MATH 380: Course Project Instructions

The project is an important part of this course - not just in terms of the grade, but for the sake of comprehensive, practical understanding of how to apply modeling framework to an open ended real-life problem. This is why the problem statements that I have given to you are short and open-ended descriptions of the certain real-life situations. You have complete freedom in mathematical interpretation of the problem and how you "solve" it. The only requirement is that you use the mathematical modeling process, and justify your model and its conclusions as they apply to the problem. Its a test of your creativity in formulation of models and solution methods, and your ability to find, understand, and use relevant mathematical and computational knowledge.

Here are some important suggestions to keep in mind as you choose and work on a project problem.

1. **Break the problem into three main parts**: What are you modeling and how can you formulate it in mathematical terms? What ideas for algorithms/ solvers do you have? How will you compare algorithms/ models/ solutions? Think about how you will use specific mathematical tools (our textbook has a pretty long list such mathematical tools - just look at the table of contents) to describe and solve each of these aspects. Discuss with your team members. Note that there could be more than one way for approaching each of these parts.

2. Based on your discussion in step 1, you should have some **preexisting mathematical techniques/ methods/ models** that you want to focus on initially. First look up your textbooks (from this course and from your other math courses), and then go to the library (in person(!) or online) to look up textbooks and papers that discuss the mathematics of your approach, and to learn as much about your (interpretation of the) problem as possible. What approaches have others considered for similar problems?

Note that you are allowed to use older research as your motivation (with proper attribution and reference, of course). But you can not simply plagiarize (exactly or after some minor modification) other's work/ ideas. Such plagiarism is fairly easy to catch through online search and will result in zero credit for your project in addition to any further disciplinary action by the University.

3. Re-discuss Step 1 after your literature survey for relevant mathematical methods/ models. What are reasonable assumptions and what are not? What is a reasonable Mathematical model for problems of this kind and what is not? **Finalize your mathematical model** and keep track of all the simplifying assumptions you are making for it. Don't worry if this is a simple model as you can modify this model as needed later on.

4. **How will you "solve" this model?** Can you use a standard algorithm or do you have to write your own algorithm combining other algorithms? You might have to write your own computer code that incorporates appropriate solvers from Mathematica/ Matlab. In any case, you have to use a computer program to solve your model.

5. Do you need **input data** to solve this model? Where can you get this data? Search online for data depositories. If nothing is available, then discuss how you can create your own "synthetic data" (data that does not come from real-life but is created, possibly with help of a computer program, to mimic real-life data). You have three possibilities here: (1) Use preexisting data that fits your needs (with proper reference as always); (2) Modify preexisting data to fit your needs (explain how and why); (3) Write a mini-model incorporating the properties of the data you need,

and then use this model/rules generate your own data on a computer.

6. **Get solutions** from your model and data. Note the plural - you must aim for multiple solutions so that you can compare them and pick the best. How well does your solution answer the project question? Now, you should go through the whole modeling process again, and possibly "unsimplify" your model to get better results. Analyze the strengths and weaknesses of your modeling approach, and **modify your model** to eliminate as many of these weaknesses as possible.

7. Quantitatively and qualitatively **compare** the various solutions and models. Computationally examine the **sensitivity** of your solutions and models to the input parameters and data. Finally, come up with your **conclusions** about the problem, your work , and what you have achieved. Think about what are the best ways to display and discuss your solution data and conclusions.

8. Put together the **project report** and all that you need for it - description of the models, data, algorithms/ programs, tables and graphs of the output solutions, references to textbooks and papers that you have used, etc. See below for the requirements for the project report.

Requirements for the Project Report and Submission:

Your typed project report should be 20 to 25 pages long (in 11pt font), excluding the appendix. It should include the following sections. Each of these sections will be evaluated individually in addition to overall evaluation of the project as well as the new and innovative ideas introduced by your team.

To see a simple example of such a report, look through the appendix B of https://m3challenge.siam.org/sites/default/files/uploads/siam-guidebook-final-press.pdf**.**

In addition to your report in PDF format, you will also submit all your code, programs, data, etc. in a zipped file through email.

1. Introduction: This should give a 1-2 page summary of all that follows in the report, including a short description of the problem in your own words, and your final conclusions. You can have the final conclusions as an introduction in the beginning of the report or as summary at the end of the report.

2. Statement and Analysis of the Problem: Quote the problem statement and then restate it in your own words, describing how and what you interpret the problem to be and how you are going to approach solving it. Also describe the results of your literature survey about how others have modeled and solved same or similar problems (with references of course) and how your interpretation is similar to and different from them.

3. Description of the Model: List and discuss the assumptions of your model. Discuss how you designed it and your justification for the model. List all the variables/ parameters used in the model. Note that you could have more than one model here if your approach is to compare results from more than one model.

4. Analysis and Testing of the Model: List/describe/present your input data and output results; and the conclusions from your model. Discuss how you tested the model, including, if relevant, the error analysis and stability of your solutions (if you change some of the data/ parameters, does that change your solution drastically?). You should give an step-by-step outline and detailed description of any algorithms/ computer programs that you are using, explaining their purpose and intent. Standard mathematical algorithms can be referenced through textbooks but you should still give a short description of what they do.

5. Results and Quality of the Model: Discuss your results and analysis, with graphs, figures, etc. to make your final conclusions. And also discuss, how you would improve the models you studied - in particular the strengths and weaknesses of your model(s).

6. References: You have to explicitly list all sources of your Mathematical tools, models, algorithms, data, etc. This should be as extensive as needed. Each of these references must be cited within the main report wherever appropriate.

7. Appendix: Include the computer programs that you wrote, numerical calculations, proofs, anything ancillary that you did not include in the main report.

Resources: In addition to your textbooks from this course and all other Math/CS courses that you have taken, here are some sources for important algorithms for fundamental Mathematical models/ tools:

1. Numerical Recipes: The Art of Scientific Computing, by William H. Press and others.

2. Numerical Methods that Work, by Forman Acton.

3. Introduction to Algorithms, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.

4. Operations Research: Applications and Algorithms, by Wayne Winston.

You may not need them for actual programming since you can utilize appropriate solvers from Matlab/ Mathematica, but they can serve as sources of ideas and inspiration for your computational models and related computations.