Modeling, Simulation and Analysis
CS 250, Spring 2018
Homework 3
Due in part **BY 11:59PM** Monday, April 30, and
in part **AT THE BEGINNING OF CLASS** Tuesday, May 1
(Please see the notes below!)

- This contains some more exercises than were on the LOOKAHEAD HW3. For clarity, instructions also additionally emphasize that you are to turn in printouts of both your code and your write-up document. Please let me know if you have any questions about instructions for this assignment!

Project: Runge-Kutta and Hodgkin-Huxley!

Some introductory notes, some of which are new, and some of which are reminders and repeats of notes from the previous assignment:

- This assignment is about implementing Runge-Kutta 4 simulations, building to a simulation of a classic model, the Hodgkin-Huxley model of action potentials in neurons! As with your previous assignment, all code is to be written in Matlab.
- Please submit code answers to every exercise except for those marked as **write-up only**. The code for one exercise may be *very similar* to the code for another exercise for the exercises below, but please do not combine responses—submit different, individual code files for each coding exercise. (It's fine to submit a script that runs multiple other scripts, i.e., running multiple simulations with a single script, but each coding exercise should be in its own file(s), not combined with others.)
- Your write-ups for these exercises should explain how the intended model is implemented in your simulations. Some decisions may be subtle and require some explanation, so please explain as needed! For simulations, *unless values are given to you for an exercise* (which is frequently the case on this assignment!), your write-up should include the values of constants / parameters employed for each run of the simulation and a very brief explanation of why you chose to run those particular values for simulations. Descriptions of results should be concise and information-heavy; feel free to include figures (e.g., Matlab plots) in write-ups to illustrate your observations.

  If there are any questions about what might be good to include in a write-up, please let me know!
- As always, readability is an essential part of the assignment: Make sure both your code and your write-up are easy to read and understand. As part of this, be sure to follow the style guidelines presented in lecture for implementing a Runge-Kutta 4 simulation: Make sure it is easy to tell exactly what function is used to compute the derivative and exactly what arguments are provided to that function for each estimate $\partial_1, \partial_2, \partial_3, \partial_4$.

For example, if you are modeling the dynamics (change in value) of some variable P, then you should have a function `dPdt(...)` somewhere in your code (perhaps it could be defined using anonymous function syntax `dPdt = @(...)  ...`?), and in your simulation loop, you would have lines that call that function, such as

```
dp1 = dPdt( ...  )  * dt;
dp2 = dPdt( ...  )  * dt;
dp3 = dPdt( ...  )  * dt;
dp4 = dPdt( ...  )  * dt;
```

to calculate the estimates $\partial_1, \partial_2, \partial_3, \partial_4$.

Structure like this makes it easy to read and understand an RK4 simulation loop. Code that is not *at least* this easily readable may not merit full credit. Please feel free to ask me any questions about code style and readability for these exercises!

- In all exercises, for full credit, your work must be sufficiently documented to demonstrate an understanding of the relevant concepts and questions for each exercise. In general, always explain your answers and document the process you used to arrive at those answers.

  The purpose of a write-up document is to contribute to the clear, explanatory documentation of your work. It should be employed to complement the commenting in code, providing a description in English prose; it should not be simply a repeat of code comments.

  If there are questions about this or anything else regarding these exercises, please feel free to ask your Prof.!

- For each exercise, please electronically submit your code in `.m` files to the course dropbox, using the `submit250` script: `submit250 hw3 <your-directory-name>`. In addition, on or before the due date, please turn in printouts of your code and your write-up document (with answers to non-programming questions) on paper.

- Your programming as submitted to the course dropbox should contain one `.m` file for each part of these exercises that requires coding. In particular, for this assignment, that means one file each for the following exercises: 1a, 1b, 1c, 1d, 2a, 2b, 2c, 2d, and 2e.

  If for some reason, you feel that it is stylistically important to have more than one file for your approach to one of these exercises, please see me to discuss it! As always, please make stylistic choices that emphasize functionality for the reader / viewer of your code and its output.

- The non-programming portions of this homework, including printouts of all code and your write-up, are due at the beginning of class on May 1.

- The programming portions of this assignment are to be electronically submitted to our course dropbox by 11:59pm on April 30.

- You will need to demo your code with me **before the end of Study Period** as part of the evaluation of your work on these exercises. (Failure to do so will result in an automatic 50% penalty.) Please prepare for that demo, and contact me to schedule it where you are ready!

Exercises

1. **Have A Ball!** These exercises are based on examples from Chapter 3.1: the example of a ball being thrown straight up from the side of a bridge, from the section titled "Acceleration, Velocity, and Position"; and the example of a ball being dropped from a height of 400m, from the section titled "Friction during Fall."

   (a) Implement an Euler simulation of the example of a ball being thrown straight up from the side of a bridge, from the section titled "Acceleration, Velocity, and Position." You should use exactly the parameters, variables, and initial values given in the textbook. Simulate it for the length of time used to get Figure 3.1.2 in the text, using the same timestep used for that figure ($\Delta t = 0.25s$), and graph the values of position and velocity.

   (b) The simulation resulting in Figure 3.1.2 was not an Euler method, however—it was Runge-Kutta 4. So, implement an RK4 simulation of the same system, and simulate it to generate the graph of position and velocity in Figure 3.1.2.

   (When you implement this simulation, please keep in mind exercise 1c below—ideally, your code for this exercise and exercise 1c would be very similar!)

   (c) Now imagine this simulation occurred in a strange world in which gravity (represented by gravitational constant g) was not the only factor in acceleration of the ball. Instead, acceleration varied with velocity and time: for time t, height $h(t)$, and velocity $v(t)$, $a(t) = g + 0.01 * (v(t) + h(t)) + 0.3 * t^2$. (Note: This will also affect the equation for velocity—it will no longer be $v(t) = 15 - 9.8t$!)

   Model and implement an RK4 simulation of this new environment, with the same simulation parameters as above (i.e., simulated for 4 seconds, $\Delta t = 0.25s$).

   If your implementation for exercise 1b is implemented in a way that fits the general form of RK4 simulations, modifying your simulation for exercise 1b to make it work for this exercise will be very straightforward!

   (d) Implement an RK4 simulation of the system given in the section "Friction during Fall," described in Equation Set 3.1.1. As described in the text, run the simulation for 15s; use a timestep of $\Delta t = 0.01s$. Graph the values of position and speed in the same figure—the result will not look exactly like Figure 3.1.5, because you will not have to represent two different scales on the same y-axis, but it will contain the same information as Figure 3.1.5.

   As noted in exercise 1e below, your code for this exercise should be as similar to exercises 1b and 1c as possible.

(e) **(For Write-up Only)** It is often important for simulation code to be easily modified and used for different simulations, so in this case (which I admit is somewhat contrived!), maximal credit answers for the above three RK4 simulations will be as similar to each other as possible. That is, if the code is more different from exercise to exercise, it will earn less credit; if the code is more similar from exercise to exercise, it will earn more credit. (This is not always a good criterion for programming style, it's just being used for these exercises!)

In your write-up, list all changes made to transform your code from exercise 1b to exercise 1c, and from exercise 1c to exercise 1d. Also, explain why your code answers for the three exercises are *maximally similar to each other*, so code couldn't be structured to be correct for those simulations with fewer changes.

This may be a bit of an unusual exercise, so please feel free to ask for clarification on any of it! As always, questions are welcome!

(**IMPORTANT NOTE**: If you feel you need to make slight sacrifices to readability or style to ensure that your three RK4 simulation code answers are as similar as possible, then for this somewhat contrived exercise, please make those slight sacrifices. Be sure to document them in comments in your code, however, so readers understand why you made the choices you did!)

2. **Brainpower: The Hodgkin-Huxley Model!** These exercises are based on the Hodgkin-Huxley model of activation potentials in neurons, presented in Chapter 7.9 in your textbook; in particular, they are related (but not identical!) to project 2, on pages 287–288, so please read that project as preparation for these exercises. You will be implementing Runge-Kutta 4 simulations of variations of the Hodgkin-Huxley model (*HH model*, for short), as described below!

Chapter 7.9 presents the model along with a lengthy list of parameters, variables, and equations needed to implement a simulation of the model—please see Table 7.9.1 on page 287 of your textbook! You can use most of the values in that table when you implement your simulations, but with the last eight symbols in that table, use the values / formulas given in Table 1 in this document, instead of the ones in the textbook. (Note: Do not use T or ϕ in your simulation at all!)

(a) Implement the HH model as in project 2, Chapter 7.9, but include only the sodium, potassium, and leakage channels, not the Na^+-K^+-ATPase pump described on page 288. Thus, the model will be consistent with the HH model equation presented on page 285: $\frac{dV}{dt} = (I_{ext} - I_K - I_{Na} - I_L)/C$, with values / formulas for I_K, I_{Na}, and I_L as given in Table 7.9.1 in the textbook. Do not bother (yet!) keeping track of $[Na^+]$ and $[K^+]$ (see the top of page 288), but do implement the voltage-gating on the sodium and potassium channels (as described in the last paragraph of project 2).

Simulate this as described in the textbook (simulating for 3ms, with timestep 0.001ms, and a stimulus current of 15nA, starting at 0.5ms, lasting for 0.5ms), **except that** the threshold for *Na* channels closing and *K* channels opening should

Symbol (meaning—units)	Formula / value
a_n (opening rate constant—ms^{-1})	$0.01 \cdot (V + 55)/(1 - e^{-(V+55)/10})$
a_m (opening rate constant—ms^{-1})	$0.1 \cdot (V + 40)/(1 - e^{-(V+40)/10})$
a_h (opening rate constant—ms^{-1})	$0.07 \cdot e^{-(V+65)/20}$
b_n (closing rate constant—ms^{-1})	$0.125 \cdot e^{-(V+65)/80}$
b_m (closing rate constant—ms^{-1})	$4 \cdot e^{-(V+65)/18}$
b_h (closing rate constant—ms^{-1})	$1/(e^{-(V+35)/10} + 1)$
T (temperature—°C)	**Do not include in your simulation**
ϕ (factor for temperature correction)	**Do not include in your simulation**

Table 1: Table of some values and formulas for your Hodgkin-Huxley simulation.

be 49.3mV rather than the 50mV value given in the textbook. That is, the *Na* channels should be voltage-gated to close, and the *K* channels voltage-gated to open, at 49.3mV. Show graphs of V, m, n, and h, as in your textbook.

In your write-up, describe the results of the simulation. How well does this simulation match the Hodgkin-Huxley simulation graphs given in your textbook? Be sure to describe your observations of similarities and differences in terms of their neuroscience meanings (e.g., "Without the Na^+-K^+-ATPase pump, the voltage V was ... and the value of sodium gating value m was .... Also, the value of ...")—the point is not just to point out different shapes on graphs, but to explain how various factors affect the properties and functioning of neurons!

(b) Modify the previous simulation to include the Na^+-K^+-ATPase pump (*Na-K pump*, for short) as if it's always on—that is, you will need to modify the equation for $\frac{dV}{dt}$ to include the pump current (call it I_P) in it, but do not bother (yet!) keeping track of $[Na^+]$ and $[K^+]$ concentrations.

As described in the text for project 2 in the textbook, the pump current I_P should have a constant value that counteracts *the initial value* of the leakage current I_L. That way, as long as voltage V is the initial voltage (i.e., the resting voltage, -65mV), the I_P and I_L terms should cancel each other out in the modified $\frac{dV}{dt}$ equation. (Note that as V changes, the value of I_L will change, because I_L is a function of V, but the value of I_P is a constant and will remain unchanged.)

With this modified model, run a simulation with the same parameters and values as in exercise 2a. Show graphs of V, m, n, and h, as in your textbook.

In your write-up, describe the simulation. How well does this simulation match the Hodgkin-Huxley simulation given in your textbook? The graphs from exercise 2a? Be sure to explain the reasons *why* the differences between this simulation and the simulation in exercise 2a are what they are. (As in exercise 2a, be sure to describe your observations in terms of their neuroscience meanings!)

(**NOTE**: Your textbook suggests adding *two* terms to the $\frac{dV}{dt}$ equation, one for current from the Na part of the pump, and one for current from the K part of the pump. For this project assignment, that will not be necessary; combining those into a single term I_P is sufficient.)

(c) Modify the model from exercise 2b to implement the *Na-K* pump as if it's *not* always on, as specified in the text book: the pump is only on when the concentration of $[Na^+]$ inside the axon and $[K^+]$ outside the axon are both greater than 0. Note that this does not add additional terms to the equation for $\frac{dV}{dt}$, although it require changes in the way your code handles voltage-gating for the terms already present there.

Simulate this modified model with the same parameters and values as for exercise 2b. Show graphs of V, m, n, and h, as in your textbook. In your write-up, describe the simulation. How well does this simulation match the Hodgkin-Huxley simulation given in your textbook? The graphs and values from exercise 2b? (As always, please explain the reasons behind your observed similarities and differences, relating them to their neuroscience meanings whenever appropriate.)

(d) Now, try simulating the model from exercise 2c with the Na channels voltage-gated to close at 50mV and K channels voltage-gated to open at 50mV, as specified in the textbook. (Please use the rate constants from the table in this handout, though, not the ones from the book.) Implement and run the simulation. What happens? In your write-up, describe the results. (As always, please explain the reasons behind your observations, relating them to their neuroscience meanings whenever appropriate.)

(e) Here's a hypothesis: In these HH models, having a voltage-gated leakage channel can result in a higher maximum value for the action potential. To test this, implement the leakage channel so that it is only open when the membrane potential V is such that $V \leq -54.4mV$, and then run a simulation of the system (with the same parameters / values as for exercise 2d).

How does this new voltage-gating of the leakage channel affect the simulation when the K and Na channels are voltage-gated at 50mV, as in exercise 2d? How well does this simulation match the Hodgkin-Huxley simulation given in your textbook? The results of previous exercises? In your write-up, describe your results. (As always, please explain the reasons behind your observations, relating them to their neuroscience meanings whenever appropriate.)

(f) **(For Write-up Only)** What are other ways to get the action potential to a higher maximum value? In your write-up, give some suggestions, and explain the reasons that they would result in a higher maximum action potential. Implement your suggestions, and in your write-up, report the results—what difference does each suggestion make? (You do not need to submit code for this exercise, just a write-up describing how you modified code and describing simulation results. You *may* submit short code fragments if you'd like, but not full code files.)

NOTE: In the process of implementing some of the Hodgkin-Huxley simulations with the above specifications, you might find concentrations of some quantities (e.g., sodium or potassium) dropping to zero, or even to negative values. If that happens in your simulations, please make a design decision about how to handle it—e.g., will you sacrifice biological realism and permit such values, or will you do something to prevent them?—and in your documentation, explain what your decision way, why you thought it was the best design choice, and how you implemented it in your code.