Computational Quantum Physics and Applications: Calculating the Informational Entropy of Atoms

Nikolaos Zafeiriadis

July 2025

Introduction

The aim of this project was to study the information entropy of the first 10 atoms of the periodic table (excluding Hydrogen) and its dependence on their atomic number. After creating a database rearranged in an optimal way, a script for reading it and returning the desired physical entities was written in Python programming language.

Theory

The main units that need to be evaluated are different types of information entropy. The first units of this matter is the Shannon's information entropy in position space S_r and in momentum space S_k, which are evaluated via the following formulas:

$$S_r = -\int \rho(\vec{r}) \cdot ln(\rho(\vec{r}))\, d\vec{r} \quad \text{and} \quad S_k = -\int n(\vec{k}) \cdot ln(n(\vec{k}))\, d\vec{k}$$

where $\rho(\vec{r})$ is the electron density distribution normalized to unity and $n(\vec{k})$ is the momentum density distribution normalized to unity. Another key unit is the total information entropy, which is given by:

$$S = S_r + S_k$$

The maximum information entropy accessible to the system S_{max} is given by:

$$S_{max} = 3(1 + ln(\pi)) + \frac{3}{2} ln\left(\frac{4}{9}\langle r^2\rangle \cdot \langle k^2\rangle\right)$$

Landberg's order parameter Ω is evaluated using:

$$\Omega = 1 - \frac{S}{S_{max}}$$

The last unit of measurement of information context is Onicescu's Information Entropy, which is given by the following formula:

$$O = \frac{1}{E_r E_k}$$

where E_r and E_k are evaluated via:

$$E_r = \int_0^\infty \rho^2(r) 4\pi r^2 dr \quad \text{and} \quad E_k = \int_0^\infty n^2(k) 4\pi k^2 dk$$

From the previous formulas, it becomes apparent that the mandatory functions for calculating the information entropy of a quantum system are the $\rho(\vec{r})$ and $n(\vec{k})$. In this project, those functions are obtained by applying the Roothaan-Hantree-Fock method. The atomic wave orbitals are considered spherically symmetric and the radial part is expanded as a finite superposition of primitive radial functions:

$$R_{nl}(r) = \sum_j C_{jnl} S_{jl}(r)$$

where the $S_{jl}(r)$ are taken as a Slater-Type Orbital (STO) set:

$$S_{jl}(r) = N_{jl} r^{n_{jl}-1} e^{-Z_{jl} r}$$

where n_{jl} is the principal quantum number, Z_{jl} is the orbital exponent, l is the azimuthal quantum number and N_{jl} is a normalization factor that is given by:

$$N_{jl} = (2Z_{jl})^{(n_{jl}+1/2)} / [(2n_{jl})!]^{1/2}$$

Implementation

The first step of the implementation was to define all the Slater-Type Orbitals as functions on a python file named *wavefunctions.py*. On this file, three dictionaries were also defined. The first two had as keys the names of the STOs and their values were the corresponding defined functions. The third dictionary had as keys the name of the radial orbital and returned the position that the script should look for the corresponding electron number. Those three dictionaries made the extracting of data from the database more efficient.

The second and hardest step of this project was "designing" and creating the database. Initially, the python OCR package *pytesseract* was used in order to extract the C_{jnl} and Z_{jl} of the atoms Z=2-10 from the scientific paper *Roothaan-Hartree-Fock Ground-State atomic wave functions: Slater-Type Orbital Expansions and expectation values for Z=2-54*. The OCR did not work perfectly so the procedure was not automated since it required a lot of corrections by hand. After the extraction of data, the database was designed and created using the following form:

Z	**orb**	**sto**	**zeff**	**coeff**
2	r1s	1s	1.4595	1.3479
2	r1s	3s	5.3244	-0.001613
⋮	⋮	⋮	⋮	⋮

Table 1: *The structure of the database.*

Where the first column contains the atomic number Z of the element, *orb* containing the name of the radial orbital of that the following STO belongs to, *sto* is the type of STO and *zeff* and *coeff* are the corresponding C_{jnl} and Z_{jl} of the atom. This database is found on the *df.xlsx* file. Along with that, a second database called *electronic_structures.xlsx* was created, that its first column is the Z atomic number of the element and the following columns are the names of the atomic orbitals, with their values being the number of electrons that this specific element has.

At the next step, since the database was created, the python script that read the database and evaluated the desired entities was written (*main.ipynb*). The script contained a function, called *information*, with its given parameters being the database it has to look, the electronic structure database and the atomic number Z of interest. The first thing that this function does is to read the databases and create the required radial orbitals and merge them together to create the $\rho(r)$ and $n(k)$ of the element. After that, the desired information entropy entities are easily evaluated and returned after the execution.

Results and Discussion

The results of this analysis are presented in the following graphs.:

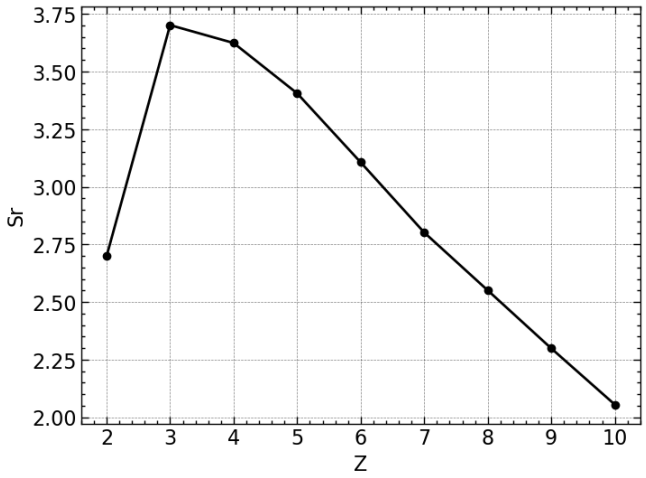

Figure 1: *The value of S_r with respect to Z.*

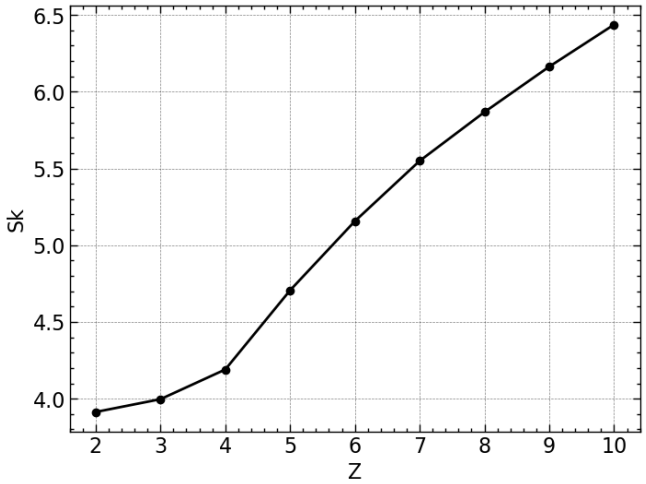

Figure 2: *The value of S_k with respect to Z.*

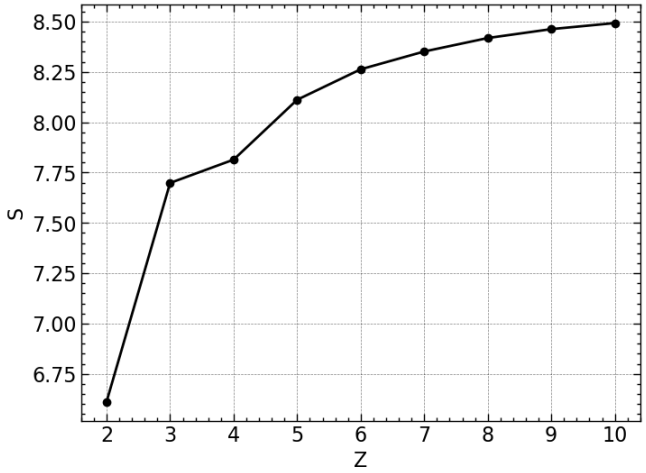

Figure 3: *The value of S with respect to Z.*

As it becomes clear from Figure (3), the total Shannon Information Entropy seems to have an increasing trend as the Z increases. The relationship between them seems to be a logarithmic and not linear. In the following figures, the other information entropy entities are presented as well.

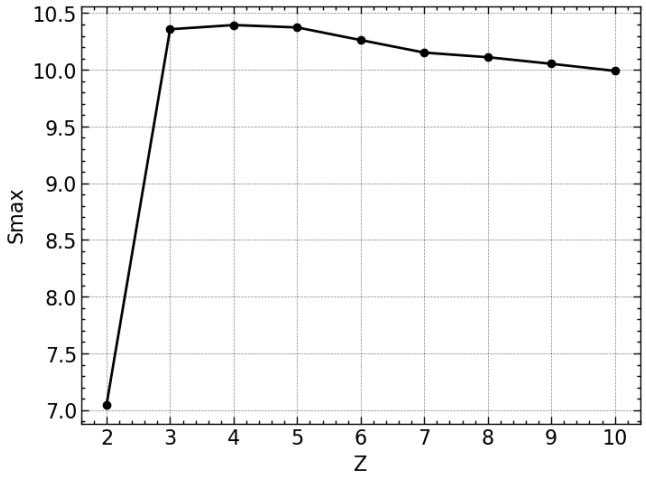

Figure 4: *The value of S_{max} with respect to Z.*

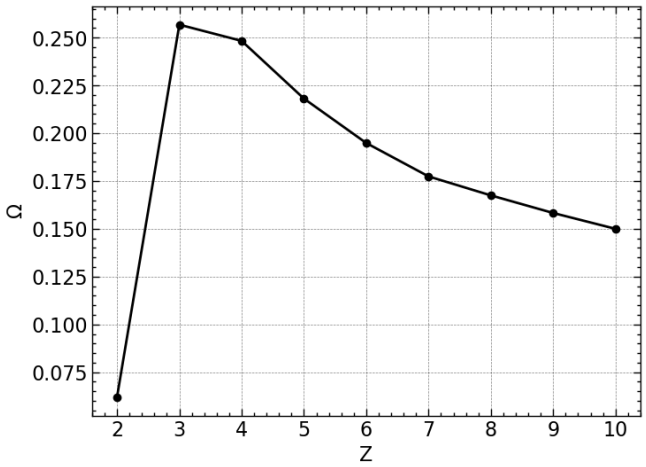

Figure 5: *The value of Ω with respect to Z.*

The last entity that is presented is the Onicescu's Information Entropy. Along with this entity, the experimental values of the ionization potential are shown on the same graph.

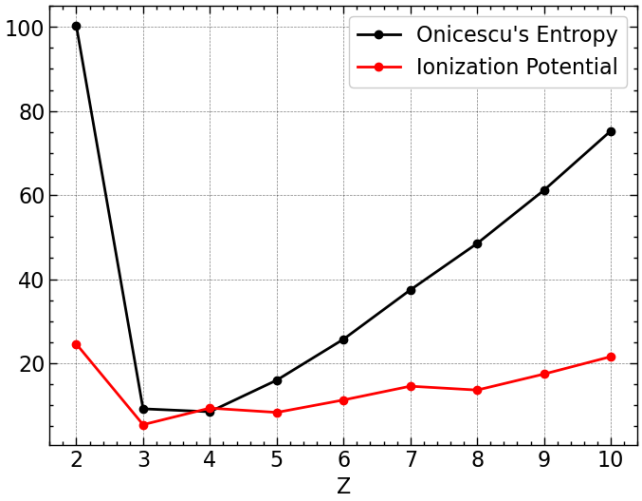

Figure 6: *The value of* Ω *with respect to* Z*, along with the ionization potential.*

By presenting both of these entities on the same graph, it becomes clear that the follow the same trend. Both of these have a sudden decrease in value moving from Z=2 to Z=3 and they also follow the same linear increasing trend from Z=4 to Z=10. That means that perhaps the information entropy of a quantum system can be correlated with other properties of the elements, that can even be measured experimentally.

Regression Model

As mentioned earlier, the S entropy and the Z atomic number seem to be correlated with a logarithmic relationship. Thus, an exponential regression relationship could be generated:

$$Z = \alpha e^{\beta S}$$

By some algebraic manipulation, the relationship took the following form:

$$ln(Z) = ln(\alpha) + \beta \cdot S$$

which is a linear form. Therefore, a simple linear regression model could be implemented in order to obtain a relationship between Z and S. Instead of using only the data that were evaluated using this analysis, the results of the scientific paper *Information entropy, information distances, and complexity in atoms* were used, which contained data for Z=2-54. Using the function *linregress* from the library *scipy* and this imported data, the following regression model was created:

$$Z = 0.0026 \cdot e^{0.9439 \cdot S}$$

The whole regression script is found on the *regression.ipynb*. The regression function is presented in the following figure, along with the data of the paper.

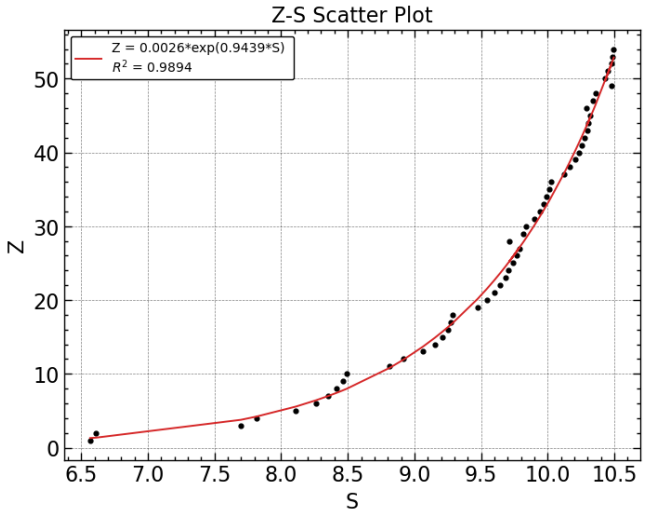

Figure 7: *Regression model using data from Z=2-54.*

Bibliography

1. Information entropy, information distances, and complexity in atoms (DOI: 10.1063/1.2121610)

2. Roothaan-Hartree-Fock Ground-State atomic wave functions: Slater-Type Orbital Expansions and expectation values for Z=2-54 (DOI: 10.1006/adnd.1993.1003)

3. Python Libraries: numpy, scipy, pandas, tesseract, matplotlib, science-plots