

Computational Modeling & Simulation I (Interdisciplinary Science)  
CS 346

Spring 2020

Homework 2—**Revised, after Break**

(Please see the notes below!)

As there have been many changes in the semester, and many concerns have been voiced about team assignments for the remainder of the semester, **there have been several changes to the context and instructions for this assignment**. Please read the revised instructions:

- The exercises here are exactly those that were on the HW2-Lookahead posted before Break. They are now the entire HW2; no other exercises will be added to HW2. The proposed exercise on mutualism, to follow up the guest lecture of Prof. Chris Moore, will be made available on the course website as an *optional, ungraded* exercise; it is not part of HW2.
- A major change: **HW2 is now optional**. To be clear, you **do not need to do HW2 for your grade for CS346** this semester. Please read on for details.
  - If you do not submit HW2, your HWs grade will be based on the other two projects assigned for CS346—i.e., your final grade for HWs will be the average of your other two grades.
  - If you want to submit HW2 to be graded, it must be submitted by **11:59pm Maine time on Friday, May 8**, i.e., the end of the last day of classes. Late submissions will not be graded (but there will be no penalty—see below!). You may submit your work before the deadline—if you do, you must email me when it's submitted, or else I will assume the files are not a final submission—but you may only submit it once (not, e.g., submit it then revise it and resubmit it).
  - If you do submit HW2 to be graded, it will be counted for your grade **only if it raises your HW grade for the semester**. If it doesn't raise your HW grade, your HW grade will be calculated as if HW2 were not submitted.
  - If you do choose to work on this assignment, you may either do so **in your originally assigned team of two**, or **as an individual**. Work submitted by teams (i.e., groups of two or more) of other than the originally assigned team will not be graded.
  - If you want to submit all or part of HW2 just to get feedback on it, without the possibility of it counting toward your grade, I am happy to give you that feedback! Please talk with me, however, before you submit your work.
- If you submit work for HW2, please use the string HW2 as part of the folder name (along with names / userids / etc., as required) for submitting this assignment. See documents on the course website for general HW guidelines and submission instructions. (Not following guidelines may result in small deductions on this HW—please see your Prof. with any questions!)

# Project: Predator / Prey and Other Interactions!

- 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; as usual, looking at a classmate's code or sharing specifics about code on an assignment is a violation of course policy. Please note that some of this assignment is specifically intended to be completed without any collaboration (although you may always ask your Prof. any questions you might have!), even about general ideas; please see the exercises below for details.
- 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!** These exercises are based on Project 3, Chapter 4.2 in your textbook—here, you'll be modeling predator / prey interactions with 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  $Y$ .

- (a) Develop a predator-prey model for this scenario, under the assumption 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. Moreover, assume that the timeframe to be modeled will be on the order of months (rather than, e.g., decades), so we can assume the population of  $H$  stays constant over the time being simulated.

In your write-up (code is not required for this part of the exercise), give the differential equations for this model. As always, be sure to document your process, which includes listing the assumptions and sources used in developing the model!

- (b) Analytically solve (using mathematics, not computational tools such as Matlab!) for the equilibria solutions for the equations you developed. As always, explain your answers!
- (c) 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 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) Run more simulations as in Exercise 1c, 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 Exercise 1b are helpful to inform your answer, please include that in your explanation.

In submitting your code, you may have the same `.m` file contain your coding for both Exercise 1c and 1d.

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 the revised *HW Notes & Guidelines* document about guidelines when including a representative subset of your tests in a write-up.)

- (e) In Project 1 of Chapter 4.2 in your textbook, a framework for *periodic* variation of a parameter in a model (there, it was prey birth fraction) is presented. Adapt that idea to implement periodic *seasonal fishing* in your model of  $H$ ,  $P$ , and  $Y$ . In particular, the period of the fishing cycle should be 12 months: for the 6-month *off-season* span in the cycle, the fishing rate is 0; for the 6-month *fishing season* span in the cycle, the fishing rate starts at 0, rises to its maximal value 3 months into the fishing season, and falls again to 0 at the end of the 6-month season. Use the cosine function form given in Project 1, Chapter 4.2, with parameters  $f$ ,  $a$ , and  $p$ , to implement seasonal fishing. (As always, explain your work in developing your model!)

Run simulations of your model with several different values of model parameters. Describe and discuss your results. Are there any values of the relevant constants / parameters that lead to particularly interesting results?

(Please be sure to submit code and write-up documentation—both of which should include visualizations—for this part of the exercise!)

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 Exercise 1a, 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 Exercise 1a.

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) **(Collaboration is not permitted on this exercise—other than with your teammate, of course!)** Do *hypothesis testing* with your 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 newly-updated *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!