

Contents

Meta-Rubric	1
Standard Rubrics	2
Formulation Rubric	2
Solution Rubric	3
Analysis and Interpretation Rubric	4
Figure Rubric	4

These rubrics are intended as a description for students and a guide to the TA for how to assign partial credit.

Meta-Rubric

- Individual problems may vary from these standard rubrics based on the assessed learning outcomes for the problem, but they should provide an overview of what features students ought to be included in a given solution.
- Each bullet point should appear as a separate rubric item in Gradescope. We will use *positive* grading (points awarded for each component).
- Each standard rubric describes partial credit for a problem with 10 points. Partial credit for (sub)problems worth less than 10 points should be scaled appropriately. Full problems usually combine a formulation, its solution, and some interpretation, so the rubric will be a combination of the standard rubrics below.
- Generally, rubrics for 10 point “entire problems” can be summarized with the following:
 - ▶ +4 points for an answer with a correct implementation (including model setup).
 - ▶ +2 points for the correct solution (including details such as labels, units, etc.).
 - ▶ +4 points for the interpretation. These points may be broken up across subproblems, but the general assignment of points should follow this summary.
- Note that the points may not scale with the distribution of work for a given problem. That’s life! It may take more work to derive and implement your model than it does to interpret your results, but in this class we care a lot about your ability to critically evaluate and interpret modeling results.
- Sometimes problems will be broken up into subproblems. Each subproblem is graded with the parts of these rubrics that apply to it, scaled to its points.
- A few questions ask for your own judgment: a recommendation, a tradeoff, or what information would change your answer. These have no single right answer. They are graded on your reasoning and how it uses your results, not on which option you choose.
- The lists under *What this looks like* are not separate rubric items. They describe what earns each item for the kinds of problems we work on in this course.
- **The graders cannot read your mind and will not try to.** Submissions that are unclear for any reason, including but not limited to unclear syntax (English or code), uncommented code, lack of reasoning or derivation, too much detail or writing, will not be given credit. The TA has complete discretion here. If something in your solution is ambiguous, the TA has been instructed to interpret it the “wrong” way. Clear responses are a sign of understanding, which is part of what we’re assessing.
- You will not be doubly penalized for getting the “right” solution to the “wrong” setup (you would have been penalized above), but this requires the TA to be able to *easily* identify that your implementation is correct given the wrong model. If your implementation is unclear, you are likely to lose points here because we have no way of knowing that your answer is “right” without re-coding your problem.
- You will not be given credit for your code (this isn’t a programming class); the code is a means to solving the problem, and we’re more interested in how you set up the problem and interpret the solution. This does mean that code that is sloppy but *works* is perfectly acceptable. However, your code may get you

partial credit if the TA can **easily** find where you made a mistake, so make sure your submitted code is well commented. It's important to write down the mathematics of what your code is trying to implement, because the TA will only take a cursory look at your code.

- Some rubrics include “**deadly sins**” which will cause an immediate zero to be given for the problem. Do not do these!
- Regardless of the problem type, the following penalties will be applied **per incident**:
 - -1 for missing units.

Standard Rubrics

Formulation Rubric

10 points =

- +3 for clear specification of the variables;
 - This includes units and meaning for each variable used. For decision variables, say whether each is continuous, integer, or binary, and for a binary, what 1 and 0 mean.
- +3 for the correct model equations or objective;
 - You must write down the derivation, starting with the principle or the quantity being optimized (a mass balance, a cost built up from its parts) and explaining each term. Don't treat any part of this as though it's obvious; part of technical communication is communicating to people who aren't as clever as you.
- +3 for the correct constraints or conditions;
 - Make sure you include derivations for all constraints, starting with the meaning and explanation in intuitive terms for the constraint. Don't treat any constraints as though they're obvious.
 - -2/-3 for an incomplete set of constraints, even if the constraints that are presented are correct.
- +1 for stating the assumptions the formulation relies on.

What this looks like

- **Simulation models**: the state variables; the rates of change, from a mass balance; the initial conditions; and where inputs enter, such as an effluent mixing into a river.
- **Systems diagrams**: every stock and flow, named and with units; arrows in the direction of influence; and each feedback loop traced around, with its polarity (amplifying or dampening).
- **Optimization models**: the decision variables, an objective built up from its costs or benefits, and a constraint for each requirement or limit, including upper bounds. Common structures:
 - linking a continuous decision to a binary one ($x \leq Cy$), and fixed costs (Fy);
 - logical rules between binaries (at most one, if-then), checked against every combination of the binaries involved;
 - flow conservation at every node of a network;
 - balances that link one time period to the next, such as storage;
 - standard form, when asked: a maximization with every constraint written as $\leq$, each one a row of **A** and an entry of **b**.
- **Decisions under uncertainty**: the first-stage decision, the scenarios and their probabilities, and the recourse decisions in each (a scenario tree, where asked). For a sequential decision: the stages, the state, the decision, and the value function.

Worked Example: Optimization

An optimization problem that asks for the formulation and its solution is graded as

- +3 for clear specification of decision variables;
- +2 for the correct objective;
- +3 for the correct constraints;

- +2 for the correct solution.

Solution Rubric

10 points =

- +4 for a method that fits the problem, shown so that it can be followed;
 - Show the steps, or give an explicit sketch of what your code does. If the sketch is right, a difference in the result is likely to be a minor bug rather than a misunderstanding.
- +4 for the correct result, with labels and units;
 - -1 if the error is from a minor bug.
- +2 for checking the result.
 - For example: refining a step size, comparing a solver's answer with a hand calculation or a simple case, or giving the Monte Carlo standard error.

What this looks like

- **Equilibria and stability:** set the rate of change to zero and show the algebra, reporting every equilibrium (including trivial ones, such as zero populations). Then apply the criterion that fits the model at each equilibrium: the sign of $f'(\hat{x})$ for a continuous-time model, $|f'(\hat{x})| < 1$ for a discrete map, or the eigenvalues of the Jacobian $\mathbf{J}$ for a system.
- **Simulation:** a discretization with a justified step size, and a check that the answer does not change much when the step is refined.
- **Graphical solution of a linear program:** every constraint's boundary drawn and labelled, with the feasible region shaded; every corner, with its coordinates; and the optimum, found from level lines of the objective or by evaluating the objective at each corner, with its value.
- **Solving with a solver:** a model that matches your written formulation (with constraints named when you need their shadow prices), the solution and the objective value with units, and a check against a hand answer or a simple case.
- **Integer programs:** the LP relaxation, solved and used as a bound on the integer optimum. For branch and bound: each node's added bound, its LP solution and value, and the rule that prunes it.
- **Sequential decisions:** at each stage and state, every feasible decision compared, with its value; then the policy and its value.

Worked Example: Simulation

- +4 for the model derivation.
 - This includes any relevant reasoning from mass-balance or other principles.
- +4 for discretizing the model correctly.
 - -1 if no justification is provided for the chosen step-size(s).
- +2 points for obtaining the correct solution.
 - -1 if the error is from a minor bug.

Worked Example: Monte Carlo

- +4 for a clear English explanation of the sampling plan.
 - You *must* justify each distribution used, including your choice of mean and standard deviation.
- +2 for justification of Monte Carlo sample size.
 - **Deadly Sin:** No points will be given for a Monte Carlo solution with an arbitrary sample size.
- +2 for an estimate of the Monte Carlo standard error.
- +2 for correct estimate.
 - Deviations from the posted solution are ok if they're within the Monte Carlo standard error.
 - -1 if no seed is set for reproducibility.

Analysis and Interpretation Rubric

10 points =

- +4 for specific reference to the modeling results;
 - -2 if the results are not specifically referenced and the reader has to refer to the previous problems to understand the interpretation.
- +6 for thoughtfulness of the interpretation;
 - -4 for not specifically referencing relevant model assumptions;
 - **Deadly Sin:** Do not neglect fundamental engineering principles in any of your interpretations. You will be given a zero if you make a recommendation which e.g. violates an engineering code or standard.
 - **Deadly Sin:** Completely generic interpretations will result in zero points.

What this looks like

- **Shadow prices and sensitivity:** the binding constraints and the slack in the others, with units; each shadow price with its units (objective units per unit of the constraint) and the sign convention you use; what it is worth in the decision being asked about, adding up, over every constraint the change touches, its shadow price times how far the change moves it; and a re-solve to check the prediction, with why the two answers differ when a change is too large for the shadow price to hold.
- **Model evaluation:** a measure of fit, with units; the residuals plotted against the variable that matters (distance, time), with any pattern named (a bias, a trend, a structure); whether the pattern points to a wrong parameter value or a missing process, and whether calibration could fix it; and whether the model is fit for the use it is intended for.
- **Value of information:** the expected value of perfect information and the value of the stochastic solution, with units, and what each one is worth to the decision-maker.
- **Tradeoffs between objectives:** which options are dominated, and by which option; the tradeoffs among the rest, quantified; and a recommendation that says which priorities it reflects.

Figure Rubric

10 points =

- +2 for appropriate choice of axes.
 - **Deadly Sin!** No points will be given for the figure if the axes are not labelled.
- +4 for the correct data series.
- +2 for a descriptive legend.
- +2 for a *succinct* description or caption.
 - -1 for a description which is too wordy but contains the relevant information.