Atomic Scale Simulations
(MSE485 / CSE485 / PHYS466)

- **Location:** 4101 MSEB
- **Date and Time:** Tuesday/Thursdays 9:30 – 10:50am

- **Instructor:** Dallas Trinkle (dtrinkle@illinois.edu)
- **Teaching Assistants**: Kevin Daniel (kdaniel5@illinois.edu)

- **Office hours:** Tuesdays/Thursdays 11:00-11:30am (in person, location MSEB 308)

- **Literature:**
  - "Understanding Molecular Simulation", Frenkel and Smit
  - "Computational Materials Science", Richard LeSar
  - "A Guide to Monte Carlo Simulations in Statistical Physics", David P. Landau, Kurt Binder, Cambridge University Press
  - "Computer Simulation of Liquids", Michael Allen, Dominic Tildesley, Oxford Science Publications
  - Antonia Statt online notes
  - Links/Notes posted on course website

My background

- **Professor** (joined U. Illinois in 2006) and Associate Head

- **Interests:** Computational materials science
  - Crystalline defects (dislocations, point defects, interfaces) from density-functional theory
  - Development of new algorithms and computational tools
  - Solid solution softening/strengthening
  - Theory of diffusion

- **Languages:** python, C/C++

Email: dtrinkle@illinois.edu Office: MSEB 308

Class Structure

Web:

- schedule, syllabus: https://courses.engr.illinois.edu/mse485/fa2026/
- Homework, Project Reports, Surveys, Quizzes, Gradebook: GradeScope (https://www.gradescope.com, G64YK2)
- Announcements, Links, Notes, Online discussion, Communication: Slack
- Code testing: PrairieLearn (https://www.prairielearn.org/pl/enroll)

Prereqs:

- Statistical physics/thermodynamics; programming experience

Goals of this course:

- Fundamental and rigorous introduction and **motivation** (Why?)
- **Outline** important concepts and techniques (What?)
- Provide first insight into **atomistic simulations** (How?)

- **Caveat:** Cannot cover everything in depth

Class Structure

Assessment:

- Homework (50% with 25% Code, 25% Report)
- Project (40% with 5% Abstract/proposal/status report, 15% final presentation, 15% final report)
- Peer reviews (15%)

Homework:

- Due: every two weeks Thursdays 6pm (dates: see syllabus website) on GradeScope
- 50% penalty per day late (need to contact me directly if late)
- Language: Python
- Proper typed up report needs to be submitted to GradeScope **written in Overleaf**
- Use PrairieLearn to test your code. You can submit and test as many times as you want.

Class Structure

Project:

- Group project (will assign teams in late September)
- Due dates: see syllabus website, 6pm
- 50% penalty per day late (need to contact me directly if late)
- Language: your choice – code is part of your report!
- Topic: your choice (will provide list of old topics for inspiration)
- Come up with a project idea, submit abstract/proposal and present to class as status report
- Carry out project, write report and present to class
- Must contain: Scientific research, algorithm development + testing, presentation
- Submit all presentation slides and reports on GradeScope

What you have to do as soon as possible:

- sign up for MSE485 in GradeScope
- get on Slack once I send you an invite

- read syllabus
- ask any questions you may have

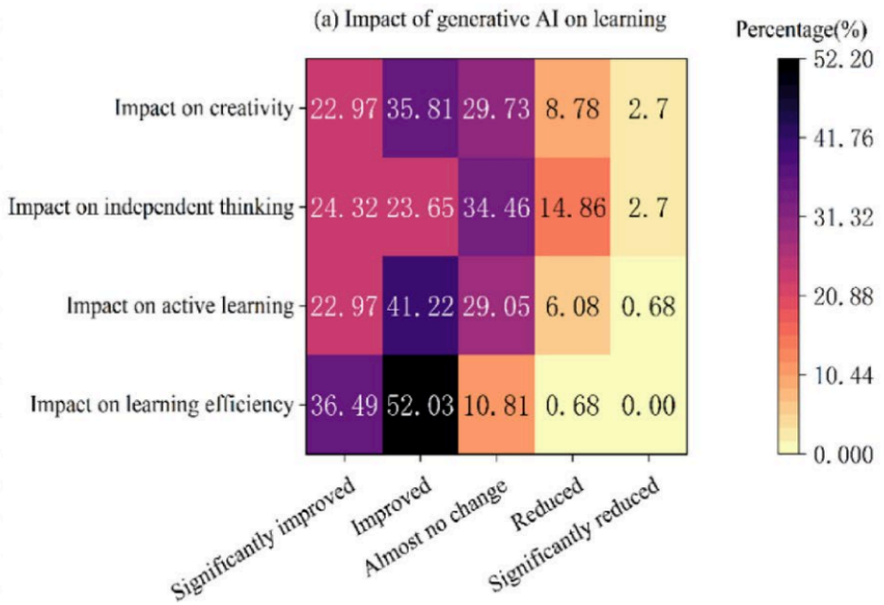
Class Structure

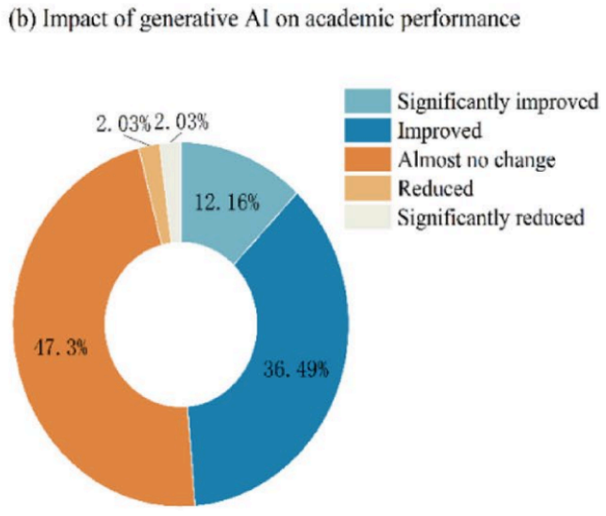
What you should do on a regular basis:

- Ask questions in class
- Reading (website) and homework (Prairielearn/Gradescope)
- Study and think about material; ask questions
- Seek out help when necessary
- Check grades in GradeScope

- DRES information: see syllabus + make contact ASAP
- You are bound by university honor code, NO exceptions
- Harassment or discrimination will not be tolerated
- Don’t come sick to class

Comments on AI





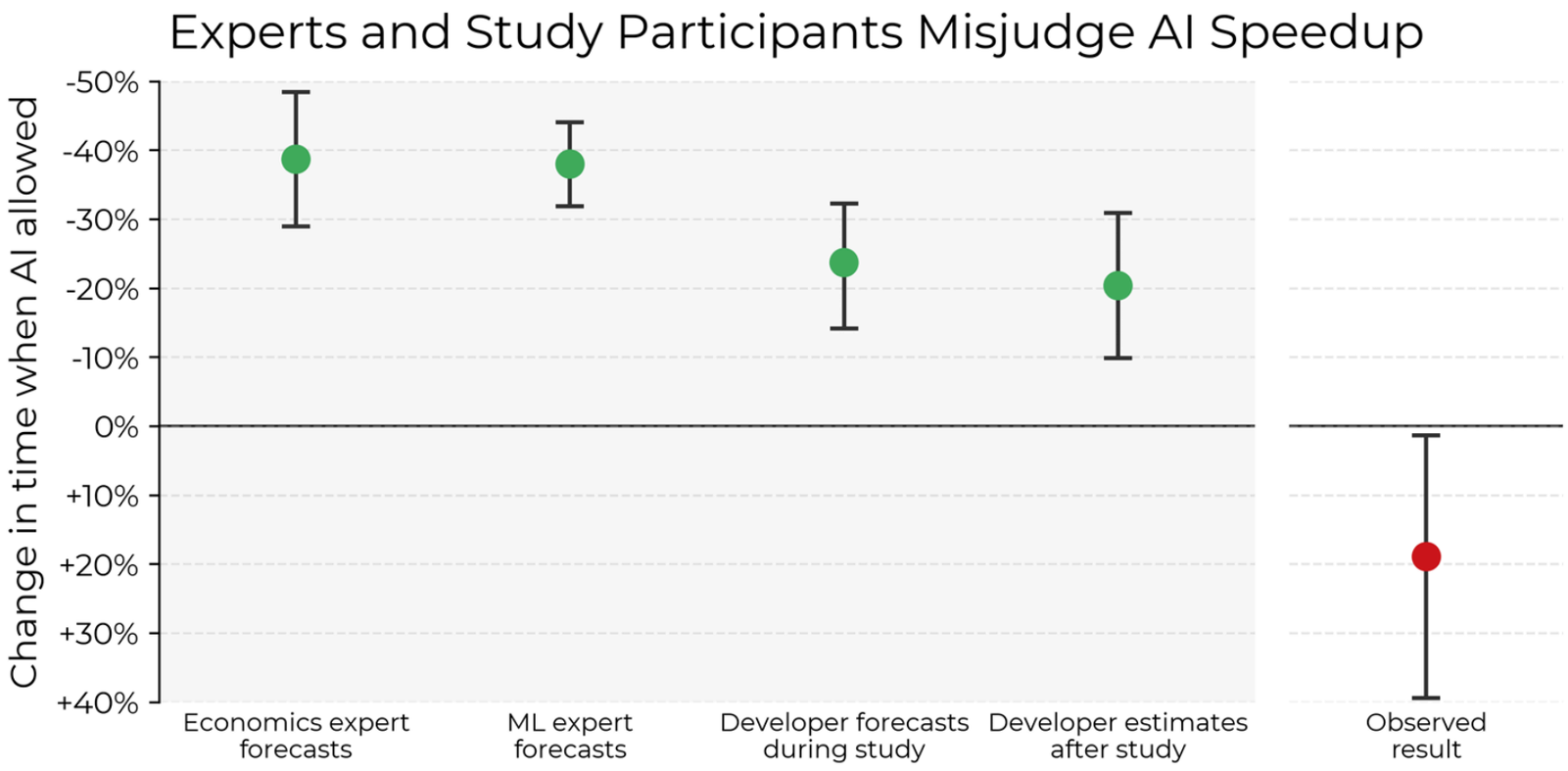
47% observed almost no change on academic performance, while 36% saw some improvement.

Biggest improvement: “Learning efficiency”

OPEN

Educational impacts of generative artificial intelligence on learning and performance of engineering students in China

Lei Fan✉, Kunyang Deng & Fangxue Liu



Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity

Joel Becker*, **Nate Rush***, **Beth Barnes**, **David Rein**

Model Evaluation & Threat Research (METR)

Expected time improvement: ~20% faster, actual observed time spent: 20% more time!

Biggest hurdle: AI doesn't know code context, time spent on cleanup/ debugging

The Lazy Student's Dream: ChatGPT Passing an Engineering Course on Its Own

Gokul Puthumanaillam, Timothy Bretl, Melkior Ornik

University of Illinois Urbana-Champaign, Urbana, IL, USA
(e-mails: {gokulp2, tbretl, mornik}@illinois.edu)

"a student who puts in minimal effort, showing no effort to learn the material, could use ChatGPT exclusively, get a B. The problem is the passing grade might be the combination of A+ in simple math and D- in analysis. They haven't learned much."

OPEN

The role of generative AI tools in shaping mechanical engineering education from an undergraduate perspective

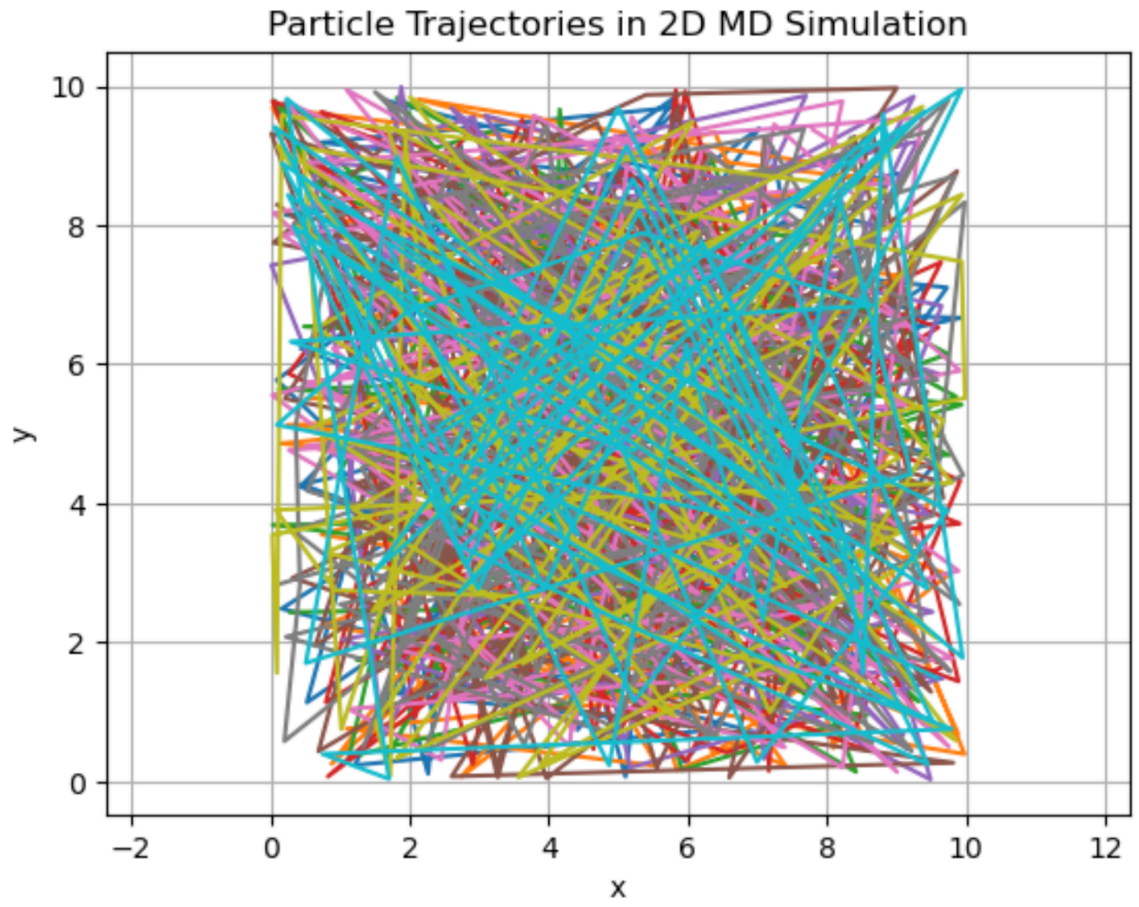
Harshal Akolekar[1,2], Piyush Jhamnani[1], Vikash Kumar[1], Vinay Tailor[1], Aditya Pote[1], Ankit Meena[1], Kamal Kumar[3], Jagat Sesh Challa[4]✉ & Dhruv Kumar[4,5]

AI shows "strong performance in theory-based questions, they struggled with numerical problem-solving, particularly in areas requiring deep conceptual understanding and complex calculations. Among them, Copilot achieved the highest accuracy (60.38%),"

- **Academic Integrity:** How do you credit a source if you don't know what the original source is? Better: find the actual source where the code is from and cite that.
- **Learning:** You want to learn – you need to practice/ exercise your brain. We are here to support you in your learning, not improve AI training/data set.
- **Potential Bias in AI Algorithms, Security Concerns,…**
- We are not going to debug your AI code/prompt!

AI MD Code

Copilot MD:



You will learn…

- Fundamentals of general methods used in simulations
  - Theoretical underpinnings (class)
  - Simple code implementation (homework, project)
- Some statistical mechanics/thermodynamics background
- How to analyze simulations, compute quantities,..
- Python: coding, debugging, testing,…
- Some information on commonly used MD packages (LAMMPS, HOOMD)
- Focus on MD, MC

You will NOT learn…

- Machine learning, AI
- Quantum Mechanics, DFT, etc.
- Computational research in specific fields

Atomic Scale Simulations
(MSE485 / CSE485 / PHYS466)

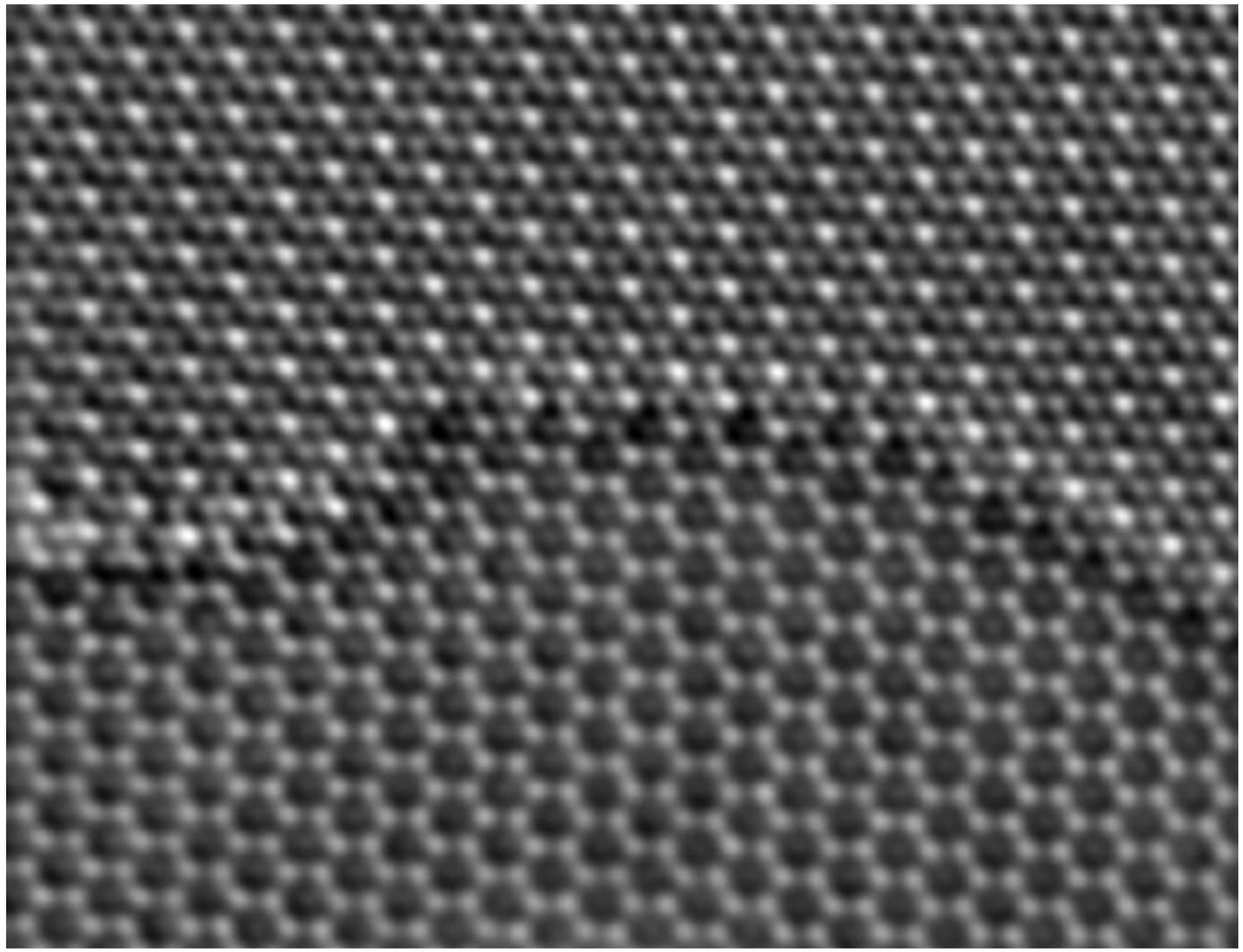
- Molecular Models and Interactions
  - Periodic Boundary Conditions, Intermolecular Potentials, short-ranged potentials, long-ranged potentials
- Statistical Mechanics and Thermodynamics
  - Ensembles, thermodynamic properties, structural and dynamical correlations/transport coefficients
- Molecular Dynamics
  - Verlet integration, thermostats, Langevin dynamics, MD packages, neighbor lists, HPC
- Monte Carlo
  - Sampling Methods, Metropolis, Gibbs ensemble

Atomic Scale Simulations (MSE485 / CSE485 / PHYS466)

Introduction/Motivation

Why Atomic Scale?

Nature Materials 10, 165–166 (2011)



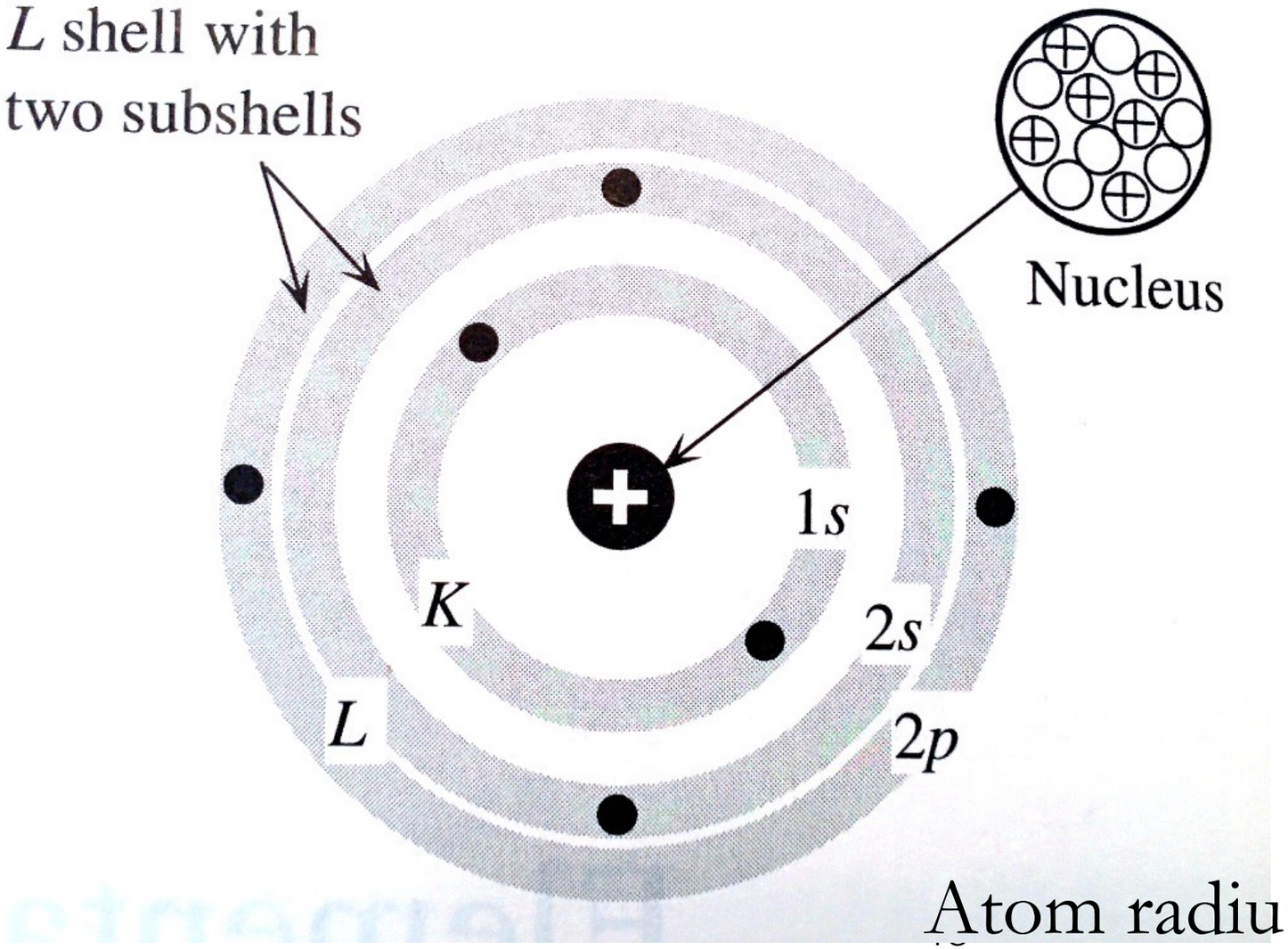
HRTEM image of single and bilayer graphene with atomic resolution

Atomic Scale?

- first “atomistic” theories of matter: Democritus (460 - 370 BC), Aristotle (384 - 322 BC), Plato, (428 - 348 BC)

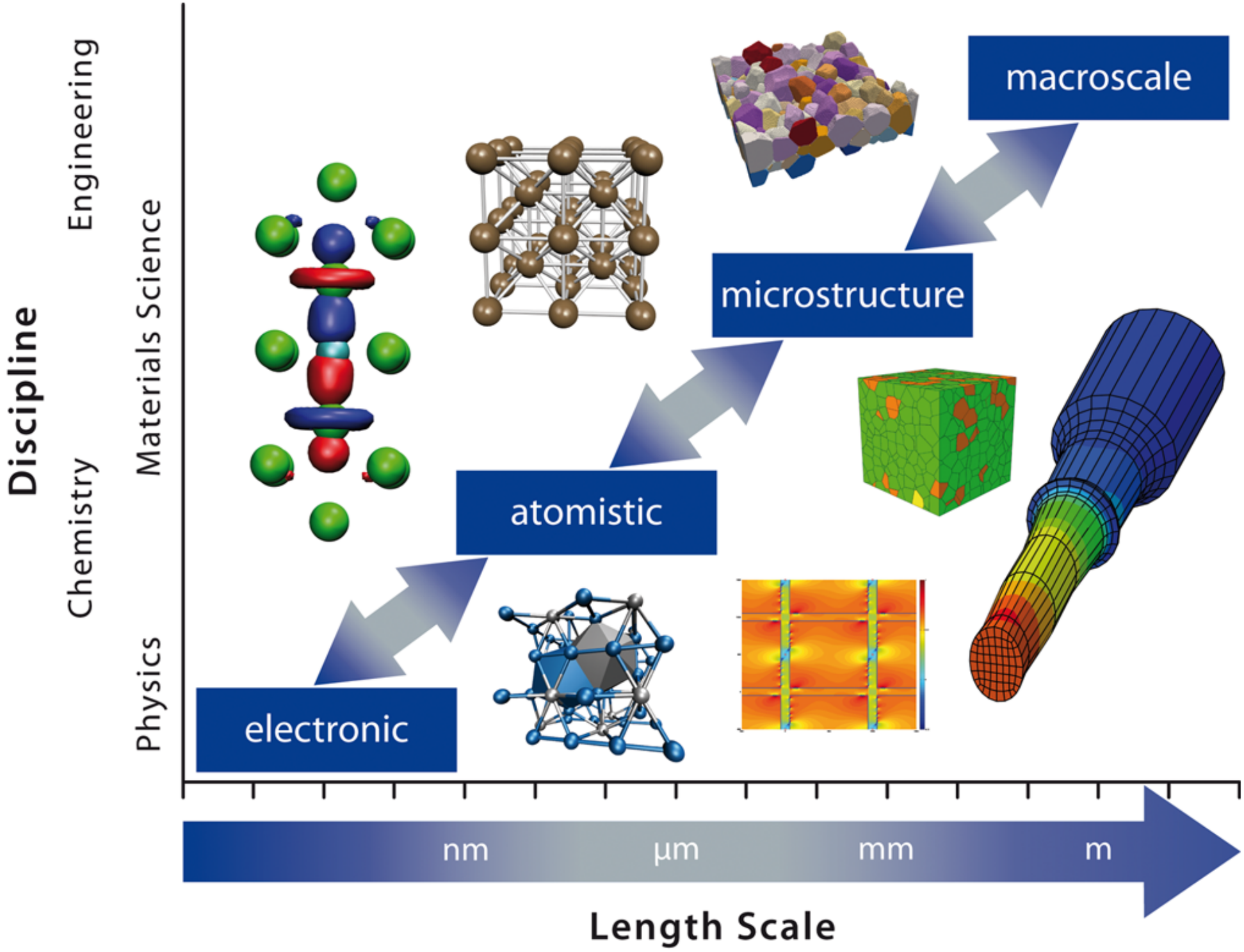
- More modern but still heavily simplified: Bohr model (1913)

Nucleus radius: 10 fm = 10^{-5} Angstrom



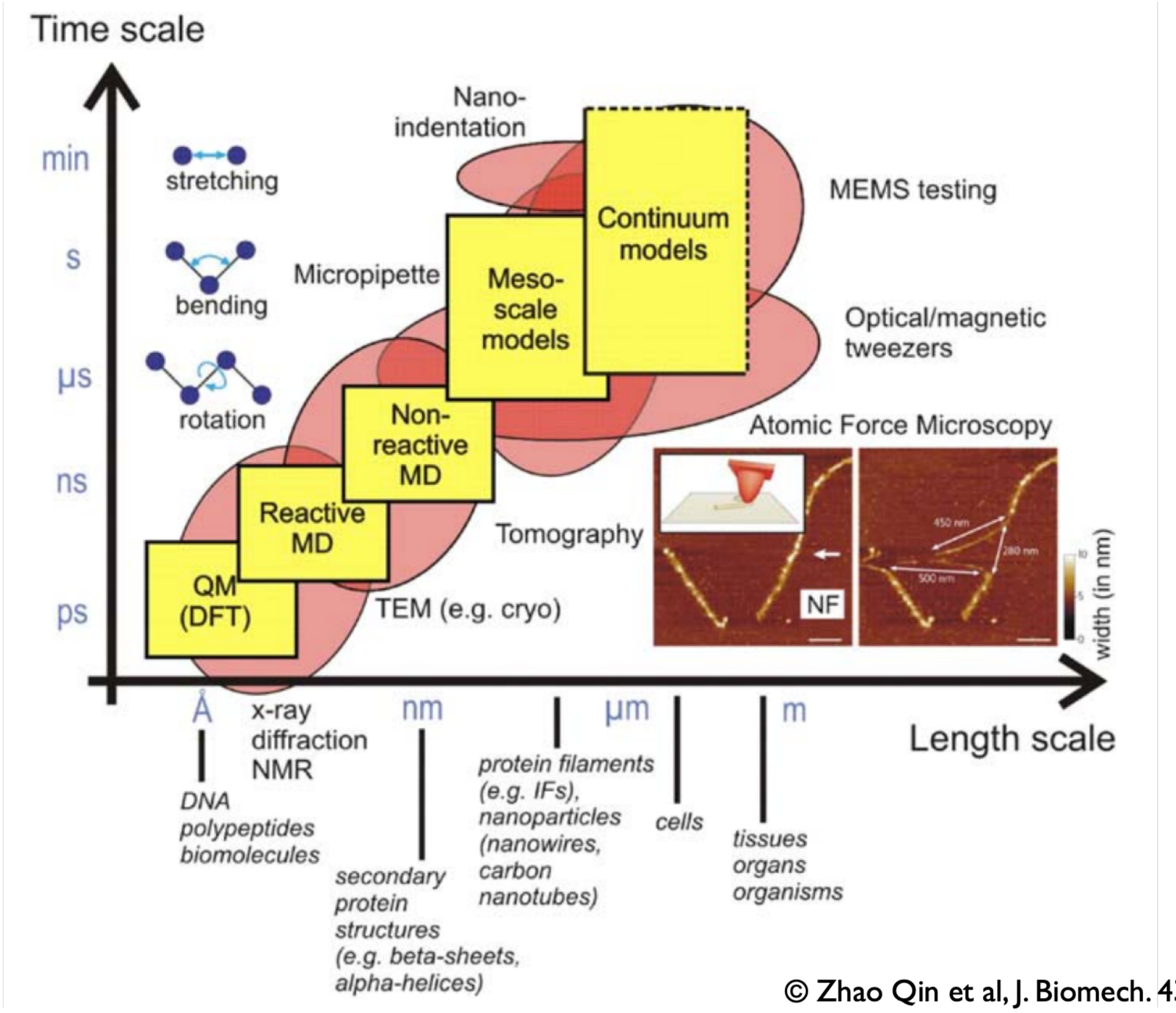
Atom radius: 1…few Angstrom

Atomic Scale?



(c) ICAMS

Atomic Scale?



© Zhao Qin et al, J. Biomech. 43, 15 (2010)

Atomic Scale?

- “The general theory of quantum mechanics is now almost complete. The underlying physical laws necessary for the mathematical **theory of a large part of physics and the whole of chemistry are thus completely known**, and the difficulty is only that the exact application of **these laws leads to equations much too complicated to be soluble**.” (P. A. M. Dirac, Proc. R. Soc. Lond. A 1929, pg. 123)

- “A profound joke is that one can measure the progress of physics by the problems that **can't** be solved. In Newtonian gravity, the three-body problem is difficult, but the two-body problem can be done exactly; in general relativity, two bodies are difficult but one body can be done exactly; **in quantum gravity, the vacuum is intractable**.” (Frank Wilczek, Physics Today 2003, pg.11)

Atomic Scale Simulations?

1. Use experimental data, invent a model to mimic the problem
   - semi-empirical approach
   - **But:** one cannot reliably extrapolate the model away from the empirical data.

2. Use foundation of physics
   - Maxwell, Boltzmann, and Schrödinger
   - numerically solve the mathematical problem
   - determine the properties (first-principles/ab initio methods)

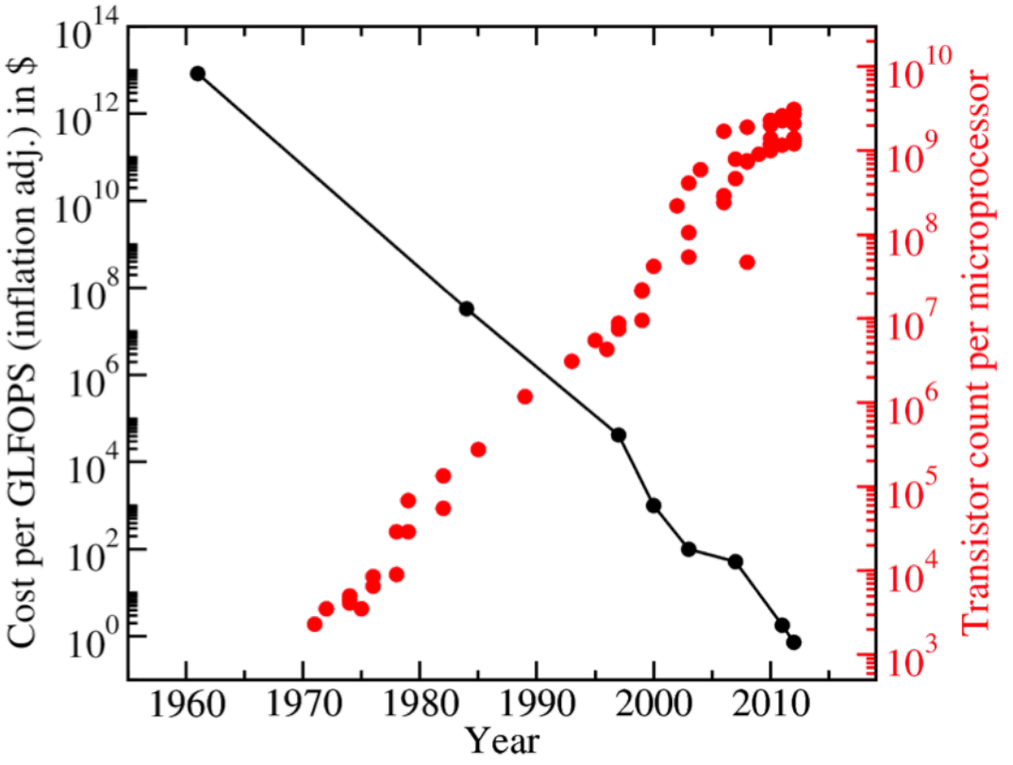
Motivation

- Simulations are the only general method for “solving” many-body problems

- Experiment is/can be limited and expensive
- Simulations can complement the experiment

- Simulations are “easy” even for complex systems

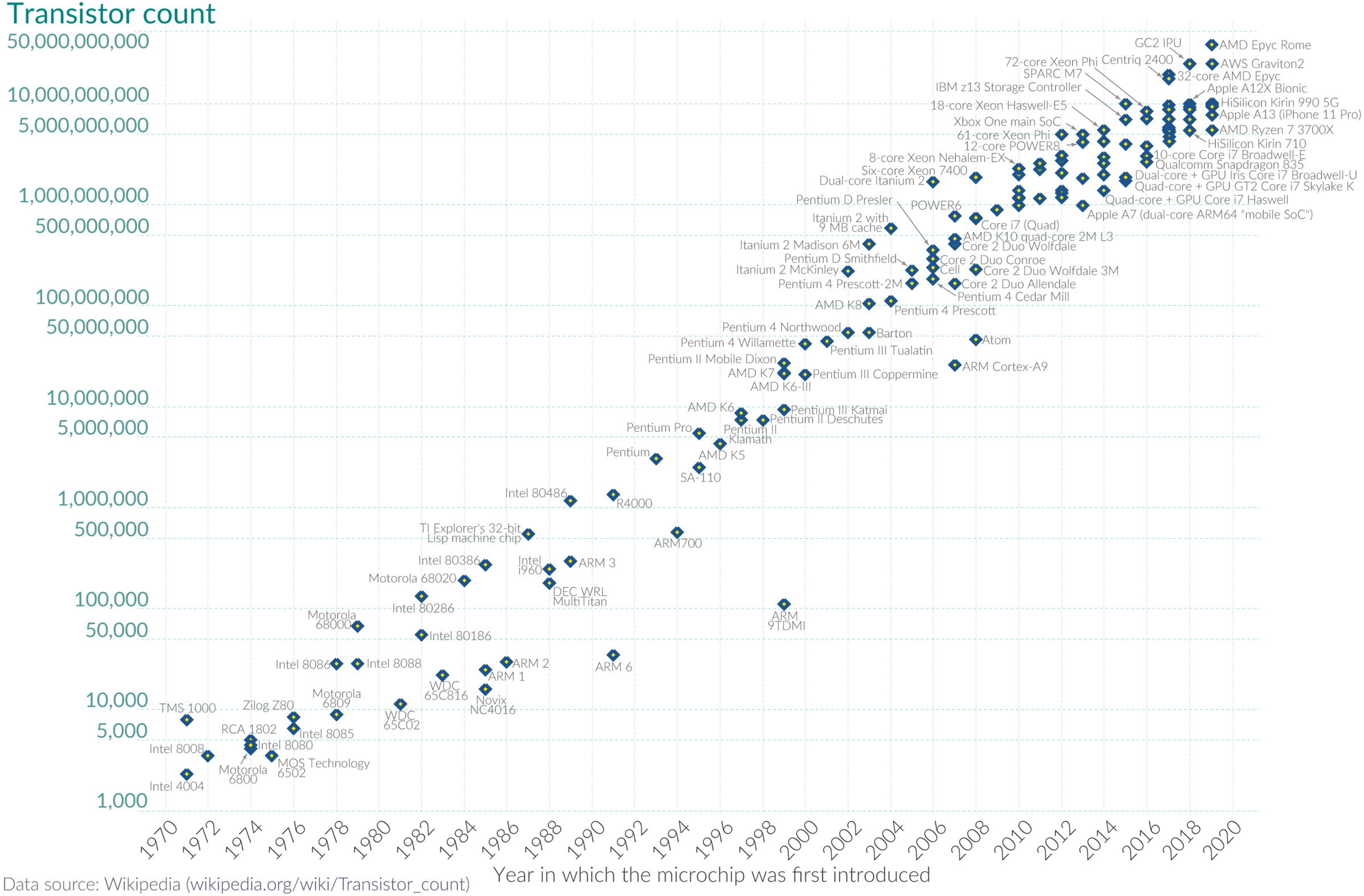
- Some methods scale up with the computer power which is growing more powerful every year—Moore’s law
- 1965: Gordon E. Moore extrapolates data from 1958 - 1965
- doubling of transistor count every two years “for at least ten years” (still mostly valid today)



Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

https://en.wikipedia.org/wiki/Moore%27s_law

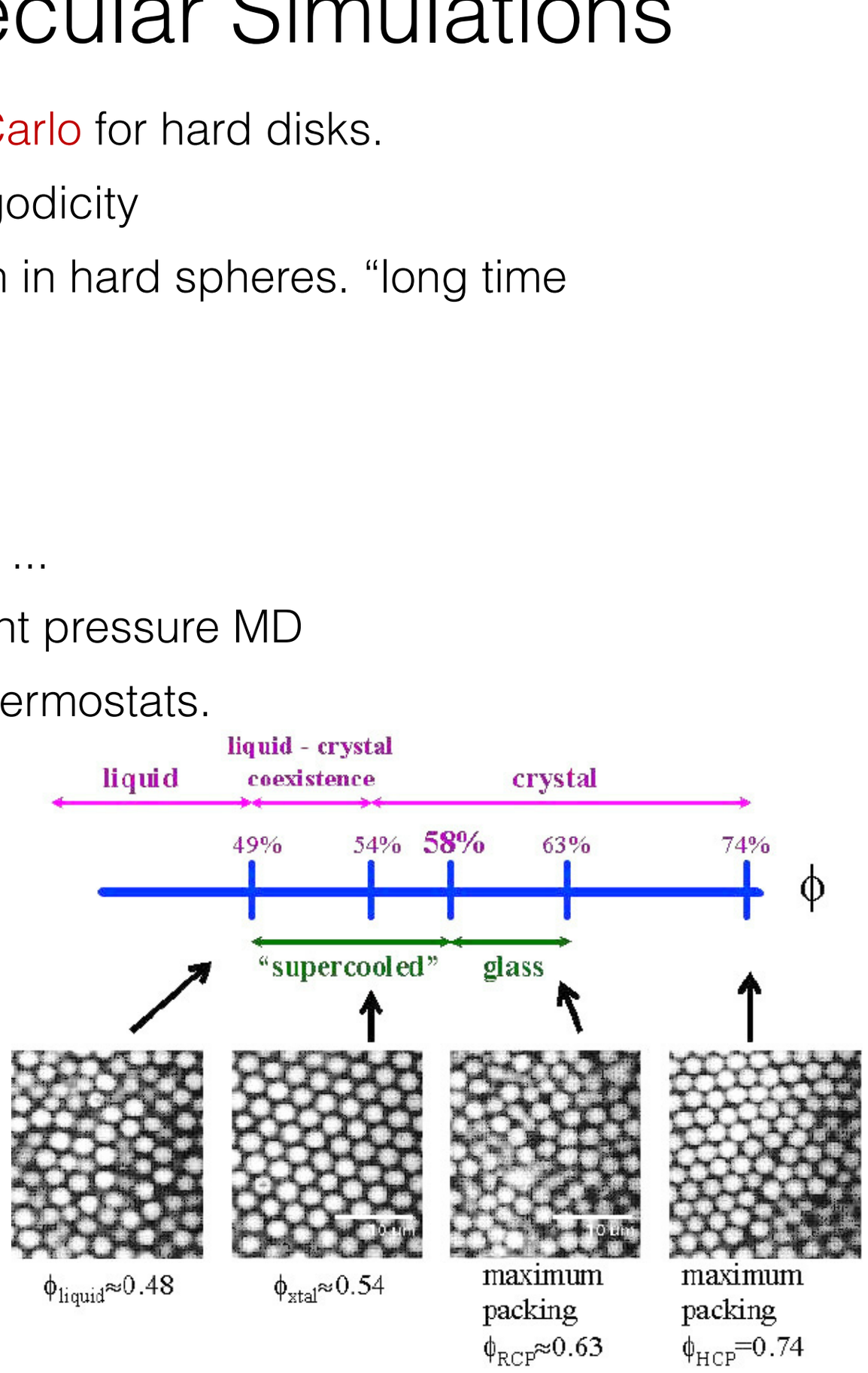
Motivation

- Computational physics per se is interesting and challenging:
  - good (sometimes the only) way to understand the physics

- **But:**
  - Computer time is limited: few particles for short time
  - Systems with many particles and long-time scales are problematic
  - Physical and mathematical underpinnings: What approximations come in? Hamiltonian is unknown, until we solve the quantum many-body problem!
  - How do we estimate errors? (Statistical and systematic)
  - How do we manage ever more complex codes? (Parallelism, testing)

Short history of Molecular Simulations

- Metropolis, Rosenbluth, Teller (1953) Monte Carlo for hard disks.
- Fermi, Pasta, Ulam (1954) experiment on ergodicity
- Alder, Wainwright (1958) liquid-solid transition in hard spheres. “long time tails” (1970)
- Vineyard (1960) radiation damage using MD
- Rahman (1964) liquid argon, water (1971)
- Verlet (1967) correlation functions, algorithms ...
- Andersen, Rahman, Parrinello (1980) constant pressure MD
- Nose, Hoover (1983) constant temperature thermostats.
- Car, Parrinello (1985) ab initio MD.

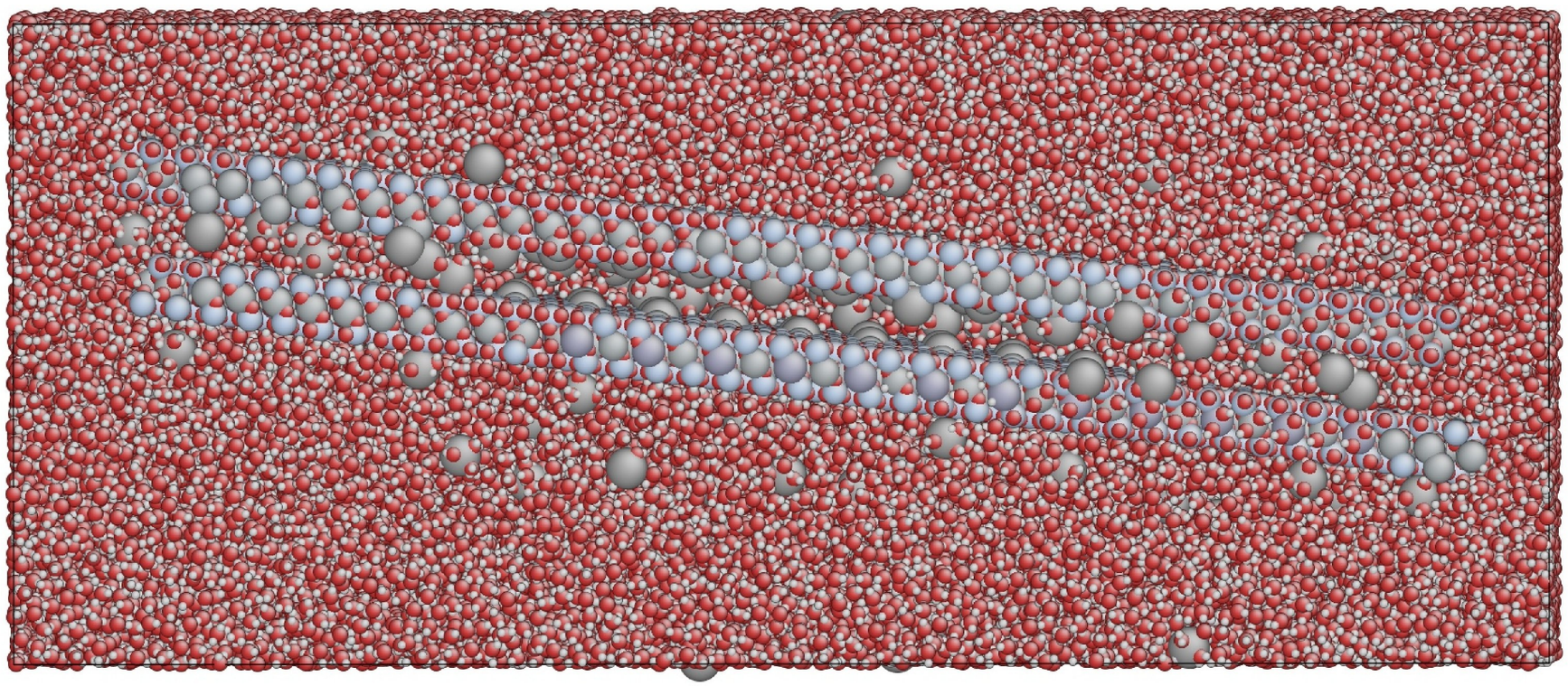


Prasad, Vikram & Semwogerere, D & Weeks, Eric. (2007). Confocal microscopy of colloids. Journal of Physics: Condensed Matter. 19. 113102. 10.1088/0953-8984/19/11/113102.

Course content

- **Molecular Dynamics:** integration algorithms, static and dynamic correlations functions and their connection to order and transport
- **Monte Carlo and Random Walks:** variance reduction, Metropolis algorithms, Kinetic Monte Carlo, heat diffusion, Brownian motion, etc.
- **Phase Transitions:** melting-freezing, calculating free energies
- **Polymers:** growth and equilibrium structure
- **Quantum Simulation:** Zero temperature and finite temperature methods.
- **Optimization techniques:** such as simulated annealing.

From microscopic to macroscopic



(c) ANL: Snapshot from a molecular dynamics simulation of an isolated clay platelet (consisting of two sheets) immersed in water.

How to learn about the system from this information?

Atomic Scale Simulations
(MSE485 / CSE485 / PHYS466)

Statistics/Phase space

How to characterize a simulation

- ***phase space*** S is used:
  - entire phase space: all possible states a system can be in
  - each point in phase space represents system in a possible state
  - each point in phase space contains “everything you can know” about system at that time
  - Example: Classical mechanics: positions $\{q_i\}$ and momenta $\{p_i\}$
  - Example: Ising model: spins (up/down), rolling dice (1/2/3/4/5/6), coin flip (head/tail)

Phase space: Example - Harmonic oscillator

- Newton's equations of motion:

- solve this differential equation

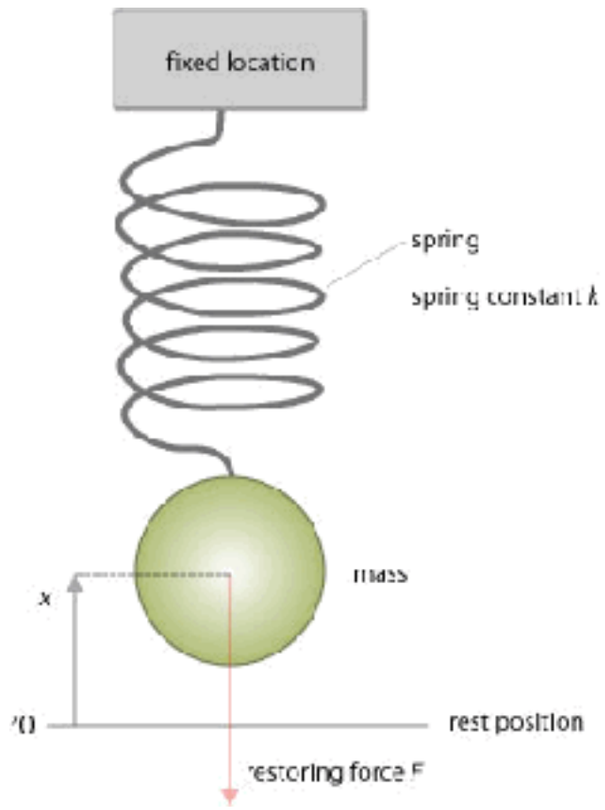
- constants determined by initial condition (velocity and position)
- after that, the ***entire*** trajectory fully determined
- kinetic energy:

- potential energy:

Phase space: Example - Harmonic oscillator

- Newton's equations of motion: F=ma

$$m\frac{d^2z}{dt^2} = -kz$$



- solve this differential equation

$$z(t) = A\cos(\omega t) + B\sin(\omega t)$$

- constants determined by initial condition (velocity and position)
- after that, the entire trajectory fully determined
- kinetic energy: $K = \frac{1}{2}mv^2$
- potential energy: $U = \frac{1}{2}k\left(x - x_0\right)^2$

Phase space: Example - Harmonic oscillator

- fully equivalent: write down the “Hamiltonian” (energy function)
- when kinetic energy only depends on velocity and potential energy depends only on position: Newtonian system

$$\mathcal{H}\left(\mathbf{r}^N, \mathbf{p}^N\right) = K\left(\mathbf{p}^N\right) + U\left(\mathbf{r}^N\right)$$

- using this, we know what a “trajectory of constant energy” looks like in phase space

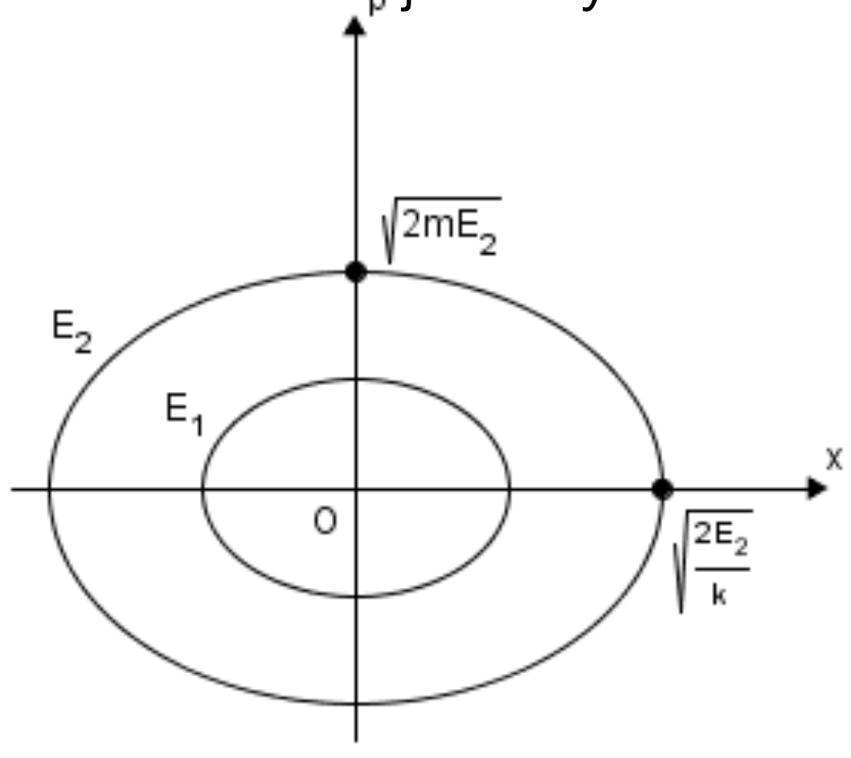
Phase space: Example - Harmonic oscillator

- fully equivalent: write down the “Hamiltonian” (energy function)
- when kinetic energy only depends on velocity and potential energy depends only on position: Newtonian system

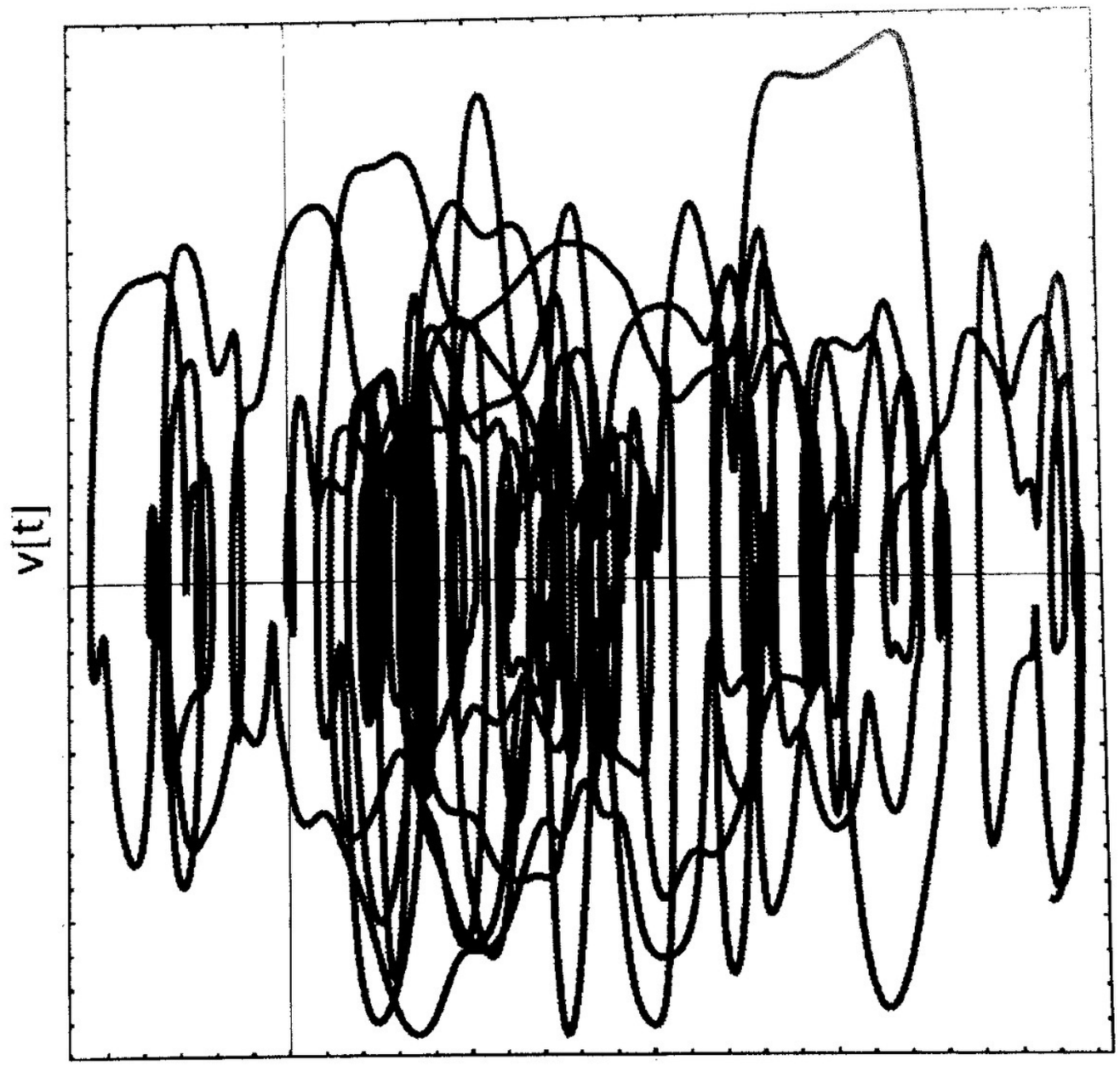
$$\mathcal{H}\left(\mathbf{r}^N, \mathbf{p}^N\right) = K\left(\mathbf{p}^N\right) + U\left(\mathbf{r}^N\right)$$

$$\frac{1}{2}mv^2 \qquad \frac{1}{2}k\left(x - x_0\right)^2$$

- using this, we know what a “trajectory of constant energy” looks like in phase space



Basic concepts of Statistical Mechanics



(b)

- one atom over a sub-femtosecond time span from a molecular dynamics simulation

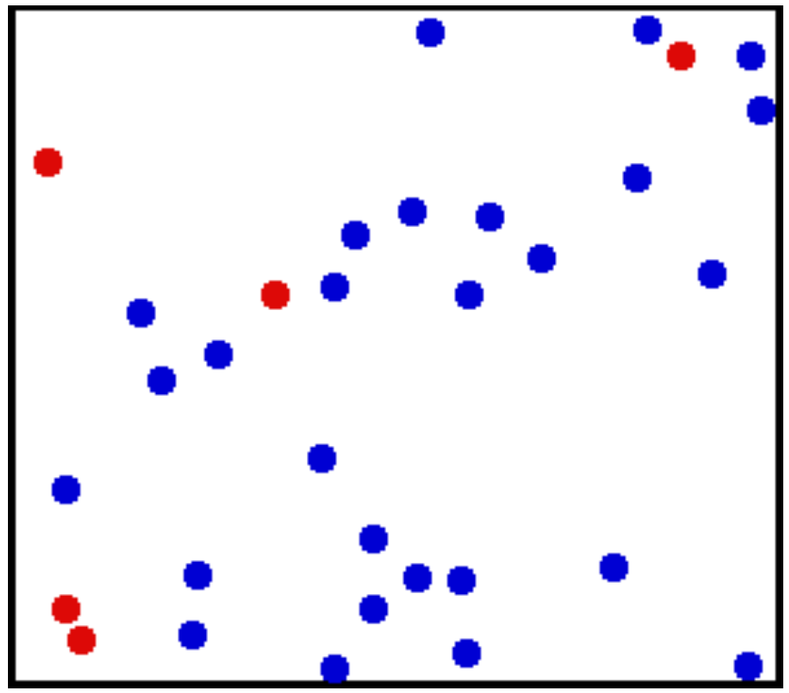
Basic concepts of Statistical Mechanics

- Micro state:
  - *N* point particles (e.g. gas molecules)
  - set of all positions and velocities is one micro state

- (Equilibrium) macro state:
  - characterized by only a few “state variables”
  - examples for state variables: pressure, temperature
  - example: isolated system <-> characterized by *E*, *V*, *N* (=const.)

Basic concepts of Statistical Mechanics

- Micro state:
  - *N* point particles (e.g. gas molecules)
  - set of all positions and velocities is one micro state



- (Equilibrium) macro state:
  - characterized by only a few “state variables”
  - examples for state variables: pressure, temperature
  - example: isolated system <-> characterized by *E*, *V*, *N* (=const.)

What is a simulation?

- simulation: changing the state: $S_{n+1}=T(S_n)$
  - deterministic: set of rules
  - random case: draw samples from a distribution

$S_0 \longrightarrow S_1 \longrightarrow S_2 \longrightarrow S_3 \longrightarrow \ldots \longrightarrow S_n \longrightarrow \ldots$

- index n can be time, pseudo-time, “iteration index”, …
- rules of simulation can be simple, outcome: unpredictable
- https://www.youtube.com/watch?v=C2vgICfQawE

- we will use tools from statistics and probability theory to characterize these situations

What is a simulation – Example: Conway's Game of Life

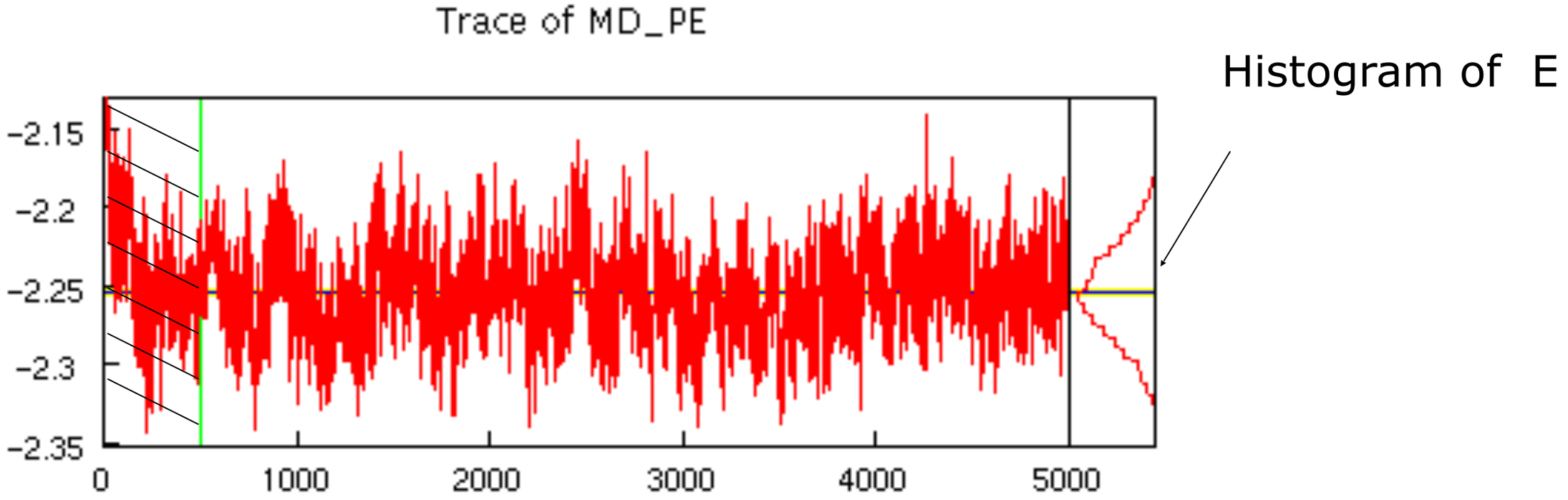
Simple rules:

- 2D grid, each cell is live/dead
- Interactions with 8 neighbors:
  - Any live cell with two or three live neighbors survives.
  - Any dead cell with three live neighbors becomes a live cell.
  - All other live cells die in the next generation. All other dead cells stay dead.

https://www.youtube.com/watch?v=C2vgICfQawE https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

Characterizing a simulation

- Trace and Histogram
- Equilibration time (transient, warm-up) to be defined
- Average and variance
- Law of large numbers: Error goes to zero for long trace



Summary and Outlook

- **This lecture:**
  - “Atomic scale” inherently related to statistical treatment
  - Language of statistical physics: Phase space, averages
  - What is a simulation: S_0 —> S_1 —> S_2 —> S_3 —> … —> S_n —>…
  - Trace, Equilibration, Histogram, Mean, Variance

- **Next lectures:**
  - Interatomic potentials
  - Python review
  - Statistical mechanics and statistics
  - Ergodicity
  - Autocorrelation and bias

Jupyter/Python Intro - Links

Useful Links:

- Python related:
  - https://www.python.org/about/gettingstarted/
  - Learn Python the Hard Way by Zed A. Shaw

- Libraries and Math:
  - http://matplotlib.org/
  - http://www.numpy.org/
  - http://www.scipy.org/
  - http://numba.pydata.org/

- Jupyter notebook:
  - https://jupyter.org/