PHOTOELECTRIC HEATING ON INTERSTELLAR GRAINS:
Impact of the fractal structure of grains by Monte-Carlo

I- Projet description

I-1 Context

Photodissociation regions (PDRs) are regions of the interstellar medium where an exposition to a spectrum rich in UV photons leads to a high dissociation rate of interstellar molecules. A typical example is the Orion bar (Fig.1) where the massive star θ^1 Ori within the so-called Trapezium cluster irradiates the surface of its parent cloud. The structure and composition of these bright and compact regions is governed by a wealth of physical and chemical processes (magnetohydrodynamics, radiative transfer, light-matter interaction, heating/cooling of the gas and dust grains, chemistry in the gas and at the surface of dust grains, ...), making these regions an exceptional laboratory of interstellar physics.

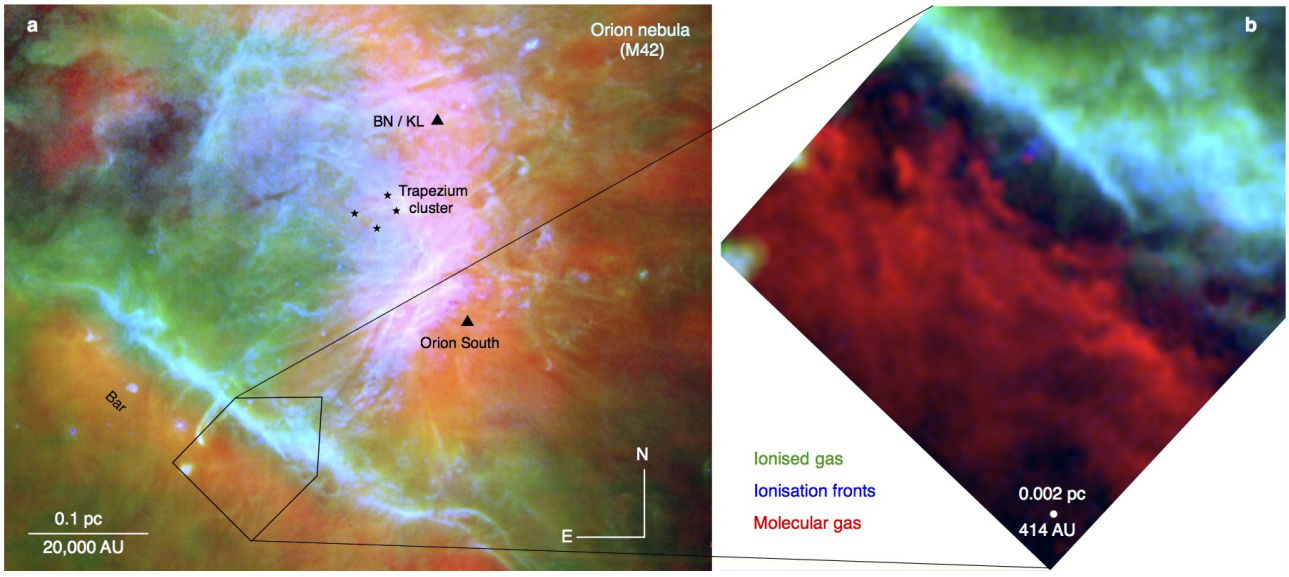

Figure 1 Multiphase view of the Orion nebula and molecular cloud. *(a) Overlay of the HCO^+ (J=3-2) emission (red) tracing the extended Orion molecular cloud. The hot ionised gas surrounding the Trapezium stars is shown by the [S II] 6,731 Å emission (green). The interfaces between the ionised and the neutral gas, the ionisation fronts, are traced by the [O i] 6,300 Å emission (blue), both lines imaged with VLT/MUSE. The size of the image is ~5.8' ×4.6'. (b) Close-up of the Bar region imaged with ALMA in the HCO^+ (J=4-3) emission (red). The black-shaded region is the atomic layer. From Goicoechea et al. (Nature, 2016)*

A major element in the modeling of PDRs is the thermal balance of the gas. It is determined by the cooling processes, dominated by the fine structure line emission of atomic oxygen (λ=63 µm) and of ionized carbon (C^+, λ=158 µm), and by heating. The latter is dominated by photoelectric heating due to interstellar dust grains: absorbing a UV photon by a grain can lead, photoelectric effect, to ejecting a high energy electron. The collision between this electron and atoms and molecules of the gas will transfer the electron's kinetic energy to the gas, which indeed corresponds to a heating process.

Photoelectric heating is often modeled based on the work by Bakes & Tielens (ApJ, vol. 427, p.822, 1994) where dust grains are idealized by graphite spheres. Yet, it is well-known that grains dense molecular clouds and in PDRs, may have a fractal structure, which may impact the efficiency of photoelectric heating.

I-2 Aim of the projet

This project consists in modeling, in a simple way, the photoelectric heating on a dust grain with a fractal structure irradiated by a UV flux. The students will use the provided python script to generate 2D

grain models. They will develop a python program to compute the velocity distribution of the electrons emitted by the grain (the “photoelectrons”). The computations will then be repeated for different fractal properties, to discuss the impact on the photoelectric heating.

II- Development environment

The project will be conducted with the python langage.

III- Description of delivrables

Python program:

- Input: ASCII file containing an array with 0 and 1 (integers). The first line will contain 2 integers corresponding to the dimensions (rows,columns) of the array.
- Output: ASCII file containing the list of energy values of photo-electrons in eV, at a rate of one value per line. When no photo-electron is emitted, one will use the values -1 to indicate that the UV photon was not absorbed, -2 to indicate that no photoelectric effect occurred after absorbing the photon, and -3 to indicate that the photo-electron recombined in the grain and therefore did not escape the grain.

IV- Description of tasks

Please read all the steps before starting the project. During the program development, it can be useful to start by a very simple grain, e.g. circular or squared, for which the results can be computed analytically, in order to check the numerical results.

III-1 Generate the grains

Use the python script `GenGrain.py` to generate an ASCII file containing a matrix of 0 and 1. This matrix represents a 2D image of the grain, the 1 (resp. 0) representing the image pixels occupied by the grain (resp. not occupied by the grain). Observe that multiple executions of the script lead to grains with different geometries, but with similar fractal properties. These properties are parametrized by 2 values that can be chosen at the beginning of the script. Explore the evolution of the grain geometry as a function of the values of these parameters.

III-2 Modeling the photoelectric effet

Create a python program in order to model the photoelectric effect. Your should follow the following steps:

a) Search the useful numerical values in the article by Bakes & Tielens.
b) Load the grain in a numpy array.
c) Randomly draw the initial position of a UV photon using one of python’s random number generators. For the sake of simplicity, all the photons will initially be in the first column.
d) Randomly draw the energy E of the photon. We will simply assume a white spectrum between 3 and 15 eV.
e) Identify the point of first contact of the photon with the grain. For the sake of simplicity, we will assume that photons move along the pixel rows, as if the grain were irradiated by a distant star in the direction of the matrix rows.
f) Randomly draw the distance d_a traveled in the grain by the photon before being absorbed. This distance follows an exponential probability distribution $P=\exp(-d_a/l_a) / l_a$, where l_a is the so-called “attenuation distance”, i.e. the characteristic distance of absorption. Derive the pixel in which the photon is absorbed. Only the distance traveled within the material must be counted. If the distance d_a is greater than the length of the grain along the photon trajectory, there is no absorption.
g) Randomly draw whether an electron is ejected. We will assueme that the ionization probability (also called *ionisation yield*) is parametrized by $Y=0.5\ (1+\mathrm{Th}[(E-E_0)/2])$, where $E_0=8$ eV.
h) If an electron is ejected, randomly draw the distance d_e traveled by the electron in the material before recombining with the grain. This distance follows an exponential probability distribution $P=\exp(-d_e/l_e) / l_e$, where l_e is the escape distance, i.e. the characteristic distance of recombination

of the electron with the grain. Randomly draw the direction in which the electron is ejected. Conclude whether the electron escapes the grain.

i) The kinetic energy of the ejected electron is $E_e = E - E_i$ where E_i is the ionization energy of the grain. Store this value in an ASCII file.

N.B. : A greater number of electrons leads to a better characterization of the distribution. It will therefore be useful, as long as the computation times remain reasonable, to maximize the number of UV photons.

III-3 Analysis of the results

Create a new python script to read the energy values of the photo-electrons, and represent their distribution, for example using a histogram. To discuss the effect of the fractal structure of the grains, it is possible to compare:

- the distributions obtained for several different grains with similar fractal properties,
- the distributions obtained for a fractal grain fractal and a spherical grain (circular, in 2D…),
- the distributions obtained for several grains with different fractal properties,
- the distributions obtained for several grains with different sizes but the same fractal properties.
- ...

III-4 More advanced version of the program

If the previous points are completed before the end of the “swimming pool”, more advanced versions of the program could include:

- The generalization of the initial position of the UV photon to any original location and propagation direction;
- Replace the spectrum of UV photons by a realistic stellar spectrum;
- The usage of the Bokeh python library (https://bokeh.org/) to propose an interactive view of your results.
- Make a 3D grain model.

V- Physics notions necessary for the project

Physical notions: ionization, recombination, light-matter interaction
Statistical notions: probability distribution function

VI- Description of the algorithms

For most of the project, common sense should be sufficient. To draw random numbers following a complex density distribution function the following steps can be used.
To randomly draw a value x following the probability density function $f(x)$ from a uniform random number generator:

1. Randomly draw a value of x with a uniform generator.
2. Randomly draw a number u between 0 and 1 with a uniform generator.
3. If u<f(x), keep x, otherwise, redo steps 2 and 3, until u<f(x).

It is advised to develop your non-uniform random number generator in a dedicated program, and to test it by plotting the distribution of the generated numbers. For a large amount of numbers, the distribution should tend to the curve of $f(x)$.

Supervisor: J. Montillaud