

# Math160S

## Project Draft

### Synchronization of oscillators in Sand Crab: How do numerical methods change system stability and consistency?

Tony Yingte Huang

March 29th, 2009

#### Abstract

Synchronization has been widely observed in biological systems. One of the most well known synchronization phenomena was observed in Asian fireflies, which flash rhythmically either with an external source or with other fireflies (Ermentrout et al, 1984.)

Another interesting synchronizing phenomenon is observed in sand crabs. During a digging process, a sand crab shows a synchronizing rhythm: its uropod, the fourth right leg was in bilateral synchrony, being in anti-phase with its uropod. In the end of the digging process, however, three legs are exactly 1/3 out of phase with each other (A. Hodge et al, 2006). Other legs, in the meanwhile, don't exhibit any kinds of coordination.

In the research by Hodge et al, a model assuming that each leg is controlled by a central pattern generator (CPG) was proposed. Each CPG generates an intrinsic rhythm, which is then influenced by the rhythms of other oscillators through coupling. The model is widely applied to other crustaceans.

Our project adopted the model by Hodge et al and aimed to provide an analysis on the efficacy of various numerical methods to this particular model. The project researched four different explicit numerical methods: Euler's method, Adam-Bashforth second, third and fourth order (denoted as AB-2, AB-3, and AB-4, respectively). It was conducted in two stages: the first part we investigated that given an initial condition, whether different numerical methods would change the entry points of the limit cycle, and the second part we examined how stability broke down with respect to different timesteps under various numerical methods. The results showed that, under small timesteps, Adam-Bashforth methods predict value with higher consistency across various sizes of timesteps. Yet, counterintuitively, Euler's method demonstrated higher stability than other explicit multi-stage numerical methods. We concluded that AB-2 is the optimal method for an oscillatory system, as it's consistent and relatively more stable than AB of higher orders.

#### Literature review: introduction to synchronization

A basic synchronization model involves two oscillators, with certain networking in between, either excitatory or inhibitory. Each oscillator has its initial frequency and initial phase, which is then fastened or dampened through the networks between contagious oscillators' phase. A simple mathematical model, where  $N_1$  and  $N_2$  are two oscillating nodes and  $\theta_1, \theta_2$  are the phase of each oscillatory. They can be modeled as following:  
Excitatory network

$$N_1 \rightarrow^+ N_2 \tag{1}$$

$$\frac{d\theta_1}{dt} = w + \cos(\theta_1) \tag{2}$$

$$\frac{d\theta_2}{dt} = w + \cos(\theta_2) + A\cos(\theta_1) \tag{3}$$

where A is a positive constant, and  $\theta_1, \theta_2$  represent the phase of  $N_1, N_2$ , respectively.

Similarly,  
Inhibitory network

$$N_1 \rightarrow^- N_2 \tag{4}$$

$$\frac{d\theta_1}{dt} = w + \cos(\theta_1) \tag{5}$$

$$\frac{d\theta_2}{dt} = w + \cos(\theta_2) + A\cos(\theta_1) \tag{6}$$

where A is a negative constant.

### Synchronization in biological systems

As behaviors exhibited in living organisms are controlled by membrane potentials, which oscillate over stimuli, oscillating models which imitate the neuronal stimulation are widely adopted in synchronization problems in biological systems. Hodgkin-Huxley model, developed by Physiology or Medicine Nobel Laureates Alan Lloyd and Andrew Huxley, successfully expressed the molecular interactions and corresponding voltage changes during neural stimuli. The model treated a neural stimulating process as a circuit, where the case lipid layer is represented as a capacitance; two ion gated channels (including Potassium, and Calcium conductance,  $g_K, g_{Ca}$ , respectively) as nonlinear conductances; one leak channel  $g_L$  as leak conductance; and chemicals from near synapses as batteries. Figure 1 summarizes Hodgkin-Huxley's assumption. The four-

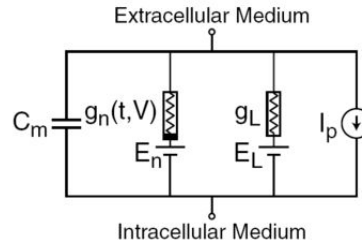


Figure 1: A summary of Hodgkin-Huxley's assumptions<sup>1</sup>

variable Hodgkin-Huxley model was later reduced to a two-variable Morris-Lecar model, which reduces the complexity in implementing the model. It consists of two parts,  $v$  and  $n$ , denoting cell membrane potential and fraction of open potassium channels, respectively:

$$\epsilon \frac{dv}{dt} = f(v, n) \tag{7}$$

$$\frac{dn}{dt} = g(v, n) \tag{8}$$

The equation for a central pulse generator of our interest is shown below:

$$\frac{dV_i}{dt} = F(V_i) \quad (9)$$

$$F(V_i) = \begin{bmatrix} \frac{dv_i}{dt} \\ \frac{dn_i}{dt} \end{bmatrix} = \begin{bmatrix} i_{app} - (g_l(v_i - v_l) - g_{Ca}m_\infty(v_i - v_{Ca}) - g_K n(v_i - v_K)) \\ \lambda_n(n_\infty(v_i) - n_i) \end{bmatrix} \quad (10)$$

$$\text{where} \quad (11)$$

$$m_\infty(v) = \frac{1}{2} \left( 1 + \tanh\left(\frac{v - v_a}{v_b}\right) \right) \quad (12)$$

$$n_\infty(v) = \frac{1}{2} \left( 1 + \tanh\left(\frac{v - v_c}{v_d}\right) \right) \quad (13)$$

$$\lambda_n(v) = \phi_n \cosh\left(\frac{v - v_c}{2v_d}\right) \quad (14)$$

Since each oscillator has an impact on the frequency of adjacent oscillators, the networks between oscillators and the coupling strength jointly determine the equilibrium frequency. For instance, a basic model of three oscillators, connected as following can be expressed in the following way, as shown in Figure 2:

We model the interaction term as following:

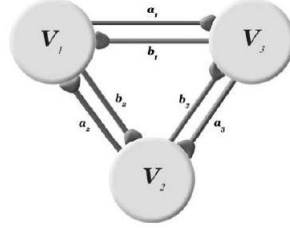


Figure 2: Interactions between three oscillators,  $V_1, V_2$  and  $V_3^2$

$$\frac{dV_i}{dt} = F(V_i) + g_{syn} G_i(V_1, V_2, V_3) \quad (15)$$

$$G_i(V_1, V_2, V_3) = \begin{bmatrix} (b_i S(v_{i-1}) + a_{i+1} S(v_{i+1}))(v_{syn} - v_i) \\ 0 \end{bmatrix} \quad (16)$$

$$\text{with } v_{i+3} = v_i, n_{i+3} = n_i, S(v) = \frac{1}{2} \left( 1 + \tanh\left(\frac{v}{15}\right) \right) \quad (17)$$

Numerical values are provided in Appendix A.

### Defining entry point and stability

Given the complexity of the model, we believe that explicit methods will be reasonable for numerical integration. Figure 3 shows an implementation of Euler method, setting initial conditions of  $[V_1 \ V_2 \ V_3]$  to be  $\begin{bmatrix} 100 & -200 & 150 \\ 0.6 & 0.9 & 0.1 \end{bmatrix}$  will give the following phase diagram, on which we define entry point as the point at which  $V_i$  enters the limit cycle.

We observe that, a great timestep may lead to the disappearance of the limit cycle, and the whole system became instable. As a consequence, stability of a system under a specific numerical method is defined as the maximum value of timesteps under which the system oscillates with its limit cycle. For example, a clear distinction can be seen between Figure 3, which is a stable system and Figure 4, which is unstable.

### Stability check

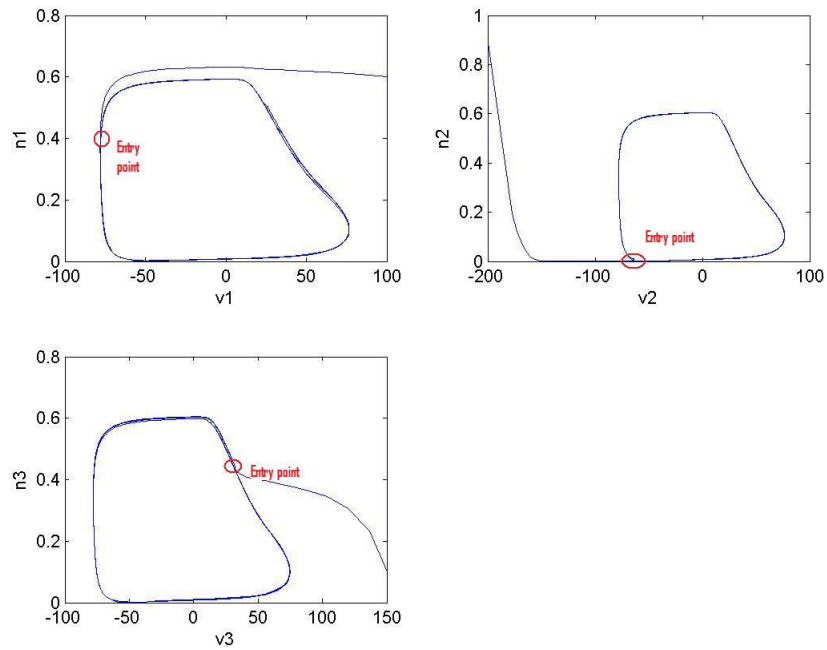


Figure 3: Defining entry point as the point at which the desired variable enters the limit cycle.

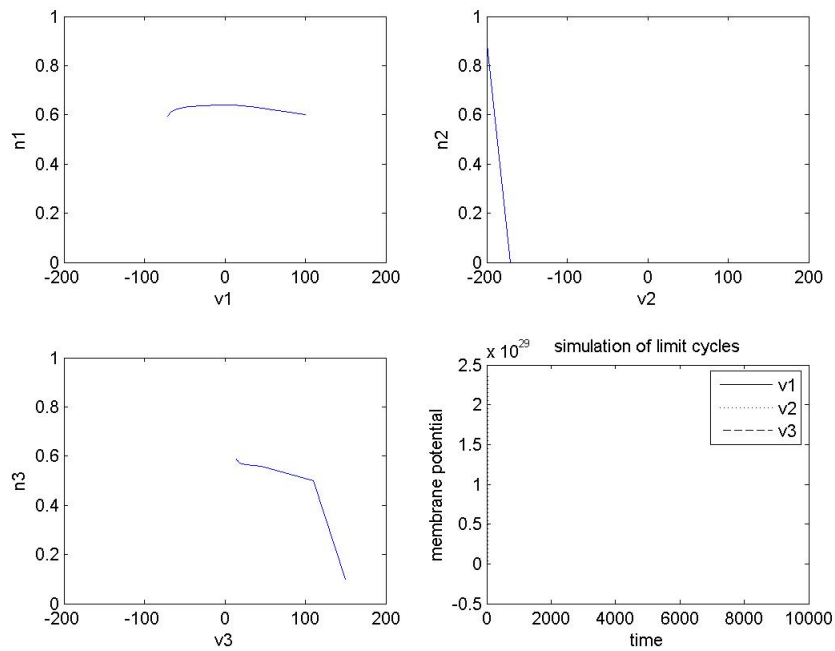


Figure 4: Unstable system with the same set of initial condition and Euler method, under a different timestep

A further investigation showed that the stability broke down, with the same set of initial condition, as aforementioned, happened when timestep=2.644. We then examined the stability breakdown points of AB-2, AB-3 and AB-4. Our results indicated that, surprisingly, conditional on the initial conditions, the use of AB-2 decreased the breakdown timestep to 2.11, or lower than Euler method by 20%. AB-3 further showed a stricter timestep for the system to reach stability. It required the timestep to be not greater than 1.93, which is lower than the Euler method by 27%. AB-4 showed the lowest tolerance, with maximum timestep being 1.67.

To test the robustness of our finding, we randomly picked other sets of initial conditions and conducted a similar analysis. A comparison between three sets of initial conditions tested and corresponding conditions required for stability is exhibited in Table 1. We found that, first, stability breakdown point varied with initial values, and second, the finding was robust, as Adam-bashforth method of higher orders consistently required smaller timesteps to reach stability. Table 1 summarizes our finding

Method	Euler's method	AB-2	AB-3	AB-4
Initial Condition I <sup>a</sup>	2.644	2.11(-20%) <sup>d</sup>	1.93(-27%)	1.67 (-37%)
Initial Condition II <sup>b</sup>	6.50	5.42(-16%)	4.02(-38%)	3.80 (-41%)
Initial Condition III <sup>c</sup>	7.18	5.78(-19%)	4.04(-43%)	1.67 (-53%)

<sup>a</sup> Initial condition= $[V_1, V_2, V_3] = [100, -200, 150; 0.6, 0.9, 0.1]$

<sup>b</sup> Initial condition= $[V_1, V_2, V_3] = [150, -120, 190; 0.3, 0.4, 0.7]$

<sup>c</sup> Initial condition= $[V_1, V_2, V_3] = [190, 130, -20; 0.6, 0.1, 0.4]$

<sup>d</sup> Percentage decrease from Euler's method

Table 1: Stability analysis with various initial conditions

### Consistency check

We conducted Euler's method to the system, with aforementioned initial conditions and various timesteps. In Figure 5, red, blue and green curves show simulations with three timesteps: 1, 1.5 and 2. As clearly shown in Figure 5, entry points clearly vary with different sizes of timesteps, especially in oscillator 3, whose entry points exhibited stronger variability, whereas in oscillator 1 and 2 they are relatively more consistent.

Given the symmetric setup of the equation, asymmetric consistency of the entry points between oscillators indicate that initial conditions may alter the consistency of entry points. To confirm this, we changed oscillator 1's initial condition,  $n_1$  from 1 to 0.15 and conducted an alternative simulation. Consistent with our conjecture, we observed that oscillator 1 exhibited greater consistency under the new initial conditions, as shown in Figure 6.

Even though Adam-bashforth methods with higher orders showed smaller tolerance for timesteps, they improved the consistency problem of entry point. In Figure 7 and 8, the red, blue and green cycle indicated simulation under three sizes of timesteps: 1.9, 1.5 and 1, respectively. AB-4 was eliminated from our discussion since 1.9 was above the tolerance of AB-4 method. A comparison between Euler's method, AB-2 and AB-3 showed that the inconsistency in oscillator 3 was greatly diminished in the AB-2 scheme(Figure 7). In Figure 8, AB-3 showed a higher inconsistency than AB-2, yet the inconsistency was still smaller than that of Euler's method.

With initial condition clearly playing a role in the system's consistency, we further investigated how initial conditions influence the consistency of the entry point. We simulated the system over a broad range of initial conditions of oscillator 1, while fixing oscillator 2 and 3, and then graphed different limit cycles under each initial condition. An interesting characteristic was observed: two broad sets of initial conditions unanimously converged into two particular points on the limit cycle. As shown in Figure 9, initial points in region A, B entered the cycle from point a, and b, respectively. The last set of initial condition over region C, on the other hand, entered the cycle from a broad range of initial conditions over the "c band". This explains why in Figure 5 that oscillators 1 and 2 showed a greater consistency than oscillator 3, as oscillator 1 and 2 start from region A and B, respectively and thus are guaranteed unanimous entries into the cycle.

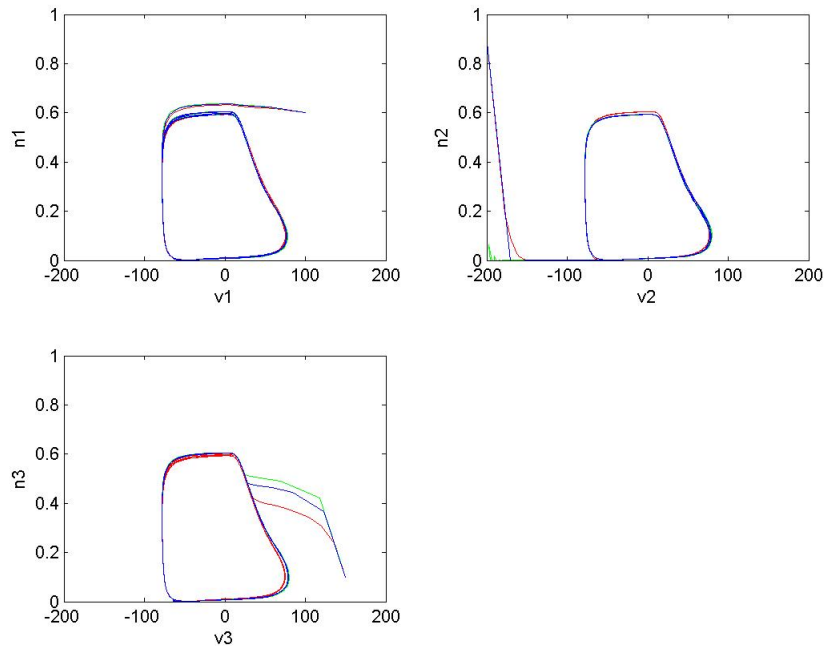


Figure 5: Oscillator 3 exhibits stronger variability in its entry point than oscillators 1 and 2.

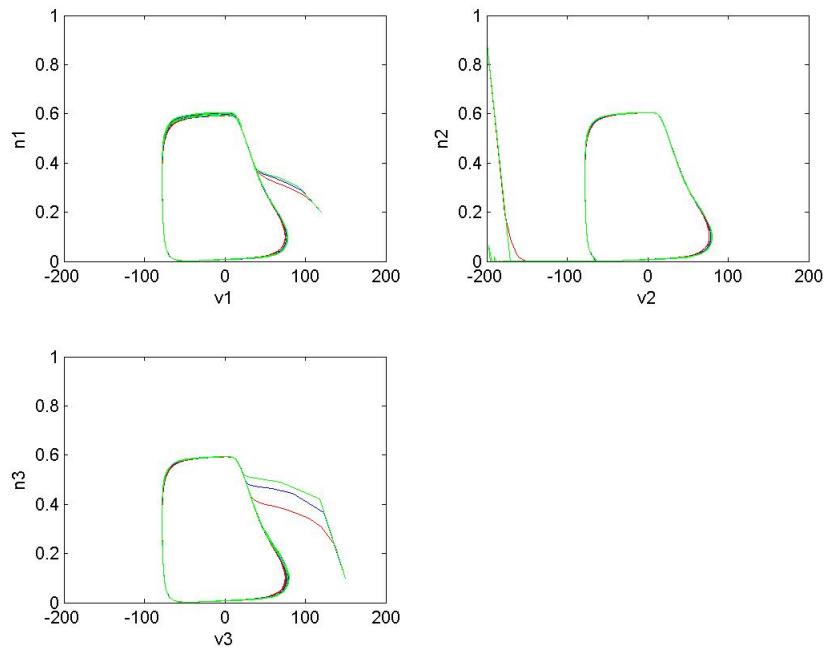


Figure 6: Oscillator 1 exhibits stronger variability after changing its initial conditions.

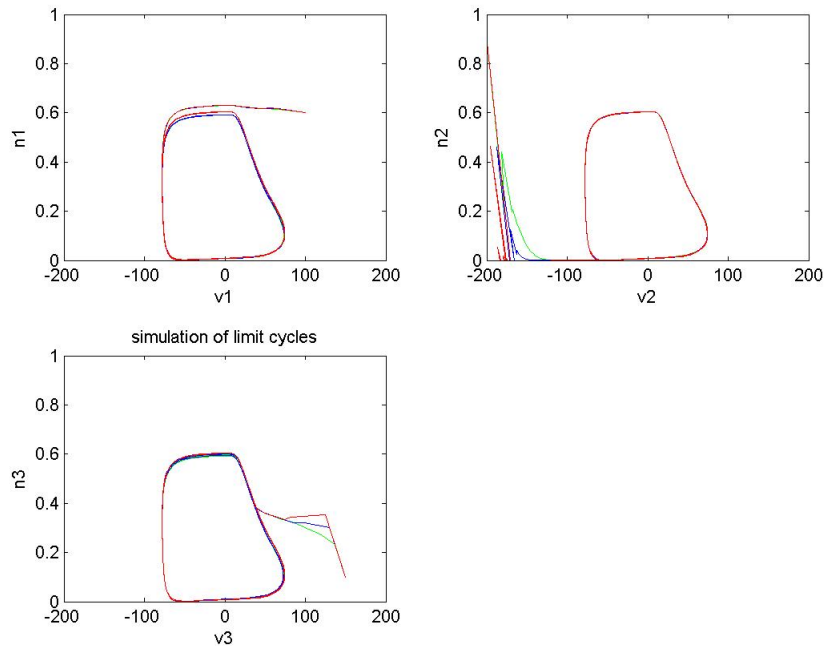


Figure 7: Inconsistency in the entry points oscillator 3 were greatly diminished

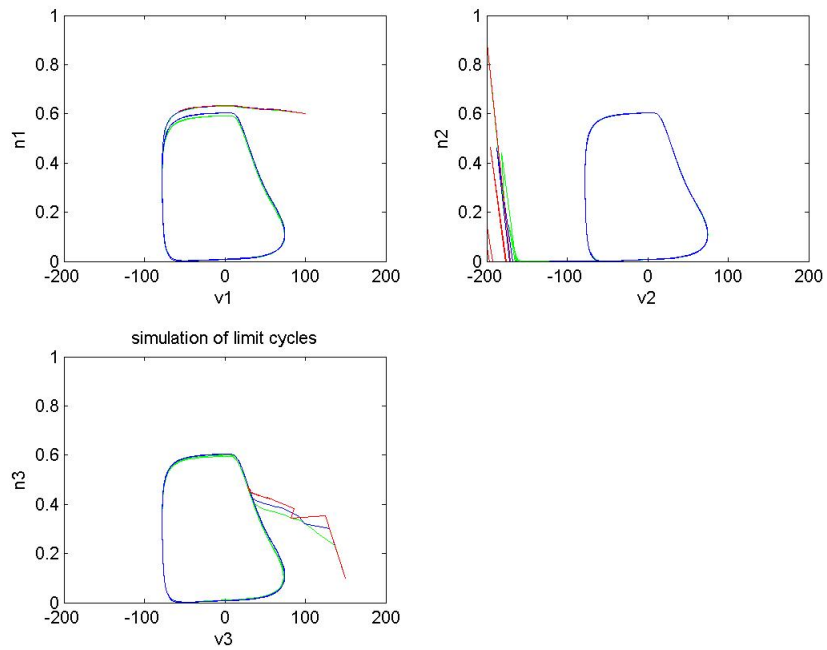


Figure 8: Oscillator 3 showed a greater inconsistency than AB-2, yet still smaller than Euler's method

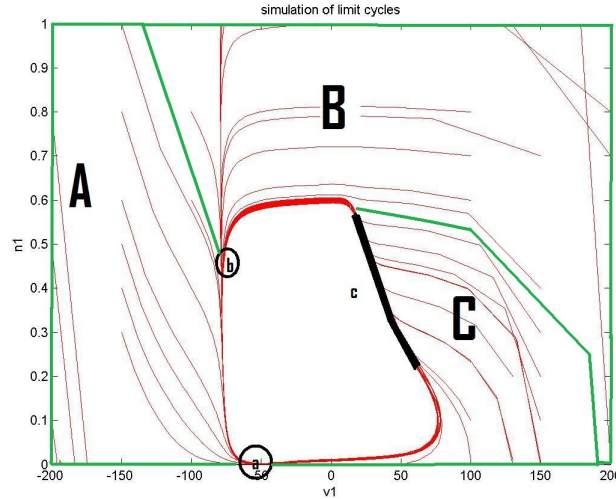


Figure 9: The set up of the model makes three distinct entry regions: 'point a, b' and 'c band'

## Discussion

It's been argued that AB-2, AB-3 and AB-4 are good numerical methods as they exhibit higher order of accuracy, as shown in class. Adam- Bashforth method offers a good approximation during non-oscillatory regions. For instance, it offered more consistent entry point predictions under various sizes of timesteps. AB-2 was clearly effective in terms of eliminating inconsistency of entry points over the region C. However, when an oscillatory stability is of our interest, the intrinsic characteristic of AB method made it a bad candidate, as it takes the average value of previous time points. It may not accurately reflect the oscillatory characteristic of the system. As Morris-Lecar model has been widely adopted in the biological research, this project offered a comprehensive comparison between different explicit numerical methods. Our results suggested that, to compromise between stability and consistency, low-order of AB method will be optimal for an oscillatory system.

## Reference

- G.B Ermentrout, J.Rinzel(1984), Beyond a pacemakers entrainment limit: phase walk-through, Am J Physiol Regulatory Integrative Comp Physiol 246:102-106
- A. Hodge, R. Edwards, D.H.Paul, P. van den Driessche(2006), Neuronal network models of phase separation between limb CPGs of digging sand crabs, Biological Cybernetics 95:55-68.
- Frances K. Skinner, Nancy Kopell, Eve Marder(1994), Mechanisms for Oscillation and Frequency Control in Reciprocally Inhibitory Model Neural Networks, Journal of Computational Neuroscience1, 69-87
- G.B Ermentrout, N. Kopell(1991), Multiple pulse interactions and averaging in systems of coupled neural oscillators J. Math Biol 29: 195-217.
- F.C Hoppensteadt C.S. Peskin, Mathematics in Medicine and the Life Sciences
- Catherine Morris, Harold Lecar(1981), Voltage oscillator in the barnacle giant muscle fiber,Biophys.J Vol35 193-213.

## Notes

<sup>1</sup>Source:<http://upload.wikimedia.org/wikipedia/commons/thumb/c/cf/Hodgkin-Huxley.jpg/300px-Hodgkin-Huxley.jpg>

<sup>2</sup>Source:A Hodge et al(2006)

## A Numerical Values

$$w = 1.9$$

$$g_{syn} = 0.0025$$

$$g_l = 0.005$$

$$g_k = 0.02$$

$$g_{ca} = 0.015$$

$$v_{syn} = -80$$

$$v_l = -50$$

$$v_k = -80$$

$$v_{ca} = 100$$

$$v_a = 0$$

$$v_b = 15$$

$$v_c = 0$$

$$v_d = 15$$

$$\eta_n = 0.002$$

$$i_{app} = 100$$

$$a_1 = a_2 = a_3 = 1$$

$$b_1 = b_2 = b_3 = 1$$

## B Simulation with Euler's method

```
1 clear
2 w=1.9
3 g_syn=0.0025;
4 g_l=0.005;
5 g_k=0.02;
6 g_ca=0.015;
7 v_syn=-80;
8 v_l=-50;
9 v_k=-80;
10 v_ca=100;
11 v_a=0;
12 v_b=15;
13 v_c=0;
14 v_d=15;
15 eta_n=0.002;
16 iapp=100;
17 v(1,1)=100
18 an(1,1)=0.6
19 v(2,1)=-200
20 an(2,1)=0.9
21 v(3,1)=150
22 an(3,1)=0.1
23 z=[1:w:10000];
24 for i=1:length(z)
25 kw=[0.005*(-50-v(1,i))+0.2*an(1,i)*(-80-v(1,i))+0.075*(1+tanh(v(1,i)/15))...
26      *(100-v(1,i));0.002*cosh(v(1,i)/30)*(1/2*(1+tanh(v(1,i)/15))-...
27      an(1,i))+g_syn*[(1*0.5*(1+tanh(v(3,i)))+1*0.5*tanh((v(2,i))))*...
28      (v_syn-v(1,i));0];
29 kx=[0.005*(-50-v(2,i))+0.2*an(2,i)*(-80-v(2,i))+0.075*(1+tanh(v(2,i)/15))...
30      *(100-v(2,i));0.002*cosh(v(2,i)/30)*(1/2*(1+tanh(v(2,i)/15))-...
31      an(2,i))+g_syn*[(1*0.5*(1+tanh(v(1,i)))+1*0.5*tanh((v(3,i))))*...
32      (v_syn-v(2,i));0];
33 kz=[0.005*(-50-v(3,i))+0.2*an(3,i)*(-80-v(3,i))+0.075*(1+tanh(v(3,i)/15))...
34      *(100-v(3,i));0.002*cosh(v(3,i)/30)*(1/2*(1+tanh(v(3,i)/15))...
35      -an(3,i))+g_syn*[(1*0.5*(1+tanh(v(2,i)))+1*0.5*tanh((v(1,i))))*...
36      (v_syn-v(3,i));0];
37 v(1,i+1)=kw(1)*w+v(1,i);
38 an(1,i+1)=kw(2)*w+an(1,i);
39 v(2,i+1)=kx(1)*w+v(2,i);
40 an(2,i+1)=kx(2)*w+an(2,i);
41 v(3,i+1)=kz(1)*w+v(3,i);
42 an(3,i+1)=kz(2)*w+an(3,i);
43 end
44 subplot(2,2,1)
45 plot(v(1,:),an(1,:), 'r')
46 axis([-200 200 0 1]);
47 xlabel('v1')
48 ylabel('n1')
49 hold on
50 subplot(2,2,2)
51 plot(v(2,:),an(2,:), 'r')
52 axis([-200 200 0 1]);
```

```

53 xlabel('v2')
54 ylabel('n2')
55 hold on
56 subplot(2,2,3)
57 plot(v(3,:),an(3:),'r')
58 axis([-200 200 0 1]);
59 xlabel('v3')
60 ylabel('n3')
61 z=linspace(1,10000,i+1)
62 hold on
63 % subplot(2,2,4)
64 % plot(z,v(1:),'k-',z,v(2:),'k:',z,v(3:),'k--')
65 % legend('v1','v2','v3')
66 % xlabel('time')
67 % ylabel('membrane potential')
68 title('simulation of limit cycles')
69

```

## C Simulation with AB-2 method

```
1 clear
2 w=1.9
3 g_syn=0.0025;
4 g_l=0.005;
5 g_k=0.02;
6 g_ca=0.015;
7 v_syn=-80;
8 v_l=-50;
9 v_k=-80;
10 v_ca=100;
11 v_a=0;
12 v_b=15;
13 v_c=0;
14 v_d=15;
15 eta_n=0.002;
16 iapp=100;
17 v(1,1)=100
18 an(1,1)=0.6
19 v(2,1)=-200
20 an(2,1)=0.9
21 v(3,1)=150
22 an(3,1)=0.1
23 z=[1:w:10000];
24 for i=1:length(z)
25 kw=[0.005*(-50-v(1,i))+0.2*an(1,i)*(-80-v(1,i))+0.075*(1+tanh(v(1,i)/15))...
26      *(100-v(1,i));0.002*cosh(v(1,i)/30)*(1/2*(1+tanh(v(1,i)/15))-...
27      an(1,i))+g_syn*[(1*0.5*(1+tanh(v(3,i)))+1*0.5*tanh((v(2,i))))*...
28      (v_syn-v(1,i));0];
29 kx=[0.005*(-50-v(2,i))+0.2*an(2,i)*(-80-v(2,i))+0.075*(1+tanh(v(2,i)/15))...
30      *(100-v(2,i));0.002*cosh(v(2,i)/30)*(1/2*(1+tanh(v(2,i)/15))-...
31      an(2,i))+g_syn*[(1*0.5*(1+tanh(v(1,i)))+1*0.5*tanh((v(3,i))))*...
32      (v_syn-v(2,i));0];
33 kz=[0.005*(-50-v(3,i))+0.2*an(3,i)*(-80-v(3,i))+0.075*(1+tanh(v(3,i)/15))...
34      *(100-v(3,i));0.002*cosh(v(3,i)/30)*(1/2*(1+tanh(v(3,i)/15))...
35      -an(3,i))+g_syn*[(1*0.5*(1+tanh(v(2,i)))+1*0.5*tanh((v(1,i))))*...
36      (v_syn-v(3,i));0];
37 v(1,i+1)=kw(1)*w+v(1,i);
38 an(1,i+1)=kw(2)*w+an(1,i);
39 v(2,i+1)=kx(1)*w+v(2,i);
40 an(2,i+1)=kx(2)*w+an(2,i);
41 v(3,i+1)=kz(1)*w+v(3,i);
42 an(3,i+1)=kz(2)*w+an(3,i);
43 end
44 subplot(2,2,1)
45 plot(v(1,:),an(1,:), 'r')
46 axis([-200 200 0 1]);
47 xlabel('v1')
48 ylabel('n1')
49 hold on
50 subplot(2,2,2)
51 plot(v(2,:),an(2,:), 'r')
52 axis([-200 200 0 1]);
```

```

53 xlabel('v2')
54 ylabel('n2')
55 hold on
56 subplot(2,2,3)
57 plot(v(3,:),an(3:),'r')
58 axis([-200 200 0 1]);
59 xlabel('v3')
60 ylabel('n3')
61 z=linspace(1,10000,i+1)
62 hold on
63 % subplot(2,2,4)
64 % plot(z,v(1:),'k-',z,v(2:),'k:',z,v(3:),'k--')
65 % legend('v1','v2','v3')
66 % xlabel('time')
67 % ylabel('membrane potential')
68 title('simulation of limit cycles')
69

```

## D Simulation with AB-3 method

```
1 clear
2 w=1.9
3 g_syn=0.0025;
4 g_l=0.005;
5 g_k=0.02;
6 g_ca=0.015;
7 v_syn=-80;
8 v_l=-50;
9 v_k=-80;
10 v_ca=100;
11 v_a=0;
12 v_b=15;
13 v_c=0;
14 v_d=15;
15 eta_n=0.002;
16 iapp=100;
17 v(1,1)=100
18 an(1,1)=0.6
19 v(2,1)=-200
20 an(2,1)=0.9
21 v(3,1)=150
22 an(3,1)=0.1
23 z=[1:w:10000];
24 for i=1:length(z)
25 kw=[0.005*(-50-v(1,i))+0.2*an(1,i)*(-80-v(1,i))+0.075*(1+tanh(v(1,i)/15))...
26      *(100-v(1,i));0.002*cosh(v(1,i)/30)*(1/2*(1+tanh(v(1,i)/15))-...
27      an(1,i)]+g_syn*[(1*0.5*(1+tanh(v(3,i)))+1*0.5*tanh((v(2,i))))*...
28      (v_syn-v(1,i));0];
29 kx=[0.005*(-50-v(2,i))+0.2*an(2,i)*(-80-v(2,i))+0.075*(1+tanh(v(2,i)/15))...
30      *(100-v(2,i));0.002*cosh(v(2,i)/30)*(1/2*(1+tanh(v(2,i)/15))-...
31      an(2,i)]+g_syn*[(1*0.5*(1+tanh(v(1,i)))+1*0.5*tanh((v(3,i))))*...
32      (v_syn-v(2,i));0];
33 kz=[0.005*(-50-v(3,i))+0.2*an(3,i)*(-80-v(3,i))+0.075*(1+tanh(v(3,i)/15))...
34      *(100-v(3,i));0.002*cosh(v(3,i)/30)*(1/2*(1+tanh(v(3,i)/15))...
35      -an(3,i)]+g_syn*[(1*0.5*(1+tanh(v(2,i)))+1*0.5*tanh((v(1,i))))*...
36      (v_syn-v(3,i));0];
37 v(1,i+1)=kw(1)*w+v(1,i);
38 an(1,i+1)=kw(2)*w+an(1,i);
39 v(2,i+1)=kx(1)*w+v(2,i);
40 an(2,i+1)=kx(2)*w+an(2,i);
41 v(3,i+1)=kz(1)*w+v(3,i);
42 an(3,i+1)=kz(2)*w+an(3,i);
43 end
44 subplot(2,2,1)
45 plot(v(1,:),an(1,:), 'r')
46 axis([-200 200 0 1]);
47 xlabel('v1')
48 ylabel('n1')
49 hold on
50 subplot(2,2,2)
51 plot(v(2,:),an(2,:), 'r')
52 axis([-200 200 0 1]);
```

```
53 xlabel('v2')
54 ylabel('n2')
55 hold on
56 subplot(2,2,3)
57 plot(v(3,:),an(3:),'r')
58 axis([-200 200 0 1]);
59 xlabel('v3')
60 ylabel('n3')
61 z=linspace(1,10000,i+1)
62 hold on
63 % subplot(2,2,4)
64 % plot(z,v(1:),'k-',z,v(2:),'k:',z,v(3:),'k--')
65 % legend('v1','v2','v3')
66 % xlabel('time')
67 % ylabel('membrane potential')
68 title('simulation of limit cycles')
69
```

## E Simulation with AB-4 method

```
1 clear
2 w=1.9
3 g_syn=0.0025;
4 g_l=0.005;
5 g_k=0.02;
6 g_ca=0.015;
7 v_syn=-80;
8 v_l=-50;
9 v_k=-80;
10 v_ca=100;
11 v_a=0;
12 v_b=15;
13 v_c=0;
14 v_d=15;
15 eta_n=0.002;
16 iapp=100;
17 v(1,1)=100
18 an(1,1)=0.6
19 v(2,1)=-200
20 an(2,1)=0.9
21 v(3,1)=150
22 an(3,1)=0.1
23 z=[1:w:10000];
24 for i=1:length(z)
25 kw=[0.005*(-50-v(1,i))+0.2*an(1,i)*(-80-v(1,i))+0.075*(1+tanh(v(1,i)/15))...
26      *(100-v(1,i));0.002*cosh(v(1,i)/30)*(1/2*(1+tanh(v(1,i)/15))-...
27      an(1,i))+g_syn*[(1*0.5*(1+tanh(v(3,i)))+1*0.5*tanh((v(2,i))))*...
28      (v_syn-v(1,i));0];
29 kx=[0.005*(-50-v(2,i))+0.2*an(2,i)*(-80-v(2,i))+0.075*(1+tanh(v(2,i)/15))...
30      *(100-v(2,i));0.002*cosh(v(2,i)/30)*(1/2*(1+tanh(v(2,i)/15))-...
31      an(2,i))+g_syn*[(1*0.5*(1+tanh(v(1,i)))+1*0.5*tanh((v(3,i))))*...
32      (v_syn-v(2,i));0];
33 kz=[0.005*(-50-v(3,i))+0.2*an(3,i)*(-80-v(3,i))+0.075*(1+tanh(v(3,i)/15))...
34      *(100-v(3,i));0.002*cosh(v(3,i)/30)*(1/2*(1+tanh(v(3,i)/15))...
35      -an(3,i))+g_syn*[(1*0.5*(1+tanh(v(2,i)))+1*0.5*tanh((v(1,i))))*...
36      (v_syn-v(3,i));0];
37 v(1,i+1)=kw(1)*w+v(1,i);
38 an(1,i+1)=kw(2)*w+an(1,i);
39 v(2,i+1)=kx(1)*w+v(2,i);
40 an(2,i+1)=kx(2)*w+an(2,i);
41 v(3,i+1)=kz(1)*w+v(3,i);
42 an(3,i+1)=kz(2)*w+an(3,i);
43 end
44 subplot(2,2,1)
45 plot(v(1,:),an(1,:), 'r')
46 axis([-200 200 0 1]);
47 xlabel('v1')
48 ylabel('n1')
49 hold on
50 subplot(2,2,2)
51 plot(v(2,:),an(2,:), 'r')
52 axis([-200 200 0 1]);
```

```
53 xlabel('v2')
54 ylabel('n2')
55 hold on
56 subplot(2,2,3)
57 plot(v(3,:),an(3:),'r')
58 axis([-200 200 0 1]);
59 xlabel('v3')
60 ylabel('n3')
61 z=linspace(1,10000,i+1)
62 hold on
63 % subplot(2,2,4)
64 % plot(z,v(1:),'k-',z,v(2:),'k:',z,v(3:),'k--')
65 % legend('v1','v2','v3')
66 % xlabel('time')
67 % ylabel('membrane potential')
68 title('simulation of limit cycles')
69
```