Problem Set 1

Debendro Mookerjee
GitHub Account Name: Debendro-dm5790

September 10, 2023

Abstract

This document gives a personal introduction and outlines my goals and future plans. This is followed by a captioned image of a Gaussian distribution function generated solely from Python's numpy and matplotlib packages.

1 Introduction

I wish to improve my computational skills by learning how to perform Monte Carlo simulations and Fourier analysis on Python. I also am curious to learn about the currently unknown topics reserved for the December lectures. I have had some introduction to Python and its applications to many-body systems while doing my Master's at Rutgers University in New Brunswick, New Jersey, where I learned how to use the Density Matrix Renormalization Group (DMRG) algorithm provided in Python's TeNPy library to study the zero-temperature phase diagrams of the Bose-Hubbard and Majorana-Bose-Hubbard models; however, from a coding perspective I only learned and used for loops, if statements, matplotlib, and TeNPy statements to study the phase diagrams. I hope to get a postdoc in soft matter theory after completing my doctoral studies at New York University.

2 Methods

The code file "plot_gaussian.py" creates and plots a normalized Gaussian with mean 0, which is stored in the variable mu, and standard deviation 3, which is stored in the variable sigma, over the range [-10,10]. The plot is saved as an .png file called "gaussian.png." A variable called dataList stores an array of 1000 numbers that range over $[-10, 10]$. This array is generated by using numpy's linspace function. Feeding dataList into the formula for a Gaussian distribution yields another array of 1000 elements which is stored in a variable called DoP, short for "density of probablity." These two arrays are plotted by using the plot function from the matplotlib package.

2.1 Formulation of the problem

The relationship between the arrays dataList and DoP and the constant "variables" mu and sigma can be summarized as follows:

$$DoP = \frac{1}{sigma \times \sqrt{2\pi}} e^{-\frac{1}{2}(dataList-mu)^2/sigma^2}$$

2.2 Computational methods

We used the standard matplotlib.pyplot functions such as xlabel(), ylabel(), title(), show(), and savefig() functions.

3 Results

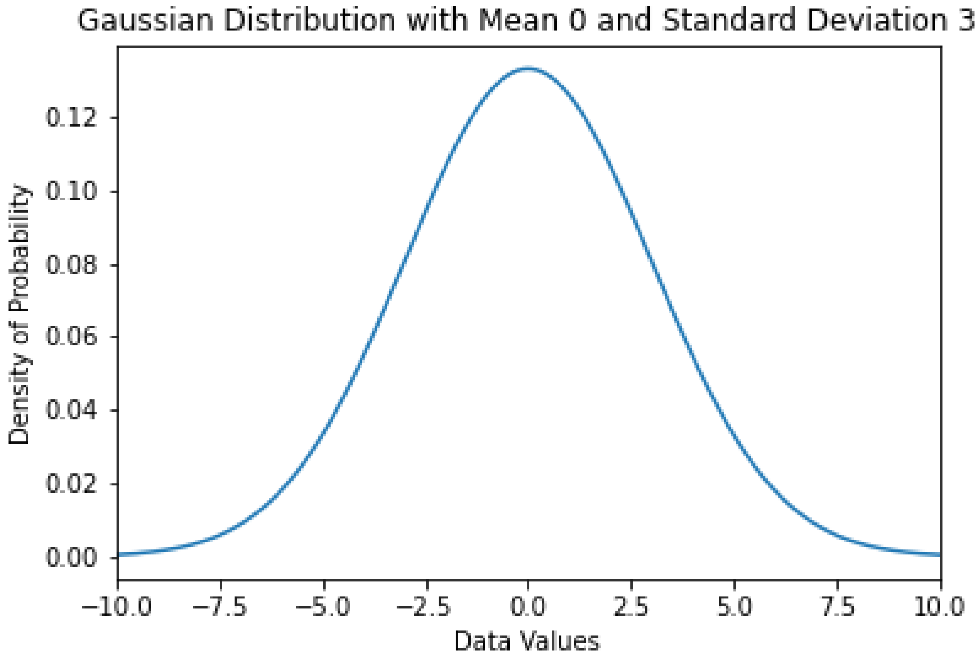

Figure 1: A normalized Gaussian distribution with mean 0 and standard deviation 3