Written Homework Format

Each homework assignment will typically involve writing a program to test a numerical method or to simulate some engineering physics topic. In a professional academic or industrial research programming environment the program alone is never the total finished product. You must always document your program (how to use it and how it works internally), and you must always organize the scientific component of the program and its scientific results (in English) for others to read without necessarily understanding your program in detail. Although there is not enough time to write a long report for each homework set, you still need to pay attention to the documentation side of programming and computational engineering physics. The program documentation and the program are equally important and the grading will be based roughly on equal components of each.

In a practical sense you will be allowed to use several different programming environments and a variety of computer systems. This makes it very difficult to grade the homework without adequate (English) documentation. For this reason the English and mathematical equation part of your homework write-up should be self-contained. The grader should not need to read the program to find out what you did. Note however, that the grader will also read your program to see how well it is written. Your write-up should contain a description of the background for your work, your results and your interpretation of the physical meaning of your results (where appropriate). You may discuss the problems among yourselves but what you turn in must be your work and not someone else's. To be more specific your written report should contain the following (in your own words):

1. A few pages (typically 1-3) of English **text** and mathematical **equations** describing the engineering physics, mathematical background and numerical methods used in the assigned problem and **how your program works** to solve this problem (in addition to the problem statement). If the numerical methods are given in the text then you may just reference the text and give a one or two sentence description of the method. You do not need to copy a lengthy derivation into your report if it is already in the text. You should include **proper references** to the source of the methods you use. References may be in the text as: "The attached program (hw2.cpp) uses the extended trapezoid rule (equation 4.1.11 in Numerical Recipes , 3rd edit, by Press et al) to numerically evaluate this integral." or collected at the end and referred to in abbreviated form in your discussion (as in equation 4.1.11 of Press[1] et al). Although a clear and neat handwritten report may be acceptable, you are encouraged to use a professional document preparation package such as LaTeX (available in the AEP computer room, free versions for nearly all computer platforms). LaTeX is initially harder to use than MS-word, but does a much better job with equations and is easier for long sophisticated documents. The book, by H. Kopka and P. W. Daly[2] is a good place to start, or just go to Google and search the web for 'latex tutorial' and you will find a variety of on-line tutorials (some are listed on the course web site). You also need a plain text editor to use with LaTeX (the "Crimson Editor" also highlights LaTeX/TeX syntax). The TeXmaker (MS-windows, linux and mac) and TeXShop (for Mac) packages are free GUI front ends for LaTeX that make it much easier to use.

2. The **results** of running the program. These should be presented in a convenient, easy to understandable format. This will usually be one or more graphs, or small tables of key numbers. You should learn to use a scientific graphing program, should as python, matlab,

etc. MS excel is a poor scientific graphing program, please do not use it for this course. You should not turn in lengthy lists of number printed by the computer.

3. A few paragraphs discussing (using English and/or equations) the **physical meaning** of your results where appropriate.

4. A **source listing** of the program(s) used to solve the problem. The program should be self-documenting to some extent. The source code should be **organized and readable**. Avoid overly clever programming tricks that make it hard to follow. It should also contain well-written comments at appropriate positions in the code. Too many comments may however make is hard to read. The comments at the top of the listing should include **your name, date, title, CPU, op. system, and compiler or interpreter type and version number**. You may also find that a well written and documented program is also easier to debug.

References

[1] W. H. Press, S. A, Teukolsky, W. T. Vetterling and B. P. Flannery *Numerical Recipes, The Art of Scientific Computing, 3rd edit.*, Camb. Univ. Press 2007.

[2] H. Kopka, P. W. Daly *A Guide to LaTeX, 3rd edit.*, Addison-Wesley 1999.