Part 1: Mathematical Preliminaries

Questions and Solutions:

1. Consider the following sinusoid:

$$x(t) = 4\cos(2t)$$

   a) What is the amplitude of $x(t)$?
   b) What is the frequency of $x(t)$?
   c) What is the phase of $x(t)$?
   d) Is $x(t)$ an even or an odd function?
   e) Draw $x(t)$ on the interval $[0, 2\pi]$.

Solution:

a) The amplitude is **4** m.

b) The frequency is $1/\pi$ radians/s.

c) The phase is **0** radians.

d) The function is **even** since cosine is even.

e) See Figure 1 below.

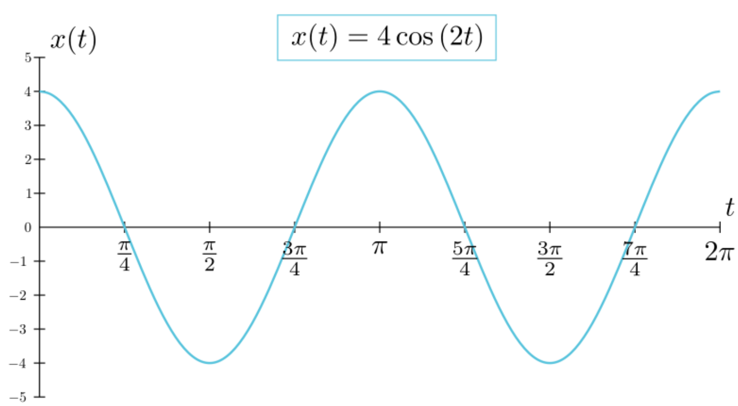

Figure 1: Solution to Part 1, Question 1e.

2. Let's practice some manipulations with $z_1 = 2 + 3i$ and $z_2 = -1 - 5i$.
   a) Compute the sum and differences $z_1 + z_2$ and $z_1 - z_2$.
   b) Compute $z_1 \cdot z_2$

Solution:

a) $z_1 + z_2 = 1 - 2i$ and $z_1 - z_2 = 3 + 8i$

b) $z_1 \cdot z_2 = 13 - 13i$

3. Again, we will practice with $z_1 = 2 + 3i$ and $z_2 = -1 - 5i$.
   a) Compute z_1^* and z_2^*
   b) Compute $z_1 \cdot z_1^*$

Solution:

a) $z_1^* = 2 - 3i$ and $z_2^* = -1 + 5i$

b) $z_1 \cdot z_1^* = 13$

4. Using $e^{ix} = \cos(x) = i\sin(x)$ show that...
   a) $e^{i\pi} = -1$
   b) $e^{-ix} = \cos(x) - i\sin(x)$
   c) $\frac{1}{2}(e^{ix} + e^{-ix}) = \cos(x)$

Solution:

a) $e^{i\pi} = \cos(\pi) + i\sin(\pi)$
$e^{i\pi} = -1 + i \cdot 0$
$e^{i\pi} = -1$

b) $e^{-ix} = \cos(-x) + i\sin(-x)$
Since cosine is even, $\cos(-x) = \cos(x)$
Since sine is odd, $\sin(-x) = -\sin(x)$
$e^{-ix} = \cos(x) - i\sin(x)$

c) $\frac{1}{2}[e^{ix} + e^{-ix}] = \frac{1}{2}[(\cos(x) + i\sin(x)) + (\cos(x) - i\sin(x))]$

$\frac{1}{2}[e^{ix} + e^{-ix}] = \frac{1}{2}[2\cos(x) + 0]$

$\frac{1}{2}[e^{ix} + e^{-ix}] = \cos(x)$

5. Consider the function $f(x, y) = y\sin(x)$.
   a) Using the above example as a guide, try to compute $\int_0^\pi f(x, y)\,\mathrm{d}y$?
   b) What do you observe? Is the solution a function of x or y?
   c) Now suppose that you are given some new function $h(\omega, t) = e^{i\omega t} f(\omega)$. If we integrate h with respect to t, we will get a new function $\hat{h}$. Will $\hat{h}$ be a function of ω or t?

Solution:

a) $\int_0^\pi f(x, y)\,\mathrm{d}y = \sin(x)\int_0^\pi y\,\mathrm{d}y$

$\int_0^\pi f(x, y)\,\mathrm{d}y = \frac{1}{2}\sin(x)y^2\Big|_0^\pi$

$\int_0^\pi f(x, y)\,\mathrm{d}y = \boxed{\frac{1}{2}\sin(x)\pi^2}$

b) The solution is a function of x.

c) The function $\hat{h}$ will be a function of ω.

6. The FT of a function $f(t)$ is given by

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} e^{i\omega t} f(t)\, \mathrm{d}t$$

   a) Using the results of Question 5, justify the fact that the left-hand-side of the Fourier Transform equation is *not* a function of time, t.
   b) Let's get back to the idea of FT-NMR. Ultimately, our goal is to understand how we transform our FID signal from the 'time domain' to the 'frequency domain'. As a first step, let's label the following inputs and outputs on the Fourier Transform equation: frequency, time, complex exponential, FID (time-domain), frequency spectrum.

Solution:

a) Since the Fourier transform takes the integral of $e^{i\omega t} f(t)$ with respect to t, the result will be a function of ω.

b) See Figure 2 below.

$$\underset{\text{Frequency Spectrum}}{\boxed{\hat{f}(\omega)}} = \int_{-\infty}^{\infty} \underbrace{e^{i\overset{\text{Frequency}}{\omega}\overset{\text{Time}}{t}}}_{\text{Complex Exponential}} \underset{\text{FID (time-domain)}}{\boxed{f(t)}}\, \mathrm{d}t$$

Figure 2: Solution to Part 1, Question 6b.

Part 2: Google Colab

There are no questions associated with Part 2 of the pre-lab.

Part 3: Python Fundamentals

Questions and Solutions:

4. Create a new cell by clicking the "+ Code" button at the top, then enter `np.power(2,3)` and, after pressing play, the answer should be displayed below the code block.
   a) What does `np.power` do?
   b) Does changing the order of the arguments (the numbers in parenthesis) change the output? If so, how does the position of the arguments relate to the result of the mathematical operation being calculated?

Solution:

a) `np.power` raises the first argument to the power of the second.

b) Yes, in this case, changing the order changes the output. Since `np.power(a,b)` represents a^b, changing the order would result in b^a which is often not the same value.

5. Use `numpy` to calculate the cosine (cos) of π, as well as the sine (sin) of $8\pi/9$.
   a) What code did you use to caluclate the cosine of π?
   b) What value did you get for the sine of $8\pi/9$?

Solution:

a) `print(np.cos(np.pi))`

b) $\sin(\frac{8\pi}{9}) = 0.34202\ldots$

7. Practice defining variables:
   a) Write code to assign the value 0.1 to a variable named `dt`.
   b) Variables can represent things other than numbers. Write code which stores your time array from before as `time_array`, and use the `dt` value from part a as your `STEP`.

Solution:

a) `dt = 0.1`

b) `time_array = np.arange(0, 2, dt)`

10. Using the following for-loop,

```
for i in range(0, 10, 2):
    print(i)
```

   a) What do each of the three values of the `range` function represent?

Solution:

a) The first argument represents the value to *start* at; the second value is the value to *stop* before reaching; and the final value is the *step-size*.

11. Using for-loops to implement Σ-sums:
   a) Write code to implement the following Σ-sum and show that the answer is 99.

$$\sum_{k=0}^{10}(2k-1) = 99$$

Solution:

```
total = 0
for i in range(0, 11, 1):
    total = total + (2 * k - 1)
print(total)
```

Part 4: Plotting a Cosine Wave

Questions and Solutions:

1. Using the cosine wave $x(t) = 4\cos(2t)$ fill in the following variables:

```
start = 0
duration = 2 * np.pi
dt = 0.1
amplitude = ?
frequency = ?
phase = ?
```

a) Use `np.arange` as before to create the `time_array` for this cosine wave.

b) In python, if you include an array in a mathematical operation, that operation is applied to all values of an array. Since our function takes the cosine of $2t$, multiply the time array by 2, then store this variable as `updated_time_array`.

c) Now take the cosine of all the time values and store that as a variable named `cosine_wave`.

d) Finally, multiply the entire cosine wave by 4.

Solution:

```
start = 0
duration = 2 * np.pi
dt = 0.1
amplitude = 4
frequency = np.pi
phase = 0
```

a) `time_array = np.arange(start, duration, dt)`

b) `updated_time_array = 2 * time_array`

c) `cosine_wave = np.cos(updated_time_array)`

d) `cosine_wave = 4 * cosine_wave`

2. Plot the time array and the cosine wave.

Solution:

```
plt.plot(time_array, cosine_wave)
plt.show()
```

The wave should match exactly the one from above, if everything was defined properly. (See Figure 3 for the exact image which is generated).

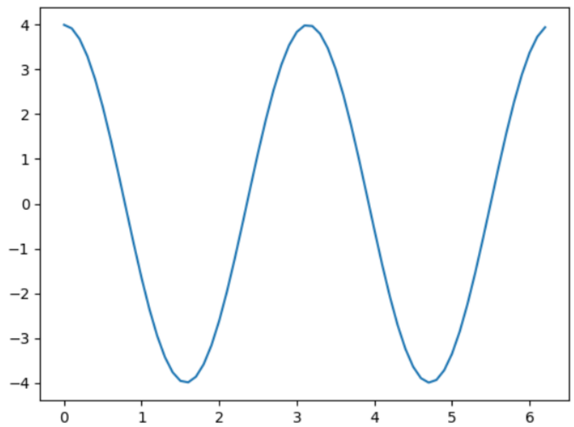

Figure 3: The graph generated at the end of Part 4, via `matplotlib`.