

# Introductory Computational Physics

(Physics 420/520)

This course is intended to give an introduction to main computational tools, techniques and methods used in contemporary physics. Student will practice writing, compiling, and running computer programs, together with analysis of results, and presentation of their results as scientific reports.

This is not a course in computing science, or in programming. It focuses specifically on practical methods for solving physics problems. The course is therefore designed such that a significant fraction of the students' time is spent actually programming specific physical problems. In this course the students should **learn by doing**.

<b>University Catalog</b>	420/520 Introductory Computational Physics. CRN 23834. Credit hours 3. Introduction of computational methods and visualization techniques for problem solving in physics.
<b>Prerequisite</b>	Students are expected to be familiar with at least one programming language (Fortran, C, C++, Basic, or Java). Physics 231/232 and Math 211/212
<b>Instructor</b>	Dr. Alexander Godunov Office: OCNPS 0219 (Oceanography and Physics) Phone: 683-5805 email: agodunov [at] odu.edu The course web page: <a href="http://www.odu.edu/~agodunov/teaching/phys420_09">http://www.odu.edu/~agodunov/teaching/phys420_09</a>
<b>Classes</b>	Monday, Wednesday 13:00 – 14:15 Oceanography & Physics, Room 303
<b>Office Hours</b>	Monday, Wednesday 12:00 – 13:00 Oceanography & Physics, Room 219 (office) and by appointment
<b>Materials (recommended)</b>	Lecture notes: <a href="http://www.odu.edu/~agodunov/teaching/notes/index.html">http://www.odu.edu/~agodunov/teaching/notes/index.html</a> <i>Introductory Computational Physics</i> by A. Klein and A. Godunov, Cambridge University Press (2006) <i>An Introduction to Computer Simulation Methods: Applications to Physical Systems</i> (3 <sup>rd</sup> edition) by H. Gould, J. Tobochnik, W. Christian, Addison Wesley (2006) <i>A Survey of Computational Physics: Introductory Computational Science</i> by R.H. Landau, J. Paez, and C.C. Bordeianu (Princeton University Press 2008)
<b>Course structure</b>	Different people learn in different ways. Therefore this course offers a learning environment with a diversified set of options that you can tailor to your individual learning style. There are class meetings, homework assignments, computational projects, and practice. There are many ways to get assistance with the material in this course: the office hours, appointments, e-mail, phone

- Keys to success** Right motivation, working diligently, effectively and efficiently is the key to success. If you work regularly and allocate enough time each day to practice and complete the assignments on time and keep up with the course, you will get the most out of the course both intellectually and grade-wise.
- You should invest about **6-9 hours per week** outside of class to succeed in this course. This is consistent with university guidelines (i.e. two to three hours of outside preparation time for every credit hour). Students with little or no programming experience (or those who love the subject and wish to do extremely well in it) may want to put in more hours. Please, let me know as soon as possible if you have difficulties and may need extra help.
- You are recommended to start your assignments well before the last night when your assignments are due. It is a general experience that a computer program usually does not work correctly (if works at all) at the beginning. A search for a problem, or a computer bug, may take more time than you expect.
- Homework** Doing the homework problems is one of the best ways to learn the art of computational physics. Credit for homework is given to encourage practicing and thinking about computational physics on a regular basis. This credit influences the final grade for this course. Assignments may be submitted early but will not be accepted late.
- Projects** There will be midterm computational projects and one final project. The projects will aim to solve specific problems in physics. You will be required to submit a report on each project. All reports should have the following sections: title, description of the problem, equations and computational model, testing, example of input parameters, results (figures, tables, analysis), and conclusion.
- The report and your computer program should be submitted electronically by the due date and time (as an attachment, including an example with input/output files). This will enable the assessors to check and run your program if necessary. Please, remember that "no submitted program = no credit for the assignment". Recommended formats for reports are MSWord and PDF, with all figures embedded in the body of the document. A carbon copy of your report should be in the instructor's mailbox as well.
- Each assignment will be graded for completeness and correctness. Clarity of your presentation may affect your grade.
- No make-up projects will be given. In case you have a legitimate reason for missing a project deadline, consult with me before, or within 24 hours after the deadline.
- Regrade: Requests for correction of grading mistakes on projects can be made when the work is returned to you. The requests must be made within three days after getting your grade. Regrade requests should be written. In their request, students must explain why they believe there is a mistake in grading and why they deserve more credit. It is not a plea for more points

<b>Exams</b>	<p>Since this is a laboratory like course there will be no traditional exams but midterm projects, and one final project. The final exam will consist of an oral presentation of the final project and an accompanying written paper (in the format of a research article).</p> <p>The final examination is mandatory and will be given only at the scheduled time.</p>
<b>Grades</b>	<p>The final grade is calculated on an absolute scale. There are 100 points possible for this course of which 25 points are for the homework, 50 points for the midterm projects and 25 points for the final project and examination (the presentation 5 points and the research article 20 points).</p> <p><b><i>A letter grade is determined only at the end of the term.</i></b></p>
<b>Attendance</b>	<p>Attendance in lectures is optional; however, you are responsible for all information discussed in class, including any announcements.</p>
<b>Collaboration</b>	<p>Collaboration in class is strongly encouraged. Because the course is graded on an absolute scale, you will never reduce your grade by helping others — on the contrary, by doing so you will reinforce your own knowledge and improve your performance.</p> <p>Activities for which collaboration is NOT permitted are: homework assignments, projects and the final examination.</p>
<b>Professional Conduct</b>	<p>In Physics 420/520 high professional standards, including ethical standards, are promoted. Plagiarism and cheating are serious offenses and may be punished by failure in the course. The academic integrity code is to be maintained at all times.</p>

## Tentative Course Outline

1. Basic
  - The nature of computational physics
  - Tools of computational physics
  - Machine representation, precision and errors in computations
2. Short introduction to computer languages
  - Computer languages for scientific computations and available compilers
  - C/C++
  - Fortran 90
3. Basic tools of numerical analysis in science and engineering
  - Nonlinear equations
  - Interpolation
  - Differentiation
  - Integration
  - Available numerical libraries for Fortran and C/C++
4. Matrix calculations and applications in physics
  - Linear systems
  - Eigenvalue problem
5. Ordinary differential equations and applications in science and engineering
  - Initial value problem
  - Boundary value problem
  - \*Eigenvalue problem
6. Nonlinear systems, chaos and applications
  - Nonlinear differential equations
  - \*Fourier analysis of nonlinear systems
7. Random processes
  - Pseudo-random number generators
  - Random walks
  - Monte-Carlo simulation and applications
8. Data analysis and modeling with application to physics
9. Complex systems and other issues
  - Cellular automata
  - Percolation
  - Fractals
10. Introduction to scientific software (Maple, MathCad, Mathematica)
11. Introduction to scientific visualization and animation techniques.
12. \* High-performance computing

Some topics may be added and some topics omitted at the instructor's discretion, and depending on student's interest and time constraints.

\* Indicates "optional" sections (if we get that far)