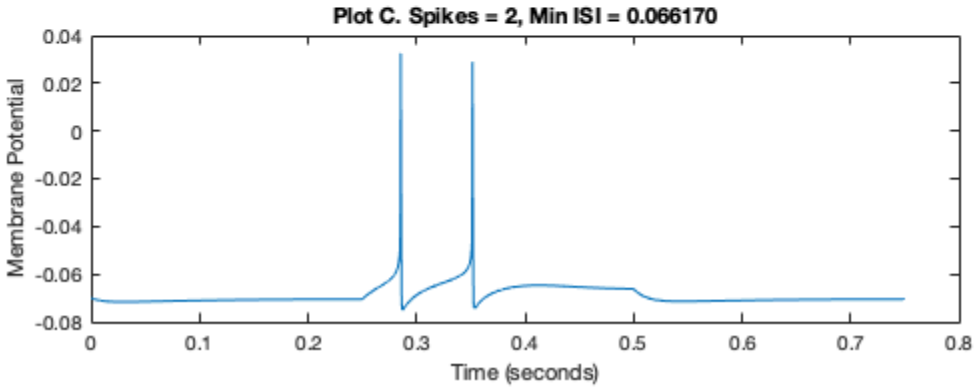

Applied Current vs. Time
×10^-10
Applied Current
Time (seconds)
-1.35
-1.4
-1.45
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

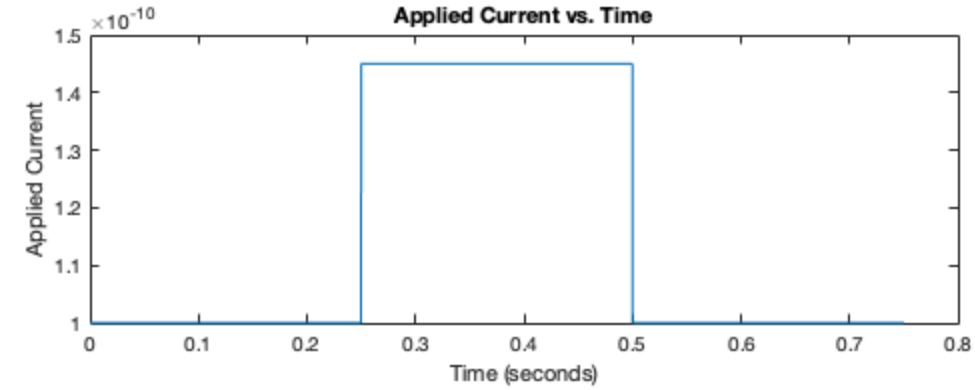

Plot A. Spikes = 0, Min ISI = 0.000000
Membrane Potential
Time (seconds)
-0.07
-0.075
-0.08
-0.085
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

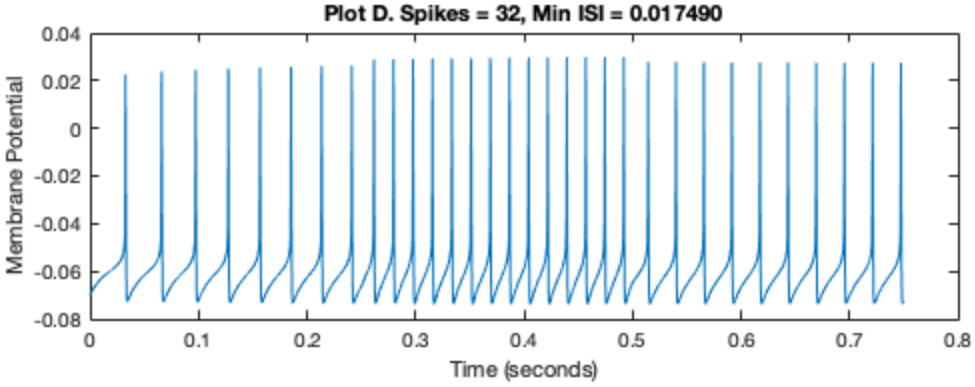

Applied Current vs. Time
×10^-11
Applied Current
Time (seconds)

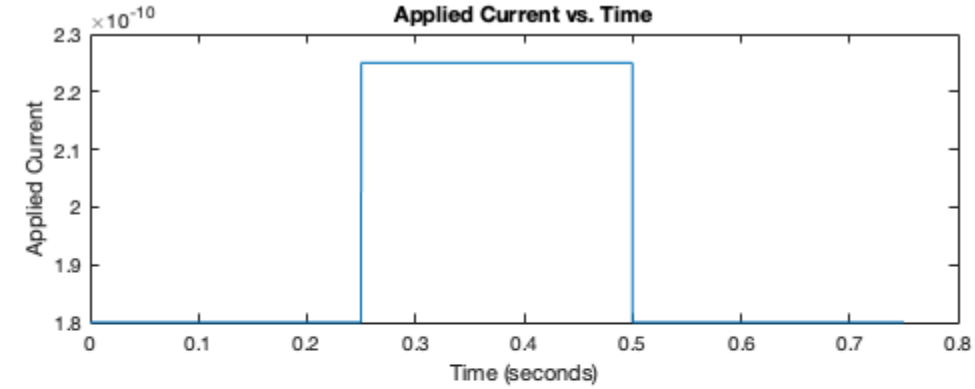

Plot B. Spikes = 11, Min ISI = 0.014430
Membrane Potential
Time (seconds)

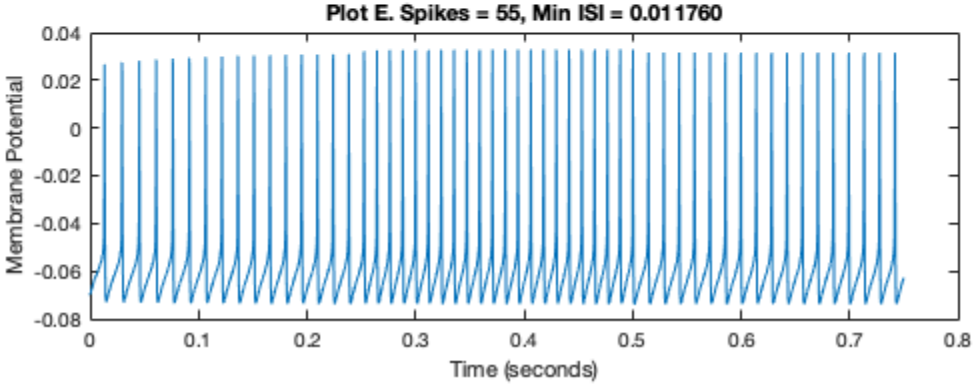

Applied Current vs. Time
×10^-11
Applied Current
Time (seconds)


Plot C. Spikes = 2, Min ISI = 0.066170
Membrane Potential
Time (seconds)


Applied Current vs. Time
×10^-10
Applied Current
Time (seconds)


Plot D. Spikes = 32, Min ISI = 0.017490
Membrane Potential
Time (seconds)


Applied Current vs. Time
×10^-10
Applied Current
Time (seconds)


Plot E. Spikes = 55, Min ISI = 0.011760
Membrane Potential
Time (seconds)

Published with MATLAB® R2018a