Basic Numerical Techniques for Astrophysics and Cosmology

This problem set has been compiled by Suman Majumdar and Bhargav Vaidya

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.

1.2 Prerequisites:

There are plenty of prerequisites in this section of the lab. You will need to spend a significant amount of time on these prerequisites (outside the lab hours) if you are not already familiar with them.

1.2.1 Operating System

The preferred operating system for this lab is **Linux**(`https://en.wikipedia.org/wiki/Linux`). Before attending these classes you should know how to use a Linux 'Terminal' and some basic Linux commands. A large variety of tutorials and help are available online on this topic. So please ask 'Google'(`https://www.google.co.in`) before asking your instructor!

1.2.2 Programming Languages

Our preferred programming languages are **C** and **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. Make use of the fast internet connection available on campus!

1.2.3 Editors

There are many editors that you can use for writing your codes in Linux. We will prefer GVim, Emacs and jupyter notebook.

1.2.4 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 '**gnuplot**' and/or '**matplotlib**'. 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.

1.2.5 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 ('.c' or '.py' files) that you have written to solve these problems. These codes should compile in a Linux computer having just gcc, python, scipy and numpy. We will not install any special libraries or packages (unless stated otherwise) to compile or run your codes! If your code does not compile or run, you will get **ZERO** credit for that specific problem.

1.3 Precautions:

Always remember to backup your work/codes etc. in some cloud storage at the end of the lab classes everyday or more frequently. Institute provides you with an unlimited storage in the Google drive. You can use it for this purpose. Try to write reports in Overleaf, so that they are backed up instantaneously. The department is not responsible if you loose your codes etc. due to the malfunction in a computer/server, sudden power blips etc.

1.4 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 1.006571×10^{-9}.

3 Interpolation

3.1 Problem 1

Given the three data points (x, y) = (1.0, 8.0), (2.1, 20.6) and (5.0, 13.7), write a program to return the value of y for any arbitrary x in the range [1.0, 5.0]. The program should exit if x is outside this range.

3.2 Problem 2

Generalize the above program to input data points from a given ascii data file. The data file will be provided in the class.

3.3 Problem 3

Repeat Problem 1 using second order polynomial. Compare the y values from the linear interpolation with those from the second order polynomial. Plot the polynomial and the input data points.

4 Integration

4.1 Problem 1

Describe the motion of a planet for each of the following sets of initial condition. The sun is assumed to be at the origin and its motion is not considered. The only force involved being the force of gravitation.

Assuming $GM_\odot = 1$, where G is gravitation constant and $M_\odot$ is the mass of sun, one can avoid dealing with astronomical magnitudes for positions:

- $x_0 = 0.5$; $y_0 = 0.0$; $v_{0,x} = 0.0$; $v_{0,y} = 1.63$.
- $x_0 = 0.5$; $y_0 = 0.0$; $v_{0,x} = 0.0$; $v_{0,y} = 1.80$.
- $x_0 = 0.5$; $y_0 = 0.0$; $v_{0,x} = 0.0$; $v_{0,y} = 1.40$.
- $x_0 = 0.5$; $y_0 = 0.0$; $v_{0,x} = 0.50$; $v_{0,y} = 1.80$.
- $x_0 = 0.5$; $y_0 = 0.0$; $v_{0,x} = -0.50$; $v_{0,y} = 1.40$.
- $x_0 = 0.5$; $y_0 = 0.50$; $v_{0,x} = -0.50$; $v_{0,y} = 0.50$.

4.2 Problem 2

Write two functions which will perform integration following Trapezoidal and Simpson's rules respectively. Write these functions in such a way that they can take in any function as their integral.
Hints: If you are familiar with the concepts of pointer and function pointer, that will help you in writing these functions. Once you have managed to do that, evaluate the following integrals numerically:

- $\int_0^1 x dx$
- $\int_0^1 x^2 dx$
- $\int_0^1 \sin(x) dx$
- $\int_0^1 x \sin(x) dx$

using both Trapezoidal and Simpson's rules for $N = 2, 4, 8, ...1024$, where N is the number of intervals used in the integration. As all of these integrals can be evaluated analytically, estimate the relative (with respect to the analytical value) error $\mathrm{Err}(N)$ in the numerical integration as a function of N. Show plots of $\mathrm{Err}(N)$ with varying N for all the integrals. Are these results consistent with how you expect the error to scale with N?

4.3 Problem 3

Consider the integral

$$\int_{-\infty}^{\infty} \frac{\sin^2(x)}{x^2} dx \,. \tag{7}$$

How will you evaluate it analytically? How will you evaluate it numerically? Compare your numerical and analytical results.

4.4 Problem 4

- Compute the following integral using 2, 3 and 5 point Gaussian quadrature method:

$$\int_{-1}^{1} x^2 e^{-x^2} dx \,. \tag{8}$$

- The arc length along a curve $f(x)$ is define as:

$$s = \int \sqrt{1 + f'(x)^2} \tag{9}$$

find the arc length from 0 to π along a curve $f(x) = sin(x)$ analytically and also by using 2, 3 and 5-point Gaussian quadrature method and determine the % error with respect to the analytical value.

4.5 Problem 5

We consider the 1D motion of a particle of mass m in a time independent potential $V(x)$. The fact that the energy E will be conserved allows us to integrate the equation of motion and obtain a solution in a closed form:

$$t - C = \sqrt{\frac{m}{2}} \int_{x_i}^{x} \frac{dx'}{\sqrt{E - V(x')}} \,. \tag{10}$$

The particle is at the position x_i at time $t = 0$ and it is located at the position x at any arbitrary time t. For simplicity, let us assume that $C = 0$.

We consider a particular case where the particle is in bound motion between two points a and b where $V(a) = E$ and $V(b) = E$ and $V(x) < E$ for $a < x < b$. The time period of the oscillation T is given by

$$T = 2\sqrt{\frac{m}{2}} \int_{x=a}^{x=b} \frac{dx'}{\sqrt{E - V(x')}} \,. \tag{11}$$

First consider a particle with $m = 1$ kg in the potential $V(x) = \alpha x^2/2$ with $\alpha = 4\,\mathrm{kg/sec}^2$. Numerically calculate the time period of oscillation and check this against the expected value. Verify that the frequency does not depend on the amplitude of oscillation.

Next consider a potential $V(x) = \exp(\alpha x^2/2) - 1$. Numerically verify that for small amplitude oscillations you recover the same results as the simple harmonic oscillator. The time period is expected to be different for large amplitude oscillations. How does the time period vary with the amplitude of oscillations? Show this graphically.

5 Ordinary Differential Equations

5.1 Problem 1

Write a program using Euler's Method and using the 2$^{\text{nd}}$ and 4$^{\text{th}}$ order Runge-Kutta Method to find the phase space trajectory of a particle of unit mass in a 1-D potential $V(x) = (x^2 - 1)^2$.

- Show the trajectory for the initial conditions $(x, P_x) =$ (1.0, 0.1), (-1.0, 0.1), (1.0, 10.0).
- Check the conservation of energy for different step sizes, for both the Euler Method and the 2$^{\text{nd}}$ and 4$^{\text{th}}$ Order Runge-Kutta method. Determine the step size where you have at least 1% accuracy in the energy over one whole period of the particle.

5.2 Problem 2

Write a program to follow the motion of an electron in an electric field $E(x, t)$ and a magnetic field $B(x, t)$. Numerically determine the trajectory of an electron which starts at the origin with velocity $v = (1.0, 1.0, 1.0)\,\mathrm{m/sec}$ for the following field configurations:

- Uniform magnetic field 10^{-4} Tesla along the z axis.
- Uniform magnetic field 10^{-6} Tesla along the z axis and an uniform electric field 1V/m along the Y axis.
- Uniform Electric field 5×10^{-3} V/m ($|\vec{B}| = 0$) along the X-axis. Use first the Euler method and then the Leap frog method.

5.3 Problem 3

Solve the two dimensional trajectory of a projectile fired from a cannon shell located at the origin using the Euler method. Assume the initial projected speed is 700 m/s, and using different firing angles starting from 20° to 60° with an interval of 5°. Neglect the effects of the air resistance. Plot the trajectories of the projectile for different firing angles. Also plot the range of the projectile against the firing angles and show numerically that the maximum range of the projectile corresponds to a firing angle of 45°. Compare the numerical results with the analytical solutions obtained for the range and the duration of the projectile.

5.4 Problem 4

Extend the above problem to include the effect of the air resistance by assuming a value of $b/m = 4 \times 10^{-5}\mathrm{m}^{-1}$, where the air drag force is expressed as $F_{\text{drag}} = bv^2$. Plot the trajectories for a given firing angle for the case of no air resistance and with air resistance.
Repeat this for different firing angles. Plot the range of the projectile against the different firing angles and find numerically the firing angle which corresponds to the maximum range of the projectile.

5.5 Problem 5

Now consider the air resistance is not uniform. Consider the air density varies as $\exp(y/y_0)$, where $y_0 = 10^4$ m. Now plot the trajectories for a given firing angle without air resistance, with constant air resistance and with varying air resistance as given above. Repeat this for different firing angles. Also plot the range of the projectile against the different firing angles and find numerically the firing angle which corresponds to the maximum range of the projectile.

6 Random Processes

6.1 Random Walk

It is late night and a drunkard is walking along a very long street. The drunkard is not sure which is the way home, so he/she randomly takes steps of length 1.0 m forward or backward. He/She takes one step every second continuously for 1 hr.

- Simulate this process by using a random number generator from python or C library.
- Generate 100 different realizations of the random sequence of steps by using a different seed each time.
- Graphically show the random walks for two realizations.
- Calculate the displacement of the drunkard for each realization and show graphically how these are distributed using a histogram.
- Calculate the mean and the root mean square (r.m.s.) displacement of the drunkard after 1 hr.
- Increase the time (number of steps) and show that the mean displacement tends to 0 and the r.m.s. displacement scales as the square-root of the time. Also try to derive the same results analytically.

6.2 Monte-Carlo Integration

Calculate the following integrals (I) using Monte-Carlo method (you can use a random number generator from the python or C library or GNU Scientific library):

$$I = \int_0^{\pi} \cos(x)dx \tag{12}$$

$$I = \int_0^{1} x^2 dx \tag{13}$$

- How will you calculate the standard deviation σ of the integral I?
- Plot I and σ as a function of N (where N is the number of random points used to estimate the integral).
- What is 'standard deviation of the means' ($\sigma_{\rm M}$) in the context of Monte-Carlo integration? Estimate $\sigma_{\rm M}$ for the above integrals.
- From the numerical results that you have obtained, try to show that for large N

$$\sigma_{\rm M} \approx \sigma/\sqrt{N}\,. \tag{14}$$

7 Astrophysics Related Problems

7.1 Polytopes and Lane-Emden Equation

In theoretical studies of star formation and its structure, an important concept is that of *hydrostatic balance*. The forces due to gradient of gas pressure balances the gravity pull of the star. This is given as -

$$\frac{dP(r)}{dr} = -\frac{Gm(r)\rho(r)}{r^2} \tag{15}$$

where r is the radial co-ordinate of the spherical star and the mass $m(r)$ is related to density ρ as-

$$\frac{dm(r)}{dr} = 4\pi r^2 \rho(r) \tag{16}$$

Further the relation between pressure P and density ρ called the Equation of state is assumed to be of that of a polytrope -

$$P = \mathcal{K}\rho^{1+1/n} \tag{17}$$

where $\mathcal{K}$ is a constant and n is the poly-tropic index.

- Using the scaling $\Theta = (\rho/\rho_c)^{1/n}$ and $\zeta = r/\alpha$ with ρ_c being the density at the center of the star show that Eq. 15 can be written as -

$$\frac{1}{\zeta^2}\frac{d}{d\zeta}\left(\zeta^2\frac{d\Theta}{d\zeta}\right) = -\Theta^n \tag{18}$$

  and find the expression for α.
- Equation 18 is called *Lane-Emden* equation and our aim in this exercise is to solve the above equation numerically given the boundary conditions. Applying this equation to study the structure of star, specify the boundary conditions that would be required and justify your choices.
- Using the ODE methods you know, solve the Eq.18 and show that the solution for n = 1 satisfy $\sin\zeta/\zeta$ and the outer radius of the star (outer boundary) is at $\zeta_1 = \pi$
- Repeat your numerical solution for n = 1, 1.5, 2, 2.5, 3.0 and plot the value of ζ vs Θ and finally estimate the values of ζ_1 for different values of n.

7.2 Cosmology Calculator

Write a code to design a calculator that will estimate few basic cosmological parameters given the standard inputs. A very popular cosmology calculator based on *Java* is available on `http://www.astro.ucla.edu/~wright/CosmoCalc.html`. You can use the same to verify the results of your code.

The code should take the following inputs -

1. Hubble Constant H_0.
2. $\Omega_{\rm vac}$: cosmological constant.
3. Ω_M : the ratio of the matter (dark + baryonic) density of the Universe to the critical density.
4. The redshift - z.

The code should output the following quantities

1. Age of the universe since the Big Bang.
2. Age of the universe at the given redshift z.
3. The light travel time.
4. Co-moving radial distance.
5. Co-moving volume within the given redshift z.
6. Angular size distance (D_A) and Luminosity Distance (D_L).

7.3 Parker Solar Wind

The Sun has a solar wind which blows out of its surface. Material from the million degree outer atmosphere of the Sun, the corona, continually expands, eventually reaching supersonic velocities of ~ 1000 km/s.

Eugene Parker suggested a model based on purely radial flow in isothermal conditions to explain the solar wind, this model is called the *Parker wind model*. This will be (or is) covered in the Astrophysical Fluids class [also see `http://www.scholarpedia.org/article/Parker_Wind` for more details.]

The momentum conservation equation for the Parker-wind model can be expressed as follows -

$$\frac{dV}{dr}\left(\mathcal{M} - \frac{1}{\mathcal{M}}\right) = \frac{2}{r}\left(1 - \frac{R}{r}\right) \tag{19}$$

where, $\mathcal{M}(r) = \frac{V(r)}{c_s}$ is the radially dependent *Mach* number with c_s being the isothermal sound speed at corresponding to temperature $T(r)$. The critical radius $R = \frac{2GM_0}{c_s^2}$, with M_0 as the mass of the Sun.

Solve the above differential equation and plot the solution i.e., curves for the variation of Mach number - $\mathcal{M}(r)$ as a function of r/R. Explain the significance of critical radius R.