PHYS–467 Machine Learning for Physicists
Scientific Programming with NumPy
September 30, 2022

Exercise 1: Vectorized Operations: Writing Optimized Code

1. Generate a 1000×1000 matrix of random integers between 0 (included) and 20 (excluded).
2. Compute the sum of diagonal elements using a `for` loop.
3. Compute the sum of diagonal elements using a NumPy linear algebra function.
4. Compare the running times of 2. and 3. with `%%timeit` (the `%%` syntax allows timing multiple lines).

Hint: use `numpy.random.randint`.

Exercise 2: Fibonacci Numbers with Binet Formula

1. Using NumPy, compute the first 20 numbers of the Fibonacci series F_n ($n = 1, 2, \ldots, 10$) with *Binet formula*

$$F_n = \frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{2^n\,\sqrt{5}}.$$

Exercise 3: Quantum Harmonic Oscillator

In this exercise, you will find the bound states of the 1d quantum harmonic oscillator of mass m by solving the time-independent *Schrödinger equation* numerically. This equation is an eigenvalue problem

$$H\psi_n = E_n\psi_n$$

where $H = K + V = -\frac{\hbar^2}{2m}\partial_x^2 + \frac{1}{2}m\omega^2 x^2$ is the Hamiltonian operator, ψ_n is the wavefunction (eigenfunction), and E_n the energy level (eigenvalue). In the following, set $\hbar = m = \omega = 1$.

1. **Space discretization**. Consider the interval $[-10, 10]$ and discretize it generating a mesh of uniform spacing $\Delta x = 0.01$ with N points, i.e., $x_i = -10 + i\,\Delta x$ ($i = 0, 1, \ldots, N-1$). In this space, the wavefunction ψ is a vector with components $(\psi)_i = \psi(x_i)$.
2. **Hamiltonian discretization**. Using this discretization, the Hamiltonian becomes a $N \times N$ matrix. In particular, the kinetic energy term is

$$\hat{K} = -\frac{1}{2\,\Delta x^2}\begin{pmatrix} -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix},$$

   where we used the finite-difference approximation of the second derivative, and the potential energy term is

$$\hat{V} = \frac{1}{2}\begin{pmatrix} V(x_0) & 0 & 0 & 0 & \cdots & 0 \\ 0 & V(x_1) & 0 & 0 & \cdots & 0 \\ 0 & 0 & V(x_2) & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & V(x_{N-2}) & 0 \\ 0 & 0 & \cdots & 0 & 0 & V(x_{N-1}) \end{pmatrix}.$$

   Generate the Hamiltonian matrix $\hat{H} = \hat{K} + \hat{V}$.

3. **Diagonalization**. Diagonalize $\hat{H}$ using `numpy.linalg.eig` and `numpy.linalg.eigh`, which is specific for symmetric matrices. Use `%%timeit` to compare running times. Check that the eigenvalues are sorted in increasing order, and if not, do so.
4. Normalize the wavefunctions (eigenfunctions) of $\hat{H}$ computing the normalization constant

$$Z = \sqrt{\int_{-10}^{10} \psi^*(x)\psi(x)dx} \approx \sqrt{\sum_{i=0}^{N-1} \psi^*(x_i)\psi(x_i)\Delta x}.$$

5. Plot the first 10 (normalized) wavefunctions and print the values of the corresponding energy levels (eigenvalues[1]). Compare with the theoretical predictions $E_n = (n + \frac{1}{2})$.

Hint: use `numpy.arange`, `numpy.diag`, `numpy.argsort`, `matplotlib.pyplot.plot`.

Exercise 4: Eigenvalue Statistics of Random Matrices

In this exercise, you will study the distribution of the spectrum of certain random matrices. Random matrix theory is a branch of mathematics with numerous applications in, e.g., physics, machine learning, and finance. The simplest random matrix is the Wigner one

$$W = \frac{1}{\sqrt{2N}}(A + A^\top),$$

where A is a random matrix of size N with i.i.d. elements distributed from a standard Gaussian, i.e., $A_{ij} \sim \mathcal{N}(0, 1)$.

1. Generate a Wigner matrix with $N = 1000$.
2. Find the eigenvalues λ_n $(n = 1, 2, \ldots, N)$ and sort them in increasing order.
3. Plot the probability density of a single eigenvalue $\rho_W(\lambda)$ and compare it to the Wigner semi-circle law

$$\rho_W(\lambda) = \begin{cases} \frac{1}{2\pi}\sqrt{4 - \lambda^2}, & \text{if } |\lambda| < 2 \\ 0, & \text{otherwise} \end{cases}$$

Hint: use `matplotlib.pyplot.hist`.

[1] Be careful that you renormalized the eigenfunctions and therefore should update the eigenvalues accordingly!