Course Administration -- ATM 150

Page last modified: 1October 2018

- Instructor:
  - Richard Grotjahn,
  - room 231 Hoagland Hall,
  - 752-2246,
  - grotjahn'at'ucdavis.edu
  - Office Hours: TBA
- Class Meetings: SCHEDULE
  - Lecture: T, Th 2:10 p - 3:30 p in 124 Hoagland
  - Discussion: T 3:40 p - 5:00 p, Th 3:40 p - 4:00 p in 124 Hoagland
  - Note: at present, no class scheduled Tuesday: 9 October
- Course Goals
  - Provide a basic introduction to commands in linux OS environment.
  - Provide a basic working knowledge of Fortran 95 programming elements commonly-used in physical sciences.
  - Provide a basic introduction to NCO commands
  - Provide a basic working knowledge of elements of NCL scripting language commonly-used in physical sciences.
  - Provide a basic introduction to tools and applications involving numerical analysis techniques. The techniques have broad applicability to the physical sciences and examples are drawn from techniques used in fluid dynamics, oceanography and meteorology.
  - Emphasis is *not* on mathematical theory, but upon learning the techniques.
  - Each student will write several computer programs that illustrate applications of numerical and graphical tools that are used in many fields of study.
  - Note: Some material on objective analysis, initialization, parameterization, operational forecast model design, or predictability is incorporated in the courses ATM 111 and ATM 255 and **not** in ATM 150.
- Text:
  - There is no text.
  - The book: *Fortran 90/95 for Scientists and Engineers* by Stephen Chapman has been a useful Fortran reference for me.
    Note that this class will use several primers available online. These will cover some basic issues. However, it is likely you will want to use other sources, perhaps printed and/or online, for more complete reference reference materials.
- Entry Level:
  - Knowledge of a higher level programming language like C is desirable **but not required**. The course is intended to provide basic skills to program in Fortran.
  - While the numerical techniques discussed in the course can be solved using C or Matlab, or even spreadsheet software, those languages may be inefficient at doing so. Furthermore, a goal of this course is to learn how to program in Fortran, so

students are required to do the *Fortran* exercises in Fortran. *Nota bene:* If you use C or Matlab, I may not be able to assist you with debugging your code.
    - We will make a reasonable effort to provide you with access to Atmospheric Science Group (ASGG) computers.
    - Familiarity with the basic equations of atmospheric dynamics and thermodynamics as studied in Atmospheric Science courses 120, 121A and 121B is helpful to provide a context. **But this background is not required.**
- Course Format:
    - This 4 credit course has 3 hours of lecture and 2 hours of discussion per week.
    - In practice the lecture will sometimes include in-class coding.
    - The weekly homework assignments will often include practice at the computer as well as simple derivations.
- Evaluation:
    - Following the wishes of most students who have taken the course in the past, the grading is based on 4/5 from the homework, 1/5 from the final exam.
    - To facilitate discussion during lab, a 10% penalty is applied per week day that any assignment is late.
    - The course is graded P/NP. So what is sufficient to pass? >= 65%.
    - The homework is graded partly on the accuracy. For computer programs, the grading also includes the program logic. Even if you got the right answer despite an error, points may be deducted for that error.
      Homework is usually turned in using a combination of hard copy (print out) and email attachments. A typical situation is:
        - Turn in:
            1. Print outs of
                - all plots,
                - character output, &
                - your program file
            2. If the assignment involves writing a program, you are requested to place a copy of all files created by your program in the 'replies' directory under the subdirectory of your kerberos name. Be sure to include a copy of the Fortran program file you write in that subdirectory.
               If you are using some other programming language (C++, MATLAB) you are requested to attach a copy of the file to an email all **listing** files.
        - Fortran compiler options can checks bounds and cross references.

          ncargf90 -fcheck=bounds template.exe template.f
          where template.for is the file being compiled. Bounds checks array bounds at time of execution, e.g. **./template.exe**

    - There will not be a midterm.
    - The final is scheduled for Thursday, 6 December 2018 (3:00 pm - 4 pm).
    - Required: **ACADEMIC CODE OF CONDUCT** – Please see the website http://sja.ucdavis.edu/files/cac.pdf for campus-wide rules.