Introduction to Computer Programming

Exercises – Week 8. Mathematical computations with Python

Part 1. SymPy

Exercise 1 - Getting familiar with SymPy (Essential)

1. In SymPy, the keywords `pi` and `E` define the special numbers π and e. Use the `pprint` or `display` functions to show that the floating-point representations of these numbers can be computed using `pi.evalf()` and `E.evalf()`. **Fun fact**: Typing `pi.evalf(n)` computes the first n digits of π. Try executing `pi.evalf(100)`.
2. Compute $\cos(\pi/2)$ and show that it is equal to 0. Compute $\log(e^2)$ and show that it is equal to 2. **Note:** the `log` function in SymPy computes the natural logarithm.
3. The exponential function e^x is defined in SymPy as `exp(x)`. Compute `exp(1)` and assign it to a variable `my_e`. Use the `evalf` method to compute a floating-point approximation to `my_e`. You should find that the value is the same as `E.evalf()`.
4. Suppose that we want to assign the variable `a` the exact value of 1/13. Try to do this by executing `a = 1/13` and then printing the result using the `pprint` or `display` functions. You should see that a float is computed instead of an exact symbolic expression. To get around this issue, SymPy has a special variable type for fractions called Rational. Use the command `a = Rational(1, 13)` to create a variable of type Rational with value 1/13 and assign it to `a`. Then print this value to see that the exact representation of 1/13 has been retained.
5. Compute the exact value of $43/452 + 54/725 - 10/98$.

Exercise 2 - Functions (Essential)

1. Define the symbol x and the function

$$y(x) = \frac{x^2 - 4x + 3}{1 + x^2} \tag{1}$$

   Compute the values $y(0)$ and $y(4)$ exactly. Compute the floating-point approximation to $y(4)$.
2. Plot $y(x)$ over the range from $x = -10$ to $x = 10$.
3. Compute the roots of the function y; that is, determine the values of x such that $y(x) = 0$.
4. SymPy can also compute limits. Look up how to do this using `help(limit)`. Compute

$$\lim_{x\to\infty} y(x). \tag{2}$$

5. Use the `diff` function to compute the derivative of $y(x)$, which we denote by $y'(x)$. Compute $y'(0)$.

6. Compute the integral given by

$$A = \int_0^1 y(x)\,dx. \tag{3}$$

   Then, determine a floating-point approximation to A.

7. (Advanced) Determine the maximum and minimum values of the function y. **Hint**: The positions of the maximum and minimum can be determined by solving $y'(x) = 0$. Then evaluate $y(x)$ at these points. Show that these points are maxima and minima by computing $y''(x)$ at these points and checking the sign.

Exercise 3 - Modelling a pandemic (Essential)

The increase in the number of individuals infected with a virus can be described by the logistic function

$$P(t) = \frac{KP_0}{P_0 + (K - P_0)e^{-rt}}. \tag{4}$$

In this equation, P is the number of infected individuals, t is the number of days since the outbreak of infection, P_0 is the number of infected individuals at the time of outbreak, r is the rate of infection, and K is the maximum number of individuals who can be infected.

1. Create symbols `t, r, K, P_0` for the variables that appear in the logistic function. Store the function $P(t)$ in a variable called `P`.
2. Create a dictionary called `params` that stores the parameter values $K = 67$, $r = 0.1$, and $P_0 = 0.5$. **Hint**: treat the symbols as keys when creating the dictionary.
3. Substitute the dictionary of parameter values into the function $P(t)$ and plot it over the range from $t = 0$ to $t = 100$, You should see that after the outbreak there is a period of rapid, exponential growth. Then, the growth of the infected population begins to slow because there is no one left to infect.
4. Use the `diff` function to compute the derivative of P, which we'll denote by $P'(t)$. Substitute the parameter values into $P'(t)$ and plot it from $t = 0$ to $t = 100$. (Advanced) Estimate when the population of infected individuals is increasing the fastest.
5. Using the `solve` function, obtain an equation for the time at which half of the population has been infected by the virus. **Hint**: Half of the total population is $K/2$ so the equation we want to solve is $P(t) = K/2$, which can be reformulated as $P(t) - K/2 = 0$.
6. Substitute the parameter values into your expression to show that it takes about 50 days (7 weeks) for half of the population to be infected by the virus. Then, create a second parameter set where the infection rate is half of the original, so $r = 0.05$. Show that this *doubles* the time it takes for half of the population to become infected. Hence, reducing the infection rate can substantially delay the spread of the virus.

Exercise 4 - The roots of unity (Advanced)

The n-th roots of unity are the solutions z of the equation $z^n = 1$. The fundamantal theorem of algebra tells us that there must be n solutions to this equation. This question will look at computing these solutions.

1. Use SymPy to compute the tenth roots of unity by setting $n = 10$ and then solving for z.
2. Use a list comprehension along with the `evalf` method to convert the list of exact values that SymPy computes into a list of floats.
3. Use another list comprehension to compute floating-point approximations to the numbers $z = e^{2\pi i m/10}$ where $i = \sqrt{-1}$ is the imaginary number and $m = 0, 1, \dots, 9$. You should see that these are the exactly the roots of unity computed by SymPy (but they are ordered differently in the lists).
4. Explore this problem using different values of n. Can you think of a formula for the n-th roots of unity?

Exercise 5 - The period of a pendulum (Advanced)

Consider a pendulum that is made up of a sphere of mass $m = 1$ kg that is attached to a string of length $L = 0.5$ m. Suppose that the mass is displaced so that it makes an angle θ_0 with respect to the vertical. When the mass is released, it will swing back and forth. The purpose of this question is to calculate the period of the oscillations, which is the time required for the pendulum to swing one full cycle, that is, to return to its initial position once it has been released.

1. If the initial angle θ_0 is small, then the pendulum undergoes simple harmonic motion. The period of oscillations, denoted by T, is given by the equation

$$T = 2\pi\sqrt{\frac{L}{g}}, \tag{5}$$

   where $g = 9.8$ m/s^2 is the gravitational acceleration. Using Equation (5), compute the period of oscillations with provided parameter values. **Hint:** use the function `sqrt()` to compute the square root.
2. The general expression for the period of a pendulum, which is valid for any value of θ_0, is given by the integral

$$T = 2\sqrt{\frac{L}{g}}\int_{-\theta_0}^{\theta_0} \frac{d\theta}{\sqrt{2(\cos\theta - \cos\theta_0)}}. \tag{6}$$

   By taking the initial angle to be $\theta_0 = \pi/4$ (this is equivalent to 45°) and using the values of the length L and gravitational acceleration g, compute a floating-point approximation to this integral. **Note**: this integral cannot be evaluated exactly, so a floating-point approximation is the best we can do!
3. Using a loop, compute the period when $\theta_0 = 0.1\pi$, 0.2π, ..., 0.9π and print the result at each iteration. What happens to the period T as θ_0 increases? **Note:** these calculations may take a few minutes depending on the speed of the computer you are using.

Part 2. NumPy

Exercise 6 - Creating arrays and accessing elements (Essential)

1. Create a variable a that stores the array [5, 4, 9, 2, 0, 4, 7, 2].

2. Print the first entry of a and the last entry of a (**Hint**: You can use the index -1 to access the last entry of an array). The colon operator : can be used to access sequential elements of an array. Print the values of `a[1:6]` and explain the output.
3. Change the last entry of a to -9 and print the result. Now run the command `a[0:3] = 1` and print the result. How has this altered the array a?
4. Create an array r that contains 20 random integers between 1 and 9 (inclusive). Print the result. This array will be used in the next question.
5. (Advanced) **Logical indexing** provides a quick way to access and modify entries in an array that satisfy certain criteria. In this question, we'll use logical indexing to replace all of the entries in r that are smaller than 5 with 0. First, run the command `idx = r < 5`. Print the value of `idx`. Explain the result you see. Now run the command `r[idx] = 0` and print the value of r. What has happened?
6. Create a variable A to store the matrix

$$\begin{pmatrix} 6 & 2 & 3 \\ 4 & 4 & 1 \\ 8 & 5 & 6 \end{pmatrix} \tag{7}$$

   as a NumPy array.
7. Change the entry in the second row, first column of A to 9. Then change the entry in the last row and last column of A to 0. Print the updated array A. The n-th row of A can be accessed using the colon operator as `A[n-1,:]`. Similarly, the m-th column of A can be accessed using `A[:,m-1]`. Use the colon operator to print the entries in the second row of A.
8. Create a 2×2 array of zeros and assign it to a variable A. Use the colon operator to set the first row of A to 1 and the second row of A to 2. **Hint**: The operation `A[n-1,:] = q` will set all of the entries in the n-th row of A to the value q. Using a `for` loop, create a 5×5 matrix where the entries in row i are equal to i.

Exercise 7 - Performing operations on NumPy arrays (Essential)

1. Create two NumPy arrays to store the vectors $a = (3, 5, 2)$ and $b = (6, 3, 1)$. Calculate $c = a + 2b$. Calculate the dot product $a \cdot b$ using the `dot` method or the `dot` function. Can you also compute the dot product using element-by-element multiplication along with the `np.sum` function? Recall that the dot product is defined as $a \cdot b = \sum_i a_i b_i$.
2. Create an array called t that contains 500 values between 0 and 5. Create a second array called y that stores the values of $y = t^2 e^{-2t}$. **Hint**: use the `exp` function to compute the exponential of a NumPy array. Find the maximum value of y. **Note**: this is a simple way of finding the maximum of a function. (Advanced): Use logical indexing or otherwise to find the value of t at which y is maximal. (Optional but helpful): Use SymPy to plot the function $y(t)$ and check that your answers are correct.
3. This question will demonstrate that NumPy can also integrate functions. Unlike SymPy, NumPy can only produce approximate values of integrals. Create a NumPy array x that stores 50 values between 0 and 5. Create the array $y = x/(x + 1)$. Look up how to use NumPy's `trapz` function, which uses the trapezoid rule to approximate integrals. Use `trapz`

to compute $\int_0^5 y(x)\,dx$. Use SymPy to compute this integral exactly and compare with the NumPy result. What happens if you repeat the calculation using 500 points between 0 and 5?

4. The table below provides the gravitational acceleration g of each of the planets:

Planet	g [m/s^2]
Mercury	3.7
Venus	8.9
Earth	9.8
Mars	3.7
Jupiter	25
Saturn	10
Uranus	8.9
Neptune	11

   Use NumPy functions to compute the maximum, minimum, mean, and median values of g.

5. Create NumPy arrays to store the matrices

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 4 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 5 & 0 \\ 0 & 1 & 1 \\ 4 & 3 & 1 \end{pmatrix} \tag{8}$$

   Calculate $C = A + 2B$. Then compute AB and BA. You should notice that $AB \neq BA$.

6. A common operation to perform on matrices is to turn the rows into columns and the columns into rows. This is called transposing the matrix. Use the function `transpose` to compute the transpose of A and print the result.

7. Solve the linear system of equations $Ax = b$ where

$$A = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 2 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 2 \end{pmatrix} \tag{9}$$

   Print the array x. Then compute $Ax - b$ and print the result.

Exercise 8 - Weather prediction (Advanced)

In this example we'll use a **Markov chain** to create a simple model for weather prediction. To start, we will assume that there are three states of weather: sunny, cloudy, and rainy. We will use the state vector $x = (x_0, x_1, x_2)$ to describe the probabilities of the weather being sunny (x_0), cloudy (x_1), or rainy (x_2). We use a transition matrix P to describe how the weather changes from one day to the next. The entries of the transition matrix, $P_{i,j}$, describe the probability of going from state i to state j. For this problem, we will assume that

$$P = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.4 & 0.2 & 0.4 \\ 0.6 & 0.2 & 0.2 \end{pmatrix} \tag{10}$$

The entry $P_{1,1} = 0.5$ means there is a 50% chance that if a day is sunny, then the next day will be sunny. Similarly, $P_{3,1} = 0.6$ means there is a 60% chance that if a day is rainy, then the next day will be sunny.

1. Suppose that today is sunny. Then we can write the state vector as $x^{(0)} = (1, 0, 0)$. The weather tomorrow can be predicted by computing the product $x^{(1)} = x^{(0)}P$. What is the probability that tomorrow will be sunny?
2. The product $x^{(2)} = x^{(1)}P = x^{(0)}P^2$ can be used to predict the weather in two days. What is the probability that it will rain in two days?
3. Provide a prediction of the weather for seven days from now. That is, compute $x^{(7)}$.