Computational Methods in Many-Body Physics | Tutorial 0
Prof. M. Knap, Prof. F. Pollmann | Summer Term 2023
A. Romen, W. Kadow, R. Morral Yepes, P. Zechmann | 2023-04-21

Note: The purpose of this exercise sheet is to give a brief review over the basic and necessary functions in python. Please make sure you can solve all the problems.
Note: This exercise sheet is not counted for the bonus. Therefore, you do not need to show or send the solutions.

Exercise 0.1: Plotting

In this exercise, you will get familiar with *numpy.array* and *matplotlib* package.

Plotting the function $f(x) = x^2$ for $x \in [-1, 1]$.

a) Create a numpy array X of evenly distributed points in $[-1, 1]$. For example, $X = [-1., -0.9, -0.8, \ldots, 1.]$

b) Create a numpy array Y of $f(x_i)$ with point x_i from the array X we created.

c) Plot the function. *Hint: matplotlib.pyplot.plot*

Now, we repeat the same procedure but considering $f(x) = e^{-x}$ for $x \in [0, 10]$.

d) Repeat step (a)-(c)

e) Plot now the Y in log scale. *Hint: matplotlib.pyplot.semilogy*

Suppose in additional to the data Y, we also have the standard deviation ΔY to the data resulting from some random error,

f) Create a numpy array ΔY and put in some small random numbers. *Hint: numpy.random.rand*

g) Repeat the steps before, but plot the errorbar ΔY with the data Y.
Hint: matplotlib.pyplot.errorbar

Exercise 0.2: Numerical linear algebra methods

In this exercise, you will get familiar with *numpy.array* and *numpy.linalg* package.

Consider the Heisenberg model of two spins,

$$H = \sum_{\alpha=x,y,z} \sigma_1^\alpha \sigma_2^\alpha = \sigma_1^x \otimes \sigma_2^x + \sigma_1^y \otimes \sigma_2^y + \sigma_1^z \otimes \sigma_2^z$$

where the Pauli matrices are given as

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

a) Construct the Pauli matrices as numpy array.

b) Construct the Heisenberg model as numpy array.

c) Diagonalize the Hamiltonian. What are the eigenstates? *Hint: numpy.linalg.eigh*