% This MATLAB file generates figure 1

steps = 500; %This simulation runs for 500 steps

R = [];

a=0.02; b=0.25; c=-65; d=6;

V=-64; u=b\*V;

VV=[]; uu=[];

tau = 0.25; tspan = 0:tau:steps; %tau is the discretization time-step

%tspan is the simulation interval

spikes = 0;

T1=50; %T1 is the time at which the step input rises

for I = 0:0.25:20

V=-64; u=b\*V;

VV=[]; uu=[];

spikes = 0;

for t=tspan

if (t>T1)

Input=I; % This is the input which you will change in your simulation

else

Input=0;

end;

V = V + tau\*(0.04\*V^2+5\*V+140-u+Input);

u = u + tau\*a\*(b\*V-u);

if V > 30

spikes = spikes + 1;

VV(end+1)=30; %VV is the time-series of membrane potentials

V = c;

u = u + d;

else

VV(end+1)=V;

end;

uu(end+1)=u;

end;

R(end+1) = spikes/300;

end;

plot(0:0.25:20, R);

title('R vs. I');

xlabel('I (current) [mA]');

ylabel('R (mean spike rate)');

% This MATLAB file generates figure 2, figure 3, figure 4, figure, 5 and

% figure 6 when executed multiple times for a different hardcoded input current

steps = 500; %This simulation runs for 500 steps

a=0.02; b=0.25; c=-65; d=6;

V=-64; u=b\*V;

VV=[]; uu=[];

tau = 0.25; tspan = 0:tau:steps; %tau is the discretization time-step

%tspan is the simulation interval

T1=50; %T1 is the time at which the step input rises

for t=tspan

if (t>T1)

I=1.0; % this input is increased to produce different plots

else

I=0;

end;

V = V + tau\*(0.04\*V^2+5\*V+140-u+I);

u = u + tau\*a\*(b\*V-u);

if V > 30

VV(end+1)=30; %VV is the time-series of membrane potentials

V = c;

u = u + d;

else

VV(end+1)=V;

end;

uu(end+1)=u;

end;

plot(tspan,VV); %VV is plotted as the output

axis([0 max(tspan) -90 40])

title('I = 1');

% This MATLAB script generates figure 7, figure 8, and figure 9

steps = 500; %This simulation runs for 500 steps

a=0.02; b=0.25; c=-65; d=6;

VB=-64; uB=b\*VB;

VVB=[]; uuB=[];

wB = 100; % this is weight for the synapse between neuron B and C

VA=-64; uA=b\*VA;

VVA=[]; uuA=[];

wA = 100; %this is the wight for the synapse between A and C

VC=-64; uC=b\*VC;

VVC=[]; uuC=[];

tau = 0.25; tspan = 0:tau:steps; %tau is the discretization time-step

%tspan is the simulation interval

spikeA = 0; %these become 1's when either A and B fire.

spikeB = 0;

T1=50; %T1 is the time at which the step input rises

counta = 0; % these count how many times there is a spike

countb = 0;

countc = 0;

Ra = []; % these will store the mean spike rates per each individual neuron

Rb = [];

Rc = [];

for INPUT = 0:0.25:20

counta = 0;

countb = 0;

countc = 0;

VB=-64; uB=b\*VB;

VVB=[]; uuB=[];

VA=-64; uA=b\*VA;

VVA=[]; uuA=[];

VC=-64; uC=b\*VC;

VVC=[]; uuC=[];

for t=tspan

% Script does the necessary calculations for neurons A and B first.

% Then, if there is a spike, a variable is set which will affect the

% input to neuron C.

if (t>T1)

IA=INPUT;

IB=5;

else

IA=0;

IB=0;

end;

VA = VA + tau\*(0.04\*VA^2+5\*VA+140-uA+IA);

uA = uA + tau\*a\*(b\*VA-uA);

VB = VB + tau\*(0.04\*VB^2+5\*VB+140-uB+IB);

uB = uB + tau\*a\*(b\*VB-uB);

if VB > 30

spikeB = 1; % neuron B is firing off down this synapse to neuron C

countb = countb + 1;

VVB(end+1)=30; %VV is the time-series of membrane potentials

VB = c;

uB = uB + d;

else

spikeB = 0;

VVB(end+1)=VB;

end;

uuB(end+1)=uB;

if VA > 30

spikeA = 1; % neuron A is firing off down this synapse to neuron C

counta = counta + 1;

VVA(end+1)=30;

VA = c;

uA = uA + d;

else

spikeA = 0;

VVA(end+1)=VA;

end;

uuA(end+1)=uA;

if (t>T1)

IC = wA\*spikeA + wB\*spikeB; % input to neuron C dependent on weighted spike signals from the other two neurons. Note: the charging is very quick due to the fact that input is only applied for one tau time, but the weight makes up for it.

else

IC = 0;

end;

VC = VC + tau\*(0.04\*VC^2+5\*VC+140-uC+IC);

uC = uC + tau\*a\*(b\*VC-uC);

if VC > 30

countc = countc + 1;

VVC(end+1)=30;

VC = c;

uC = uC + d;

else

VVC(end+1)=VC;

end;

uuC(end+1)=uC;

end;

Ra(end+1) = counta/300;

Rb(end+1) = countb/300;

Rc(end+1) = countc/300;

end;

plot(0:0.25:20, Ra);

title('R vs. IA');

xlabel('IC (current through neuron A) [mA]');

ylabel('R (mean spike rate)');

figure()

plot(0:0.25:20, Rc)

title('R vs. IC');

xlabel('IC (current through neuron C) [mA]');

ylabel('R (mean spike rate)');

figure()

scatter(Rc, Ra)

title('Ra vs. Rc')

xlabel('Rc (mean spike rate for neuron C)')

ylabel('Ra (mean spike rate for neuron A)')

% This MATLAB script generates figures 10, and 11 when you hardcode the

% input to different values (5 and 15)

steps = 500; %This simulation runs for 500 steps

a=0.02; b=0.25; c=-65; d=6;

VB=-64; uB=b\*VB;

VVB=[]; uuB=[];

wB = 100; % this is weight for the synapse between neuron B and %C

VA=-64; uA=b\*VA;

VVA=[]; uuA=[];

wA = 100; %this is the weight for the synapse between A and C

VC=-64; uC=b\*VC;

VVC=[]; uuC=[];

tau = 0.25; tspan = 0:tau:steps; %tau is the discretization time-step

%tspan is the simulation interval

spikeA = 0; %these become 1's when either A and B fire.

spikeB = 0;

T1=50; %T1 is the time at which the step input rises

for t=tspan

% Script does the necessary calculations for neurons A and B first.

% Then, if there is a spike, a variable is set which will affect the

% input to neuron C.

if (t>T1)

IA=15; % This is the input which you will change in your simulation

IB=5;

else

IA=0;

IB=0;

end;

VA = VA + tau\*(0.04\*VA^2+5\*VA+140-uA+IA);

uA = uA + tau\*a\*(b\*VA-uA);

VB = VB + tau\*(0.04\*VB^2+5\*VB+140-uB+IB);

uB = uB + tau\*a\*(b\*VB-uB);

if VB > 30

spikeB = 1; % neuron B is firing off down this synapse to neuron C

VVB(end+1)=30; %VV is the time-series of membrane potentials

VB = c;

uB = uB + d;

else

spikeB = 0;

VVB(end+1)=VB;

end;

uuB(end+1)=uB;

if VA > 30

spikeA = 1; % neuron A is firing off down this synapse to neuron C

VVA(end+1)=30;

VA = c;

uA = uA + d;

else

spikeA = 0;

VVA(end+1)=VA;

end;

uuA(end+1)=uA;

if (t>T1)

IC = wA\*spikeA + wB\*spikeB; % input to neuron C dependent on weighted spike signals from the other two neurons. Note: the charging is very quick due to the fact that input is only applied for one tau time, but the weight makes up for it.

else

IC = 0;

end;

VC = VC + tau\*(0.04\*VC^2+5\*VC+140-uC+IC);

uC = uC + tau\*a\*(b\*VC-uC);

if VC > 30

VVC(end+1)=30;

VC = c;

uC = uC + d;

else

VVC(end+1)=VC;

end;

uuC(end+1)=uC;

end;

subplot(3,1,1)

plot(tspan,VVA); %VV is plotted as the output

axis([400 max(tspan) -90 40])

title('I\_A = 15');

ylabel('V\_A (action potential of neuron A)');

subplot(3,1,2)

plot(tspan,VVB);

axis([400 max(tspan) -90 40])

ylabel('V\_B (action potential of neuron B)');

subplot(3,1,3)

plot(tspan,VVC);

axis([400 max(tspan) -90 40])

ylabel('V\_C (action potential of neuron C)');

xlabel('timespan (last 100 steps)');

% This MATLAB script generates figure 12, figure 13, and figure 14

steps = 500; %This simulation runs for 500 steps

a=0.02; b=0.25; c=-65; d=6;

VB=-64; uB=b\*VB;

VVB=[]; uuB=[];

wB = 50; % this is weight for the synapse between neuron B and C

VA=-64; uA=b\*VA;

VVA=[]; uuA=[];

%wA = 0; %this is the wight for the synapse between A and C

VC=-64; uC=b\*VC;

VVC=[]; uuC=[];

tau = 0.25; tspan = 0:tau:steps; %tau is the discretization time-step

%tspan is the simulation interval

spikeA = 0; %these become 1's when either A and B fire.

spikeB = 0;

T1=50; %T1 is the time at which the step input rises

counta = 0; % these count how many times there is a spike

countb = 0;

countc = 0;

Ra = []; % these will store the mean spike rates per each individual neuron

Rb = [];

Rc = [];

for wA = 50:10:200

counta = 0;

countb = 0;

countc = 0;

VB=-64; uB=b\*VB;

VVB=[]; uuB=[];

VA=-64; uA=b\*VA;

VVA=[]; uuA=[];

VC=-64; uC=b\*VC;

VVC=[]; uuC=[];

for t=tspan

% Script does the necessary calculations for neurons A and B first.

% Then, if there is a spike, a variable is set which will affect the

% input to neuron C.

if (t>T1)

IA=5;

IB=15; % This is the input which you will change in your simulation

else

IA=0;

IB=0;

end;

VA = VA + tau\*(0.04\*VA^2+5\*VA+140-uA+IA);

uA = uA + tau\*a\*(b\*VA-uA);

VB = VB + tau\*(0.04\*VB^2+5\*VB+140-uB+IB);

uB = uB + tau\*a\*(b\*VB-uB);

if VB > 30

spikeB = 1; % neuron B is firing off down this synapse to neuron C

countb = countb + 1;

VVB(end+1)=30; %VV is the time-series of membrane potentials

VB = c;

uB = uB + d;

else

spikeB = 0;

VVB(end+1)=VB;

end;

uuB(end+1)=uB;

if VA > 30

spikeA = 1; % neuron A is firing off down this synapse to neuron C

counta = counta + 1;

VVA(end+1)=30;

VA = c;

uA = uA + d;

else

spikeA = 0;

VVA(end+1)=VA;

end;

uuA(end+1)=uA;

if (t>T1)

IC = wA\*spikeA + wB\*spikeB; % input to neuron C dependent on weighted spike signals from the other two neurons. Note: the charging is very quick due to the fact that input is only applied for one tau time, but the weight makes up for it.

else

IC = 0;

end;

VC = VC + tau\*(0.04\*VC^2+5\*VC+140-uC+IC);

uC = uC + tau\*a\*(b\*VC-uC);

if VC > 30

countc = countc + 1;

VVC(end+1)=30;

VC = c;

uC = uC + d;

else

VVC(end+1)=VC;

end;

uuC(end+1)=uC;

end;

Ra(end+1) = counta/300;

Rb(end+1) = countb/300;

Rc(end+1) = countc/300;

end;

plot(50:10:200, Ra);

title('R\_A vs. w\_A');

xlabel('w\_A (the synoptic weight of the neuron A on neuron C)');

ylabel('R\_A (mean spike rate of A)');

figure()

plot(50:10:200, Rb)

title('R\_B vs. w\_A');

xlabel('w\_A (the synoptic weight of the neuron A on neuron C)');

ylabel('R\_B (mean spike rate of B)');

figure()

plot(50:10:200, Rc)

title('R\_C vs. w\_A');

xlabel('w\_A (the synoptic weight of the neuron A on neuron C)');

ylabel('R\_C (mean spike rate of C)');