



Computational Physics: Introductory Course

Assignment 2:

Programming with Python

Per Jönsson & Jörgen Ekman
Faculty of Technology and Society
Malmö University

What the report should contain

The report should contain:

- The py-files, with carefully commented Python code, that were used to solve the exercises. All code should start with a sentence or two describing what the code is doing. Each py-file should be embedded in the LaTeX or Word document.
- When applicable, you should include the output or plots produced by your code as part of the exercise.
- When you, as part of an exercise, are asked to draw physical conclusions it is important that these conclusions are included in the report. If you are unable to draw any conclusion you should explicitly write 'I was not able to draw any conclusion'.

Important things to keep in mind for producing a good report that adheres to the general rules in scientific writing.

- All variables in the report should be in math font, e.g. the distance was $d = 1.5$ (do not write the distance was $d = 1.5$, wrong).
- All units in the report should be in correct font, e.g. the acceleration is $a = 1.5 \text{ ms}^{-2}$ (do not write the acceleration is $a = 1.5 \text{ ms}^{-2}$, wrong).
- Write $a = 1.5 \cdot 10^{-3}$ instead of $a = 1.5 * 10^{-3}$ (wrong, never use $*$ in equations or text)
- If you give computed values, round them to a reasonable number of decimals.
- The font size in plots should be large enough for the axis texts and legends to be read.

Reading instructions for assignment 2

Start to read chapter 7 about programming in the lecture notes.

Exercises

1. Write a script file that uses a loop and the `print` command with format codes (see section 3.9) to exactly produce the table below. Having done this, modify your script file so that it produces a table that starts with 5 degrees and ends with 85 degrees, in steps of five degrees ($x = 0^\circ, 5^\circ, 10^\circ, \dots, 80^\circ, 85^\circ$). You can use the function `radians` to convert from x in degrees to radians before calling the trigonometric functions, see section 2.3.

x	sin(x)	cos(x)	tan(x)
0.00000	0.00000	1.00000	0.00000
1.00000	0.01745	0.99985	0.01746
2.00000	0.03490	0.99939	0.03492
3.00000	0.05234	0.99863	0.05241
4.00000	0.06976	0.99756	0.06993
5.00000	0.08716	0.99619	0.08749

6.00000	0.10453	0.99452	0.10510
7.00000	0.12187	0.99255	0.12278
8.00000	0.13917	0.99027	0.14054
9.00000	0.15643	0.98769	0.15838
10.00000	0.17365	0.98481	0.17633

2. You can generate random integers using the command `random.randint`, see section 5.12.
 - (a) Test the command above and generate and print a vector with 40 random integers between 1 and 10.
 - (b) Run the command above to generate a vector u with 10000 integer random numbers between 1 and 6. Run the command above once more to generate a vector v with 10000 integer random numbers between 1 and 6. Add u and v and save in a vector w . The vector w will contain the results from 10000 throws with two dices.
 - (c) Make a histogram with 11 bins (we have data from 2 to 12) using the data in w .
 - (d) Compute the mean and standard deviation of the elements in w (use the built in commands).
 - (e) Write a small program (script file) that loops through all the elements in w and counts how many of the elements equals 7 (how many times the eyes of the two dices adds to 7). Divide this number by the total number of throws, i.e. 10000 and you will get the probability that the sum of the eyes of the two dices is 7. Hint: inside your loop you may want to use a construction similar to the one below. `ncount` counts the number of elements equal to 7 and should be set to 0 in the beginning of the program

```

if w[i] == 7:
    ncount = ncount + 1

```

- (f) Use you knowledge in probability theory to compute the probability to get the result 7 when throwing two dices?

This exercise is the first encounter with random numbers. We will continue with random numbers in Assignment 8 on Monte-Carlo methods.

3. We have the following values of the amount of CO₂ (in ppm) in the atmosphere from beginning of 2000 to end of 2025.

Data has been taken every week for 26 years. Most years have 52 weeks, but five years, including 2000, have 53 weeks and therefore we have 1357 data values in total. These values are given in the file **co2.txt** that can be downloaded from the course home page.

- (a) Open the file **co2.txt** with an ordinary editor to see how it looks.
 - (b) Read the data with the `loadtxt` command, see lecture notes section 5.17.
 - (c) Plot the data using `+` to mark the points. Add title and text on the axes.
 - (d) Determine and write out the largest and smallest amount of CO₂. As a check the largest value should be between 429 - 432 ppm and the smallest value between 366 - 368 ppm.
 - (e) Determine the mean of CO₂ amount during 2000 with 53 weeks (lower and upper bound for slice is 0 and 53, see section 4.3 in lecture notes) and during 2025 with 52 weeks (lower and upper bound for slice is 1305 and 1357). As a check you should get a mean value of 369 - 370 ppm and 427 - 428 ppm, respectively.

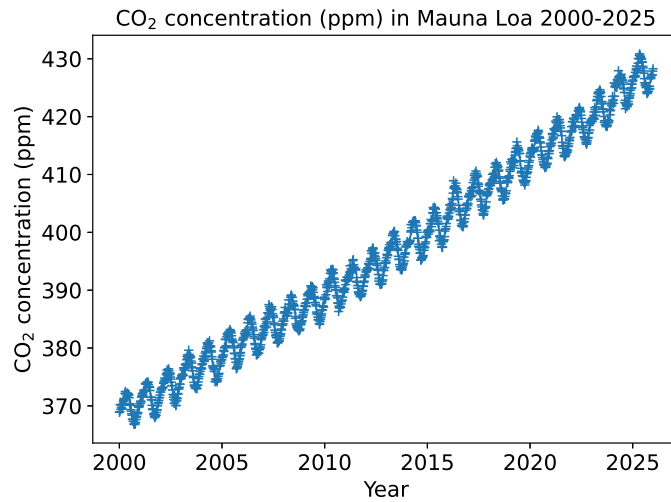


Figure 1: *Amount of CO_2 (in ppm) from 2000 to the end of 2022.*

- (f) Use the mean of CO_2 amount during 2000 and 2025 to make a prediction of the mean amount during 2050. As a check you should get between 483 - 486 ppm.
4. This is the starting exercise on nested loops. The figure below displays and interference pattern recorded with a CCD-detector. Some of the elements in the CCD-detector are faulty so that black spots appear in the image.

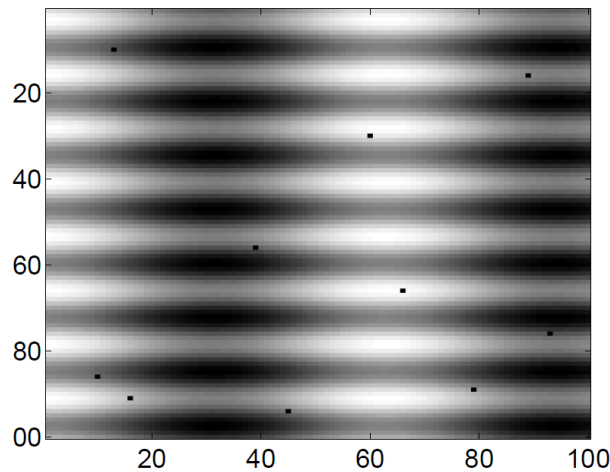


Figure 2: *Interference recorded with CCD-array.*

The interference pattern is saved in a 100×100 matrix C in the file **CCD.txt** (available at Canvas).

- Read the matrix C from the file with the `loadtxt` command.
- Display the matrix with the `imshow` method and scaling `[3, 7]`. Choose the colormap 'gray'.
- The faulty black spots all have pixel values 0. Write a program that loops through the matrix C . If a pixel $C(i, j)$ has the value 0 then it should be replaced with the mean of the eight

neighbors

$$C_m(i, j) = \frac{1}{8}(C(i-1, j-1) + C(i-1, j) + C(i-1, j+1) + C(i, j-1) + C(i, j+1) + C(i+1, j-1) + C(i+1, j) + C(i+1, j+1))$$

(d) Display the corrected matrix. Use the same scaling as above.

- The temperature (in Kelvin) has been recorded in 10×10 points distributed over a heated steel plate with dimension $1 \text{ m} \times 1 \text{ m}$. The temperature data is in a matrix T in the file **stalplat.txt** that can be downloaded from the course homepage.

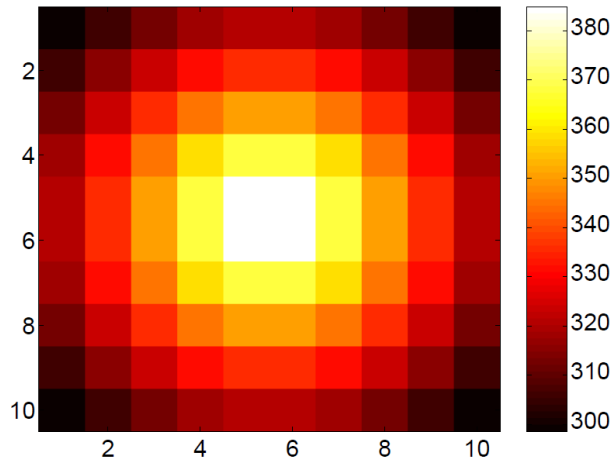


Figure 3: *Temperature for a heated steel plate.*

According to the Stefan-Boltzmann law the net radiated power from a surface element is given by

$$P = Ae\sigma(T^4 - T_0^4).$$

Here T are T_0 , respectively, the temperature of the element and the surrounding air in Kelvin. A is the area of the surface element, e is a material constant that can be set to 0.6 and $\sigma = 5.67 \times 10^{-8} \text{ Wm}^{-2}\text{K}^{-4}$ is the Boltzmann constant.

Write a script file that reads the temperature matrix T with the `loadtxt` command and computes the total emitted power from the plate under the assumption that the surrounding temperature is $T_0 = 300 \text{ K}$. Important thing to think of: the plate has area 1 m^2 and is divided into 100 elements, what is the area A of a surface element you should use in your calculation? As a check you should get a value between 300 - 305 W. Don't forget that the plate radiates from both sides

- Modify the program for computing the temperature distribution in a plate, section 7.12 in the lecture notes. Your program should compute and display the temperature distribution in the plate given the same edge temperatures as before. In addition your plate has quadratic region corresponding to rows with index 30 to 50 and columns with index 20 to 40, where the temperature is constant and $2225 \text{ }^\circ\text{C}$.

```
T0 = (650/4)*np.ones((100,100))
T0[0,:] = 100; T0[99,:] = 200
```

```

T0[:,0] = 100; T0[:,99] = 250
T0[30:51,20:41] = 225           % set middle temp.
T1 = np.copy(T0)

```

You can still have a double loop from 1 to 99, but inside the double loop you need some if-statements to handle the inner region i.e. rows with index from 30 to 50 and columns with index from 20 to 40 for which there should be no temperature update. Running with $tol = 0.01$ your temperature distribution should look like the one in figure 4.

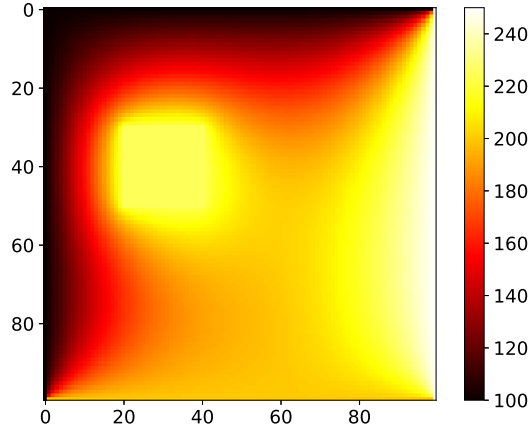


Figure 4: *Stationary temperature distribution.*

Physics background

The course is in computational physics, and we provide the physics background for many of the exercises together with links, that make it possible to read more if you find it interesting.

Stefan-Boltzmann law

The Stefan–Boltzmann law describes the power radiated from a black body in terms of its temperature T . Specifically, the Stefan–Boltzmann law states that the total radiated power across all wavelengths is given by

$$P = Ae\sigma T^4,$$

where A is the surface area, e the emissivity that is equal to 1 for a perfect black body, and $\sigma = 5.67 \times 10^{-8} \text{ Wm}^{-2}\text{K}^{-4}$ the Boltzmann constant. The spectral energy density (the distribution of the energy with respect to wavelength) of the radiation from a black body follows the Planck distribution, see figure 5. The wavelength λ_{peak} for the peak of the radiation follows Wien’s displacement law

$$\lambda_{\text{peak}} = \frac{b}{T},$$

where $b = 2.89 \times 10^{-3} \text{ mK}$.

Stefan–Boltzmann’s law and Wien’s displacement law are important in astronomy to determine stellar properties. The total radiated power L of a star is known as the luminosity and is given by

$$L = 4\pi R^2 \sigma T^4,$$

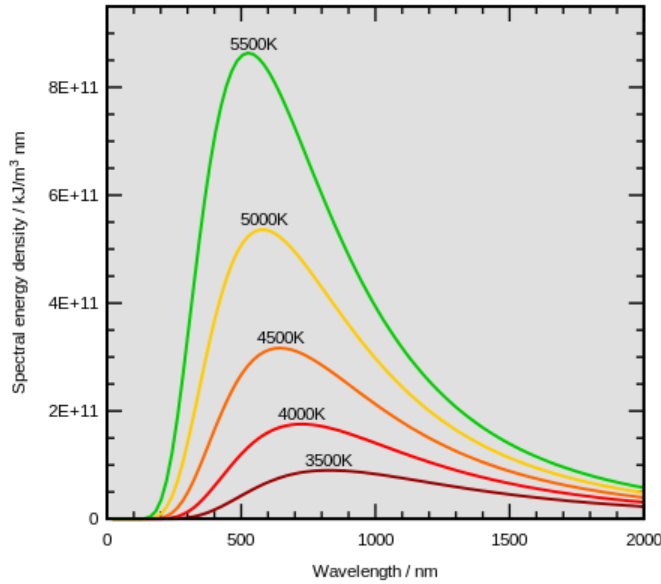


Figure 5: *Spectral energy density for different temperatures*

where R is the radius of the star ($4\pi R^2$ is the surface area of the star). Rearranging we have

$$R = \sqrt{\frac{L}{4\pi\sigma T^4}}.$$

By measuring the luminosity of the star and estimating the temperature according to Wien's displacement law, see figure 6, we can infer the radius. The cool star Betelgeuse is one of the largest stars we know and it's radius is around 900 times the radius of the Sun.

The earth breaths

The earth breaths! During the spring and summer the vegetation of the northern hemisphere absorbs CO_2 to build biomass (leaves, grass etc) and the CO_2 concentration in the atmosphere drops. During the autumn and winter the leaves fall off and molder, CO_2 is released into the atmosphere and the concentration increases. There is a similar process at the southern hemisphere, shifted half a year in time, but less pronounced as the landmass is considerably smaller. The breaths of the earth can be followed by looking at satellite images of the land cover during a year (see video below). See also https://en.wikipedia.org/wiki/Carbon_dioxide_in_Earth%27s_atmosphere#Current_concentration



Figure 6: *The color of a star is determined by its temperature, according to Wien's law. In the constellation of Orion, one can compare Betelgeuse ($T = 3300$ K, upper left), Rigel ($T = 12100$ K, bottom right), Bellatrix ($T = 22000$ K, upper right), and Mintaka ($T = 31800$ K, rightmost of the 3 "belt stars" in the middle).*

In addition to the natural breath of the earth CO_2 is constantly released into the atmosphere, mainly due to burning of coal and oil. The concentration of CO_2 is measured at the Mauna Loa observatory <https://gml.noaa.gov/ccgg/trends/> and in figure 7 we see the concentration the last 5 years and since 1960. The concentration is increasing at an alarming rate!

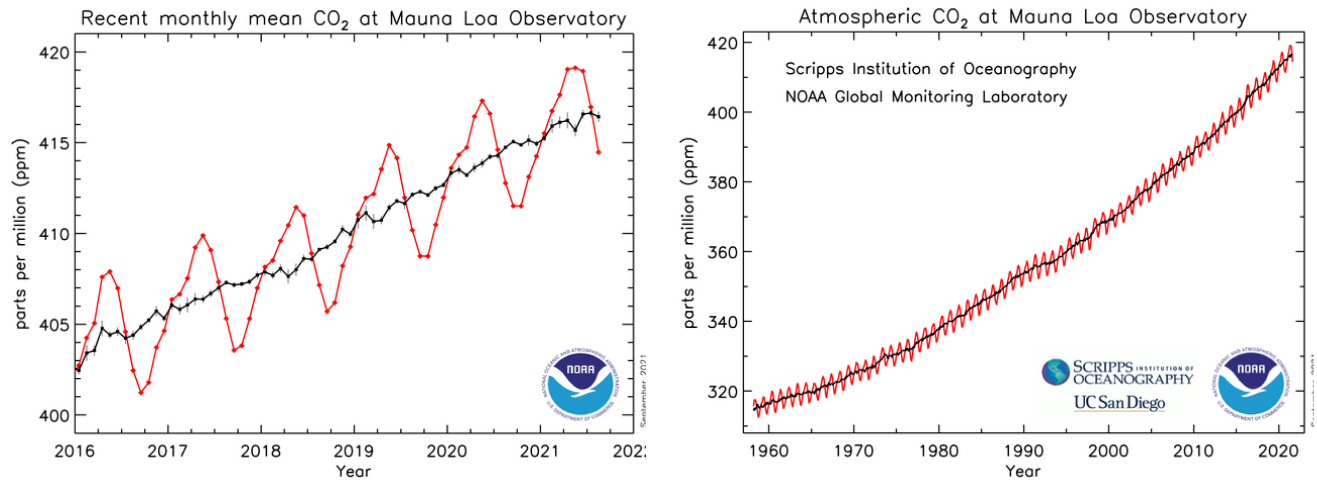


Figure 7: CO_2 concentration in the atmosphere. In addition to the natural variations between spring and summer and autumn and winter, we see an almost constant increase.