Computer Simulations in Physics 2021/2022

Kacper Cybiński

24 marca 2022

Lecture 1

Python Introduction

Python Introduction, very general. Course website
Lecture 1 presentation
Packages needed for the course:

- NumPy
- SciPy
- MatPlotLib

Tools needed:

- Python > 3.0
- Jupyter Notebook *(optional)*

Interesting submodules of **scipy**:

- `scipy.constants` - Physical constants
- `scipy.special` - Special Functions
- `scipy.integrate` - `scipy.integrate.quad` is an interface for `QUADPACK` FORTRAN integration package

The presentations for tutorials: Basic Python Tutorial Intro to NumPy and MatPlotLib **Cheatsheets**:

- Python Cheatsheet
- NumPy Cheatsheet
- Matplotlib Cheatsheet

Lecture 2

Complexity: Crackling noise, avalanches, and hysteresis

2.1 Grading

- lecture and lab combined
- one lab - 1.0 points for all excercises finished on the spot
- bring the excercises next week - 0.8 points
- Bonus, extra excercises (take-home), 0.2 points each
- Pass - 50% points
- Presentation (5 min, obligatory) - i.e. presentation of last week results. - 1.0 point
- Last week ($8 - 9.06$) final presentation time slot, and written exam (8 questions - 4 Daniel, 4 Tworzydło) - $8 \cdot 0.4 = 3.2$ pts, which accounts for 20% of course points, 50% needed for pass.

2.2 Crackling

- The system responds through discrete interactions,
- Events span a large range of sizes,
- We only look at macroscopic effects, not the microscopic.

Crackling systems examples:

- Fireplace
- Earthquakes
- Crampled Paper
- Magnetic material in external field

2.2.1 Earthquakes - Gutenberg-Richter Law

The relation frequency versus magnitude:

$$N \propto 10^{-\alpha M} \propto E^{-\frac{2\alpha}{3}}$$

It's a power law, and they are realted to scale invariance.

2.3 Avalanches

Avalanches happen because the systems naturally end up at the critical point. śelf organised criticality"(SOC). It was formulated by Bak.

Crackling studies in physics, i.e. : Bubbles in foams, fluids in porous materials, fractures of discordered materials, fluctuations in stock markets, cascading failures in power grids.

2.3.1 Magnet with random fields

Barkhausen noise experiment - magnetic domains flipping in external $H(t)$, which is turned into electric signal, and audible through the speaker.

What we will use here is a **Random field Ising Model**. Its energy function is:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j - \sum_i (H(t) + h_j) s_i$$

Its properties:

- local, Gaussian distributed h_j with std deviation $\sigma^2 = R$
- As spins initially point down ($s_j = -1$)
- The magnetic field $H(t)$ is slowly increasing, slowly enough for the system to reach the ground state
- The spins only flip to decrease energy
- It is all located on a 2D grid, with spind being the knots of the net
- Magnetization lags behind the field - we get histeresis, formulated as a series of small, sharp jumps

$$M = \frac{1}{c^2} \sum_i s_i$$

2.4 Lab Introduction

Process:

- Each spin flips when it can get more energy
- Local field $J \sum_{j(i)} s_j$[1] $+ h_i + H(t)$ at site $i = (\cdot, \cdot)$ changes sign
- Spin change can be trigerred by
  - one of the neighboring flips
  - Increase of $H(t)$, to $H(t) = 0$

Lecture Slides Labs Introduction

[1] neighbours of spin

Lecture 3

Computational Complexity once again

3.1 Complexity Classes

Theory of computation - study how resources scale with size N of the problem. IRL - what's the behavour of the least efficient part of algorithm when size goes to N
Paste graphic from Tworzydło slides Problems with exponential time ($t \sim 2^N$) are called łooking for a needle in a haystack"

3.1.1 Boolean computing - SAT problems

Here we have two terms which will appear in this

- CNF (Conjunctive normal form)
- kSAT - SAT problem with at most k variables

Profit? We can reduce an exponential problem to a polynomial (here: quadratic) time problem

3.1.2 Spin Glass

The problem leading to spin glass mapping being a NP-complete problem is Geometrical Frustration Lecture Slides Lab notes

Lecture 4

Introduction to active matter

Active matter system types
We divide them into two types:

- Coloid systems
- Dry active matter systems

An example of dry active matter system is a one moving with **Brownian Motion**. It is governed by **The Langevin Equation**:

$$m\ddot{x} = -\gamma\dot{x} + F(t)$$

F(t) is a random force, we will assume it to be gaussian-distributed. The mean squared displacement for particles will be:

$$\left\langle (x(t) - x(0))^2 \right\rangle \sim \frac{\alpha}{\gamma^2} t$$

Today we will be modelling a very viscous system, therefore negating the term $m\ddot{x}$.

New Webpage
For ex.2 Daniel recommends function `scipy.spatial.KDTree`

Indeks