NUMERICAL AND MATHEMATICAL METHODS

January 27, 2023

EXERCISE 1. Download the file star_cluster.csv, which contains information about the stars of a simulated stellar cluster. **This is a csv file (the columns are separated by commas rather than by spaces) with one header row.**
Each line contains the properties of a single star. The columns have keys **mass** (column #0), **x** (column #1), **y** (column #2), **z** (column #3), **vx** (column #4), **vy** (column #5), **vz** (column #6). They correspond to the star mass, x, y, z spatial components, and x, y, z velocity components. The star mass is in solar masses ($M_\odot$), the spatial components are in parsec (pc) and the velocity components are in km s^{-1}. Write a python script that

A. reads the csv file;

B. selects the lines in which the absolute value of each spatial component (x, y, z) is < 20 pc. The result of this selection will be a new dataframe;

C. takes the dataframe obtained in **B.** and plots a 2-dimensional histogram with **vx** and **vy** (i.e., the x and y velocity components) on the x and y axis, respectively;

D. takes the dataframe obtained in **B.** and generates two sub-data frames containing the stars with mass > 1 $M_\odot$ and ≤ 1 $M_\odot$, respectively;

E. takes the two sub-dataframes generated in **D.** and produces a 3-dimensional plot of the spatial components (**x**, **y**, **z** along the x, y and z axis, respectively) distinguishing the two sub-dataframes (the one of the massive and the one of the light stars) with two different colors.

The plots in points **C.** *and* **E.** *can be done either as two single plots or two panels of the same plot. In the latter case, use the add_subplot() function. Suggestion: To read the csv file, you can use either numpy or pandas, based on your preferences. If you choose numpy, remember to set the proper delimiter.*

The plots for this and the following exercises do not need to be saved to a file. Just add the command plt.show() inside the scripts.

Please upload the script as a unique script named ex1.py.

EXERCISE 2. Write a python script that integrates the following function

$$f(x) = \sin(x^2)\, \exp(-x^{1/2}) \tag{1}$$

in the range $x = [0, 17]$ according to the two following methodologies:

A. with the trapezoidal rule, using $N = 10^5$ intervals (please write a user-provided function instead of using the built-in function of scipy.integrate);

B. with the mean value method, using $N = 10^5$ random points.

C. Then, plot the function with a line plot.

Please upload the result as a unique script named ex2.py.

EXERCISE 3. The following system of two first-order ordinary differential equations (ODEs) describes the evolution of the semi-major axis a (in astronomical units, AU) and the orbital eccentricity (e, dimensionless) of a binary black hole that is evolving by Newtonian three-body scatterings:

$$\begin{aligned} \frac{\mathrm{d}a}{\mathrm{d}t} &= C_1\, a^2, \\ \frac{\mathrm{d}e}{\mathrm{d}t} &= C_2\, a, \end{aligned} \tag{2}$$

where $C_1 = -4.0 \times 10^{-3}$ AU^{-1} Myr^{-1} and $C_2 = 2 \times 10^{-4}$ AU^{-1} Myr^{-1}.
Write a python script that

A. integrates the system of equations with the **midpoint scheme** from time $t = 0$ to $t = 40$ Myr, assuming $a(0) = 10$ AU and $e(0) = 0.5$. You can use time-step $h = 0.1$ Myr;

B. plots the value of $a(t)$ and $e(t)$ as a function of time.

Please upload the result as a unique script named ex3.py.

EXERCISE 4. Write a python script that performs the following tasks.

A. It generates a set of $N = 10^5$ stellar masses, randomly generated according to the following probability distribution function (PDF):

$$p(m)\,\mathrm{d}m = \mathrm{const}\, m^{-1.3}\,\mathrm{d}m, \tag{3}$$

where const is the normalization constant that you have to calculate. The minimum mass is 0.1 $\mathrm{M}_\odot$ and the maximum mass 100 $\mathrm{M}_\odot$. Use the inverse sampling method.

B. It plots the histogram of this set of data in the log–log plane.

C. Then it performs a linear fit of the PDF in the log–log plane calculating the intercept A, the slope B and their corresponding uncertainties (σ_A and σ_B); the fit should be done with a user-provided routine for the least squares fitting (here, user-provided means that you should not use scipy.optimize or similar packages).

D. Finally, it plots the fitting line overlapped to the histogram.

Suggestion for the histogram: The data points that you should try to fit are the values of the logarithms of the PDF (y values) and the midpoints of the logarithmic bins (x values). If you use too many bins, some of the y values of the histogram might be zero, the logarithm of which is not defined. You can fix this problem either by using less bins, or by searching for possible zeros in the array of the y values (e.g., with numpy.where) and then transforming them into small but non-zero values (e.g., 10^{-10}).

Please upload the result as a unique script named ex4.py.