# Basic Numerical Techniques for Astrophysics and Cosmology

## 1 Introduction

### 1.1 To start with:

Now a days, it is almost impossible to avoid some amount of *scripting or coding*, if you are interested in doing research in astrophysics/astronomy/cosmology.

The aim of this section of the astronomy lab is to make you familiar with some very basic numerical techniques, error analysis, statistics and plotting tools that you will possibly (or definitely) need during the rest of your course work, specifically when you will do your year long project. In this lab we will take up some simple physics/astrophysics problems and will try to solve them numerically using a computer.

#### 1.1.1 Programming Languages

Our preferred programming language is **Python**. We would expect that you have some prior knowledge about any one or both of these programming languages! We will try to help you in your coding as much as we can during the lab, however it is your responsibility to improve your coding skills so that you are able to solve numerically most of the problems discussed in the lab classes. There are plenty of free resources available online.

#### 1.1.2 Plotting tools

It is important that you convey the results of your investigation in a comprehensive yet concise manner. One way to achieve that is by demonstrating your results in the form of a graph/plot. We would expect you to learn (during the course of these classes) how to plot using '**matplotlib**' in python. Most of the problems that we will discuss in this lab will require you to plot some functions and/or data points. Additionally, you may need to fit some data using different fitting algorithms and plot them as well.

#### 1.1.3 Report writing

You will need to submit a report on all of the problems that you have attempted to solve during this section of the lab. This report **MUST BE WRITTEN IN 'LaTeX'**. Reports written in any other text processor will not be accepted. You can find very good LaTeX templates for reports in Overleaf (`https://www.overleaf.com/gallery/tagged/report#.W0EBXZ9fjCI`). The report must reflect your understanding of the specific problem, the methodology adopted for solving it and the results with figures/plots and your conclusions. Additionally, you need to provide the source codes (python notebooks) that you have written to solve these problems. We will check and run these codes at our end as well before marking your assignments. If your code does not run, you will get **ZERO** credit for that specific problem.

### 1.2 Good practices:

- Try to follow the basic rules of clean coding! Here are some – `https://www.gnu.org/prep/standards/html_node/Writing-C.html`
- Discuss with your classmates/neighbours/friends when you are unable to understand something! Above all, learn how to ask Google! Google has solution to most/all of the problems that you may face during this course.

- You are expected to finish your previous weeks assignments outside the lab hours and come with only those queries for which you have not found any answers in Google. Lab hours are the only time slots when you can interact with the instructors and tutors. Try to utilize their and your time judiciously!
- During every lab class, you are expected to start working on a new set of assignments that has been covered in the short lecture at the beginning of the class.
- If you have managed to finish the stipulated set of assignments for the day and still have some time left during the lab classes, start on the next set of assignments or start writing the report for the set that you have just finished. Do not just sit around or chat uselessly in the lab!
- While writing your report, if you have used any information from any resources (e.g. from websites, books, articles etc.) cite them at proper places of your report. Try to build a comprehensive bibliography section in your report.
- Write your own codes. Do not try to copy codes from your friends or anywhere else in the Internet! Remember your instructors and tutors have access to Google as well!

**Enjoy the lab classes!**

# 2 Stability of Computation

## 2.1 Problem 1

Encode the following algorithm and run it to determine the smallest positive number that can be represented on the computer you are using:

```
input s <--- 1.0

for k=1,2,3,...,100 do
    s <--- 0.5*s
    t <--- s + 1.0

    if t <= 1.0 then

      s <---  2.0*s
       output k-1, s
       stop

    endif
end
```

Do this for both single precision and double precision floating point numbers.

## 2.2 Problem 2

Evaluate the expression

$$y = (x^2 + 1.0)^{0.5} - 1.0 \tag{1}$$

in two ways:

$$y = (x^2 + 1.0)^{0.5} - 1.0 \tag{2}$$

and

$$y = x^2/[(x^2 + 1.0)^{0.5} + 1.0] \tag{3}$$

for small values of x, x=0.1, 0.01 and 0.001. Determine the fractional error in both the methods of performing the subtraction. Which method is superior and why?

## 2.3 Problem 3

It is desired to calculate all integral powers of the number $x = (\sqrt{5.0} - 1.0)/2.0$. It turns out that the integral powers of $x$ satisfy a simple recursive relation

$$x^{(n+1)} = x^{(n-1)} - x^n \tag{4}$$

Show that the above recurrence relation is unstable by calculating $x^{16}$, $x^{30}$, $x^{40}$ and $x^{50}$ from the recurrence relation and comparing with the actual values.

## 2.4 Problem 4

The recurrence relation

$$y_{n+1} = e - (n+1)y_n \tag{5}$$

(where $e$ is the base of natural logarithm) can be obtained from integration by parts to the integral

$$y_n = \int_0^1 x^n e^x dx \tag{6}$$

Show that the above recurrence relation is unstable by calculating $y_{15}$ and $y_{20}$ from the recurrence relation.

## 2.5 Problem 5

Compute the dot product of the following two vectors

$x = [2.718281823, -3.141592654, 1.414213562, 0.5772156649, 0.3010299957]$
and
$y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$.

Compute the summation in four ways:

- forward order summation $\{x_i\, y_i\}$ for $i = 1, n$ .
- reverse order summation $\{x_i\, y_i\}$ for $i = n, 1$ .
- largest to smallest order (add positive numbers in order from largest to smallest, then add negative numbers in order from smallest to largest and then add the two partial sums).
- smallest to largest order (reverse order of adding in the previous method).

Use both single and double precision for a total of eight answers. Compare the results with the correct value $1.006571 \times 10^{-9}$.