Exercise 06

SUBMIT YOUR FILES BEFORE NEXT FRIDAY AT 8.00 AM TO oliver.lemke@fu-berlin.de

1 Basis set expansion

Consider a function

$$f(x) = N \cdot \exp\left(-\frac{a}{2}x^2\right) \cdot \left(\frac{m\omega}{\hbar\pi}\right)^{\frac{1}{4}} \cdot \left((3x)^3 - 1\right) \tag{1}$$

with

$$a = \sqrt{\frac{m\omega}{\hbar}} \tag{2}$$

1. Plot the function
2. Calculate the normalization coefficient N (numerically) and normalize the function (Hint: Use the integrator: scipy.integrate.trapz()).
3. Now consider that the function can be approximated by a linear combination of functions:

$$f(x) = \sum_{i=1}^{\infty} c_i \chi_i(x) \tag{3}$$

where $\{\chi_i(x)\}$ denotes an orthonormal basis set:

$$< \chi_i \mid \chi_j >= \delta_{ij} \tag{4}$$

In the following task you can see that the definition and number of the basis functions strongly influence the quality of the approximation.

(a) One of the easiest thinkable basis sets that can be applied are step functions, which are defined as:

$$\chi_i(x) = \begin{cases} 0 & x \notin S_i \\ N_i & x \in S_i \end{cases} \qquad \text{with } N_i = \frac{1}{\frac{(x_{max} - x_{min})}{n}} \tag{5}$$

To achieve such step functions the position x is divided into n equally-sized bins $\{S_1, S_2, ..., S_n\}$, which creates n stepfuntions $\{\chi_1, \chi_2, ..., \chi_n\}$.

i. Convince yourself that the basis functions are orthonormal (Use *Python*).

ii. Write a *Python* script that calculates $f(x)$ as a superposition of 10, 20 and 100 basis functions (Hint: To do this you have first to calculate the coefficients c_i). Use the parameters

$$m = 1 \qquad \hbar = 1 \qquad \omega = 1 \qquad x = [-5, 5] \qquad \Delta x = 0.01$$

iii. Calculate

$$\sum_{i=1}^{n} c_i^2 \tag{6}$$

Is the approximated function normalized?

iv. Plot the orignal $f(x)$ and the approximated $f(x)$. Fill the space between both curves (Hint: Use the command plt.fill_between()). The space between the two curves is a measure for the approximation error, i.e. the error you make by representing $f(x)$ using a finite basis set. Calculate the error as:

$$\Delta = \int_{x_{min}}^{x_{max}} \left| \left[\sum_{i=0}^{n} c_i \chi_i(x) \right] - f(x) \right| dx \tag{7}$$

Plot the error as a function of the number of basis functions.

(b) Now consider the eigenfunctions of a harmonic oscillator which are defined as:

$$\chi_i(x) = \phi_i(x) = N_i \cdot H_i \cdot \exp\left(-\frac{a}{2}x^2\right) \tag{8}$$

with

$$N_i = \frac{1}{\sqrt{2^i i!}} \cdot \left(\frac{\omega m}{\pi \hbar}\right)^{\left(\frac{1}{4}\right)} \tag{9}$$

and a as defined in equation 2. H_i denotes the Hermite polynomials (Accesible in *Python* via numpy.polynomial.hermite.hermval()). Use your *Python* script from part (a) and calculate $f(x)$ as a super position of the first 2, 3 and 5 harmonic oscillator wave functions using the same parameters. Repeat the task of (a) and add the following informations:

i. Consider the Hamiltonian of this function as:

$$\hat{H} = \hat{T} + \hat{V} = \frac{\hat{p}^2}{2m} + \frac{1}{2} m \omega^2 \hat{x}^2 \tag{10}$$

Calculate the expectation value of the energy

$$\langle E \rangle = \left\langle f(x) \left| \hat{H} \right| f(x) \right\rangle \approx \left\langle \sum_{i=0}^{n} c_i \chi_i(x) \left| \hat{H} \right| \sum_{i=0}^{n} c_i \chi_i(x) \right\rangle \tag{11}$$

A. for the original function $f(x)$ (Calculate the second derivative of $f(x)$ with pen and paper)

B. for the superposition

C. out of the coefficients c_i (Hint: use the energy eigenvalues of the harmonic oscillator)

and compare.

ii. Plot the first 5 Hermite ploynomials