Physics 344
Simulation and Inference in Stochastic Systems
Course Overview 2019

HC Eggers

1 Objectives of the course

- Practical experience with simulation of stochastic processes (Monte Carlo methods)
- Knowledge of the theory and practice of basic statistical concepts and calculations, oriented towards simulation and data processing.
- Applications, including random walks and the Ising model. Possible project work on a related topic.
- Computers, learning to program and numerical analysis as such are not part of the objectives of the course since they are covered in other modules.

2 General arrangements

2.1 Timetable and venues

		Third Term	Fourth Term
Mondays	12h–13h	Lecture Room Delta	NARGA F
Tuesdays	10h–13h	Electricity Lab	NARGA F
Wednesdays	08h–09h	Delta	NARGA F
Fridays	09h–10h	Delta	NARGA F

2.2 Theory and practice

The course consists of a **theoretical** and a **simulation** part. Initially, the development of the theory takes precedence. Applications and simulation gradually acquire more importance.

- **Theory:** The theory part is conducted in terms of the usual lectures and a test. Weekly problem sets are marked. The contents consist of **basic statistics** (lecture notes based on Spiegel Chapters 1–5) as well as further topics such as the Central Limit Theorem and the theory of Monte Carlo. **Theory of the underlying physics** such as the Ising model are discussed briefly as relevant applications arise.

- **Simulation** is done mostly in project mode. Apart from possible tutorials, formal problems and tests are used only in exceptional circumstances. Evaluation is done on the basis of a workbook; see Section 3 below. Except where indicated otherwise, material taken from the book of Newman and Barkema and other books form part of the simulation- and not the theory-component of the course, i.e. is not examined theoretically but evaluated by means of the workbook.

2.3 Attendance

In the fourth term, attendance of simulation class periods is compulsory. Normal lectures count for one hour; the tutorial slot on Tuesdays counts for two hours. An attendance register will be kept. One percent of the simulation mark (equivalent to 0.5% of the module final mark) will be deducted from the marks for Assignment 2 for each hour of absence. A student arriving more than 15 minutes late is marked absent.

2.4 Computers

- **Existing competence:** It is assumed that you already have a working knowledge of, and practical experience with, at least one operating system and one numerical computer language.
- **Computer language:** Use of C or java for fast simulations is recommended but not enforced. You may code in a language of your choice (e.g. python, matlab, fortran, C++, pascal . . .). With the exception of fortran and basic C, I cannot help you much with debugging. I can, however, read and assess all the above code.
- **Rules of conduct** as set out by NARGA apply both in NARGA and elsewhere. Specifically, eating or drinking is not permitted in the computer labs.
- **Deposition of code:** Arrangements regarding the electronic deposition of program code when the workbook is handed in will be made a few days before the due dates.

2.5 Books

The following books are available in the Physics Library:

- Spiegel, Schiller and Srinivasan, *Probability and Statistics* `G 519.2076 SPI`
- M.E.J. Newman and G.T. Barkema, *Monte Carlo Methods in Statistical Physics* `G 530.13 NEW`
- D.V. Schroeder, *Thermal Physics* Phys 314 prescribed textbook): introductory treatment of the Ising Model; good set of problems on entry level. `G 536.7 SCHR`
- K. Binder and D.W. Heermann, *Monte Carlo Simulation in Statistical Physics* (2de uitgawe): this and the Third Edition contain good ideas regarding programming of the Ising Model, but is now obsolete. `530.13 BIN`
- K. Binder and D.W. Heermann, *Monte Carlo Simulation in Statistical Physics* (3rd Edition) `530.415 BIN`
- D.P. Landau en K. Binder *A Guide to Monte Carlo Simulations in Statistical Physics*: Advanced book; use only to get ideas and to look for references. `530.159282`

For computation we recommend:

- W. Press et al., *Numerical Recipes in C*: Excellent book which covers both the theory and practice of a large number of algorithms. Use it for random number generators. `519.4028553`
- Brian W. Kernighan, Dennis M. Ritchie *The C programming language* `G 005.133 C KER`

You can also make use of the following books:

- J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer (2001): Advanced level book; use as supplementary material. `518.282`
- J.E. Gentle *Random Number Generation and Monte Carlo Methods*: complete treatment of random numbers. `518.282 GEN`
- N. Giordano, *Computational Physics*: Introductory level explanation of Ising Model and random walks `G 530.0285 GIO`
- H. Gould and J. Tobochnik, *An introduction to computer simulation methods* more advanced treatment `G 530.0724 GOU`

3 Your workbook

As mentioned, the workbook is the basis for evaluation in the simulation part of the course.

3.1 Basic principle

The goal of the workbook is to emulate the record and thoughts written down by a researcher during research. All the rules and recommendations made with respect to the workbook are based on this goal. **Write your workbook as a memorandum which you are writing to YOURSELF: What should you write, explain, document, in order to enable you to understand your own work in six months' or six years' time?** For example: Which parameter values must be recorded on your figures? How do you write yourself a memo regarding the design of your code and what the variables mean?

The workbook is not a report of the type required in physics prac modules. While readability is necessary, little value is otherwise attached to aesthetics. The primary criterion is not to be "beautiful" but rather "effective" in the sense of the above memorandum to yourself.

3.2 N- T- and O-pages

The workbook is a combination of:

N **NOTES**: Whatever you take down in class and what you note down from books, the internet and articles.

T **THINKING** *in its rough form*: Anything and everything which you write down based on your own thought processes and initiative, no matter how rough or confused or bad it may be. T-pages do not have to be beautiful, clear or well-organised; it's just rough work. You are perhaps confused, and that is OK, as long as the direction is towards more clarity.

— **Mathematics:** Mark your **independent working out of mathematical derivations or problems** from the literature as T-pages.

O The resulting **OUTCOMES**: Your O-pages should take the form of a summary which you can read and understand in six months' or six years' time. The ideal which the O-part of your workbook seeks to attain is the record of a researcher. The aim is to compile for yourself a memorandum, without excessive attention to aesthetics, but with maximal clarity. This is about **clarity** and **originality** of thought and the quality of **setting it out.**

In the fourth term, you should be **working steadily on your workbook.** Every part of the project cycle (basic work together, basic work on you project choices, creative work) requires its own writing. If you do not write up something at least a few times per week in the workbook, then you are missing the objectives of the course.

3.3 Workbook appearance

- Use a black A4 exercise book or ringbound book as your workbook. Do not use loose sheets or plastic filing covers which make it hard to turn the pages.
- Every page must be marked at the top with the **date** plus one of the three letters **N**, **T** or **O** to show in which of the above categories it falls.
- Use English or Afrikaans as you like.
- **The workbook must be in your hand writing, usually in pencil.** Use an eraser if necessary. Do not use electronic text processors.

- N- and T-pages can be on both sides of a workbook sheet, but U-pages should please stay on the **right-hand side of the sheets only**; use the left hand sides later for further notes and comments.
- Each **O**-section should have a **heading** (preferably in red). Whether N- and T-pages deserve headings depends on their content.
- **Number all your pages continuously**. This and other "**signposting**" (date, heading, number) is important for future cross-referencing and if you are looking for something later.
- Normally, you will begin with a confused combination of N- and T-pages. They don't have to look nice or be logical. They are just rough work. Really really they do not need to be perfect. If the mess on a page becomes unbearable, or if everything is just totally wrong, just draw two skew lines at the top of the page and leave it at that.
- **Earlier pages which have already been marked by the lecturer** should not be removed or changed. If it becomes clear later on that there is an error on such pages or something needs to be added or corrected, just put in a cross reference on the wrong page to the correct one.
- In addition to any electronic deposition of program code (see below), you should stick a **paper copy of your programmes** and the most important **graphics** in the book once they have attained their final form. All of these are O-pages. Always put labels on the graphics axes, if only in pencil afterwards.
- **Documentation of final versions of programs** and descriptive choice of variable and parameter names is important since the lecturer does not have time to indulge in mind-reading. Read *Six ways to write more comprehensible code* of Jeff Vogel which is available electronically.
- Results which can only be displayed on the **computer screen**: 1. Show them to the lecturer, and pass on the source code to him. 2. Put in a reference and short description into your workbook.

3.4 Levels of work

The definitions and ideas in this section are approximate only and cannot be rigidly enforced or followed. There is basic work and there is creative work.

- **Basic work should include**
  - ◇ notes which were supplied in introductory lectures or copied from the board,
  - ◇ the physics as it appears in sources, the mathematics, a short discussion of numerical techniques, graphics, listing of basic code,
  - ◇ simple questions, thoughts, conceptual and other problems, program issues etc. (Write out your question or problem.)
- **Creative work** assumes that you have done your thinking:
  - ◇ Look at the physics, results of your basic work and ask: "What's going on here?" Try to differentiate between the **physical**, the **mathematical** and the **numerical** effects in your basic results.
  - ◇ For example, think about:
    - — **assumptions** which were made in setting up the physics equations (sometimes light can be cast on them by considering solutions and results in extreme situations),
    - — **extensions** of the basic work,
    - — **comments** regarding your results.
    - — Further typical questions: "How well does the mathematics describe the true physical system that it is trying to describe? Are there refinements which would be sensible, or would they complicate matters unnecessarily? How sensitive are the equations and/or numerical aspects to the inclusion of a particular physical aspect of the system?"
  - ◇ Take your own decisions where to dig deeper and in which direction to take your questions and thoughts.

- Try not to write or make statements way beyond your level of understanding. It is better to directly state that you do not understand something. There is nothing wrong with incomplete understanding of a problem (research is nothing but the investigation of questions whose answers are unknown!); but there is much wrong with the "lie" which pretends that something has been solved which hasn't. Honest ignorance is worth more than a misleading answer.

3.5 Cooperation, use of material and plagiarism

- **Cooperation and peer learning**, in the sense that you consult your co-students how to do something, to get ideas, or to get help with debugging, **is encouraged**. It is quite acceptable to implement yourself an idea or method which you *saw* in another student's work.
- You can of course see what's available on the internet. (N-)material which comes directly from the net may appear in your workbook but is not taken into account during evaluation.
- You are supposed to **write your own code** and your **own workbook** except where code is directly provided to you.
- **Plagiarism:** The definition of plagiarism includes:
  - ◇ pretending that code, graphics, text etc. which you obtained from a co-student (inside or outside the course) is your own work,
  - ◇ having someone else write code on your behalf,
  - ◇ supplying your own code to another student,
  - ◇ putting into your book material which was copied from the internet or other sources without a complete reference to the source.
- Code, graphics etc. which was obtained from the net or any other source must be marked as such and the reference given.
- **Plagiarism is a serious offence** and is handled in terms of the university's policy on plagiarism, which includes formal letters, keeping of records and appearance before the Central Disciplinary Committee of repeat offenders.
- Punishment will be proportional to the seriousness of the offence.

4 Assessment

4.1 Overall structure

- Assessment in the theory part follows the conventional patterns of problems and test(s).
- However, in the simulation part, the methodology is very different:
  - ◇ Formal homework is kept to a minimum.
  - ◇ Every Tuesday at the end of the tutorial, I sign and date your workbook at the bottom of your current writing.
  - ◇ The workbooks are taken in and marked twice during the fourth term.
  - ◇ **Program code** is scrutinised closely and its quality forms part of the workbook mark.
  - ◇ **Questions:** If necessary, I ask you questions about your work and may adjust marks accordingly.

4.2 Mark allocation

- The module makes use of continuous assessment.
- The **final mark** is made up of **50%±5%** for the **theory part**, including problems and one or more tests, and **50%±5%** for the **simulation part**, including the workbook, programming and explanation (on the level of giving a small talk or answering questions, not a full seminar).

- The main test in the theory part is scheduled for **Tuesday 17 September**. An "exam" has formally been scheduled for Tuesday 5 November but there will be no formal exam. We will determine by consultation in class whether to use the exam date or an earlier date for the final hand-in of the workbook.
- In the simulation part of the course, the **workbook**, together with the **quality of your program code** carries the greatest weight. Criteria for marking are those of Section 3.4. Note specifically that N-pages do not count much, but the T-pages are very important. A workbook with few T-pages will not get good marks. Of course not every scrap of rough work will be marked.
- In O-parts, the goal is to demonstrate a scientific approach. Criteria include: clarity and logic of argument, quality of code and results, numbers in tables (by hand or code-generated), appropriate **use of mathematics**.
- Marks are awarded according to the results and their interpretation actually shown in the workbook, *not* according to results which your program could potentially generate but are not shown. It is not enough simply to display your code and expect the lecturer to explore its possibilities.
- Answers to my **oral questions** in connection with the workbook and your program code can influence the mark.
- Homework (if any) will of course be taken into account.
- The bottom ± 50 percent of the mark are allocated to completion of the **basic work**.
- **Additional and creative work:** Beyond the basic work, the higher marks are awarded for the quality of your additional and original work.
- As already set out, marks are deducted for absence from the simulation periods.

5 Outcomes of the course

"Outcomes" are defined as the set of skills, knowledge etc. which are supposed to measure the success of a module in actual progress of the student. The outcomes which Computational Physics C aspires to are:

- Insight and competency in the concepts and methods of stochastic systems.
- Insight and competency in the concepts and methods of inference.
- Appreciation of the importance and role played by random number generators and their pitfalls.
- Working knowledge of Monte Carlo simulation and its applications in physics and beyond.
- Familiarity with the operating system currently used in the module and a numerical computer language.
- The ability to write and debug computer simulations.
- Developed skills in compiling and maintaining a record of own work and thoughts.

ENJOY THE COURSE!