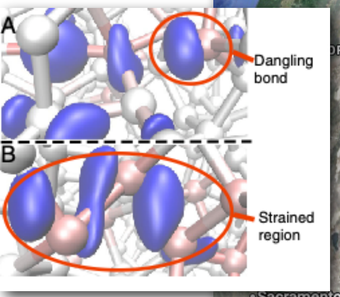
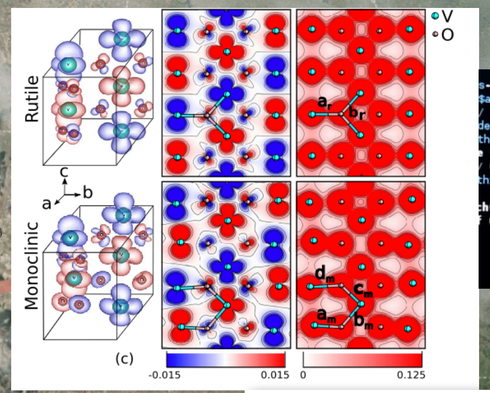
Introduction to computational physics and computing π

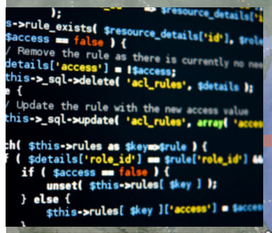
Instructor: Prof. Lucas Wagner

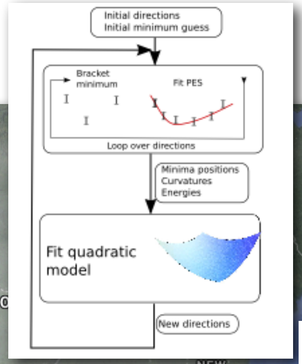
https://lkwagner.github.io/IntroductionToComputationalPhysics/intro.html

About me









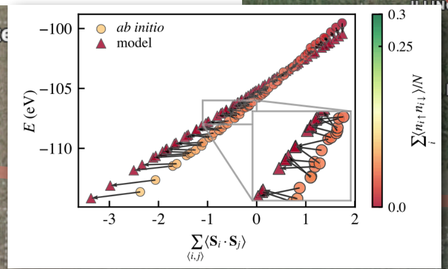
Postdoc

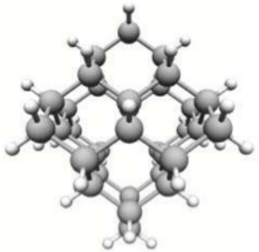
Faculty

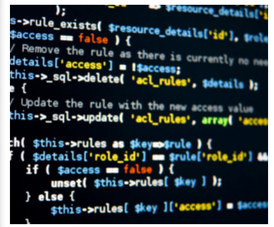
Grew up

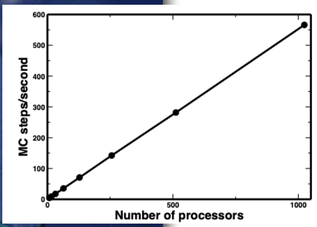
Postdoc

School

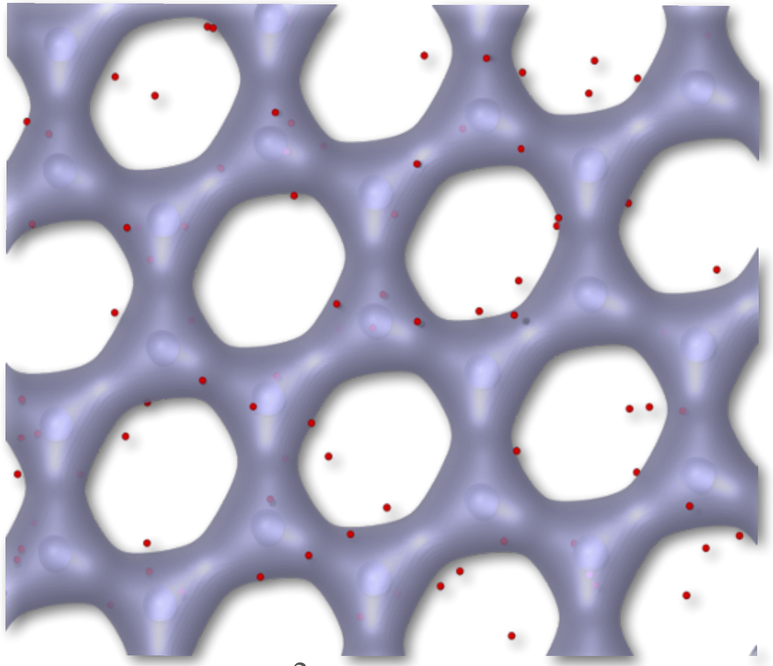








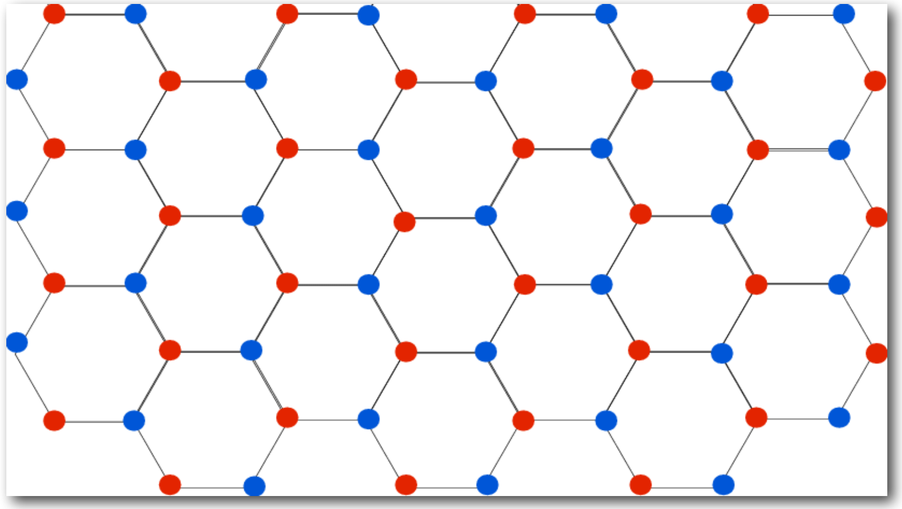
My research



$$\hat{H} = -\sum_i \frac{\nabla_i^2}{2} - \sum_{i\alpha} \frac{Z_\alpha}{r_{i\alpha}} + \sum_{ij} \frac{1}{r_{ij}} + \dots$$

First principles calculations: universal physics for all condensed matter systems.

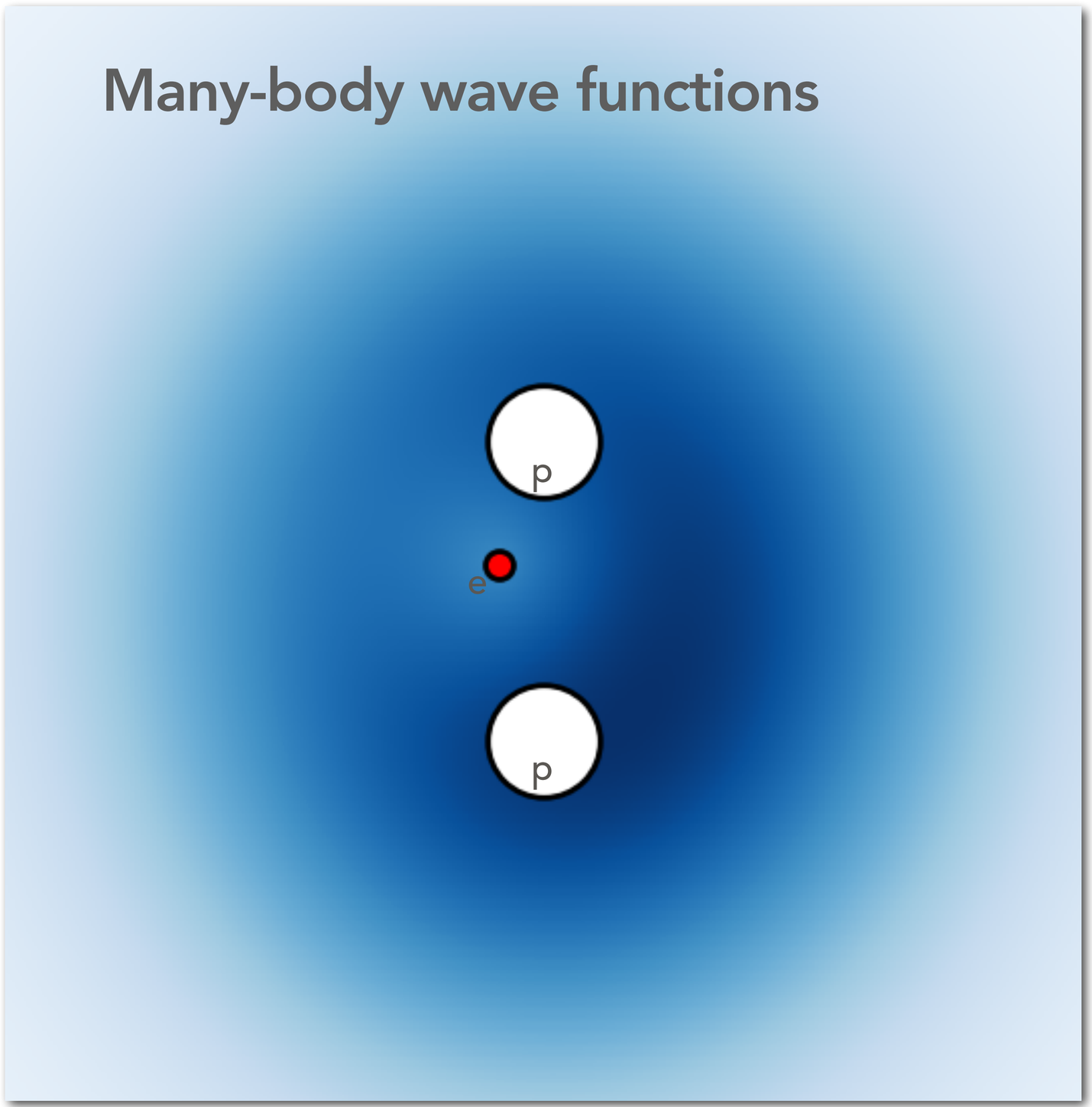
Data to bridge the gap



$$\hat{H} = \sum_{ij} t_{ij} c_i^\dagger c_j + \sum_{ijkl} V_{ijkl} c_i^\dagger c_j^\dagger c_k c_l$$

Effective models: specific, simpler models for a particular material

Many-body wave functions



Two electron and two protons

Conditional wave function: $\Psi(x_1 \,|\, x_2, r_1, r_2)$

Treatment of electron correlations are absolutely required to understand materials

Color	Wavelength (nm)	Frequency (THz)	Photon energy (eV)
violet	380–450	670–790	2.75–3.26
blue	450–485	620–670	2.56–2.75
cyan	485–500	600–620	2.48–2.56
green	500–565	530–600	2.19–2.48
yellow	565–590	510–530	2.10–2.19
orange	590–625	480–510	1.98–2.10
red	625–750	400–480	1.65–1.98

Question: what can we say about the band gap of silicon? (i.e., what is the lowest energy photon that it can absorb?)

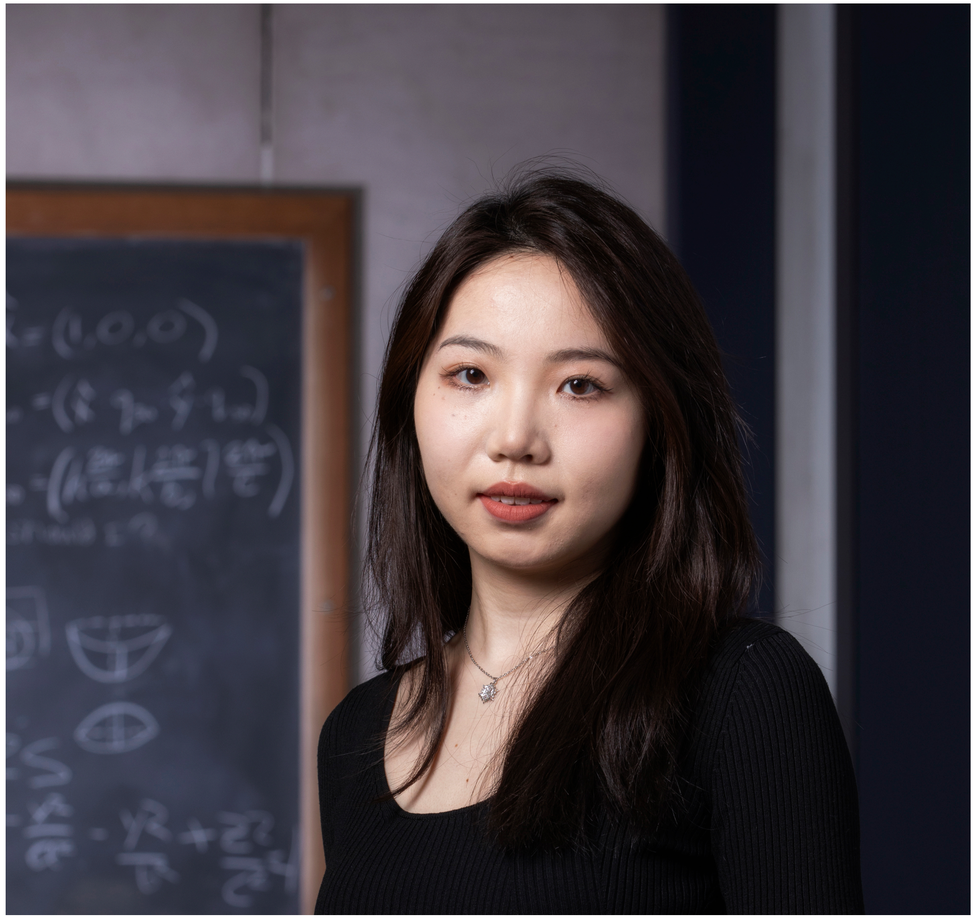
Silicon

Band gap ignoring electron correlations: 5 eV

Band gap in reality: 1.1 eV

Source: O. Madelung, U. Rössler, M. Schulz (ed.) SpringerMaterials Silicon (Si), band structure Landolt-Börnstein - Group III Condensed Matter 41A1b (Group IV Elements, IV-IV and III-V Compounds. Part b - Electronic, Transport, Optical and Other Properties) http://materials.springer.com/lb/docs/sm_lbs_978-3-540-31356-4_432 10.1007/10832182_432 (Springer-Verlag Berlin Heidelberg ©2002) Accessed: 14-01-2016

TA staff



Jia Wang

Will Huie

This class

Website: https://lkwagner.github.io/IntroductionToComputationalPhysics/intro.html

60%: 14 exercises, one per week.
20%: 13 quizzes, one per week (except today).
20%: Final project -- group project

What you will get from this class:

- How to tackle physics problems with computers
- Common approaches to computational problems
- Focuses on the numerical algorithms themselves
- Chance to work on problems with experts in computational physics
- A fantastic stepping stone into research or industry

What this class is not:

- A class on machine learning/artificial intelligence
- A basic CS course
- A class on front-end development (e.g. websites, user-interface, GUI etc)
- A class on high-performance computing

Philosophical interlude

Trainee 1

Trainee 2

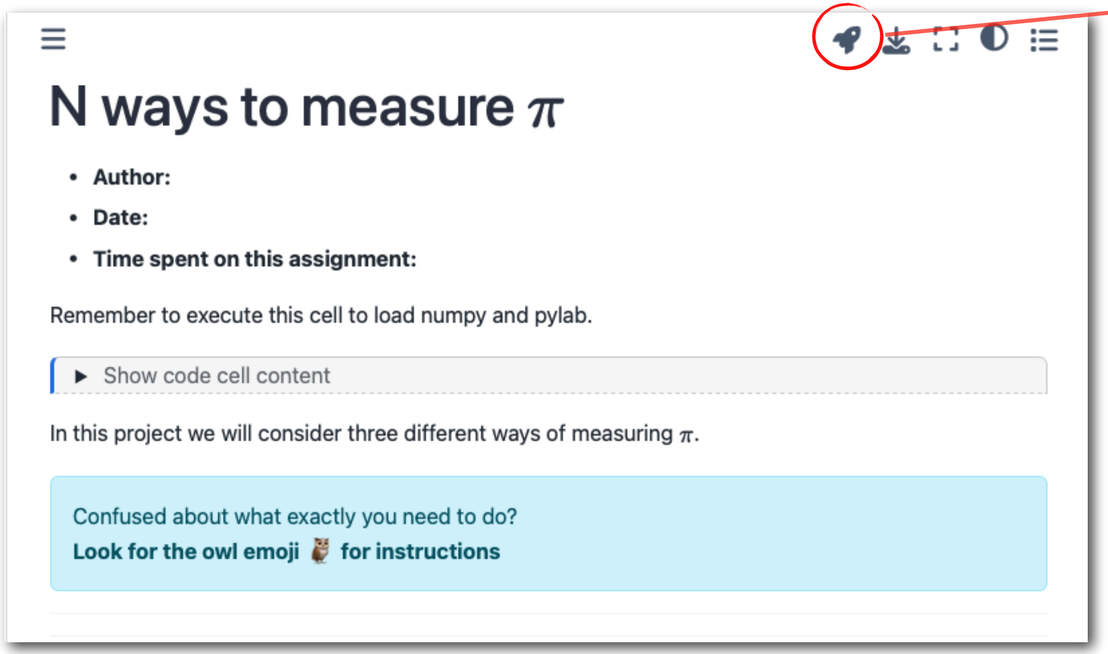
Who is faster at the end?

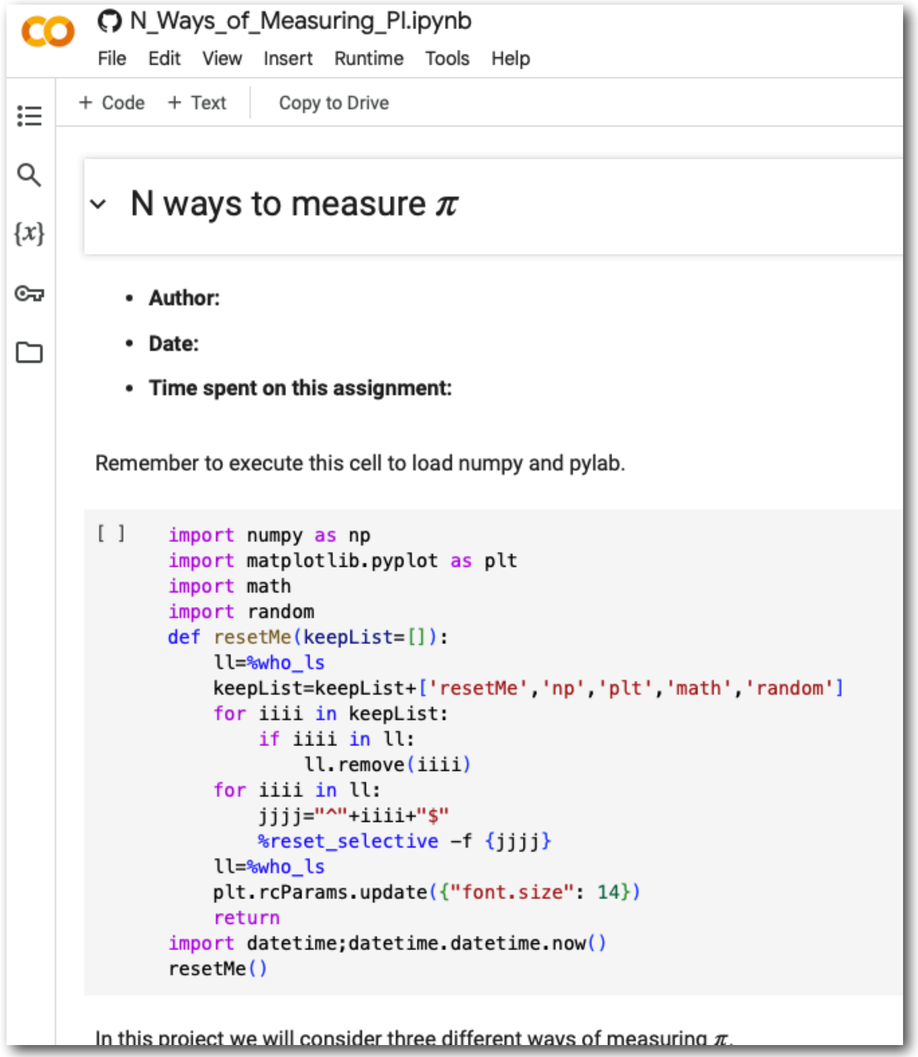
Regarding external help (including AI): You are welcome to discuss the work, google how to do things, etc.

The work you submit must be your own and YOU must have put in the miles for it.

The purpose of the quizzes is to help you finalize your understanding and also to encourage you to put in the miles.

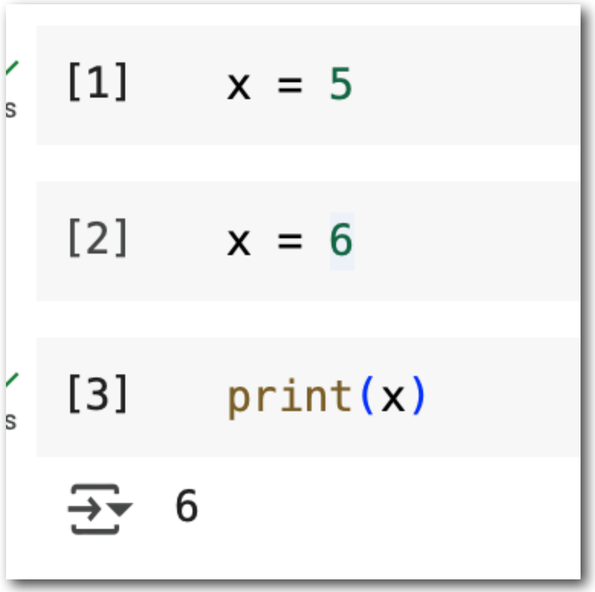
Getting started

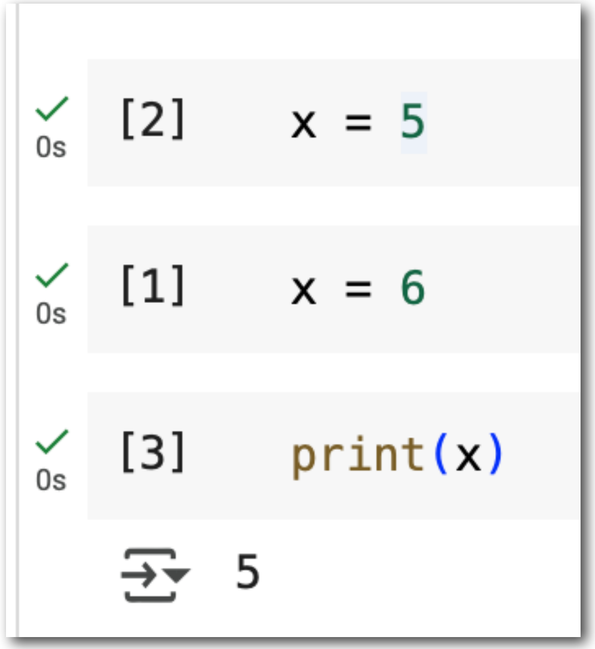




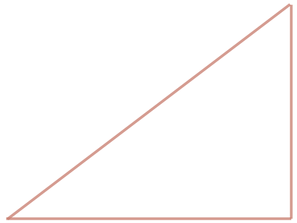
Python notebooks

RUNTIME ENVIRONMENT





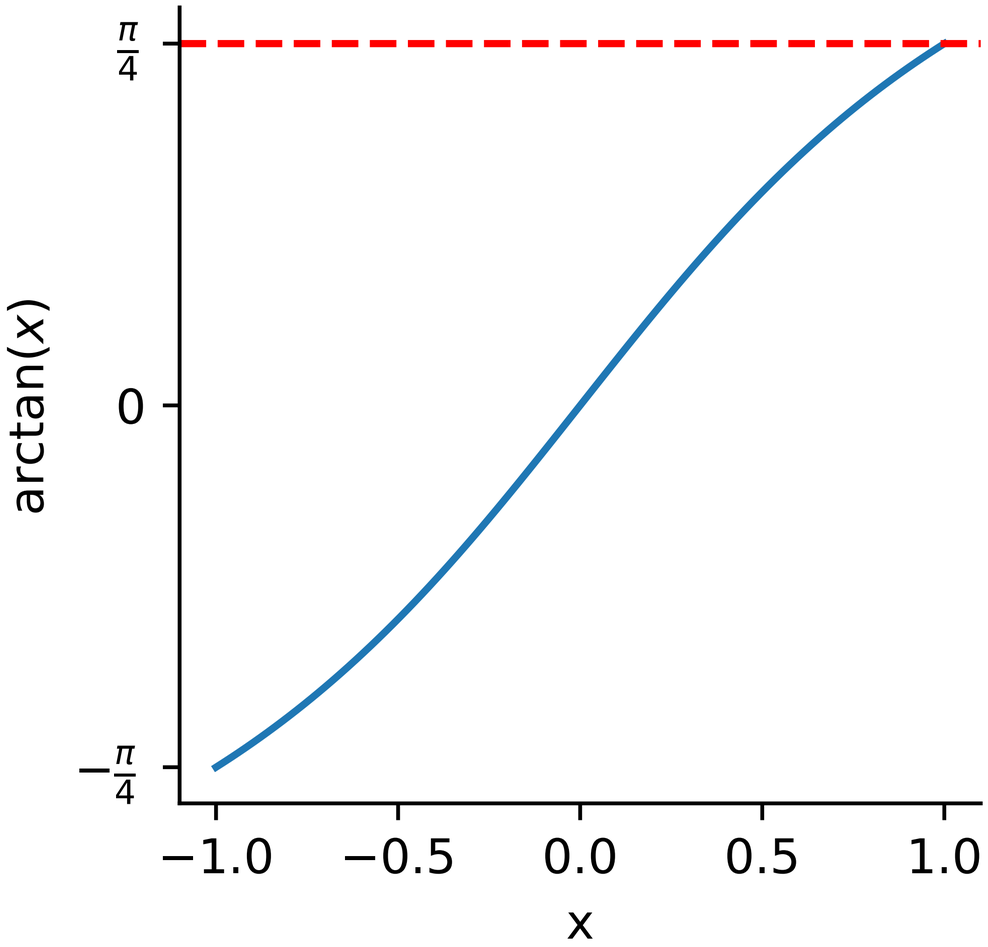
Methods to compute π



Note that $\arctan(1) = \pi/4$

Expand in a Taylor series around 0:

$$\tan^{-1}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1}; -1 < x \leq 1$$



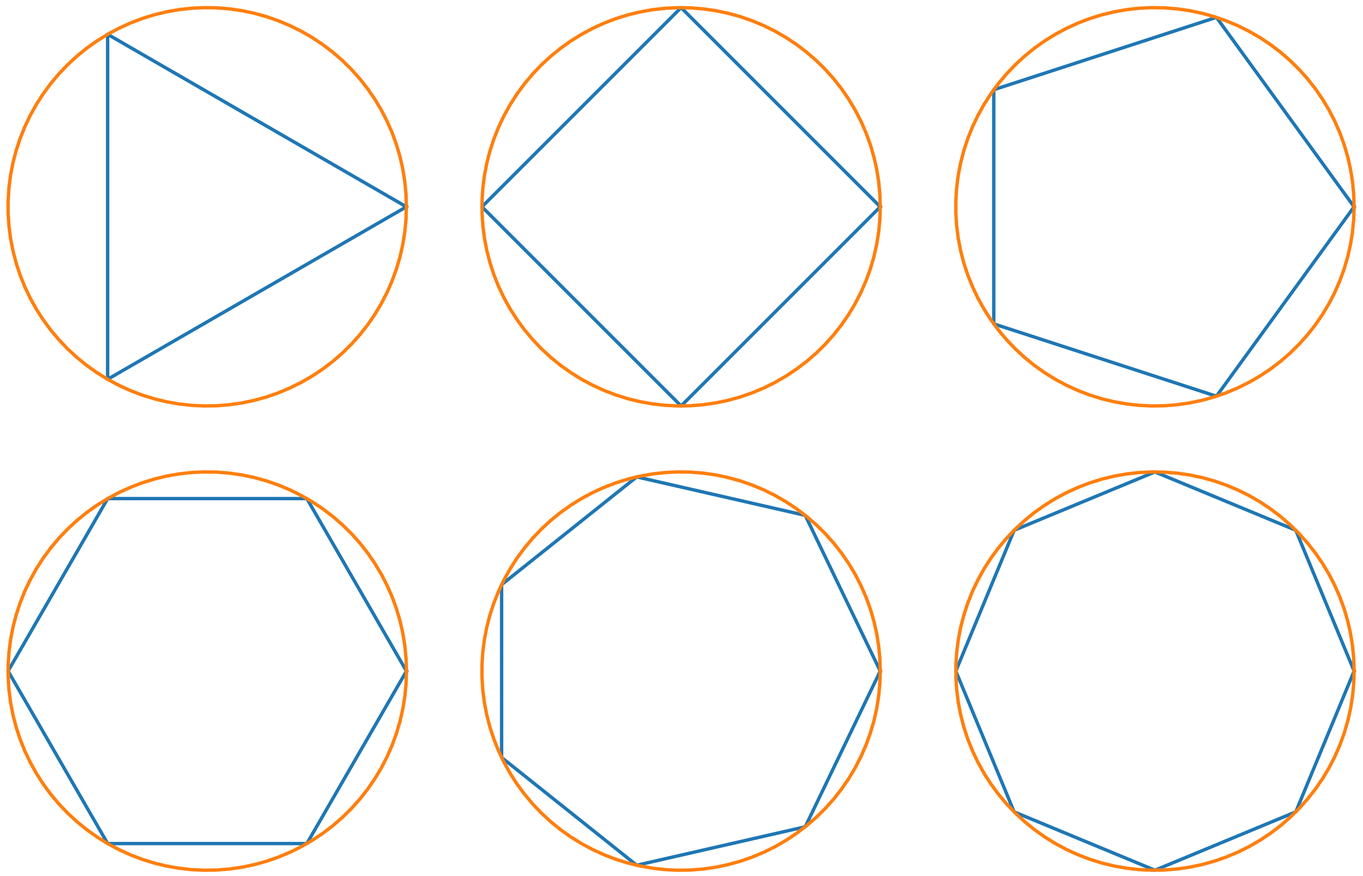
Ramanujan

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

Surprisingly difficult to show!

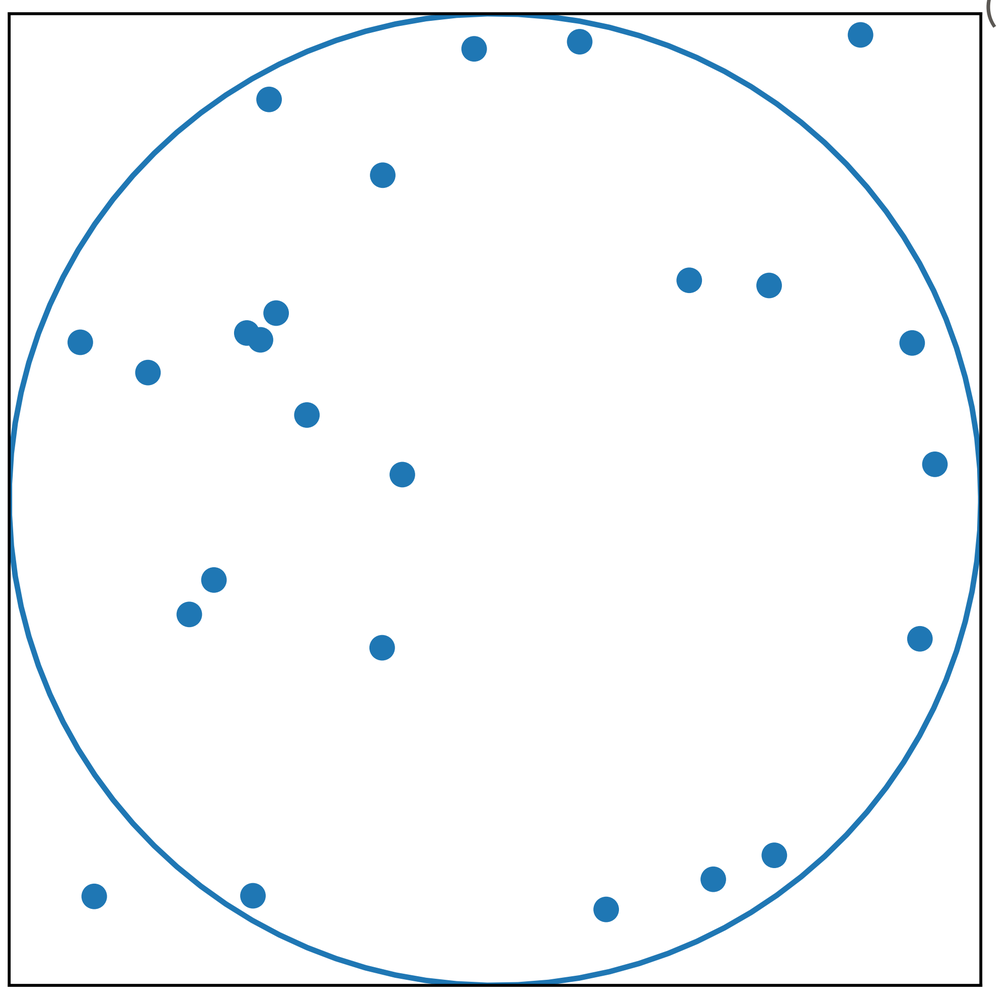
Detailed explanation here:
https://paramanands.blogspot.com/2012/03/modular-equations-and-approximations-to-pi-part-1.html?m=0

Archimedes



The circumference of a polygon of degree N approaches that of a circle as $N \to \infty$.

Monte Carlo (random numbers)



Area of the circle: π
Area of the square: 4

Proportion of random throws that land within the circle: $\pi/4$

Averaging and uncertainty

Consider a random variable x with expectation value $\langle x \rangle$ and standard deviation σ.

Construct a random variable $x_n = \frac{1}{n} \sum_{i=1}^{n} x_i$ as the average of n samples of x.

Central limit theorem:

$x_n \sim N(\langle x \rangle, \sigma_n)$, where $\sigma_n = \frac{\sigma}{\sqrt{n}}$

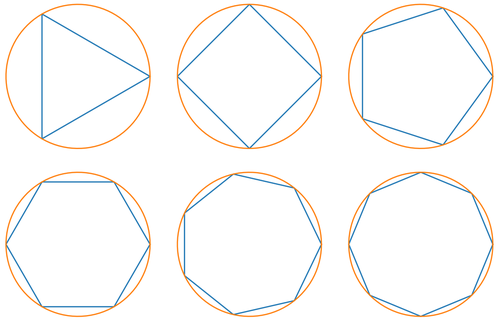
Discussion question

We saw ~four methods to compute π.

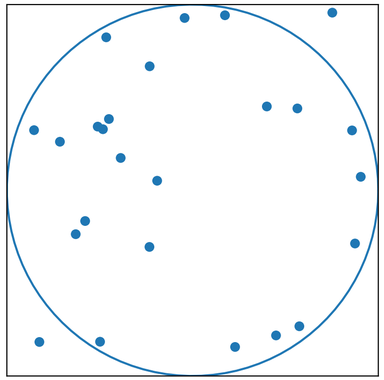
What do you think the pluses and minuses might be for these methods?

How to evaluate them against one another?

$$\arctan(1) = \pi/4$$



$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$



Coding: one style hint

```
x=5
y=6
def f():
  return x+y
z1 = f()
x=6
z2 = f()
```

What's wrong with this?

```
def f(x,y):
  return x+y
z1 = f(x=5,y=6)
z2 = f(x=6,y=6)
```

Functional programming: better style

Coding: performance

```
def func(n):
  return 2*(1/((4*n+3)*(4*n+1) ))
```

```
pilist = [4*func(i) for i in range(1_000_000)]
```

0.55 s

```
pilist = 4*func(np.arange(1_000_000))
```

0.037 s

```
pilist = []
for i in range(1_000_000):
  pilist.append(4*func(i))
```

0.646 s

Submitting your work

- Work is due one week after the class.
- There are 3 "assignments" on Gradescope
  - Upload a PDF -- instructions on web. do not just hit print!
  - Upload the ipynb file
  - Beginning of next class: short quiz on the work. (don't worry; if you do the assignment, then you should be fine)

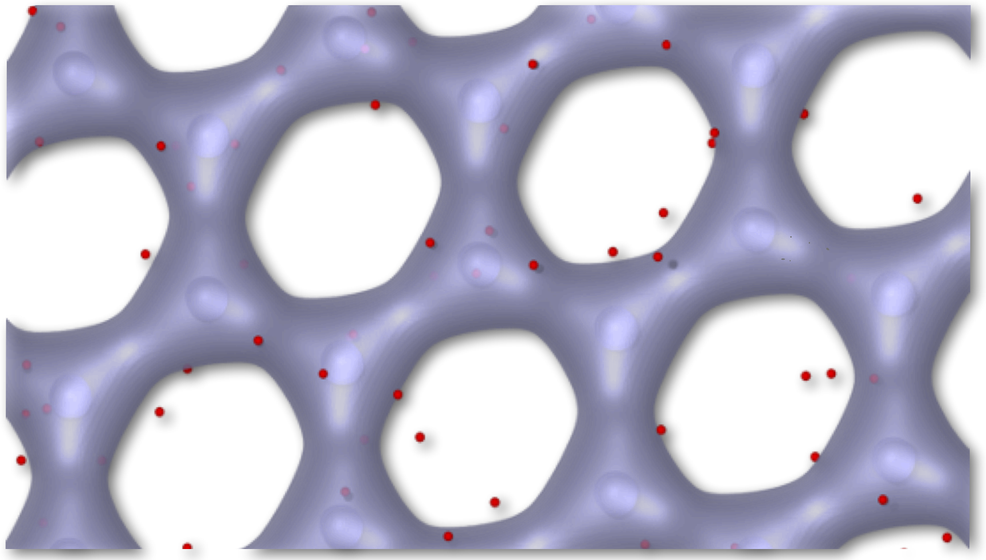
Make sure your work is gradeable -- there shouldn't be pages and pages of output and the answer boxes to the questions should be correctly assigned.

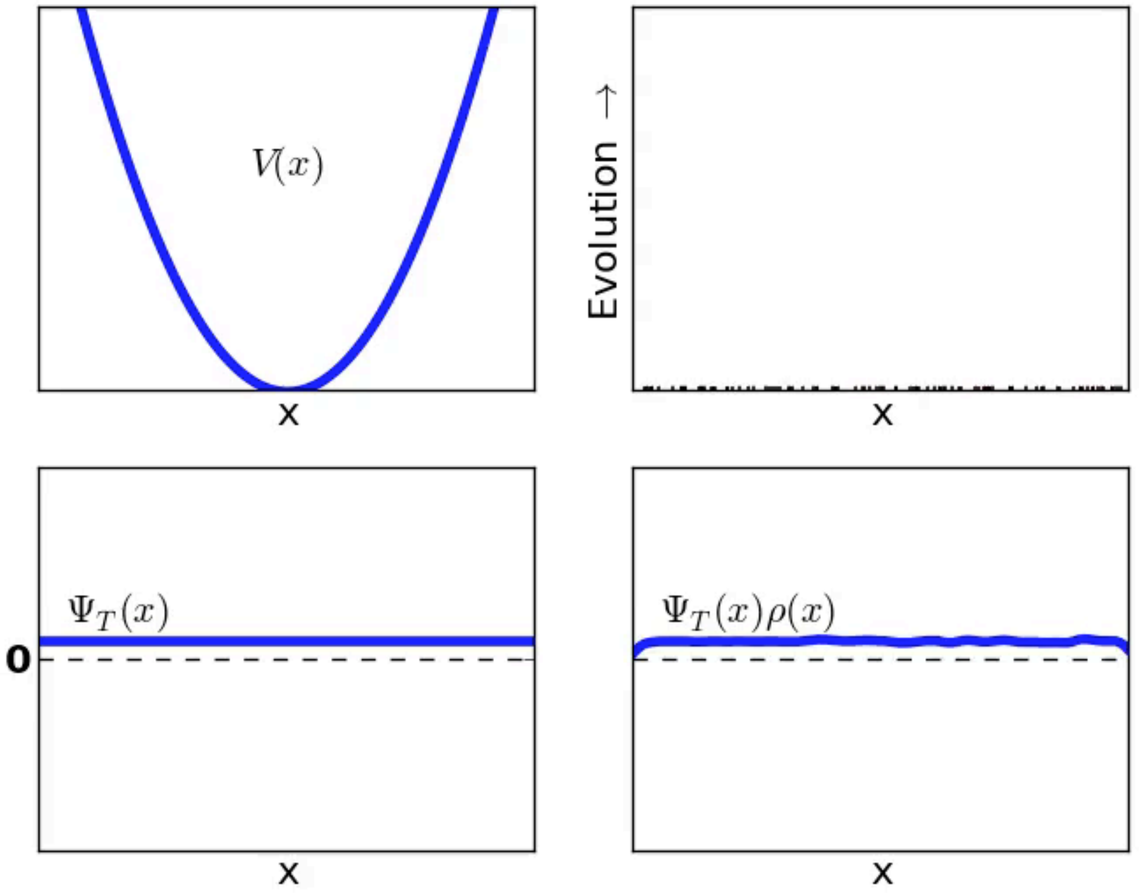
Criteria The heart of this course will be a series of computational assignments. You will work on the assignments both during class and as homework. Homework will be graded on the following criteria:

- 70% Functionality
  - The correct answer is obtained: graphs, code
  - All questions are answered and correctly answered.
- 10% Documented code
  - Well-named variables
  - Function arguments are documented
- 10% Code cleanliness
  - Junk code is removed – we don't need to know about your side quests!
  - Functions are functions – that is, they don't depend on global variables to be set to change their behavior. See below for an example.
- 10% Plot quality
  - labelled axes, colors visible, lines distinguishable, range set correctly
  - Here's a quick tutorial of what is a bad vs good plot
  - For futher resources, see Data Visualization using Matplotlib (and the links within).

```
from requests import get
from socket import gethostname, gethostbyname
ip = gethostbyname(gethostname())
filename = get(f"http://{ip}:9000/api/sessions").json()[0]['name']
print(filename)
from google.colab import files
from google.colab import drive
drive.mount('/content/drive')
#filename = "FILE_NAME_GOES_HERE"
filepath = "/content/drive/MyDrive/Colab Notebooks/" + filename
!cp "$filepath" ./
!jupyter nbconvert --to HTML "$filename"
files.download(filename.replace("ipynb","html"))
```

Efficient description of correlation using Monte Carlo





Represent the wave function as a sum of “walkers”

$$\Psi(x) = \sum_i w_i \delta(x - x_i)$$