# Computational Physics / PHYS-GA 2000 / Problem Set #1

Zak Saeed
(GitHub: Zakariah-S)

September 2024

**Abstract**

I created a public GitHub repository with the name "phys-ga2000." Working in a subdirectory of the repository titled "ps-1," I created a Python file named "plot gaussian.py" that plots and saves the image of a normalised Gaussian function with standard deviation $\sigma = 3$. The image and the two files were pushed to GitHub. Finally, a small report including both the image and a brief description of my background and goals with respect to computational physics was created and added to the GitHub repository.

## Introduction (to me!)

It's nice to meet you! I'm Zak, a BSc Physics graduate who studied at New York University Abu Dhabi. While I was there, I worked (and, to an extent, am still working) with a professor on data received from radio- and X-ray observations of an active galactic nucleus (AGN). In pursuit of this research, I learned to use Python along with the libraries NumPy, Matplotlib, SciPy, and AstroPy. I also learned to work with the Zsh and Bash shells. Working in Python, I compiled data from the observations and used it to estimate the mass of the black hole and the mass of the disc. With regard to the radio data, this involved both fitting curves to data and implementing an algorithm to find groups of data points that obeyed a Keplerian curve. For the X-ray data, I implemented the Voronoi-binning algorithm to identify and analyse regions with significant diffuse emission. My main goal for this course is to eventually challenge myself to develop well-constructed, sophisticated projects in physics (perhaps outside of the bounds of what the course requires!). Specifically, I would love to use what I learn to create educational tools for people who want to learn physics. I don't currently have a set plan for after my degree is finished, though I know that I want at least some part of my life afterward to involve helping other people learn more about the subject.

## The Gaussian function

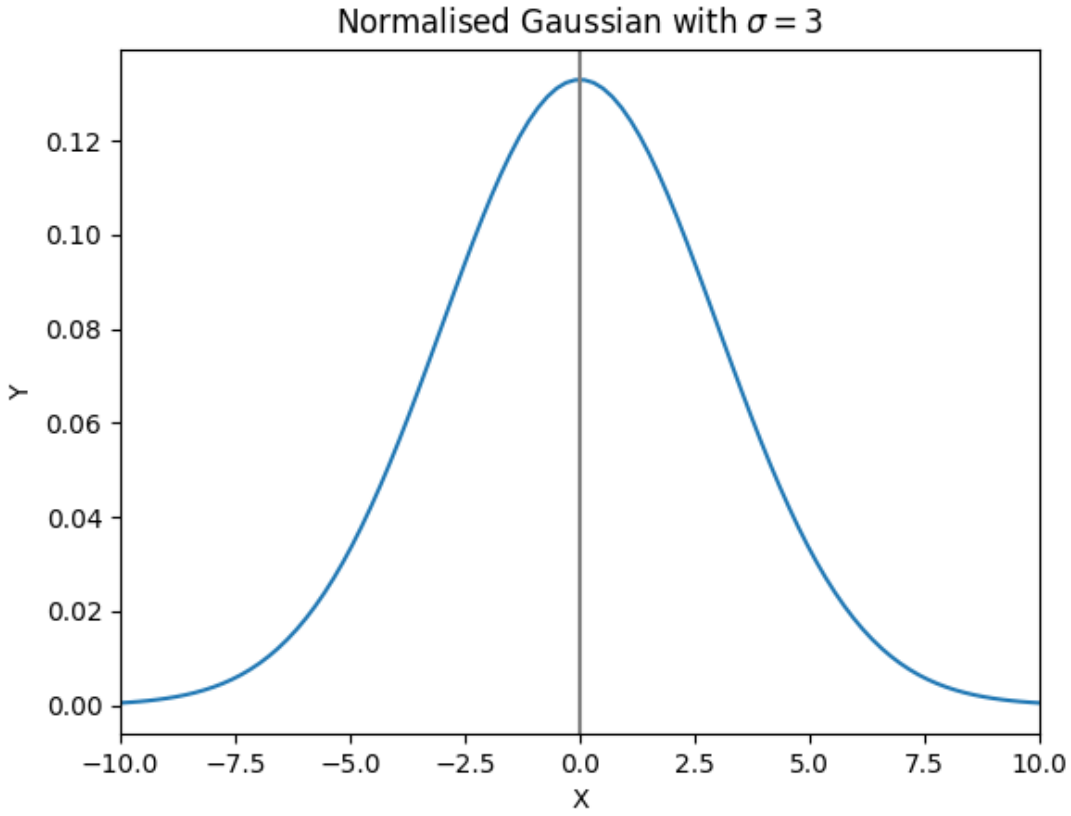


Figure 1: I used np.linspace() to generate an array of 100 equally-spaced values in the interval [-10, 10]. I then created a function that, for a given standard deviation, takes an array of floats X = $[x_0...x_{100}]$ and returns a new array Y = $[P(x_0)...P(x_{100})]$. Plotting X and Y using matplotlib.pyplot's plot() function, and setting titles and axis limits accordingly, returned the plot seen here. GitHub account with the code: Zakariah-S.