PHYS 6130: Computational Physics I

Syllabus

PHYS 6130 in conjunction with PHYS 6110 (Griesshammer) and PHYS 6120 (Haberzettl).

Teachers: Dr. Harald W. Griesshammer, Corcoran Hall 306, 202-994-3849, hgrie@gwu.edu; Dr. Helmut Haberzettl, Corcoran Hall 303, helmut@gwu.edu.

Hours: ***Friday 15:10 to 16:00 in Corcoran Hall 309***. Only the first 6 weeks will be held in a lab/studio format; afterwards, discussion of weekly project progress at these times, or as arranged with individual lecturers.

Bring your laptop to all lab/studio classes!

Prerequisites: Undergraduate Numerical Methods, undergraduate Mathematical Methods, Mechanics, Electrodynamics and Quantum Mechanics. Experience of vectors, matrices, multi-dimensional differentiation and integration, ordinary differential equations, complex numbers, etc. Basic knowledge of at least one programming language (fortran or C).
Co-requisite/Coordinated with *PHYS 6110: Mathematical Methods* (Griesshammer), *PHYS 6120: Classical Mechanics* (Haberzettl). First of a series of 3 courses, tied-in with the lectures of Semesters 1 to 3.

Goals/Learning Objectives: Introduction to Scientific Computing with problems accompanying the parallel course lectures. Focus on skill-building: clean and reproducible, well-documented and efficiently written code. This semester, focus is on applying efficient code-development tools (meta-language, debuggers, plotting, etc.); proficiency to assess usefulness and limits of numerical techniques. Final goals by semester 3 are proficiency in Scientific Computing on an advanced level, with the ability to: efficiently develop and document complex codes; critically assess powers and limitations of codes; be able to study more advanced and specialised numerical techniques on their own and as their research will necessitate. The tools and tricks we discuss form the indispensable back-bone of graduate life.

Outline of Contents, in thematic order only.

1. Studio/Demo on Good Practise in Scientific Computing (6 lectures)
   meta-language – documenting code – using debuggers – assessing numerical errors – plotting
2. Projects, e.g. root finding – eigenvalues and orthogonalisation – partial differential equations

Style: Start with 6-week studio on good practises in scientific computing, followed by 2 projects which are interlaced with the parallel course lectures Mathematical Methods and Mechanics. The grades in Computational Physics do not count towards the grades in these courses, and vice versa. Each project spans 4 weeks and has intermediate, weekly deadlines. The first project will be associated with Mechanics, the second with Mathematical Methods. A completed project consists of the submission of

- a stand-alone package which contains the complete code and all sub-routines/depending programmes, written for a common, standard, non-commercial compiler;
- the running executable;
- sample output;
- technical report/documentation in LaTeX describing the workings of the code, including which implementation was chosen and why, and its limitations;
- further requirements as specified in the project descriptions.

While structures are quite common, two programming languages account for nearly 99% of all code written or used in Physics: fortran and C. As understanding them is essential, projects have to be completed in either, as specified in the Project Descriptions.
Solutions must be *sent by email as .pdf file of documentation and results plus source code plus executable* to Drs. Griesshammer (hgrie@gwu.edu) and/or Haberzettl (helmut@gwu.edu) as specified in the project descriptions.

Grading policy: The course will be graded on an absolute scale. The final grade is a sum of:

- Exercises/Homework (30% of total grade): studio-like setting with homeworks, supervised by Dr. Griesshammer, weekly in first 6 to 7 weeks;
- Project 1 (35% of total): posed by Dr. Haberzettl (Mechanics), likely 7 to 28 Oct (subject to change);
- Project 2 (35% of total): posed by Dr. Griesshammer (Mathematical Methods), likely 4 Nov to 9 Dec (subject to change).

In order to pass, you need at least 50% of all points. An excellent score usually starts at 80% of all points. No exam. Graded solutions are **returned and discussed during class**.
While it is necessary to have the correct answer for full credit, it is not sufficient. Indeed, it may serve you only one point. What you hand in should be a tidy and efficiently short presentation of your results and how they come about, which can be understood and reproduced by your peers. Imagine it is not homework, but a research problem whose solution you are asked to explain to your peers. Electronic submission is no excuse for leaving out sketches.
We reserve the right to award zero points for any illegible, chaotic or irreproducible section of your homework.
Homework serves several purposes, e.g.: expand and solidify your "tool-chest", and deepen your understanding by applying what you learned.
We encourage you to form study groups to discuss and attack projects as team. Nothing helps you understand better than interacting with your peers. However, **the solution you hand in must be your own alone. Do not share code.**
You can best study and check your progress if you regularly discuss your project progress with the respective lecturer. Office hours are the prime tool to gauge progress and revisit material not fully digested yet.

Typical workload for this course: 4 hours per week, in addition to lectures and surgery hours. Friday Sessions may take considerably longer than the allotted 1 hr. They are used to start and continue coding of assignments.

Parameters on Project Help

(version Autumn 2018)

As discussed in the syllabus, the class will be held in a lab/studio format; discussion of weekly project progress at these times, or as arranged with individual lecturers. 2 projects are interlaced with the parallel course lectures. Each project spans 4 weeks and has intermediate, weekly deadlines.
While we lecturers provide a project problem and pointers to solve it, you will learn the techniques necessary on your own, by reading and studying. We will provide feedback and help you, but we will not tell you how to solve the project in detail. These are the detailed parameters we will employ when we help you.
We will be available for troubleshooting/debugging tips, but will not solve your problems for you. When coming for extra help, we expect you to supply:

- A clear statement of what you are trying to accomplish.
- A clear and concise description of a particular problem. "My code does not work" is neither clear nor concise.
- A list of things you have already tried before coming for help.
- Possibly (at our discretion) a minimal example of code that exhibits the issue you are having, including a pseudo-code description of the algorithm you are implementing.

We cannot read over a 200-line code and find the problem for you. You have to develop strategies to do that yourself, including "reducing code" to isolate an error.
If you receive advice and choose not to follow it, you are on your own.
Other requirements:

- Codes must be grammatically and syntactically correct, irrespective of the language used.
- Codes must be portable and standards-compliant. Non-standard or proprietary constructs are **unacceptable**.
- Codes must compile and run on any unix/linux machine using the gcc/g++/gfortran compiler. If you choose not to use gnu compiler collection, then you must compile according to industry standards. If not, your code will be returned to you ungraded.
- Whether or not you use gcc, you must code to ANSI and/or ISO standards.
- With C you can accomplish this in gcc by using one of the flags `-ansi` or `-std=c99`. Using `-pedantic` will give you both.
- With Fortran and gfortran you should code to the 95 standard (`-std=f95`). Fortran 77 is obsolete and must be avoided.
- Do not suppress error or warning messages! A code compiles only correctly if it does not produce warnings or errors when `-Wall` is set.

We will compile code with the following flags:
C: `gcc -Wall -std=c99`
Fortran: `gfortran -Wall`

It is your responsibility to make sure your code compiles when we test it. The above are the standards we apply.

Submitting code which compiles with error messages or warning will lead to significant reduction of the grade for the project.
Code which does not compile is immediately returned with a grade of "Fail" for the project.

Some Suggested Reading

There is *no required reading* for this course. You will not be able to find all aspects explained well in only one textbook. Moreover, it is an essential part of the learning process to view the same topic from different angles, i.e. using different textbooks. Here is a list of those which we found most useful. If you discover others, tell us. *In addition, each Project Description lists recommended readings for each project.*
You already received an email with instructions to download and install a workplace image, Mathematica, and various programmes and literature. That email is an integral part of the Syllabus.

Scientific Computing

[Shi] A.B. Shiflet and G.W. Shiflet: *Introduction to Computational Science: Modeling and Simulation for the Sciences*; Princeton University Press, ca. US$60. Focuses on common structures and goals of scientific computing; no particular computing language.

[Gar] A.L. Garcia: *Numerical Methods for Physics*; 2nd ed., Prentice Hall 2000, ca. US$100.

[Lan] R.H. Landau, M.J. Páez and C.C. Bordeianu: *A Survey of Computational Physics*; Princeton University Press 2008, ca. US$55; very similar as *Computational Physics*, Wiley VCH 2007.

[NR] W.H. Press, S.A.Teukolsky, W.T. Vetterling and B.P. Flannery: *Numerical Recipes* (old editions also for C, fortran 77, fortran 90); 3rd ed., online at http://www.nr.com/, or on paper: Cambridge University Press 2007, ca. US$55. The Bible.

Specific Programming Languages: C

[KR] B.W. Kernighan and D.M. Ritchie: *The C (ANSI C) Programming Language*; Prentice Hall, ca. US$55 (out-of-print).

[Lou] K. Loudon: *Mastering Algorithms with C*; O'Reilly, ca US$50.

[ANSIC] International Standard ISO/IEC: *ISO/IEC 9899:TC3 (n1256)* The ANSI specifications for the 1999 version of C, which is the most widely used flavour. It is in fact readable which is a bit surprising for a technical specification. This is *the* last word in C.

Specific Programming Languages: fortran

[Pag] C.G. Page: *Professional Programmers Guide to Fortran77*; 2005,
at http://www.star.le.ac.uk/~cgp/prof77.html

[Dur] University of Durham information Technology Service: *Guide 138: An Introduction to Programming in Fortran 90*; 2007, at *w.dur.ac.uk/resources/its/info/guides/138fortran90.pdf*

Miscellaneous

[Git] V.G. Gite: *Linux Shell Scripting Tutorial Ver. 1.0*; 2000, at http://www.elib.hbi.ir/computer/OS/os-index.htm

[Gnu] *Gnuplot FAQ*; 2008, at http://www.gnuplot.info/faq/

[Knu] D.E. Knuth: *The Art of Computer Programming*, $3+1$ volumes; Addison-Wesley 1998, ca. US$160. From the computer-guru and inventor of TEX.

A note on academic integrity: You like Physics, or you would not be here. Thus, it is trivial that you will abide by the GW Code of Academic Integrity in all graded work. An excerpt: "*Academic dishonesty is defined as cheating of any kind, including misrepresenting one's own work, taking credit for the work of others without crediting them and without appropriate authorization, and the fabrication of information.*" For the remainder of the code, see: http://studentconduct.gwu.edu/code-academic-integrity. I will deal with violations according to the Code.
A breach of academic integrity is a serious issue. The Scientific Method relies on the faith of the scientific community that the findings of its members are not fraudulent or forged. Reserachers may be wrong or sloppy, but we inherently trust they try their best to do a good job. Every researcher builds that reputation during a whole academic life – in graduate school, postdoctoral research, and thereafter. In recommendation letters, the department, your thesis advisor and collaborators all put their reputation at stake to endorse you. They all trust you. When that trust is broken, the Scientific Community feels violated and offended. It takes the only and strongest remedy: ostricisation, i.e. banishment from the scientific discourse; see e.g. wikipedia articles "Jan-Hendrik Schön", "Victor Ninov", "Andrew Wakefield".

Academic integrity is at the heart of your credibility as scientist. It is your most valuable asset. Do not risk it.

You are encouraged to collaborate on your homework and even to be inspired by a good textbook, but make sure you have understood what you hand in as your solution. Do *not* offend your own (and my) intelligence by copying other people's work (especially without referencing). The web-site, all problems and solutions are for your personal use only. Do not pass solutions or problems on to any student who has not taken the course (yet). Do not accept or solicit solutions from students who have taken the course. Other examples of a breach of academic integrity include: to facilitate cheating or help others to cheat; to obtain information for homework, exams, presentations, etc., by means other than disclosed in your bibliography; to ask for or give any kind of factual information which is not in an exam but needed to solve the problem, no matter how insignificant it may seem, except if the examiner approves; etc.
Noncompliance with these rules is a breach of integrity and will be dealt with accordingly. If you have any questions about what constitutes academic dishonesty, ask.

Absences and Excuses follow standard GW policy. It is your own responsibility to make sure you fulfil the criteria for passing, in particular that you get at least 50% of all the points available in all Problem sheets together (not per sheet). The only way around this criterion is to submit in writing documentation that you were unable to perform homework for more than half the semester due to reasons out of your control, as outlined in the GW policy on absences and excuses.
There will be no make-up exams. A missed exam will be dealt with case-by-case. Bring any potential conflicts or difficulties to my attention *before* the exam. If you miss an exam for some unexpected reason, it is your responsibility to notify in writing *within 24 hours* of the missed exam, or the grade will be zero for the missed exam. Absence for medical reasons must have formal, written documentation from the medical office providing care. DC traffic is no excuse, and no additional time will be provided for late-comers.
If you see a conflict between religious observances and the class and exam schedule, you will bring them to my attention in advance, in the first week of the semester. It is University policy to extend to these students the courtesy of absence without penalty on such occasions, including permission to make up examinations.

Recording Taking written notes in lectures has been the practise for centuries and is of course permitted. Indeed, I encourage you to at least have the manuscript printed out and annotate it vigorously during class. Taking notes makes sure your brain processes and stays awake.
However, ***electric or electronic recording*** of the lecture or any discussion in any way, shape or form (audio, video with or without sound, camera, photography, shellack, casette, dictaphone etc.) ***needs my prior written permission***, as well as prior permission by all students. I consider failing to obtain permission as a breach of GW's Students Rights and Responsibilities and Academic Freedoms policies. It also infringes on my and GW's copyright. I will call GW police, refer violators to the appropriate authorities for sanctions, and advocate for the strongest-possible prosecution permitted by criminal and civil law.

Security In the case of an emergency, if at all possible, the class should shelter in place. If the building that the class is in is affected, follow the evacuation procedures for the building. After evacuation, seek shelter at a predetermined rendezvous location.

Disability Support Services (DSS): http://counselingcenter.gwu.edu/ Any student who feels that man accommodation may be needed based on the potential impact of a disability should contact me privately to discuss specific needs. Please also contact the Disability Support Services office at 202-994-8250 in Rome Hall, Suite 102, to establish eligibility and to coordinate reasonable accommodations. For additional information, please refer to: http://disabilitysupport.gwu.edu/.

GW's Mental Health Services (202-994-5300) offers 24/7 assistance and referrals to address students' personal, social, career, and study skill problems. Services for students include:
– **Crisis and Emergency Mental Helth Consultations 202-994-5300** 24 hours, not only for emergency.
– Confidential assessment, counseling (individual & small group), referrals.

Some Hints for a Successful Graduate Life

This is by no means a complete list, but it helps me in my teaching and research. Use your own judgement!

Follow simple tricks to check that your homework solutions are focused, understandable and correct. You do not have to type your homework for this course using LaTeX, but Dr. Haberzettl collected some simple but highly efficient tricks how to structure your solution, how to avoid errors and which cross-checks are very valuable. Citation with minor modifications:

- **Indicate each problem number and the parts according to the original questions** in your solutions. Follow them in sequence. If you do not have an answer for any part, write "no answer" or leave blank. Do not repeat the question.
- **Structure your document by paragraphs**; do *not* run all parts together in one big amorphous narrative. Be concise. When necessary for cross-referencing, number the equations. Clearly identify your answers to the specific questions of the problem.
- **Clearly explain your notation, what you are doing, and why you are doing it.** Otherwise, a teacher needs to guess and you have no recourse if that guess is wrong. "But it's obvious that I meant..." is not a valid excuse. Even a Professor of Physics is not psychic, and what is obvious to you may not be obvious to me.
- **Equations that mix incompatible mathematical entities are wrong!** For example, clearly distinguish between vectors and scalars. "$\vec{F} = ma$" is wrong because it equates a vector to a scalar.
- **Always check whether your results have the right units**. This is a very simple, but extremely effective, method to avoid many mistakes. If you make a habit of it, it becomes subconscious. For example, "[kg]=[N m^3]" must be wrong.

Attack your homework early. Don't postpone it to the last few days or even minutes.

Get a first impression. Explain to you (or, better, your peers) in your own words what the problem is about and what you are asked to determine. Avoid using formulae, focus on a "story". Sketch a figure explaining the physical situation.

Make a plan of attack. First think what solution you expect from your physical intuition. This can include a sketch of the expected solution. Then ponder over a good way to find the solution. This can take even an hour. Then take a deep breath. Then think again about the problem. Then solve. The time spent on first thinking about the solution is much shorter than the time wasted with abandoned attempts when you instantly start scribbling. In particular in exams.

Form teams (see above). Nothing helps one to understand better than discussing homework and lectures with peers. But practise additional problems alone in order not to become dependent on others.

Rank craftsmanship over ingenuity. You will be outstanding soon enough, but for now, continuous, solid work is more reliable than occasional sparks of brilliance.

Don't get nailed-down. Nobody requires you to find the *best/most elegant/fastest* solution. *Any* solution will do for a start. Once you have one, you can always look for a better one – if you have the time. When stuck, discuss with your peers (and consult the lecture and books). If you get very stuck, do another problem first. It's no use to get no problem done because you wasted all your resources on the first one.

Practise sketching and plotting. Discussions, sketches and plots are a must! Not only because the homework is full of these words, and you will loose a lot of points if you do not discuss, sketch and plot. But human beings are visual beings: We understand and recollect much better when we see a figure.

Assess your answer. Does the result make sense? Compare to limits in which the problem simplifies or in which you know the solution. Does the answer match your expectations (see "first impression")? If not, why not? Check that you answered all questions in the problem description.

Scrutinise your homework when it is returned to you and reproduce a correct solution. Clean up your notes. What did you not understand? What did you miss? Was there a faster way? Where are your strengths and weaknesses? You should spend at least an hour on that, as soon as possible. It will help you with the next homework set.

Work through each lecture on the day it is delivered. If you miss that, you will have a very hard time to understand the next lecture. In that context, "Tomorrow will be another day" is a very bad motto.

"Fill in the gaps" of the lecture. Spell out the details of proofs, make sure the signs and factors are correct, etc. That already gives you a lot of free practise in math, and makes sure your thinking and notes are up-to-date and correct. And you have a set of notes you understand when you come back after weeks or months or years, for exams or research.

Consult books (plural!) after you have reviewed the lecture. It will clarify things further, show you new and different perspectives, and deepen your understanding. I usually excerpt information which I found interesting in a book, in addition to lecture notes.
"It is always useful to get a second viewpoint because it's commonly the second one that makes sense – in whichever order you read them." [Nearing: Mathematical Tools for Physics, p. vii]

Look at the Physics behind a formula. Does it make "sense" from your physical intuition? Do you understand what it means? What are its limits, i.e. regimes where it becomes particularly simple to understand? What are its limitations, i.e. where does it not work? Explain it and its underlying principle to a peer or to an undergraduate, using no math. You will believe your most beautiful mathematical proof only if you can also give a good intuitive argument why the formula should be right.

Ask yourself: What is the hidden agenda behind this topic in the lecture, homework, etc.? What can I learn that goes beyond the straightforward application? Is there a greater principle involved which I can use in different contexts? Why is e.g. a proof presented this way? In which other fields could I use similar techniques/reasoning?

Talk with your lecturers. We post out office hours not out of courtesy, and we don't bite. If you don't come to me with your problems, how can we help you? we – for one – love discussing. Have no fear to overburden us. We will tell you when we have had enough.

Have a life outside Physics.

The University Counseling Center (UCC) assists you in addressing personal, social, career, and study problems that can interfere with your academic progress and success.
Services for students include:
– **Crisis Consultations at 202-994-5300** open day and night, not only for emergency.
– Confidential assessment, counseling services (individual and small group), and referrals: http://counselingcenter.gwu.edu/counseling
– Academic Support and Peer Tutoring Services: http://gwired.gwu.edu/counsel/AcademicSupport
– Podcasts and Self-Help: http://gwired.gwu.edu/counsel/PodCast, http://gwired.gwu.edu/counsel/OutreachSelfHelp

They are also very good when you need to review your habits, like learning and exam strategies.

It's never too early to get help.