Level 5 Laboratory: Computational Physics Exercise 1

The deadline for this exercise is **Friday 17th November 2017** at 12:30 p.m. Your report and program (*.py) files should be uploaded into Blackboard at the appropriate point in the Second Year Laboratory (PHY2DLM_2017) course.
S. Hanna

Objectives of the exercise

This starter exercise is designed to be straightforward. The purpose is to make sure that you have fulfilled the following requirements needed for the rest of the course:

- Become familiar with programming in Python
- Understand how to enter and run simple Python programs (scripts)
- Understand the origins of numerical errors in computer programs
- Be able to produce output from a program and plot simple results graphically

Problem: Series expansion of $\arctan(x)$ (10 marks)

It is often necessary to calculate a logarithmic or trigonometric function in a computer program. All computer languages have built-in functions for this, which rely on series expansions. Here you will test the convergence of such an expansion by writing a Python function to evaluate the arc-tangent function, using the following Taylor series:

$$\arctan x = \sum_{n=0}^{N} \frac{(-1)^n}{2n+1} x^{2n+1} \quad \text{where} \quad N \longrightarrow \infty \quad \text{and} \quad |x| \leq 1$$

(a) Write a function that takes x and N as arguments, and returns the resulting value of the summed series. Write a short program to test your function, by asking the user to enter values of x and N, and printing $\arctan(x)$ to the screen. Check that your function produces a good approximation to $\arctan x$ for known points, e.g. $\arctan(1/\sqrt{3}) = \pi/6$ and $\arctan(1) = \pi/4$.

(b) Extend your function to work for all x. You can do this by noting that:

$$\arctan(x) = \begin{cases} \dfrac{\pi}{2} - \arctan\left(\dfrac{1}{x}\right) & x > 0 \\[2ex] -\dfrac{\pi}{2} - \arctan\left(\dfrac{1}{x}\right) & x < 0 \end{cases}$$

You will need a conditional statement in your function to choose the appropriate formula for each value of x.

(c) To proceed further, you will need a simple menu to allow the user to choose which part of the exercise to try. The following code fragment could be adapted to your need. It operates by continually looping and asking the user to enter a value, “a” to try part (a), “d” to try part (d), etc. and “q” to quit. Merge this menu with your existing program, and proceed to the next part.

```
MyInput = '0'
while MyInput != 'q':
    MyInput = input('Enter a choice, "a", "d", "f" or "q" to quit: ')
    print('You entered the choice: ',MyInput)
    if MyInput == 'a':
        print('You have chosen part (a)')
        #
        # put your code for part (a) here
        #
    elif MyInput == 'd':
        print('You have chosen part (d)')
        #
        # put your code for part (d) here
        #
```

```
    elif MyInput == 'f':
        print('You have chosen part (f)')
        #
        # put your code for part (f) here
        #
    elif MyInput != 'q':
        print('This is not a valid choice')

print('You have chosen to finish - goodbye.')
```

(d) Create a loop in this part of your program, allowing x to vary in the range $-2 \leq x \leq 2$. Ask the user for a value of N. For each x value, output the value returned by your function, as well as the result from the built-in $\arctan()$ function in Python (either from the “math” or “numpy” modules), to the screen or to a file. Repeat this for different values of N. Use your favourite plotting program (e.g. Origin or Excel etc.) to compare the two functions as you vary N? You may find it helps to plot the difference between the two $\arctan$ functions for different x and N.

(e) Use the output from your program, and the fact that $\arctan(1) = \pi/4$, to evaluate π to 7 significant figures. What value of N was needed to obtain this level of accuracy?

Report (10 marks)

Your report should include, but not be restricted to, the following:

- A *brief* statement of the problem as you understand it;
- A *brief* description of the method used to solve the problem;
- Presentation of your results, in graphical format where applicable;
- A discussion of the results, which should include a critique of the method and ideas for improvement;
- Answers to any of the questions in the script, including appropriate discussion.

Generally, you should aim for conciseness, with most emphasis being placed on the presentation of your results and the discussion.

Submitting your work

You should submit the following to Blackboard:

1. A concise report, in MS Word or pdf format;
2. The final versions of your program.

Please note the following points:

- Blackboard plagiarism checker (Turnitin) won’t accept a file ending “.py" so please rename your file with a “.txt" extension, or use the “cut-and-paste” submission option (if available).[1]
- Please also give your programs sensible distinguishing names, including your name or userid e.g. “my_userid_ex1.txt".

If you have any problems submitting your work, please contact Dr. Hanna (s.hanna@bristol.ac.uk) or ask a demonstrator.

[1] Strictly speaking, Turnitin can be instructed to accept files with the “.py” extension. However, it will only apply the automatic plagiarism checking if the “.txt’ extension is used.