Introduction to Computational Quantum Mechanics: Application-based Learning with Python

Braden M. Weight
Department of Physics and Astronomy
University of Rochester

March 9, 2023

Contents

Chapter 1

Python and Environments

No assignment here.

Chapter 2

Types of Variables in Python

1. Write a program that to evaluate e^x with $x = -5.5$ using a Taylor series,

$$e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^2 + \mathcal{O}(x^4), \tag{2.1}$$

for orders 0, 1, 2, 3, 4, 5, 10, and 100. Note that $e^x \approx 1 + x$ is the first-order expansion. How quickly does the result converge to the exact result ? At what order, can you achieve an accuracy of 1% or better (% Error = $(x_{\text{Approx}} - x_{\text{exact}})/x_{\text{exact}})$) ? Make a plot of the result with respect to the expansion order. For plotting, see example below.

2. The $N \times N$ discrete Fourier transform matrix is defined by,

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \gamma & \gamma^2 & \gamma^3 & \cdots & \gamma^{N-1} \\ 1 & \gamma^2 & \gamma^4 & \gamma^6 & \cdots & \gamma^{2(N-1)} \\ 1 & \gamma^3 & \gamma^6 & \gamma^9 & \cdots & \gamma^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma^{N-1} & \gamma^{2(N-1)} & \gamma^{3(N-1)} & \cdots & \gamma^{(N-1)(N-1)} \end{bmatrix}, \tag{2.2}$$

where $\gamma = e^{-2\pi i/N}$.

(a) Use Numpy to construct the 4×4 matrix and save to a file using the 'np.savetxt()' function.

(b) Write a function to accept an integer N and output the corresponding $N \times N$ array for the discrete Fourier transform.

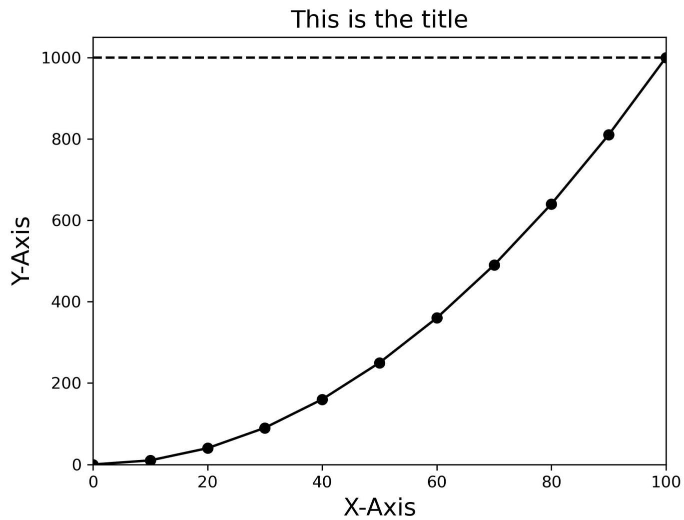

Figure 2.1: Generated from the code “example_plot.py”.

```python
# This is an example for plotting a quadratic function
import numpy as np
import matplotlib
matplotlib.use('Agg') # This is required for the NDSU cluster...not sure why
from matplotlib import pyplot as plt

X = np.arange(0,100+10,10)
Y = 0.1 * X**2
plt.plot(X,Y,"-o",c="black",label="Y(x) = $\frac{X^2}{4}$")
plt.plot(X,np.ones(len(X))*Y[-1],"--",c="black",label="Y(x) = $\frac{X^2}{4}$")
plt.xlim(X[0],X[-1])
plt.ylim(0)
plt.xlabel("X-Axis",fontsize=15)
plt.ylabel("Y-Axis",fontsize=15)
plt.title("This is the title",fontsize=15)
plt.savefig("example_plot.jpg", dpi=300)
```