Accurate 3D-Modeling of User Inputted Molecules

Ben Parr
Computer Systems Lab 2007-2008
Thomas Jefferson High School for Science and Technology
Alexandria, Virginia

June 10, 2008

Abstract

In order to better understand the structure of molecules, chemists create 3 dimensional molecular models. These models help explain reactivity, polarity and many other characteristics pertinent to chemistry. In a large introductory chemistry class, it is too costly to provide each student with a physical modeling set. Software available online can also be costly. The goal of my research project is to create a free application that will allow users to generate 3D models of simple molecules through an intuitive and user friendly interface. Then, after the model is created, the application will position the atoms in the molecule correctly by using a Nelder Mead algorithm. My project will help users better understand the geometry of different molecules.

1 Introduction

Almost every student has to take an introductory chemistry course sometime during their education. One of the keys to understanding chemistry is understanding the geometry of molecules because the shape of a molecule affects many characteristics of the molecule. Since there are so many introductory chemistry students, viable options for providing experience with molecular geometry are limited because providing each student with a physical modeling set or a professional application can be too costly. The goal of my project is to create a free application that will allow users to easily create accurate models of simple molecules. The user will not need to have any knowledge of molecular geometry prior to using the application because while running the application, users can click the "Auto-Position Atoms" option. This option will attempt to position the atoms correctly by using a Nelder Mead algorithm to minimize the energy of the current model. Therefore, by using my application, users will be able to have a complete understanding of the geometry of simple molecules, and thus have a strong foundation to build from during the rest of their chemistry courses.

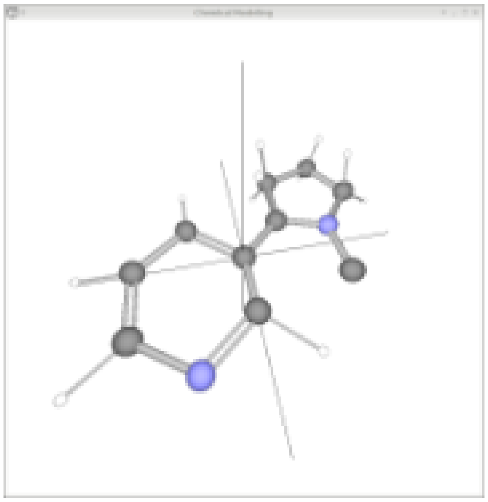


Figure 1: Model of nicotine created by my application.

2 Background

Many research labs have worked on applications to model molecules, and the techniques to do so now have become quite advanced. Pharmaceutical companies have invested thousands of dollars in applications to predict orientations of new complex molecules. Large databases have been created, storing the orientations of thousands of molecules. Chemists have modeled everything from RNA to inorganic crystals. (Haussler) These applications have become increasingly more accurate over the years. However, the cost and complexity of these applications have also increased. Alternative options, such as physical modeling sets and other applications online can cost up to hundreds of dollars each. Therefore, for a class with hundreds of introductory chemistry students, many of their options are too costly.

In order to allow users learn about the geometries of different molecules, my application will attempt to position the atoms correctly when the user chooses the "Auto-Position Atoms" option from the menu. This function will use a Nelder Mead algorithm to minimize the energy of the model. The Nelder Mead algorithm was created by Nelder and Mead in 1965, and uses the concept of a simplex in order to minimize a function in a many-dimensional space. (Price) A simplex is a polytope of N+1 vertices, with each vertex containing N dimensions; therefore, for a problem with two variables, the simplex is a triangle in 2D space, and for a problem with three variables, the simplex is a tetrahedron in 3D space. (See Figure 2) The algorithm begins with an original simplex, which must be large enough to avoid a local search that would end when a local minimum is found instead of the absolute minimum. The simplex should also not be too large because a large simplex could affect efficiency. (Kelly) The algorithm runs until the difference between maximum and minimum energies of the simplex is less than a defined tolerance. My application uses a variation of the Nelder Mead algorithm outlined in "Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions," which contains five steps for each iteration of the algorithm: order, reflection, expansion, contraction and shrink. (Lagarias) Through these steps, the algorithm finds the minimum energy.

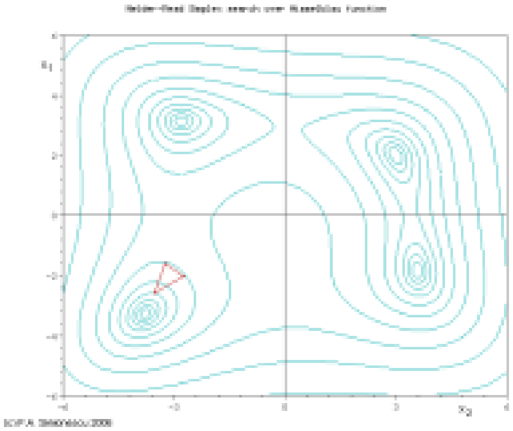
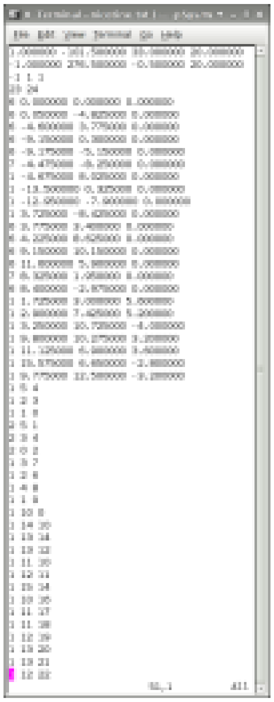


Figure 2: An example of the Nelder Mead algorithm using a simplex (red triangle) to find the minimum of a problem with two dimensions. (Image from `web.cecs.pdx.edu`)

3 Graphics and User Interface

The first part of the goal of my project was to create an easy and intuitive interface. In order to do this, I utilized OpenGL, which would handle both the graphics and the user inputs. In order to represent molecules, I took the standard route: a much simplified version where spheres represent atoms and cylinders represent bonds. In order to allow users to draw and manipulate these models, my application handles input from both the keyboard and mouse. Through this hardware, users can create atoms and bonds, select and delete atoms and bonds, import and export models, draw single, double and triple bonds, choose which element they want to draw, and position the atoms where they want. The application uses empirical values of atomic radii derived by J.C. Slater in order to draw atoms that are proportionally correct. (Slater) With these features, users can easily and quickly create models.


Terminal - nicotine.txt (~/...)
File Edit View Terminal Go Help

```
1.000000 -101.500000 55.000000 20.000000
-1.000000 276.500000 -0.500000 20.000000
-1 1 1
23 24
6 0.000000 0.000000 0.000000
6 0.850000 -4.825000 0.000000
6 -4.600000 5.775000 0.000000
6 -9.250000 0.300000 0.000000
6 -9.275000 -5.150000 0.000000
7 -4.475000 -9.250000 0.000000
1 -4.675000 8.025000 0.000000
1 -13.500000 0.325000 0.000000
1 -12.950000 -7.900000 0.000000
1 3.725000 -8.425000 0.000000
6 3.775000 3.400000 0.000000
6 4.225000 8.425000 0.000000
6 8.150000 10.150000 0.000000
6 11.800000 5.900000 0.000000
7 8.325000 1.050000 0.000000
6 8.400000 -3.875000 0.000000
1 1.725000 3.000000 5.800000
1 2.800000 7.425000 5.200000
1 3.250000 10.725000 -4.000000
1 9.800000 10.275000 3.200000
1 11.125000 6.800000 3.600000
1 15.575000 6.650000 -2.800000
1 9.775000 12.500000 -3.200000
1 5 4
1 2 3
1 1 8
2 6 1
2 3 4
2 0 2
1 5 7
1 2 6
1 4 8
1 1 6
1 10 8
1 14 10
1 13 14
1 13 12
1 11 10
1 12 11
1 15 14
1 10 15
1 11 17
1 11 18
1 12 19
1 13 20
1 13 21
1 12 22
                                52,1          All
```

Figure 3: File created when the nicotine model was exported. This file can then be imported at a later time.

4 Nelder Mead Algorithm

The second part of the goal of my project was to have the application orient the atoms of the molecules accurately for the user. In order to do this, I implemented a variation of the Nelder Mead algorithm. (Lagarias) The dimension (N) of the problem depends on how many atoms there are in the model. Since each atom has a three values (x, y, z), N = 3 * (the number of atoms) in the model. The Nelder Mead algorithm uses a simplex with N + 1 vertices in order to find the minimum of the energy function. Each vertex of the simplex contains N dimensions and represents a different model orientation. Therefore, the amount of memory needed in the algorithm grows drastically as the number of atoms increases. In order to make my code efficient, I used pointers in C. Therefore, instead of having to pass a large array of N elements, I just pass a pointer, saving time and memory.

The energy function can be broken up into three parts. The first part calculates the distance of each pair of non-bonded atoms. Then, the function adds the square of the inverse of this distance to the running total of the energy of the model (energy += 1/(distance*distance)). Therefore, by trying to minimize the energy function, the Nelder Mead maximizes the distance between non-bonded atoms, which will produce valid models because real atoms do push away from each other. The next part of the energy function calculates the absolute value of the difference between the current distance and the real distance between bonded atoms. The current distance is the distance between the two bonded atoms in the current orientation. The real distance is the distance defined in a data file which contains distinct bonds (based on what two atoms are bonded and what type of bond it is: single, double or triple bond) and the corresponding bond lengths derived from experiments. ("Bond Lengths and Energies") Therefore, the Nelder Mead algorithm tries to get the distance between bonded atoms as close as possible to the actual values. The third part of the function calculates all of the bond angles in the model and adds the negative of the angle to the energy of the function. Therefore, the bond angles are maximized, further pushing atoms away from each other. Since the Nelder Mead function tries to minimize the energy of the system, the minimum energy will occur when non-bonded atoms are as far apart and bonded atoms are the specified distance apart.

5 Results

The first part of my goal was completely met; all graphics and user interface components have been tested and work as expected. In order to test this component of my project, I tried every variation of input and observed the results. Each case was handled correctly.

The second goal of my project was not completely reached. The Nelder Mead algorithm does not work perfectly for every simple molecule. Specifically, as

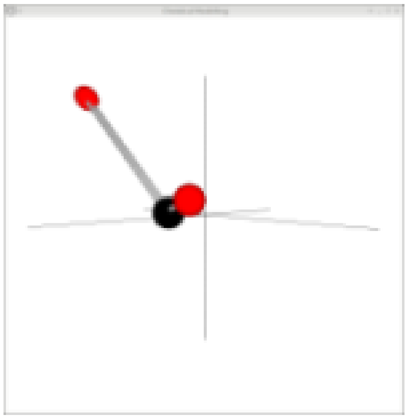
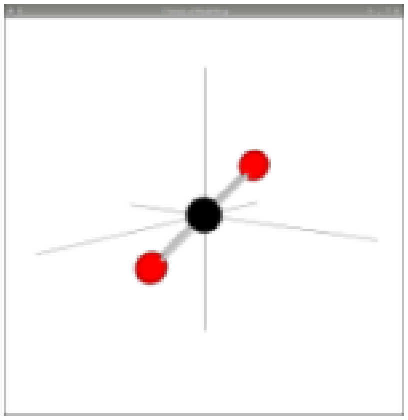


Figure 4: Above: Incorrectly oriented model of carbon dioxide before using the "Auto-Position Atoms" option. Below: Result of the "Auto-Position Atoms" option on the above model of carbon dioxide.



the number of atoms in the model increase, the error with the model created by the "Auto-Position Atoms" option increases. Molecules with only two atoms work perfectly; the two atoms are the specified distance apart after the application runs the Nelder Mead algorithm.

Molecules with three atoms are almost correct. For example, in the model of carbon dioxide, each oxygen should by 121 picometers from the carbon atom. The bond angle between the three atoms should also be 180 degrees. For a series of 5 tests on an incorrect orientation of carbon dioxide, the average percent error in the bond length was .0224%. The average percent error in the bond angle for the same 5 tests was .1822%.

The percent error increases as the number of atoms increase. For example, the result of the Nelder Mead algorithm on a model of benzene will be incorrect. However, as apparent from Figure 5, the ring of carbon atoms is created. Even though the auto-positioning of the atoms is not 100 percent correct, the output from the Nelder Mead algorithm should still produce a helpful model.

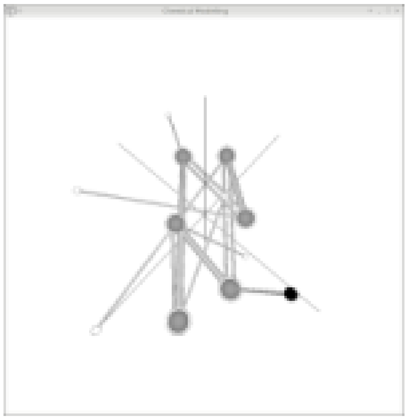
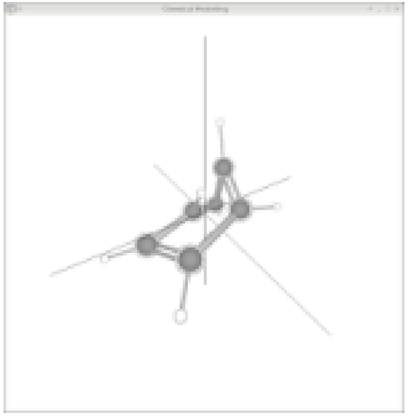


Figure 5: Above: Incorrectly oriented model of benzene before using the "Auto-Position Atoms" option. Below: Result of the "Auto-Position Atoms" option on the above model of benzene.



6 Conclusion

OpenGL was the correct choose for this application because it allowed me to easily create the graphics and user interface. By using C, I was able to make the Nelder Mead algorithm much more efficient by utilizing pointers. The Nelder Mead algorithm was also sufficient for this application. Instead of implementing complex derivatives that are needed in other minimizing algorithms, the Nelder Mead algorithm used basic math and simple steps in order to reach a minimum. The current Nelder Mead algorithm in the application, however, has two main problems. The first problem is with large molecules. The Nelder Mead should be able to find better models than it currently does. By refining the Nelder Mead algorithm, the results of the algorithm will become closer to real models, even for larger molecules. Second, the algorithm does not account for polarity in the model. Therefore, a model of water will be auto-positioned into a linear molecule. However, this would be incorrect because a water molecule is bent because of the lone electron pair on the oxygen atom. Even though there are two problems with the Nelder Mead algorithm, the application would still be helpful for introductory chemistry students because it could, at the very least, give a rough model of the molecule. However, fixing the two problems would allow the application to create more accurate models.

References

[1] ”Bond Length and Energies”, available at `http://www.science.uwaterloo.ca/ cchieh/cact/c120/bondel.html`

[2] Fox, M. A., and Whitesell, J. K., ”Bond Lengths in Organic Compounds”, *Organische Chemie*, 1994.

[3] Haussler, D., Brown, M. P. S., ”Rna modeling using stochastic context-free grammars”, University of California, Santa Cruz , 1999.

[4] Kelly, C. T., ”Detection and Remediation of Stagnation in the Nelder-Mead Algorithm Using a Sufficient Decrease Condition”, *SIAM Journal on Optimization*,1999

[5] Lagarias, J. C., Reeds, J. A., Wright,M. H., and Wright, P. E., ”Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions”, pp. 112-147, *SIAM Journal on Optimization*, 1998.

[6] Price, C. J., Coope, I. D., Byatt, D., ”A convergent variant of the Nelder-Mead algorithm”, *Journal of Optimization Theory and Applications*, 2002

[7] Shreiner, D., Woo, M., and Neider, J., OpenGL Programming Guide: The Official Guide to Learning OpenGL 6th Edition, Addison-Wesley Professional, 2007.

[8] Slater, J. C., “Empirical Atomic Radius”, *J. Chem. Phys. 39*, p. 3199, 1964.

[9] Weast, R. C., Astle, M. J., and Beyer, W. H., CRC Handbook of Chemistry and Physics, CRC Press, 1984.

[10] ”OpenGL API Documentation”, Copyright 1997 - 2008, available at `http://www.opengl.org/documentation/`