Final Project Guidelines

- Your mission is to complete a project that involves using computational methods to solve some physics (or related field) problem. The topic and computational methods you explore are completely up to you. I provide some possibilities and keywords to search at the end of this document, but you can completely ignore this and choose your own if you wish. Choose a project that is of interest to you, allows you to be creative, challenge yourself, and enjoy learning something new.

- The topic of your term project should NOT be your undergrad thesis, a topic for some other term project you have to write, or work you have done for some other course. Essentially, it should be something new you've decided to explore for this course.

- You can style your project in one of two ways:

1) You can write a 'Lab' similar to the ones we have done in this course. E.g. provide a 'Computational Background', then a 'Physics Background', then some instructions on how to solve some problems on some related theme or topic. Also provide the solutions and codes.
2) You can write a 'Term Paper'. This is more like a research paper where you provide an 'Introduction' section where you review the literature on your topic and computational methods, a 'Methods' section where you detail your methods, a 'Results' section where you provide results and a 'Discussion/Conclusion' section where you discuss your results.

- The style choice is totally up to you. Whichever you choose, make sure to provide citations to references that you use. Also provide all codes you develop.

- Your term project should somehow go beyond the scope of this course. It can do so in one of the following ways:

1) use a computational method that was not covered by the labs (e.g. some new way to numerically solve an integral, or some new finite-difference scheme for solving ODEs).
2) use a computational method for a higher-dimensional problem, as long as it involves significant changes to the algorithm (e.g. doing a 2D shallow water simulation instead of a 1D shallow water simulation is NOT trivial but solving for the volume of a hypersphere in 11 dimensions instead of 10 is trivial).
3) combine several computational techniques to solve a more complicated problem (as long as its not too similar to what we did in class).

- There is no specific length guideline because projects will be very different. Some might involve complex computing methods but only a few lines of code to implement (these would probably result in a project with a much longer introduction or computational background explaining these complex methods, but the code would be fairly short). Some might involve fairly simple methods to understand, but require a lot of programming (in this case, the intro/computational background may be shorter but the codes will be longer). **Based on the weight of the term project and the amount of time you have to do it, I am expecting an amount of work that is the equivalent of 2 regular course labs (in terms of time spent on the project)**.

- DUE DATE: Officially: Tuesday Dec. 6, 2016. However, there is a NO-PENALTY extension until **Friday Dec. 20 at 5:00pm**. This extension deadline is strict. Make sure to hand in your project long before this deadline so that there is no issue with being just over the due date/time. Work handed in after Friday Dec. 20 at 5:00pm will be assigned a grade of 0.

- How to hand in: similar to the labs, you will hand in your project via blackboard. It should include 1 pdf document for your write up and ALL python files needed to run your code.

- However you style your project, it must include the following:

  1. Explanation of any new computing methods (along with citations to where you learned about them). You can assume a knowledge base of someone who took this course (e.g. you don't have to explain the idea of numerically solving integrals to explain a new technique for numerically solving an integral. But if you decide to cover finite element methods, you should explain this from its basics since we didn't cover any in class).
  2. Some sort of efficiency/accuracy/stability/error estimates related to your project. Which to focus on will depend on your topic and methods choice, but I expect some discussion of these points (at least a paragraph).
  3. A section where you discuss how you can test (and hopefully did test) your codes to ensure there are no bugs and that it produces correct results. This can involve simple test cases, limiting cases, benchmark studies etc.
  4. Your codes will be graded based on computational efficiency, organization and readability by me. This means you should include lots of good comments, make sure your code is organized well (with modules, nice spacing) and that it minimizes the number of floating point operations (based on your accuracy goals).

1) Here is the grading scheme for the project:
   a) <u>Content</u>: 10 marks:
      i) appropriate level/amount of computational methods involved
      ii) effective (i.e. thorough, succinct and understandable) descriptions of these methods and anything else involved (e.g. your "Physics background, or your results, discussion etc.).
      iii) appropriateness of the methods for solving the physical problem you are considering.
      iv) Efficiency/accuracy/stability/error discussion.
   b) <u>Codes: 10 marks</u>:
      i) understandable code (i.e. comments, organization)
      ii) efficient code
      iii) modularization (for testing purposes)
      iv) appropriate level/amount of coding for the project.
   c) <u>Write up: 5 marks</u>:
      i) Grammar, style, lack of typos, good organization to the document.

- <u>Help, Feedback and Collaboration</u>: Since this is a 'final project', we will implement the philosophy of an open book final exam. You are required to do your OWN work (no partners) although you can use any texts/articles/online refs you like. You CANNOT ask for help on finding bugs in your codes from anyone. You CAN ask Paul and Oliver for advice on whether a particular topic is appropriate and whether you are approaching it in a good way (i.e. with appropriate methods). But make sure to do this EARLY (i.e. before the end of labs). If you want feedback on your choice of topic or computational approach, you must ask by Nov. 25. By then you are expected to have a detailed plan in place for your project.

Basically, the point of the project is for you to demonstrate, individually, what you have learned in this course and how you would apply it to some new scenario not covered in the course.

Potential Project Topics:

Here are some ideas for different topics and/or computational methods you can use in your project. Some of these are just keywords (i.e. they wouldn't be an entire project but could be a specific method to implement as part of a project). Some are just physics topics that typically involve computational methods. Do some online searching of these methods to learn what they are about. I've tried to organize them based on their relevance to the chapters of the text. Feel free to completely ignore these and come up with your own (maybe by looking through other computational text like Numerical Recipes). Again, if you want feedback/advice/guidance you must see Paul or Oliver BEFORE Nov. 25.

Some decent texts where you can look for methods and problems are:
1) Press et al., Numerical Recipes (we used this for the Lax-Wendroff method)
2) Heath, Scientific Computing
3) Giordano & Nakanishi, Computational Physics

Ideas for Topics & Methods:

- Functional Analysis (Integrals, Derivatives, Interpolation, Extrapolation…):
    - Gauss-Kronrod quadrature
    - higher order finite difference approximations
    - using Chebyshev approximations for derivatives & integrals
    - cubic spline interpolation
    - computing special functions: gamma function, error function, Bessel functions, Legendre polynomials, spherical harmonics
- Solving linear systems:
    - Singular Value Decomposition
    - Cholesky Decomposition
    - Conjugate gradient method
    - Sparse systems
    - Lanczos algorithm
    - data optimization
- FFTS:
    - dealing with aliasing
    - correlation and autocorrelation
    - filtering and data windowing
    - wavelet transforms
    - image compression
- Solving ODEs:
    - methods for stiff equations
    - predictor-corrector methods
    - integral equations
    - bifurcations in nonlinear systems
    - resonances in planetary orbits
    - chaotic tumbling of satellites
    - other chaotic systems
    - molecular dynamics simulations: dilute gas, planetary motions, planetary formation, melting transitions

- efficient numerical methods for integrating N-body problems: fast multipole method or particle-mesh methods
  - Fermi-Pasta-Ulam system
  - Galerkin method
  - Collocation method
- Solving PDEs:
  - solving for electric fields and potentials for different charge distributions
  - magnetic fields produced by current distributions
  - wave motions (on a string, in the ocean, atmosphere, …)
  - membrane vibrations
  - 2D or 3D parabolic, elliptic, or hyperbolic equations
  - Schrodinger's equation in 2 or 3D, different potentials, barriers, quantum tunneling
  - multigrid methods
  - finite element methods
  - spectral element methods
  - fluid dynamics with shocks
  - nonlinear PDEs
  - solitons
  - 2D shallow water equations
  - barotropic vorticity equations (simple atmospheric dynamics)
  - time dependent/time independent quantum mechanical calculations
- Random Processes & Monte Carlo:
  - stratified sampling
  - adaptive Monte Carlo for particle physics (VEGAS)
  - relationship between diffusion and entropy
  - fractal dimensionality of curves
  - percolation
  - 1st order phase transitions
  - 2nd order phase transitions
  - protein folding
  - neural networks
  - neurons and action potentials
  - cellular automata
  - Master's equation
  - variational monte carlo method for time independent Schrodinger equation
  - earthquake simulations: block sticking & slipping
  - kinetic Monte Carlo (Gillespie algorithm)
  - Langevin dynamics
  - Density functional theory (quantum Monte Carlo)