

In this post, I am going to show you guys how to read the atomic coordinates in the [.XYZ file format](#) using Python which is pretty much everyone's favorite programming language. With just a few lines of code, you'll be able to extract atomic symbols and coordinates from an XYZ file and even calculate some properties of the system, like the center of mass.

So, without further ado, let's get started!

To begin, we need to ensure that we have the NumPy library installed. If you haven't already, you can easily install it by running the following command in your terminal or command prompt:

```
pip3 install numpy
```

With NumPy ready to go, we can now move on to reading the XYZ file. Let's try to read the atomic coordinates of Benzene provided as an XYZ file [here](#). Download [this file](#) and save it in the same directory as the python script. We'll write a Python script that follows these three steps:

### 1. Reading the XYZ File:

We start by opening the XYZ file using the `open()` function, specifying the file name along with its extension. Then, we read the lines of the file and skip the first line, which typically contains the total number of atoms. Let's call the file object `xyz_file` for convenience.

```
with open('benzene.xyz', 'r') as xyz_file:
    lines = xyz_file.readlines()[2:] # Skipping the first two lines
```

### 2. Storing Atomic Symbols:

In the XYZ file format, each line after the first two represents an atom and contains information about its atomic symbol and coordinates. We can extract the atomic symbols by splitting each line and taking the first element. We'll store these symbols in a Python list called `atomic_symbols`.

```
atomic_symbols = []
for line in lines:
    atomic_symbols.append(line.split()[0])
```

### 3. Storing Atomic Coordinates:

Similarly, we can extract the atomic coordinates by splitting each line and selecting the second and third elements. We'll store these coordinates in a NumPy array for easy manipulation and further calculations.

```
import numpy as np

atomic_coordinates = np.array([line.split()[1:4] for line in lines], dtype=float)
```

Great job! We've successfully read the XYZ file, stored the atomic symbols in a list, and stored the atomic coordinates in a NumPy array. Now, let's move on to an example of how we can use this data for further calculations, such as finding the center of mass of the molecular system.

## Example: Calculating the Center of Mass

The center of mass is the average position of all the atoms in a system, weighted by their masses. To calculate it, we'll assume equal masses for all atoms.

```

mass = np.ones_like(atomic_coordinates[:,0]) # Assume equal masses for all atoms

# Calculate the center of mass
center_of_mass = np.average(atomic_coordinates, axis=0, weights=mass)

```

In this example, we assign a mass value of 1.0 to all atoms for simplicity. The `np.average()` function takes the atomic coordinates as input and calculates the weighted average along each axis (x, y, and z) using the given weights.

And that's it, folks! We've covered the basics of reading XYZ files in Python and demonstrated how to calculate the center of mass of a molecular system using the extracted data. With this knowledge, you can now explore and analyze molecular structures with ease.

For the sake of completeness, I will now rewrite the complete code from start to finish.

Remember to save your code in a Python file (`xyz_reader.py`) and run it with the XYZ filename of your choice.

```

import numpy as np

with open('benzene.xyz', 'r') as xyz_file:
    lines = xyz_file.readlines()[2:] # Skipping the first two lines

atomic_symbols = []
for line in lines:
    atomic_symbols.append(line.split()[0])

atomic_coordinates = np.array([line.split()[1:4] for line in lines], dtype=np.float64)

mass = np.ones_like(atomic_coordinates[:,0]) # Assume equal masses for all atoms
# Calculate the center of mass
center_of_mass = np.average(atomic_coordinates, axis=0, weights=mass)

print('Parsed contents:\n')
print('At | x | y | z |')
for iat in range(len(atomic_symbols)):
    print(atomic_symbols[iat ], atomic_coordinates[iat,0], atomic_coordinates[iat,1],
          atomic_coordinates[iat,2])
print('\n\nCenter of mass (x,y,z): ', center_of_mass)

```

Run the above code using

```
python3 xyz_reader.py
```

Feel free to modify and expand upon this code to suit your specific needs and explore other exciting possibilities in molecular data analysis.

Happy coding and happy computational material science!

## References:

- [https://en.wikipedia.org/wiki/XYZ\\_file\\_format](https://en.wikipedia.org/wiki/XYZ_file_format)
- [https://wiki.jmol.org/index.php/File\\_formats/Formats/XYZ](https://wiki.jmol.org/index.php/File_formats/Formats/XYZ)
- <https://www.wikidata.org/wiki/Q1250383>

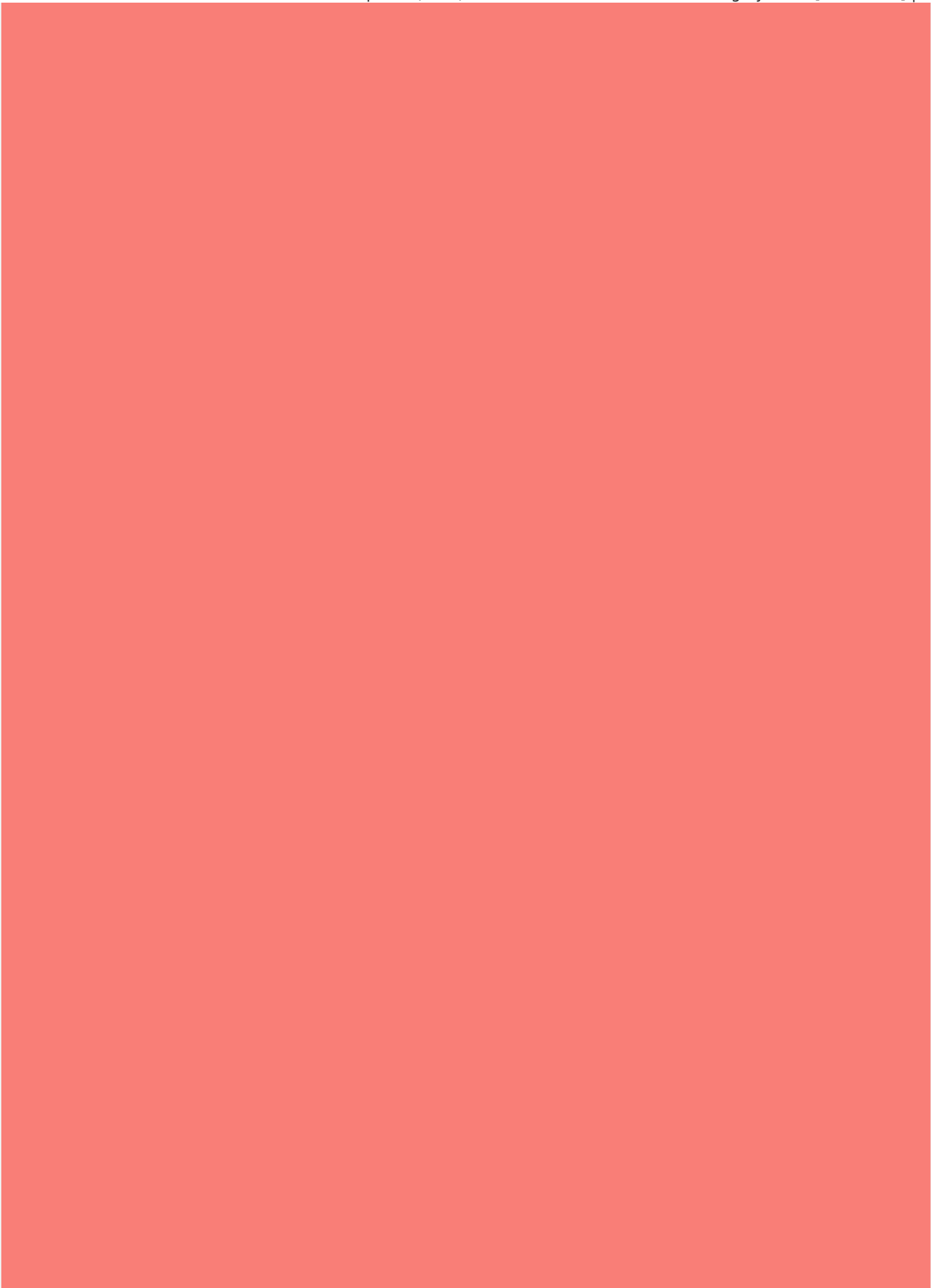


### Manas Sharma

I'm a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from time to time. Can code in most of the popular languages. I like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

[manas.bragitoff.com/](https://manas.bragitoff.com/)







**Share this:**

Click to share on Facebook (Opens in new window)

Click to share on Twitter (Opens in new window)

Click to share on WhatsApp (Opens in new window)

Click to share on Pinterest (Opens in new window)

Click to share on Reddit (Opens in new window)

Click to share on LinkedIn (Opens in new window)

Click to email a link to a friend (Opens in new window)

Click to print (Opens in new window)

Click to share on Tumblr (Opens in new window)

Click to share on Pocket (Opens in new window)

Click to share on Telegram (Opens in new window)

[wpdon id="7041" align="center"]