

# Molecular Dynamics of Benzene

Nik Awtrey, Justin Gens, Ricky Sexton

Repository URL: <https://github.com/ASU-CompMethodsPhysics-PHY494/final-2019-benzene>

# Objectives

1. Develop a MD code that solves Newton's equations of motion for a molecule in a vacuum
2. Generate initial conditions of benzene molecules based on online resources
3. Develop an algorithm that outputs initial positions of atoms for a benzene molecule, and assign velocities to all atoms in molecule based on temperature
4. Develop a force function that can be integrated via the velocity Verlet algorithm
5. Compile a list of parameters from existing force-fields for the energies and spring constants to produce a stable molecule
6. Implement restoring force to create an improper dihedral structure
7. Find parameters that produce a stable, fairly rigid model of a benzene molecule
8. Simulate interactions of multiple benzene molecules
9. Deduce heat capacity of benzene from system of molecules

# Background

We simulated the interactions of benzene ( $\text{C}_6\text{H}_6$ ) molecules using a classical approach with molecular dynamics methods to produce a stable simulation of benzene and find the parameters which reproduce experimentally determined properties of benzene. Benzene is made of an inner ring of carbon atoms with one hydrogen atom connected to each carbon atom, which means we must account for bonded (C-C single, C-C double, H-C single) and non-bonded (C-C and H-C) interactions within each molecule. The Morse potential was used to simulate bonded intramolecular interactions,

$$V_M(r_{ij}) = D_{ij} \left( 1 - e^{-\beta_{ij}(r_{ij} - R_{ij})} \right)^2.$$

Intermolecular and non-bonded intramolecular interactions were simulated using the Lennard-Jones Potential,

$$V_{LJ}(r_{ij}) = 4[(r_{ij}^*)^{-12} - (r_{ij}^*)^{-6}], \quad r_{ij} = \sqrt{(\mathbf{x}_j - \mathbf{x}_i)^2}$$

The motivation behind the project was to be able to study benzene or other molecules if a working model was able to be developed. If a realistic model of the molecule was created, it would be possible to study properties of this or other molecules on length and time scales that are otherwise not accessible.

# Methods

The interesting (and difficult) part of this project was to figure out a way to calculate the forces with different atoms. Tracking multiple masses through the calculations proved difficult but was solved by creating a labeling system for each atom in every molecule and creating a dictionary that could be used to find the neighbors of each atom in a single molecule. The atoms were assigned indices that were output to a data file along with their masses, molecule number, and the element name (H or C). Since the force due to the Morse potential only applies to bonded atoms inside a single atom, it was necessary to create a dictionary of dictionaries that contained a list of the indices of each atom's neighbors, next nearest neighbors, the neighbors of those neighbors, and so on. This dictionary was created for the first molecule and was supplemented with a function that could find the indices for any other molecule, since the index numbers would only be off by a multiple of 12. With both solutions in place, it was possible to write functions that calculated the forces due to the Lennard-Jones and Morse potentials and then integrate via the velocity Verlet algorithm.

### Bond Lengths:

- C-C: 0.139 nm
- H-C: 0.109 nm

### Dissociation Energies:

- C-C: 922 kJ/mol
- H-C: 110 kJ/mol

### Force Constants:

- C-C: 1.49E11
- H-C: 9.685E10

### Lennard-Jones Parameters:

- Epsilon = 1.198E-7 Kelvin
- Sigma = 0.341 nm

### Packages Used:

- NumPy
- SciPy

# Intramolecular Force Function:

```
def a_intra(neighb_array, dist_array, data, x):
    accel = np.zeros((3, len(x), len(x)))
    for i in range(len(x)//12):
        for j in range(len(neighb_array)):
            for k in range(j):
                if (neighb_array[j, k]==2) or (neighb_array[j, k]==3) or (neighb_array[j, k]==0):
                    continue
                elif neighb_array[j, k]==1:
                    vec = x[12*i+j] - x[12*i+k]
                    bond = data[12*i+j][2] + data[12*i+k][2]
                    r2 = dist_array[12*i+j, 12*i+k]
                    accel[:, i*12+j, 12*i+k] = functions.F_M(r2,bond)*vec/(dist_array[12*i+j,12*i+k]*data[12*i+j][3])
                    accel[:, 12*i+k, 12*i+j] = -accel[:, 12*i+j, 12*i+k]
                elif neighb_array[j, k]==4:
                    vec = (x[12*i+j] - x[12*i+k])
                    r2 = dist_array[12*i+j, 12*i+k]
                    accel[:, i*12+j, 12*i+k] = 0.5*functions.F_LJ(r2)*vec/(dist_array[12*i+j, 12*i+k]*data[12*i+j][3])
                    accel[:, 12*i+k, 12*i+j] = -accel[:, 12*i+j, 12*i+k]
                elif neighb_array[j, k]==5:
                    vec = (x[12*i+j] - x[12*i+k])
                    r2 = dist_array[12*i+j, 12*i+k]
                    accel[:, 12*i+j, 12*i+k] = functions.F_LJ(r2)*vec/(dist_array[12*i+j, 12*i+k]*data[12*i+j][3])
                    accel[:, 12*i+k, 12*i+j] = -accel[:, 12*i+j, 12*i+k]
    accs1 = np.transpose(np.sum(accel, axis=2))
    accs2 = functions.constraints(x, data)
    return accs1+accs2, accel
```

# Results

The main goal of creating a working model of benzene was not accomplished. There were several forces implemented in the code and getting the correct balance to create a stable molecule proved difficult. A seemingly stable molecule becomes unstable after a short period of time and the atoms disperse. The Morse potential was used for the force for bonded interactions and Lennard-Jones (half-weighted Lennard-Jones) interactions were added for fifth (fourth) nearest neighbors. If Lennard-Jones forces are added between all molecules, the physics becomes more realistic, but the problem of the instability persists. A harmonic restraint force was added to each atom relative to its' equilibrium position in an attempt to stabilize the molecule. This worked for a short time but again the molecule disintegrated. More fine-tuning needs to be done in order to get the molecule stabilized, which then can be tested for accuracy by comparing the parameters used and observables obtained to experiment.



## Summary

This simulation would likely be improved by implementing more forces, such as the dihedral force to account for torsional angles between the plane that carbon and a hydrogen atom reside in, and the Coulomb force to account for charges between atoms. Our project could be expounded on in the future by implementing quantum mechanical interactions to more accurately simulate the structure of benzene.

## Acknowledgements

Our group would like to thank Ian Kenney for his assistance with developing an approach for tracking individual atoms throughout the calculation process.





