

# Neuroinformatics — lab exercises manual

authors:

Eduard Bakstein (TA), Daniel Novák (Lecturer)  
<http://nit.felk.cvut.cz>, Prague 2015

March 2, 2017

## Abstract

This is supportive text for labs of Neuroinformatics course at Czech Technical University in Prague, Faculty of Electrical Engineering. Exercises for each task are provided in Matlab language.

We further recommend the book [3], which is nice and easy to read. A good summary is provided by [1]. Particular topics are covered by [2], [4] Some advanced material is covered by [2], [4].

## Contents

<b>Bibliography</b>	<b>1</b>
<b>0 Mathematical apparatus</b>	<b>2</b>
0.1 Numerical solution of differential equations . . . . .	2
<b>1 Neuron models</b>	<b>4</b>
1.1 Model of membrane and synapse: simplest case . . . . .	4
<b>2 Real data analysis, modelling of neuronal populations</b>	<b>6</b>
<b>3 Information coding and transfer in the brain</b>	<b>7</b>

## Bibliography

- [1] David Fitzpatrick William C. Hall Anthony-Samuel LaMantia Leonard E. White Dale Purves, George J. Augustine. *Neuroscience*. Sinauer Associates, Inc., 5th. edition edition, 2011.
- [2] Michael L. Hines Nicholas T. Carnevale. *The Neuron Book*. Cambridge University Press, 2006.
- [3] Thomas Trappenberg. *Fundamentals of Computational Neuroscience*. Oxford University Press, USA, June 2010.
- [4] Werner M. Kistler Wulfram Gerstner. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

## 0 Mathematical apparatus

We will introduce numerical apparatus, which will be applied during activity modelling of neurons.

### 0.1 Numerical solution of differential equations

As in other fields dealing with dynamic systems, we will use differential equations throughout this course. Due to the fact that their exact analytical solution can often be difficult or even impossible to obtain, we introduce the approximate numerical solutions in this exercise.

#### Euler's method

Let's consider the first order differential equation

$$\frac{\delta x}{\delta t} = f(x, t), \quad (0.1)$$

Euler's first order method consist of discretization  $\frac{\delta x}{\delta t} = \frac{\Delta x}{\Delta t}$ . Lets' take:

$$\begin{aligned} \Delta x &= x(t + \Delta t) - x(t) = x(t_2) - x(t) \\ \Delta t &= t_2 - t, \end{aligned}$$

then we can express eq (0.1) as

$$\frac{\Delta x}{\Delta t} = f(x(t), t)$$

finally

$$x(t + \Delta t) = x(t) + \Delta t f(x(t), t)$$

We can approximate the solution by taking into account the slope of the line at that point. However, if the slope is dependent on  $t$ , this will lead to a very rough approximation. In this case, we can refine the solution by other parameters of the Taylor series according to the formula:

$$x(t + \Delta t) = x(t) + \Delta t \frac{\delta x}{\delta t} + \frac{1}{2} (\Delta t)^2 \frac{\delta^2 x}{\delta t^2} + O, \quad (0.2)$$

where  $O$  represents all members of the higher orders. Therefore, the second order model is extended by the curvature (2nd order derivative). This approximation should be closer to the analytical solutions.

#### Runge-Kutta method

In more sophisticated methods, we can additionally refine the solution by estimation not in the point  $x$  or  $x + \Delta x$ , but in the middle of this interval - the so-called „*midpoint method*“. The resulting value should be more representative and lead to a more accurate estimate. Moreover, if we do not have analytical expression of parameters of the Taylor expansion of higher orders, we can again estimate parameters of high orders using numerical methods.

The actual Runge-Kutta method for numerical integration is the 4th order, which combines estimation in the middle of the interval with a numerical estimate of higher order. It can therefore approximate solution of arbitrary functions. This leads to a higher computational cost, but - as we shall see - leads to very accurate estimates of the solution. Runge-Kutta method is implemented in Matlab function `ode45()`.

**Exercise 0.1** Our task is numerical approximation of the following differential equation

$$\frac{dx}{dt} = t - x + 1, \quad (0.3)$$

initial conditions:  $x(0) = 1$  using Euler's method (1 and 2 order) and Runge-Kutta methods.

The analytical solution of this equation has the form  $x = t + e^{-t}$ <sup>1</sup>. Solve the equation in the interval  $(0, 5)$ , choose  $\Delta t = 0.02$ .

---

<sup>1</sup>You can check e.g. at <http://www.wolframalpha.com/>, question: „solve differential equation x '= ... “

**Task 0.1** (2 b) Plot the solution  $x(t) = f(t)$  for  $t = 0, \dots, 2s$  using Euler's method of first and second order<sup>2</sup> ( $X_{Euler}$ ) and plot into graph along with the analytical solution.

**Task 0.2** (1 b) Solve the task using Runge-Kutta  $x_{Runge}$  and add the solution to the same graph.

*Hints:* in the function `ode45`, the first parameter is a callback function, which describes the equation to be solved. In this case the callback can be easily defined using so called *function handle* and *anonymous function*<sup>3</sup> as `ode_func = @(t,x,flag) 1-x+t;`

**Task 0.3** (1.5 b) Plot the dependence of relative error of each numerical method and compare it to the analytical solution. For example (for Euler's method)  $(x_{Euler} - x_{exact})/x_{Euler} = f(\Delta t)$

**Task 0.4** (1.5 b) Plot absolute error of each numerical method as a function of the size of the integration step  $\Delta t \in (0.001, 1)s$  in time  $t = 1s$ .

---

<sup>2</sup>when solving the second order Euler method you can proceed in two ways. 1) Derivating analytic function  $f(x, t)$ .  
2) estimate value of  $f'(x, t)$  using the slope  $f(x, t)$  between points  $t$  and  $t + \text{Deltat}$

<sup>3</sup>see for example. [http://www.mathworks.com/help/matlab/matlab\\_prog/creating-a-function-handle.html](http://www.mathworks.com/help/matlab/matlab_prog/creating-a-function-handle.html)

# 1 Neuron models

In this block, we will describe properties of the action potential and behavior of individual neurons. We will implement several membrane and neuron models of differing level of complexity and investigate how the simplifying assumptions affect the resulting action potentials or spike-trains.

## 1.1 Model of membrane and synapse: simplest case

Considering that the action potentials propagate as changes in electrical potential on the cell membrane, it seems natural to model the membrane as an equivalent electrical circuit. The simplest model that we introduce in this exercise works with single type of ion channels only - the leakage chloride channels (always open) - and consists of an RC circuit and a voltage power source, representing the membrane resting potential (*Nernst potential*). As we shall see, its response to the input excitation represents only a part of the true membrane behavior and is far away from the the real action potential but we can still find it useful in understanding the basic concept and properties of the membrane itself.

**Exercise 1.1 (RC model)** Model memrabe behavior using RC model - see figure 1. Input membrane current  $I_{stim}$  is a rectangular pulse signal  $10pA$  lasting  $20ms$ . It is necessary to convert the stimulation current  $I_{stim}$  to current density to be compatible with  $I_{Cl}$  and  $I_C$ . This can be done easily by dividing the input current by membrane area  $A^4$ . The electrode is stimulating and at the same time recording the membrane current (expressed in  $cm^2$ ). Hence  $I'_{stim} = I_{stim}/A \approx 10^{-11} \cdot 10^6 \approx 10^{-5}$ . The membrane parameters are the following:

- capacity:  $C_m = 1 \mu F/cm^2$ ,
- conductance:  $g_{Cl} = 0.3 ms/cm^2$ ,
- time constant :  $\tau = C_m/g_{Cl}$ ,
- membrane surface :  $A \approx 1 \cdot 10^{-6} cm^2$
- Nernst potential of Cl:  $V_{Cl} = -68 mV$ ,
- initial conditions:  $V(0) = -68 mV$ ,  $I_{Cl}(0) = 0 \mu A/cm^2$ ,  $I_C(0) = 0 \mu A/cm^2$ .

**Task 1.1** (4 b) Plot dependence  $V(t)(= \phi_{in} - \phi_{out})$ ,  $I_{Cl}(t)$ ,  $I_C(t)$  for these time interval  $(0, 40)ms$ ,  $\Delta t = 0.01ms$

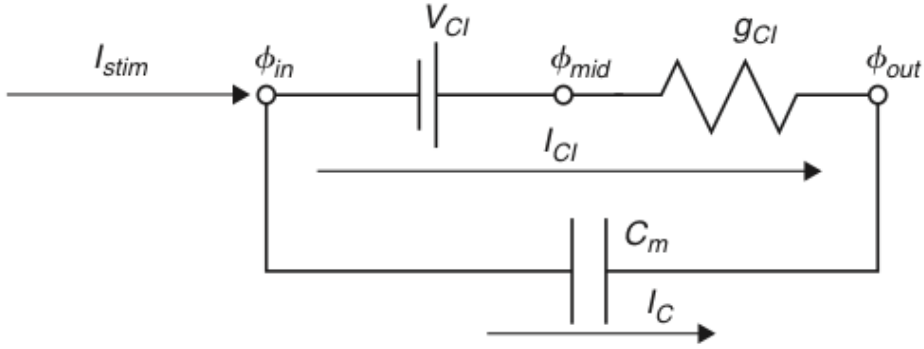


Figure 1: Membrane model with Cl leakage channel

Hence

$$I_C(t) = C_m \frac{dV}{dt}(t) \tag{1.1}$$

$$I_C(t) = \frac{I_{stim}(t)}{A} - I_{Cl}(t) \quad \rightarrow I_{stim} = A \cdot I_C(t) + A \cdot I_{Cl}(t) \tag{1.2}$$

$$I_{Cl}(t) = g_{Cl}(V(t) - V_{Cl}) \tag{1.3}$$

<sup>4</sup> $I_{Cl}$  and  $I_C$  are in fact current densities and it would be more appropriate to use  $\rho$  instead of  $I$ . However, we will stick to the terminology used in [3] and use  $I$ .

We can express the time constant of the RC circuit as

$$\tau = \frac{C_m}{g_{Cl}} \quad (1.4)$$

Combining the above equations we get

$$C_m \frac{dV}{dt} = \frac{I_{stim}(t)}{A} - g_{Cl}(V(t) - V_{Cl}) \quad (1.5)$$

Or expressed using time constant  $\tau$  from 1.4 we get

$$\tau \frac{dV}{dt} = V_{Cl} - V(t) + \frac{I_{stim}(t)}{Ag_{Cl}} \quad (1.6)$$

Euler's method (forward)

$$\tau \frac{V(j) - V(j-1)}{\Delta t} = V_{Cl} - V(j-1) + \frac{I_{stim}(j-1)}{Ag_{Cl}} \quad (1.7)$$

$$V(j) = V(j-1) + \frac{\Delta t}{\tau} [V_{Cl} - V(j-1) + \frac{I_{stim}(j-1)}{Ag_{Cl}}] \quad (1.8)$$

**Exercise 1.2 (EPSP model)** The task is analysis of a model depicted in figure 2. This model incorporates so called „excitatory postsynaptic potential“ (EPSP) simulating behavior of membrane dendrite of the postsynaptic neuron after receiving excitation. Synapse is modeled by variable conductance  $g_{syn}$ . In other words: the model contains additional channels, whose conductance is neurotransmitter-controlled and further voltage and time-dependent (literally in the moment of receiving neurotransmitter the channels are opening, hence their conductance is increasing - see result of eq (1.9)) with time constant  $\tau_{syn} = 1 \text{ mS}$ . The parameters are the same like in the previous case. Stimulation current  $I_{stim} = 0$ ,  $V_{syn} = 10 \text{ mV}$ . In time  $t = 1 \text{ ms}$  the neurotransmitter is released, hence  $g_{syn}(1 + \delta) = 1$ . Initial conditions  $V(1) = 0$ ,  $I_{syn} = 0$ ,  $g_{syn}(1) = 0$ ,  $g_L = 1$ .

$$\tau_{syn} \frac{dg_{syn}(t)}{dt} = -g_{syn}(t) + \delta(t - t_{pre} - t_{delay}) \quad (1.9)$$

**Task 1.2** (1 b) Plot dependence  $V(t)$ ,  $I_{Cl}(t)$ ,  $I_C(t)$ ,  $I_{syn}(t)$ . Explain trends and compare them with the previous task 1.1.

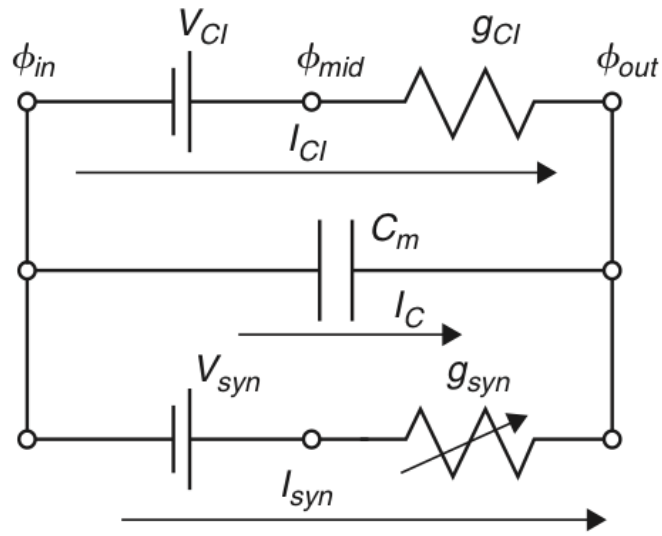


Figure 2: EPSP: Synapse model with leakage Cl channel

## 2 Real data analysis, modelling of neuronal populations

In this block, we will analyze real  $\mu EEG$  (*micro-EEG*) data, recorded from brains of Parkinson's disease patients. We will compare the data to our simulations from the previous block and we will analyze correlation with other biosignal - the electrooculogram (EOG).

Further on in this section, we will simulate more complicated network of artificial neurons and (so called *spiking network*) and analyze its behavior under varying conditions.

### **3 Information coding and transfer in the brain**

In this block, we will model some of the processes taking part in transferring and storing information in the brain. We will go through the basics of cognitive modeling and we will try out the so-called *Self-organizing maps*.