THE UNIVERSITY of TENNESSEE KNOXVILLE

Lecture 02: Python Ecosystem and Numbers in Python

Sergei V. Kalinin

Course Information

Faculty Contact Information:

Instructor:	Prof. Sergei V. Kalinin,
Office:	314 IAMM
E-mail:	sergei2@utk.edu
Teaching Assistant:	TBD

Instructor Availability:

Please don’t hesitate to email me with updates, questions, or concerns. I will typically respond within 24 hours during the week and 48 hours on the weekend. I will notify you if I will be out of town and if connection issues may delay a response.

Meeting Time: 9:10-10:00 MWF, Ferris Hall 502

The lectures and materials will be posted on Canvas and at GitHub:
https://github.com/SergeiVKalinin/MSE_Spring2026

Office Hours:

Friday 1:30 - 3:00 PM are open for 1:1 meetings to discuss any course related item. Please set time by e-mail.

Course Outline

1. Mathematical Methods in Physical Science
2. Introduction to Python and the SciPy Ecosystem
3. Classical Numerical Methods with Python
4. Regression Methods
5. Mathematical Representation of the Real World
6. Differentiable Modeling and Physics-Informed Neural Networks
7. Bayesian Data Analysis
8. Causal Methods
9. Gaussian Processes, Bayesian Optimization, and Active Learning
10. Gaussian Processes Meet Physics
11. Decision Loops, Model-Based Control, and State Estimation
12. Use of LLMs for Decision-Making Workflows

Value Proposition

1. Achieve proficiency in scientific Python, tapping into its diverse mathematical applications.
2. Gain the ability to solve both foundational and advanced equations with confidence.
3. Deep dive into data analysis, mastering tools that empower data-driven decisions.
4. Grasp the nuances of Bayesian methods, learning how to weave together data with prior knowledge.
5. Explore the intricate landscape of causal analysis within the ML spectrum.
6. Develop a solid understanding of probabilistic techniques for decision-making in uncertain scenarios.

Prerequisites

To be successful in this course you will need a general background in materials science. Python or similar programming experience, while not essential, will be extremely useful. Students without any prior programming experience should expect to spend extra time outside of class learning basic skills.

Outcomes

1. This course aims to provide students with the skills needed to link physics, numerical methods, and big data

2. Students should learn how to combine intuition from mathematics, physics, and machine learning methods

3. The course is designed to provide students with basic knowledge of numerical methods, causal analysis, and Bayesian methods

4. Preparedness for the Future of Science and Industry: With insights into automated labs, large language models in scientific workflows, and federated tools and workflows, students will be prepared for the future of industry. These skills are increasingly important as companies automate processes and incorporate AI into their workflows. This knowledge can help students stand out in the job market and be prepared for the careers of the future.

This and that

Learning Environment:

The class will be delivered as in-person lectures. The Jupyter notebooks, code libraries, and videos provided. Weekly programming exercises will be assigned via Google Colabs and those students wishing to interact with the instructor in person should attend office hours.

Use of ChatGPT:

Strongly encouraged both for programming and written assignments. However, the students have to be aware of the limitations of the generative models.

Grading & Policies:

Midterms (2)	30%
Homeworks	40%
Final Project	30%

Reference Materials

I will provide copies of lecture notes, presentations, and Colabs on GitHub and Canvas. There is no specific textbook for the course, and we will take material from a variety of sources including:

- Andrew Bird et al, Python Workshop, https://www.packtpub.com/product/the-python-workshop/9781839218859

- Oswaldo Martin, Bayesian Analysis with Python - Second Edition, https://subscription.packtpub.com/book/data/9781789341652/

- Alexander Molak, Causal Inference and Discovery in Python, https://subscription.packtpub.com/book/data/9781804612989/

Homework 1:

- Create new Colab, https://colab.google/
- Chapter 1-4 and 10, Python Workshop.

Homework, midterm, and finals format

- All homeworks, midterms, and finals will be in the Google Colab format
- Use the code for programming exercises and markdown fields for text responses
- Share in the "comment" or "editor" modes
- The Colabs should save all graph outputs
- The Colabs should be able to run from the beginning to end (e.g. if I restart the runtime and run all)
- Submit to sergei2vk@gmail.com

Homework 1:

- Create new Colab, https://colab.google/
- Chapter 1-4 and 10, Python Workshop.

THE UNREASONABLE EFFECTIVENSS OF MATHEMATICS IN THE NATURAL SCIENCES

Eugene Wigner

Mathematics, rightly viewed, possesses not only truth, but supreme beauty cold and austere, like that of sculpture, without appeal to any part of our weaker nature, without the gorgeous trappings of painting or music, yet sublimely pure, and capable of a stern perfection such as only the greatest art can show. The true spirit of delight, the exaltation, the sense of being more than Man, which is the touchstone of the highest excellence, is to be found in mathematics as surely as in poetry.

- BERTRAND RUSSELL, Study of Mathematics

Numbers, Functions, Big Data

- **Numbers:** MatLab, Python, C++, ...
- **Functions:** Mathematica, some Python libraries (SymPy)
- **Big Data:** Python, Julia, ...

How we can run code

- Google Colabs
- AWS SageMaker notebooks
- IDE: Spyder, PyCharm, etc.
- Command line interface

Key aspect of teaching now compared to 2.5 years ago: ChatGPT

- Focus on ideas more then code
- Use but verify (confabulations)
- Build foundation easier – but will have to learn advanced topics anyway

ChatGPT and coding

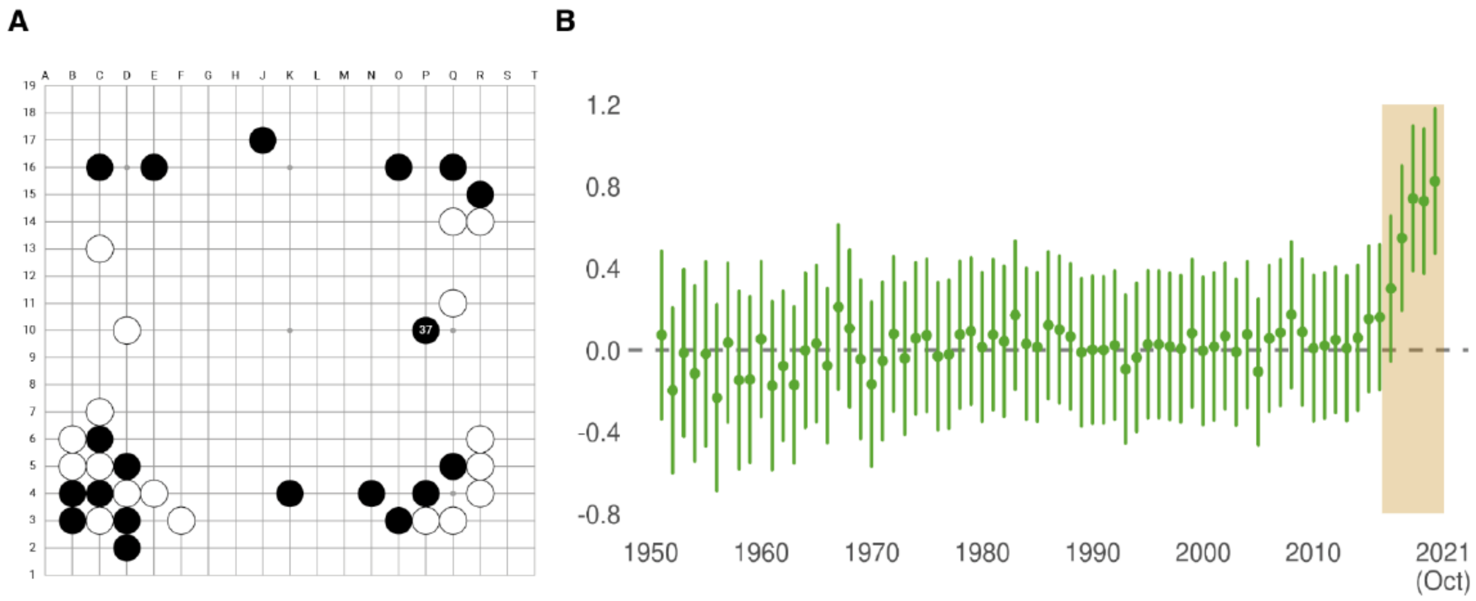
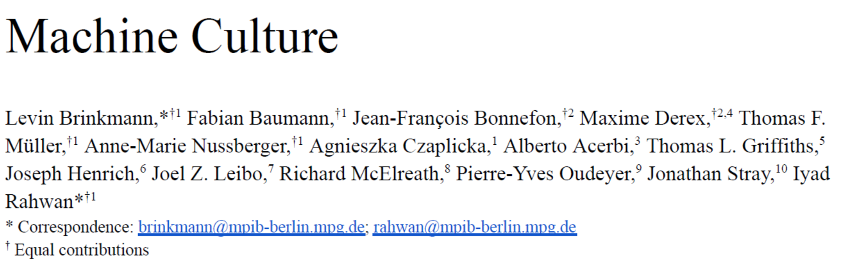


Figure 3: Go play before and after the introduction of AlphaGo. A AlphaGo, in its match against Go world champion Lee Sedol, made a highly unusual and strategic 37th move by placing its stone further from the edge, towards the center of the board, deviating from the traditional strategy of securing territory along the periphery during the early stages of the game. With this unconventional move, AlphaGo not only broke with centuries-old Go traditions but also paved the way for its ultimate victory in the match. **B** (reproduction based on [30]) Decision quality of professional Go players as evaluated by an algorithm performing at superhuman level. Decision quality significantly increased after Lee Sedol was beaten by AlphaGo



Machine Culture

Levin Brinkmann,[*†1] Fabian Baumann,[†1] Jean-François Bonnefon,[†2] Maxime Derex,[†2,4] Thomas F. Müller,[†1] Anne-Marie Nussberger,[†1] Agnieszka Czaplicka,[1] Alberto Acerbi,[3] Thomas L. Griffiths,[5] Joseph Henrich,[6] Joel Z. Leibo,[7] Richard McElreath,[8] Pierre-Yves Oudeyer,[9] Jonathan Stray,[10] Iyad Rahwan[*†1]

* Correspondence: brinkmann@mpib-berlin.mpg.de; rahwan@mpib-berlin.mpg.de

† Equal contributions

Code Repositories and Version Control

- Sharing scripts between users can be workable for immediate or short-term needs, but is not scalable nor lasting

- For reproducibility, it is better to have codes that reside in packages that are documented and well tested

- Most of you are familiar with python packages; but many are probably new to version control

- Version control systems such as git enable multiple people to work on a single software project at the same time to speed up development and ensure consistency

- Git is an open-source distributed version control system. It maintains a history of changes that have occurred in the project and allows for updates as well as reversions to older ‘commits’.

How we can share code:

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Git would take a significant amount of time to explain in detail. However, there are plenty of online tutorials, e.g.

https://www.atlassian.com/git/tutorials

Let’s just have a look!

What language do we use:

Main Python libraries we will use:

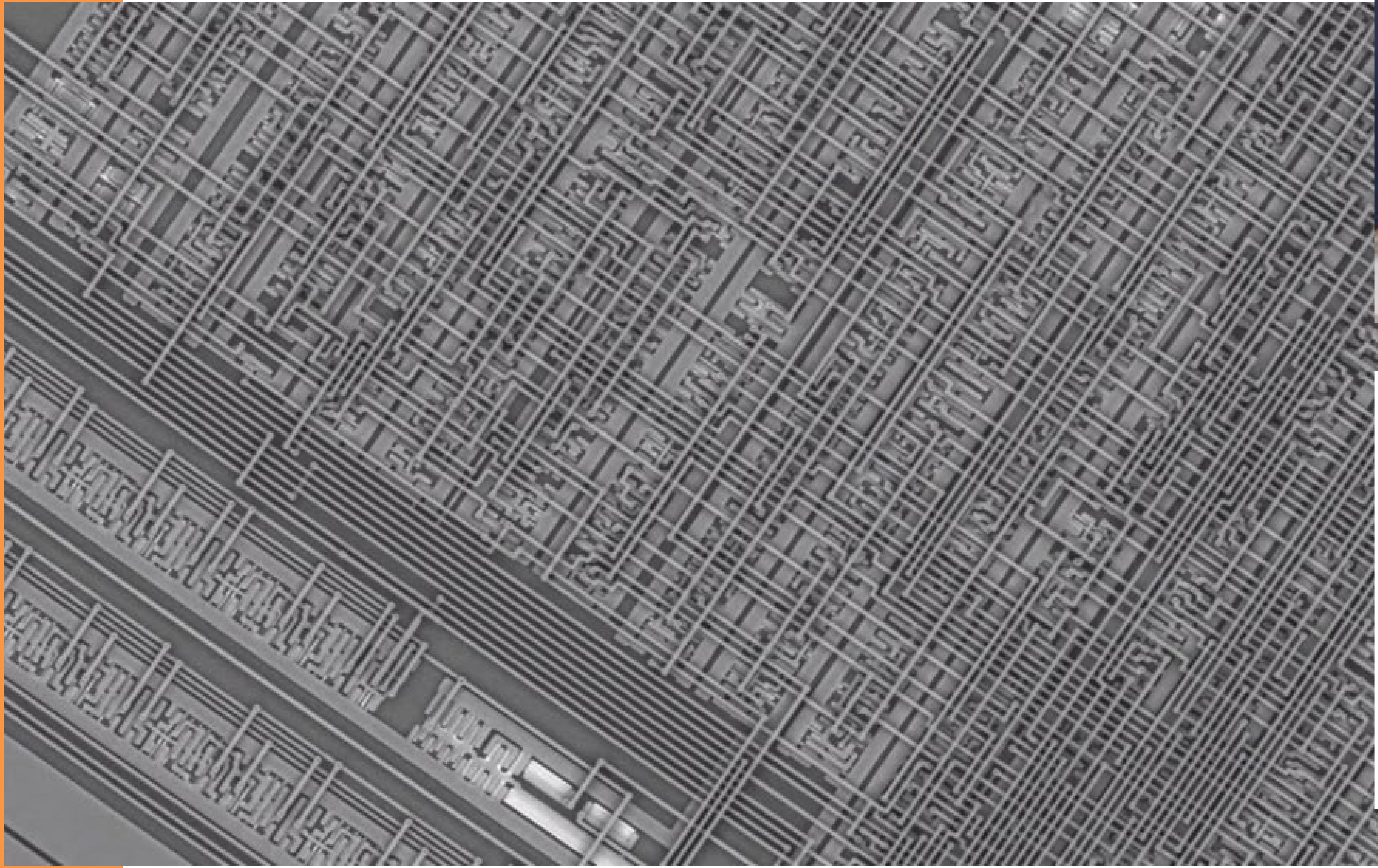
1. NumPy
2. MatPlotlib
3. Scikit-learn
4. Keras

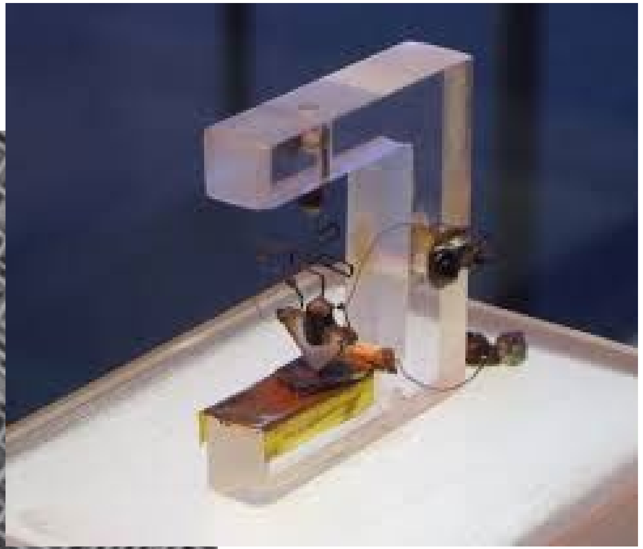
Other libraries we may use:

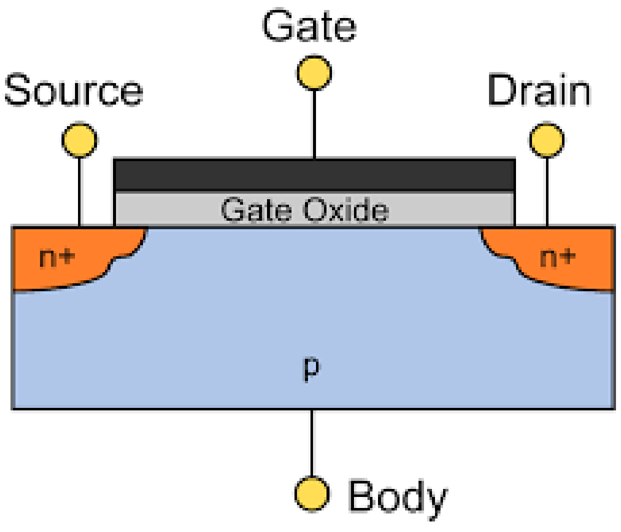
1. Seaborn
2. BOTorch
3. GPax
4. SciPy

We will learn these as we need them!

What’s inside the computer?







https://www.extremetech.com/extreme/191996-zoom-into-a-computer-chip-watch-this-video-to-fully-appreciate-just-how-magical-modern-microchips-are

https://www.britannica.com/technology/transistor/Innovation-at-Bell-Labs