

Normal-Mode-Analysis

Science-2 Project Part A for Analysing the Normal Modes of an Argon System of 108 atoms following the Lennard Jones Potential. The code generates a initial random configuration of 108 atoms based on the given conditions, implements Periodic Boundary Conditions, Reduces the system of the random configuration using the Steepest Descent Algorithm for minimisation, then it generates a Hessian Matrix , and the eigen values and eigen vectors for it. It also plots a histogram of the frequencies. All Questions (5 of them) given as the project has been solved. A detailed report is included with theory and implementation details as Report.pdf.

Running the Code

```
python3 main.py
```

The code takes about 17 mins in total to run. This is because of many computations required for the project and python is very slow in such cases. It is worth noting that many expensive processes use parallel processing to speed up.

Running this, generate all files in the folder. Submitted files are in data folder. Also since the project is highly modular you are able to split the tasks up in main.py and do it piece by piece. Also different parts of the project can be used as python modules after a few changes.

Directory Structure

Following is the directory and code structure

- **data** : This folder contains on the files that are submitted as part of the solution for this project.
 - **eigen_values.dat**- The eigen values for the submission
 - **eigen_vectors.dat**- eigen_vectors corresposing to these
 - **hessian.dat**- The hessian matrix as part of the submission
 - **hist.png** - The histogram as part of submission
 - **init.png**- The VMD Visualisation of the initial random configuration
 - **modes.xyz**- The final normal modes of the system after all calculations in the xyz format. The format is decribed in report.
 - **molecule.xyz**- The initial random configuration in the xyz format
 - **new_molecule.xyz** - The new molecule configuration after minimising the energy using the steepest descent algorithm in the xyz format.
 - **new.png** - The VMD Output for the minimised configuration
- **energy.py**: The file contains the **get_energy** method which takes in a geom and gives the Total Energy of the system following the Lennard Jones Potential.
- **frequencies.py** : Contains the **Frequencies** Class which is responsible for normal mode calculation from Hassien Matrix, and given molecular structure.
- **grad.py** : The home of the gradient descend Algorithm. It creates a type of generalised Gradient Descend Algorithm to calculate minimum energy configuration. It is worth noting that this code can be generalised and it is modular and can take in any possible function.
- **hessian.py**: Contains **Hessian** class which is responsible for Hessian Matrix calculation, eigen values and eigen vectors.The code uses the Finite Differences Method for Partial derivatives which

have been explained in the Report. Uses Parallel Processing to speed up the code.

- `init.py`: Module responsible for creating the initial Configuration
- `main.py` : Entrypoint for the whole Project
- `molecule.py`: Home of the Molecule Class, a kind of Datastore for the whole Project. It represents a system.
- `pbc.py`: Some Helper Functions that are useful in implementing the Periodic Boundary Conditions.
- `Report.pdf`: A detailed report on the theory, implementation details etc. Please read it. Took a lot of time to write it.
- `requirements.txt` : The python3 requirements to run the code.