COMPUTATIONAL PHYSICS (PHYS 4301/5322)
Spring 2008

Instructor: Igor Volobouev, i.volobouev@ttu.edu
Office: SCI 13, ph. 806-742-4572
Office Hours: Wed 15:00 – 17:00, by appointment, and open door
Meetings: TTH 14:00 – 15:20 in SCI 29

Objective: Learn to solve complex physics and engineering problems using numerical analysis

Coverage: Construction of numerical models of physical phenomena
Basic numerical methods (differentiation, integration, root finding)
Ordinary and partial differential equations
Monte Carlo methods
Data analysis and curve fitting
Understanding behavior of complex systems using computer simulations
High performance computing

Prerequisites: PHYS 2402 (required), MATH 3350/3354 (recommended)

Programming languages: Python will be the main language for the course, C++ will be used for parallel/distributed computing projects. You will be expected to pick up basic Python usage very quickly and mostly on your own. Lots of resources on the web and in print can help you with this (www.python.org is a good start).

Grading Policy: The following weighting scheme will be used:
10% class participation
30% exercises/homework assignments
60% final project

The following serves as an approximate grade scale:

100-88: A   87-75: B   74-60: C   59-50: D   < 50: F

Exercises/Assignments: Some exercises will be assigned for class/homework to support the lecture materials. The following information will be normally expected together with the solution source code: brief description of the type of problems your code can solve, description of the user interface, several test cases, example output (usually in graphical form), conclusions from the output.

Final Project: I will expect a presentation to the class (15-20 min) on an interesting physics or engineering problem of your choice. Graduate students will be expected to pick more challenging problems. Working in small groups will be encouraged. You will need to get your project topic approved on or before March 13.

Course Textbook (required): N.J. Giordano and H. Nakanishi, *Computational Physics*, 2nd ed., Prentice Hall, 2006. This book is not available from the campus bookstore, please order it elsewhere (for example, www.abebooks.com or www.amazon.com).

Advanced Sources (recommended):

1. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed., Cambridge University Press, 2007. Includes detailed descriptions of a large number of numerical algorithms and discusses their theoretical background. This book is widely regarded as the best single reference on the subject of numerical analysis.
2. H.P. Langtangen, *Python Scripting for Computational Science*, 2nd ed., Springer, 2006. A good reference on numerical programming with python. Includes lots of examples.

Programming Texts:

1. A. Downey, *How to Think Like a (Python) Programmer*, Green Tea Press, 2007. This is a free introductory level programming textbook, you can download it from http://greenteapress.com/thinkpython/ A nice book for those who want to learn to program and have little computing experience so far.
2. Guido van Rossum, *Python Tutorial*, http://docs.python.org/tut/ Good Python description from the inventor of the language for those already familiar with basic programming concepts.
3. A. Martelli, *Python in a Nutshell*, 2nd ed., O'Reilly, 2006. Reasonably complete description of the language and its standard libraries. It can be appropriate as a first Python book if you already have significant programming experience. Available online at http://proquest.safaribooksonline.com/0596100469

ADA Statement: Any student who, because of a disability, may require special arrangements in order to meet course requirements should contact the instructor as soon as possible to make any necessary arrangements. Students should present appropriate verification from Student Disability Services during the instructor's office hours. Please note instructors are not allowed to provide classroom accommodations to a student until appropriate verification from Student Disability Services has been provided.