Assignments

COURSE HOME
SYLLABUS
LECTURE NOTES
LECTURE VIDEOS
ASSIGNMENTS
EXAM STUDY GUIDES
DOWNLOAD COURSE MATERIALS

This page includes seven (7) problem sets as well as the rubric and guidelines.

PROBLEM SETS	TOPICS	LIST OF FILES
Problem set 1 (ZIP - 1.1MB)	Diffusion, equivalent circuit and Nernst potential	This zip file contains: 3 .pdf files, 3 .m files, and 1 .mat file
Problem set 2 (ZIP)	Integrate and fire and Hodgkin-Huxley	This zip file contains: 1 .pdf file, 7 .m files, and 1 .p file
Problem set 3 (PDF)	Synapses as computational devices	Reference: Vu, E.T., and F.B. Krasne. "Evidence for a Computational Distinction Between Proximal and Distal Neuronal Inhibition." *Science* 255, no. 5052 (2018): 1710–12.
Problem set 4 (ZIP)	How neurons respond to different stimuli	This zip file contains: 1 .pdf file, 5 .m files, and 1 .mat file
Problem set 5 (ZIP)	Analyzing EEG data	This zip file contains: 1 .pdf file, 2 .m files, and 1 .mat file
Problem set 6 (ZIP)	Perceptrons and PCA	This zip file contains: 1 .pdf file, 1 .m file, and 1 .mat file
Problem set 7 (ZIP)	Recurrent networks	This zip file contains: 2 .pdf files

Problem Set Rubric and Guidelines

Scope of Problem Sets

The problems sets are designed to learn basic aspects of neural computation, where the emphasis is on articulating these models and analyses into working MATLAB®. These problems are the perfect place to assimilate and build upon what you have learned in lecture and recitation.

To achieve these goals, problems will combine: building numerical simulations of neuronal models and processes, analyses and visualization of data coming from either experimental data or as output of simulations, and a bit of analytical and conceptual work.

These exercises are geared towards assessing your coding competency, your ability to report data and models' outcomes effectively, and to evaluate your conceptual understanding of the models and analyses used.

Assessment Criteria (Grading Rubric)

Problem sets will be graded according to three main criteria:

1. Accuracy (80% total grade)

This refers to the extent that your code produces the right output, plots and figures represent data accurately, analytical work is correct, and conceptual questions are well reasoned and logically sound.

Each problem set question will evaluated on a 4-point scale. The following tables give the point/assessment equivalency:

POINTS	ASSESSMENT
4	Code executes without syntax errors and produces correct results throughout. Graphs correctly represent the data. Analytical work is correct. Conceptual questions are correctly reasoned, concise and precise.
3	Code executes without or with minimal syntax errors. One or two errors in code output, graphs, analytical work or conceptual responses are present.

POINTS	ASSESSMENT
2	Code executes with syntax errors. Multiple errors in code, output, graphs, analytical work or conceptual responses are present.
1	Code executes with multiple syntax errors. There are numerous errors in code output, graphs, analytical work and conceptual responses.

2. Adherence to guidelines (10% total grade):

This evaluates the extent to which the code, graphs, equations and conceptual responses are presented per the guidelines specifications set in the assignment submission instructions.

Note that this is different than the 'correctness' criterion defined above. For example, a piece of code can produce correct results, but not in the way specified by the problem statement (wrong graphic display, etc.). Alternatively, code can meet the problem specifications, but contain errors that affect accuracy or validity.

This criterion will be evaluated on a 10-point scale for the whole problem set.

3. Readability (10% total grade):

This evaluates: if the code is well organized and human-readable; if data are plotted with appropriate labeling, scaling and explanations; and if the narrative is well organized and provides explanatory value.

This criterion will be evaluated on a 10-point scale for the whole problem set.

Other Notes

Note that code efficiency (i.e. how optimal is your code in performing the required computations) is not part of the rubric.

Writing efficient code is a skill that develops over the years and after taking several courses. We will try to teach you how to optimize code and will encourage you to produce sound and efficient code.

However, the emphasis will be on understanding the numerical techniques and developing intuitions that will carry to more advanced course work.

All 9.40 assignments should run within 5-10 minutes on your laptop, even with the most inefficient approaches. If your code is taking longer than this, there might be bugs or issues that go beyond writing efficient code.