

## your Math 615 project

**Overview.** One my goals in Math 615 is to give students practical experience in numerical analysis by working on a project which is more substantial than the homework problems. Both mathematical analysis and numerical computation are required on this project. Ideally the topic is chosen by the student—perhaps with consultation from a thesis advisor—but see subject suggestions and content requirements below.

**Project categories.** Your problem should involve one of these continuum mathematical models:

1. a partial differential equation (PDE), or
2. a delay differential equation (DDE), or
3. a stochastic (partial) differential equation (SDE or SPDE),

or it might not involve a mathematical model but instead be a

4. rigorous numerical analysis of selected ordinary differential equation (ODE) schemes.

*These are the only allowed possibilities.* Note that, especially for categories 1 and 2, the problem might involve more than one differential equation, i.e. it might be a small system.

Note that most ODE-only mathematical models *are* highly suited to black-box solutions, and thus are not allowed. Also, I am perfectly aware that DDEs and SDEs were not addressed in class, but my experience is that projects based on an “introductory case” of these slightly-out-of-the-course topics do work reasonably well.

For mathematical models (categories 1,2,3), your particular problem could be

- a. an initial value problem, or
- b. a boundary value problem, or
- c. an initial and boundary value problem [*most common case for PDE models*], or
- d. an “eigen” analysis (i.e. finding the modes of a system), or
- e. an inverse problem.

Please identify and state, in the Introduction of your project (see below), what category it falls in, e.g. “my project is 1c, a initial and boundary value problem for a PDE”.

**Subject suggestions.** These suggestions appear alphabetically, thus in no particular order. They are all basically in categories 1a, 1b, or 1c in the above taxonomy. There is varying intrinsic difficulty to the standard instances of these problems, but I know enough about each of these to be able to advise on how much you should try to do, and how to avoid becoming bogged down. I believe I also know enough to be able to “weight for difficulty” at the time of grading. Please do a web search for topics you find interesting—at least look for a wikipedia page:

- biharmonic equation
- Black-Scholes equations for security pricing
- earth deformation under load
- eigenmodes of drum heads
- eigenmodes of electromagnetic wave guides
- ice sheet equation
- Korteweg–de Vries equation
- linear (Newtonian) Stokes problem for incompressible fluid
- minimal surface equation
- morphogenesis/pattern formation

- Navier-Stokes system for incompressible fluids
- Poisson/Laplace equation on a fractal
- Schrödinger equation
- sea ice equation
- shallow water equations
- telegraph equation

**Content requirements.** Once you choose a subject (i.e. the continuum problem) you will choose a numerical scheme or schemes. There should be some explanation of your choice of scheme, and “ease of implementation” is indeed a valid reason to choose a (consistent) scheme. Other reasons might be stability or order of accuracy. It might also be the case that you are interested in learning about that kind of scheme, e.g. “I want to learn about implicit schemes for nonlinear equations” or “I want to learn about high-resolution advection schemes.”

For your chosen scheme or schemes applied to your chosen continuum problem, two actions are required:

- A. numerical analysis
- B. practical computation

On requirement A: You must make some attempt to understand and explain the efficiency, stability, convergence, and accuracy of your numerical scheme(s). In particular, for each of the following actions you should ask whether it is possible and meaningful to do so, and if so you should probably do it. Concretely, I expect you to do at least two of these analyses:

- i. compute truncation error and show consistency
- ii. do stability analysis
- iii. prove convergence [rarely possible]
- iv. assess the computational cost of the algorithm
- v. manufacture (or otherwise find) an exact solution, and verify with it.

If you are unable to accomplish two or three of the above actions then your problem is too hard.

I do not expect you to derive the equations of the continuum model, though you may do so (e.g. briefly in an introduction or appendix). Generally, derivation of the continuum equations is a subject for other courses. However, please briefly explain the physical/engineering/mathematical context as you introduce the problem. Carefully state the equations of the model, giving precise references including a source of the derivation. Explain the meaning of the symbols you use. Clearly state the role/meaning of your particular computation (e.g. what is being predicted from what data).

Regarding requirement B, you must do some practical computation in MATLAB or Python or etc., though I do not expect you to produce a production-quality code. Rather, your goal should be a functional and readable prototype. Please use MATLAB’s (or etc.) built-in numerical linear algebra, and don’t write your own. In some cases it will be appropriate to use MATLAB’s built-in ODE IVP solvers (e.g. `ode45`), or a solver for a scalar nonlinear equation (`fzero`). You do not need to use any MATLAB tricks, but vectorizing to avoid loops may help you understand what you are doing.

You are encouraged to try methods other than finite differences for your practical computation, for instance finite volume, finite element, or spectral methods. But, because these methods differ both in implementation and analysis from what is addressed in class, working with them will be intrinsically harder because the required numerical analysis will be harder. See me if you want your project to go in such a direction . . . and I can “weight for difficulty” when grading.

**Length and format of project Version 1.0 (due Monday 4/10 at start of class).** Do a little bit of everything that will go into the final product. Version 1.0 should be a useful skeleton on which to build Version 2.0 (below).

V1.0 needs to be readable but it need not be polished. The length should be between 5 and 10 pages total. This part is worth 5% of the course grade. Please include the following 7 section headings, with a start on this content:

1. **Introduction** (including scientific/engineering context and sketch/gloss of the numerical scheme, computation, and analysis you do)
2. **Continuum Model [or Continuum Problem]** (here should go the PDE itself, etc, and clear specification of the particular problem including boundary conditions and parameters; complete your statements of the meaning of symbols; include a not-completely-trivial exact solution, if you can, here or in an appendix)
3. **Numerical Scheme(s)** (state the method, discuss the reasons for this choice, discuss implementability and efficiency; small excerpts of the code may go here)
4. **Analysis** (truncation error, stability, convergence, cost of algorithm, verification results)
5. **Results** (show figures and a few key numbers; note that a section called "Conclusion" is not necessary)
6. **References** (at least three; quality matters)
7. **Appendix(ices)** (put the full codes here; also tangential analysis or computation; failed attempts if significant)

The Results part should be quite short, and may be empty, in V1.0.

I recommend having an electronic form of V1.0, of course, which can be modified into v2.0. However, turn in V1.0 and V2.0 on ordinary  $8.5 \times 11$  letter paper. Staple in the upper left corner. Please do not put it in a folder, etc.; I will have a stack of these things to grade and I don't need yours to take up extra space. Please make it double-sided if possible.

Some of the work is mine. I will give you feedback on V1.0 which is intended to improve your grade on V2.0. I am happy to talk to you about your project at any stage.

**Length and format of project Version 2.0 (due Friday 4/28 at 5pm).** Fill out the V1.0 skeleton, of course. Think through your approach given the feedback you get on V1.0. You will undoubtedly need to alter your earlier plans, so don't worry if V2.0 has different computational or analytical details than V1.0. V2.0 is final, so please make it reasonably clean and neat. The length should be between 15 and 25 pages total; *25 pages is a firm maximum*. V2.0 must include the same section headings as V1.0 (see above). This part is worth 10% of the course grade.