Exercises

For Task 1 you need knowledge from the following chapters of the course book:
Ch. 2 (numeric types),
Ch. 6 (Basic Plotting),
Ch. 7 (Passing arguments, Return Values).

Task 1

Complex valued functions

In mathematics we can express any complex number \( z \) via its real part \( x \) and imaginary part \( y \) as follows (the imaginary \( i \) is expressed in Python by \( 1j \))

\[
z = x + iy.
\]

Furthermore we can re-write \( z \) as follows,

\[
x + iy = r \exp(i\varphi),
\]

where \( r \) is the distance of the number \( z \) from the origin \((0,0)\) and \( \varphi \) is its angle from the \( x \)-axis. To make this representation easier let us define the function

\[
f(r, \varphi) = r \exp(i\varphi)
\]

which allows the user to describe any complex number \( z = r \exp(i\varphi) \) by simply providing the arguments for the input parameters \( r \) and \( \varphi \).

1. Write a python function which evaluates the mathematical function \( f \) above.
2. Compute \( f \) for \( r \) fixed while \( \varphi \) takes 150 equidistant steps between 0 and \( 2\pi \).
   
   **Hint** try the command \texttt{linspace}. **Assemble** all the real and imaginary parts of your function into two lists: zreal and zimag.

   **Note:** The real part of a complex number \( z \) is obtained by the command \texttt{z.real} and its imaginary part by \texttt{z.imag}.

3. Re-visit Task 2. **Plot** the real part of \( f \) on the x-axis and the imaginary part of \( f \) on the y-axis. What shape did you get?
4. Re-visit Task 2. Let \( r \) vary from 1 to 10 in steps of 1 and **plot** in same figure. **Include a legend** which shows the value of \( r \) for each such plot.
5. Re-visit Task 4. **Plot** each figure in red color if \( r \) is even and in blue color if \( r \) is odd. **Hint:** look into the \% operator. i.e. try something like: \texttt{5\%1}.
Newton’s Method

Newton’s method is an iterative process for finding at what \( x \) the function \( f(x) \) becomes zero. We call \( x \) the root of the function \( f \). Note that a given function may have one, many or no roots at all.

Newton’s method for a given function \( f(x) \) is defined as follows:

\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}
\]

The iteration must be started with a given value \( x_0 \).

We usually end the iteration when \( |x_{n+1} - x_n| \) is less than a given tolerance \( \text{TOL} \).

For this task you need knowledge from:
Ch. 7 (Passing arguments, Return Values),
Ch. 9 (Controlling the flow inside the loop).

Task 2

1. Write a python function \texttt{newt} which computes the root for any user provided function \( f \) according to the equation given above.

Let your python function \texttt{newt} have the following input parameters:

- \( f \), the user provided function whose root we want,
- \( f' \), the user provided function, which is the derivative of \( f \) above,
- \( x_0 \) (the start value),
- \( \text{Tol} \) (the tolerance).

As output let your function return the last obtained value \( x_{n+1} \).

2. Provide default arguments for some of the above input parameters. Decide yourself which of the input parameters could/should have default values?

3. Your function \texttt{newt} computes \( x_{n+1} \) within a for loop for a number of iterations. Make that number of iterations to be another input parameter in your \texttt{newt} function. Give a default argument of 10 for that parameter.

4. Let your python function output an extra variable \texttt{conv}, which tells the user (you decide how) if convergence towards the root was observed or not.

Note: your python function might produce error messages when the root cannot be found. This happens for instance if the numbers you get grow larger and larger (in mathematics we say that the sequence of numbers you get diverges). We will show in forthcoming lectures how these error messages can be taken care of.

Test Newton’s method on functions: a) \( f(x) = x^2 - 4 \), b) \( f(x) = \arctan(x) \).

For each function above provide your own starting value \( x_0 \) and Tol and test.

Plan your solution and write your code for this task on paper first.
Discuss the approach with your neighbors and with the teaching assistants.