

Training Assignment 04

NUMA01: Computational Programming with Python
Malin Christersson, Robert Klöfkorn

This assignment has 6 tasks.

The purpose of this training assignment is to experiment more with plotting and functions.

Warming-up Exercises

Task 1

Implement the following code:

```
from numpy import *
from matplotlib.pyplot import *

x = linspace(0, 2*pi, 1000)
f_x = cos( x )
f_y = sin( x )
plot(f_x, f_y)
xlabel('cos(x)')
ylabel('sin(x)')
show()
```

Task 2

Read Chapter 6 section Formatting on how to change the plotting style and add a legend.

Exercises

Task 3

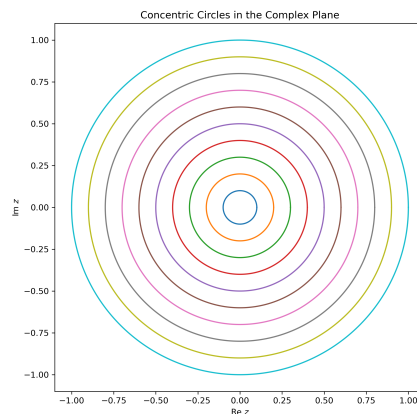
Complex valued functions

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

The complex valued function $f(\varphi, r) = r \exp(i\varphi)$ describes a circle with radius r in the complex plane, when r is kept fixed and φ varied between 0 and 2π . Set up a function which evaluates f . Plot this function for a fixed value of r in the complex plane. (Note: The real part of a complex variable z is obtained by the command `z.real` and its imaginary part by `z.imag`. Recall also that the imaginary unit i is expressed in Python by `1j`)

Let then r vary from 0.1 to 1.0 and make a plot of the corresponding concentric circles.

Expect a plot similar to the following:



Task 4

Study https://en.wikipedia.org/wiki/Euler%27s_formula and relate to Task 2.

Task 5

Newton's Method

(For this task you need knowledge from the course book Chapter 7 (Passing arguments, Return Values), and Chapter 9 (Controlling the flow inside the loop) Newton's method is an iterative process for finding a zero (root) of a given function f . It is defined as follows:

$$x_{n+1} = x_n - f(x_n)/f'(x_n).$$

The iteration is started with a given value x_0 and it is ended when $|x_{n+1} - x_n|$ is less than a given tolerance `TOL`.

Write a function `newton` which takes as arguments:

- f , the function whose zeroes we are looking for
- fp , a function, which is the derivative of f
- x_0 (the start value)
- `Tol` (the tolerance).

The function should do at most 400 iterations. It is supposed to return the last obtained value x_{n+1} together with a variable `conv`, which tells if convergence was observed or not.

Note that your function might produce error messages when the sequence diverges and the numbers grow out of the range of machine numbers. We will show in a forthcoming lecture how these error messages can be taken care of.

Write a function `myfunc` which describes a mathematical function of your choice. You should know its derivative which you are supposed to code as `myfuncp`.

Test Newton's method on these functions.

Plan your solution to this task first on paper. Discuss the approach with your neighbours and with the teaching assistants. Start programming first, when this sketch of your program was made.

Extra Task 6 (in case you want an extra challenge)

Compare the number of iterations needed to find a root within tolerance $1e-8$ for the Newton's method and for the Bisection method from Exercise 3.

Store the values $|x_{n+1} - x_n|$ for each iteration n in both methods and use both in a plot showing iteration n on the x-axis and $|x_{n+1} - x_n|$ at the y-axis.

What do you observe? Discuss with your bench neighbors or your study group.