

Computational Modeling & Simulation I (Interdisciplinary Science)
CS 346
Spring 2021

Homework 2, Stage 2

Due **BY 11:59PM** Monday, April 12

(Please see the notes below!)

- This is a **team assignment, with restricted collaboration**: You will be working in teams, and you may not consult with other students (whether or not they're in CS346) or online sources (other than those on the course website or from mathworks.com) as part of the exercise. As usual, however, you are welcome to talk with me or our TA (remember, we have TA hours now!) to ask questions or get help with the assignment!
- A little reminder: As part of CS346 naming conventions, please make sure an identifier such as your name or userid is part of every file submitted. For exercises done in teams, each filename should contain the names / userids of all people on the team.
- Please read the **HW Notes & Guidelines** document (posted on the CS346 website) carefully! For this HW, each team should submit two `.m` files—one each for the exercises numbered 1 and 2 below—and a single PDF document for this assignment. Please use the string `HW2S2` (for “HW 2, Stage 2”) in the folder name for submitting this assignment; some variant of that (e.g., `hw2s2`) could be a useful part of filenames for this assignment. See documents on the course website for additional HW guidelines and submission instructions.

Please make sure all submissions are to your Private folder!

The **HW Notes & Guidelines** document contains full instructions for submitting HWs, as well as containing other *important* guidelines about naming conventions for files you submit, and about your PDF write-up—not following those guidelines will result in deductions on your submitted work. Please read that document and let me know if you have any questions.

HW2, Stage 2: Interactions—Reflect, Revise, and Extend!

Before presenting the exercises, here are the introductory notes from Stage 1, which continue to apply:

- This assignment is about modeling system dynamics with interactions, focusing on predator / prey interactions. (See Chapter 4.2 of your textbook for additional details!) All code is to be written in Matlab.

- Some exercises ask you to develop models. When an exercise asks you to develop a model, your write-up should include not only the model itself—e.g., the differential equations (or finite difference equations), constants, and initial conditions that may comprise the model—but also an explanation of how the model was developed, including what each variable / parameter in the model stands for (“ G is the growth constant ...”), what simplifying assumptions were made, and what the reasons were for your decisions (“We assume H is constant because we are modeling on the order of months, not years ...”). If other information is pertinent (e.g., why a certain variable / parameter is *not* present in a model), explain that in the write-up as well.
- Some exercises ask you to run simulations. When an exercise asks you to run a simulation, your write-up should include the values of constants / parameters employed for each run of the simulation (e.g., in simulation 0, $G = 0.01, dt = 0.1, \dots$; in simulation 1, $G = 0.01, dt = 0.01, \dots$) and a very brief explanation of why you chose to run those particular values for simulations (“We tested five values between 0 and 1 to illustrate how the system behaves with short time parameters, which is consistent with our assumption that ...”), as well as a description of the results of the simulation (“... observed that ..., and in particular, at time $t = 3, \dots$ ”). Descriptions of results should be concise and information-heavy; feel free to include figures (e.g., Matlab plots of time series) in write-ups to illustrate your observations.
- Some exercises below ask for you to run multiple experiments / tests and come to conclusions based on the data you generate. As noted below, code submitted for these exercises would ideally, when run, perform sufficient tests (and display sufficient output) to convincingly demonstrate your conclusion, without needing to modify the code and run additional tests. Even though you may have done many, many tests to come to your conclusions—and that should be documented in your write-up (see the *HW Notes & Guidelines* document)—it is common in practice to have your code, when run, do a representative sample of the full range of tests you did; your write-up can thoroughly explain how a user of your code can modify code to run all of the tests you did, if they want to fully replicate your work.

In general, and for these exercises in particular, if it’s not stylistically possible or appropriate to automate testing that way, please explain that and give thorough instructions on how to use your code to replicate your results.

- As emphasized in other course documents, readability is essential: Please use good commenting and documentation style, and make sure both your code and your write-up are easy to read and understand.

Exercises

1. **Up The Food Chain—Reconsidered!** In HW2 Stage 1, you developed and simulated a predator-prey model for a scenario in which there are three species involved: There’s a prey species Y (e.g., tuna) and a predator species P (e.g., sharks) that consumes Y , and in addition, there’s a third species H (e.g., humans) that hunts both

P and Y . That model was created under the assumptions that H hunts P and Y equally—e.g., in the specific case mentioned above, that humans have equal fishing rates (i.e., the same rate constants in the relevant models) for tuna and sharks—and that the population of H stays constant over the time being simulated.

Now, for HW2 Stage 2, you and your teammate should work together to do the following (which is similar but not identical to what you did for HW1 Stage 2):

- (a) Read over the code that each of you submitted for HW2 Stage 1, considering the differential equations you submitted upon which that code was based. Working together, analyze each code submission: For each body of code, you as a team should identify and enumerate things that you found beneficial and things that could be improved. *Please be sure to acknowledge the strengths in each other's code, not just potential room for improvement.*

To guide you, please consider not just whether or not the code gives correct output, but also all of the criteria relevant for good code in this domain—e.g., easy to read and understand, easy to use, easy to analyze, easy to maintain or extend—that we have already discussed this semester. Be sure to consider not just your own perspectives on your team's code, but also those of potential outside readers. For example, what did you find difficult to understand in your teammate's code? What do you think that other intermediate-level programmers, not in Colby's CS346, would find difficult to understand, both in your code and in your teammate's code? Please be specific in documenting the elements of the original code submissions to which you refer. (Feel free to use line numbers from the submitted files for HW2 as part of referring to them, and if you think it improves the clarity of your report, feel free to include code with line numbers in your write-up.)

In addition, please consider and discuss any differences between the models (i.e., the systems of differential equations) underlying your code implementations, and if there are differences, explain the reasons for those differences.

Please be thorough in your analysis and discussion with your teammate, and complete in your write-up! (Please present your analysis in easily readable paragraphs, rather than bullet-pointed lists or tables.) An analysis or report that misses important elements of correctness or style—whether from your own perspective as CS346 students or from the “outsider” perspective described above—may not receive full credit.

- (b) Together, as a team, create a revised version of code for HW2 Stage 1, reflecting your team's analysis of your original submissions, both the strengths of those original submissions and the ways you identified that they could be improved. (Exercise 1c, below, will ask you to submit the code in an aptly-named `.m` file to be easily run and evaluated.) You are also welcome to make accompanying revisions to the dynamical system model from your answers in your original HW2, as appropriate.

What changes did you make in your revised code to make things easier to understand, for each other and for outside readers, and how do you believe those changes helped to make your code easier to read, understand, analyze, maintain, and extend? In your write-up, be sure to document the code thoroughly (as usual), including changes from your original versions and the reasons for those changes. Once again, please be specific in documenting the elements of code to which you refer.

- (c) Then, with your revised code (and accompanying dynamical system model), repeat your work from HW 2 Stage 1: To illustrate the predator-prey interactions between P and Y , without H , use your model to run several simulations with no fishing (but with different values of other relevant factors for predator-prey interaction). Then, run additional simulations with fishing, gradually increasing the fishing rate (i.e., the rate constant for H hunting P and Y), but keep it lower than the birth rate of Y . As always, document your process (including what values you used in simulations) and report your results.

For this part of the exercise, please submit the code for your simulations, which should include the display of visualizations to demonstrate your work (i.e., graphs of results). It could be effective and helpful to have time series visualizations both on-screen in Matlab upon running your code and also present in your write-up, as part of documenting your work.

In your write-up for this part of the exercise, for completeness, please be sure to include the differential equations for the model underlying your code, whether or not it was revised from either teammate's original HW2 submission, and give a brief explanation of the model and the assumptions underlying it.

In addition, please also answer the following questions in your write-up: As the fishing rate changes, what happens to the number of prey and predators? How do you explain the impact of predator H given your model of P , Y , and H ?

- (d) In your write-up, please reflect briefly on this process: As you reviewed each other's code and discussed style and technique, what did each of you learn about your own programming style by looking at and discussing each other's code, and writing a new version reflecting your shared perspectives? If neither teammate learned anything important in the process, please say so, but be sure you have described the elements of programming style and technique you discussed as part of the process of analysis and reflection. (This does not need to be lengthy—one or two paragraphs could be more than sufficient—although please feel free to submit longer answers if you have more to say for this part of the exercise!)

In addition, in your write-up, please also reflect on observable changes due to revising your code: Discuss whether the output or performance of your code in exercise 1c differed from the output or performance of each teammate's original code for HW2 Stage 1. Please provide a short paragraph (or more, if you'd like!) for each teammate, discussing the relevant comparisons.

- (e) Now that you've revised and reflected, you can use your newly revised code for some new explorations!

Run more simulations as in Exercise 1c above, increasing the fishing rate. In your write-up, please answer and explain the following: At what fishing rate do the predators P all die? How does this relate to the birth rate of Y ? Explain the reasons for this; if the equilibria solutions from HW2 Stage 1, exercise 1b (which you are of course invited to now share with your teammate!) are helpful to inform your answer, please include that in your explanation.

In submitting your code, please have the same `.m` file contain your code for both Exercise 1c and 1e.

Ideally, your submitted code would, when run, perform sufficient tests (and display sufficient output) to convincingly demonstrate your conclusion, without needing to modify code and run additional tests. Please do not, however, submit code that, when run, does more simulations than are needed (or shows more output than is needed) to convince a reader. Instead, your code could run a representative subset of your testing, enough to be convincing of your conclusion, and make sure it's easy for code users to modify your code to do additional testing as desired. (Please note that this is consistent with what's in the **Write-ups** section of our CS346 *HW Notes & Guidelines* document about guidelines when including a representative subset of your tests in a write-up.)

2. **Change It Up!** In Exercise 1, the population of species H stayed constant in the model. In this exercise, that restriction is removed.

- (a) Develop a model similar to that in Exercise 1, above, with two differences: species H hunts only species P , not species Y ; and the population of H should also change over time. (As a result, this model will have a differential equation for $\frac{dH}{dt}$ that was not present in the previous model!) In this model, you can treat species P as prey with respect to how its deaths are modeled, and as predator with respect to how its births are modeled.

In your write-up document, give the differential equations for this model and document your process for developing the model, as in HW2 Stage 1.

In addition, implement and run simulations using your model to illustrate changes in populations over time. Please include code and visualizations to demonstrate your work, as usual, and document and explain your results in your write-up.

- (b) Develop a model similar to Exercise 2a, except that in this model, species H hunts both species P and species Y .

As in Exercise 2a, write the differential equations and document your process for developing the model, and implement and run simulations to illustrate population dynamics. Include code and visualizations to demonstrate your work, as usual, and document and explain your results in your write-up document.

(c) Now, do *hypothesis testing* with these two models! But first, some notes about this exercise:

- **Please read through this entire exercise carefully before doing any programming** for it—you are asked to do some thinking and writing before any coding.
- Note: This exercise is conceptually about the process of coming up with hypotheses and testing them with simulations. Whether or not a hypothesis is actually supported by data is not directly relevant to evaluating your work for this exercise; the content and depth of the process of coming up with a hypothesis and testing it are primary in determining the evaluation.
- As always, please ask your Prof. if you have any questions about this exercise!

First, come up with a question you'd like to answer involving the systems being modeled, the models themselves, and the specific values of variables in those models. Then, formulate (and write down, in your write-up) a hypothesis—we'll call this the *primary hypothesis* for this exercise. For example, you might wonder if species H could survive if there were never any of species Y in the system, and your primary hypothesis could be “In this model, the survival of species H relies on the survival of species Y ”; you could test it by running simulations in which Y is extinct, to see if H goes extinct too. (Please don't use this hypothesis for your own work—please come up with something much more interesting, which isn't as readily apparent just from looking at the differential equations in the models!)

Sometimes, in practice, a primary hypothesis simply isn't supported by data. To prepare for that possibility, even before collecting or examining data, it can be helpful to come up with an alternative *competing hypothesis* about the same phenomena. For the purposes of this exercise, your competing hypothesis should have a particular property: It should be another hypothesis about the system or model, specifically one that *cannot* be true if the primary hypothesis is true although it *might* be true if the primary hypothesis is not true. (That is, it could be the case that both the primary hypothesis and the competing hypothesis are false, but it cannot be the case that both are true.) For example, given the above primary hypothesis, a competing hypothesis about the system and the model might be “In this model, the absence of species Y will result in the unbounded growth of species H .” (Again, please don't use this hypothesis for your own work—please come up with something else, and please do not choose a null hypothesis!) Note that a hypothesis “In this model, the absence of species Y will result in the unbounded growth of species P ” is *not* competing in this sense, because it could be true when the primary hypothesis is true.

For this exercise, **before doing any simulations to test your primary hypothesis**, formulate (and write down, in your write-up) an alternative competing hypothesis as well as your primary hypothesis.

Then, run simulations of both models as needed to **test both your primary and competing hypotheses**. Do the data support or contradict your hypotheses? Be sure to fully document your process, describe the results of simulations, and give an explicit answer to whether or not the data support each of your hypotheses!

As noted in the *HW Notes & Guidelines* document, you may not need to include every visualization from every test you do in your write-up—sometimes, output from a subset of the tests is sufficient to effectively communicate the main ideas, whereas including every visualization would overload your readers! For example, if you did 20 tests, you might consider showing only 4–6 of the visualizations, if that would suffice to communicate the main points to your readers. Consider for yourself, if you were a reader, would you learn more from 20 visualizations than from 4–6 that illustrate the main point? In your write-up, clear, effective communication *is* the point; please choose what to include in your write-up accordingly!

Your work for this exercise should include both a write-up and the code that was run to test your primary and competing hypotheses, with appropriate visualizations. Although it is often acceptable to make your code easy to modify and let the user determine how to use it in tests—indeed, part of good style for this kind of Computational Science is making sure code is easily modifiable for additional experiments or tests—here, please automate the testing, so that someone could simply run your `.m` file to see all the results you used to arrive at your conclusions.

If this isn't practical for some reason (because, e.g., there is *no possible way* to do that with clean, stylistically appropriate code, or because it requires too many visualizations to usefully show in a single run of code), then do two things instead: (a) please make sure your code, when run, does enough tests and shows enough output to support your conclusions—e.g., maybe 4–6 tests and visualizations, for reasons mentioned above; and (b) to ensure replicability, please be sure to give thorough instructions to enable someone with your code to fully replicate your results. Please do not, however, submit code that performs and displays more output than is needed to support your conclusions when it's run—the point is to inform your audience, not needlessly overwhelm them!

NOTE: When formulating and writing up your hypotheses, please be sure to specify the *independent variables*, whose values you will vary to see the results on the target system being studied. And please make sure you can experimentally test the hypotheses you formulate! In addition, it is often helpful to formulate hypotheses in terms of only one independent variable—complex or vaguely specified hypotheses about many variables at once can be extremely difficult to test!