Accurate 3D-Modeling of User Inputted Molecules Using a Nelder Mead Algorithm Computer Systems Lab, 2007-2008

Ben Parr

April 4, 2008

Abstract

In order to better understand chemistry, chemists create 3 dimensional models of molecules. 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 program that will allow users to generate 3D models of simple molecules through an intuitive and user friendly program. After the model is created, the program will then position the atoms in the molecule correctly by using a Nelder Mead algorithm. My project will help people, especially students, better understand the geometry of different molecules.

1 Introduction

The goal of my project is to create a free program that will allow users to easily and intuitively create models of simple molecules. While running the program, users can create and position atoms and bonds however they wish. Once they have created their model, with a click of the mouse, the user can run the "Auto-position atoms" function. This function will attempt to position the atoms correctly by using a Nelder Mead algorithm to minimize the energy of the system (molecule). The user can save the model and current configurations, such as the eye position, by exporting it. The exported model can then be imported easily at a later time. Models are rotatable and

zoomable at all times, allowing users to better understand the geometry of the molecule.

My project is coded in C using OpenGL. The first step of my project was creating the graphics for my program. The second step was creating an intuitive user interface that would allow users to easily create molecules. The third step is to create the "Auto-position atos" function. I have finished programming and lightly testing my Nelder Mead algorithm. Before I can fully test the algorithm, I need to finish programming the energy function and the original simplex.

2 Background

A lot of research has been done on modeling molecules and the techniques to do so now have become quite advanced. Pharmaceutical companies have invested thousands of dollars in programs 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. These programs have become increasingly more accurate over the years. However, the cost of these programs has also increased, and now very few people have access to them. For a beginning chemistry student there are not many options to play around with molecules and learn their different geometries. Buying physical modeling sets for everyone in a introductory chemistry course is costly. Also, software on the Internet can also be costly, sometimes a couple hundred dollars. Therefore, even today, a free program to model simple molecules would be helpful.

One of the main features of my project will be the "Auto position atoms" function. 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 is uses the concept of a simplex in order to minimize a function in a many-dimensional space. A simplex is a polytope of N+1 vertices in N dimensions; therefore, it is a triangle in 2D space and a tetrahedron in 3D space. The algorithm begins with an original simplex. This simplex can not be too small, because it could lead to a local search. A simplex that is too large could a noticeably longer time. Before each iteration of the algorithm, the difference between the energy of the maximum energy vertex and the energy of the minimum energy vertex is tested against a tolerance. If the value is less than the tolerance, than the algorithm ends, and a result has

been reached. If the value is greater than the tolerance, the program runs through another iteration. Each iteration consists of five steps: order, reflection, expansion, contraction and shrink. Through these steps, the algorithm finds the minimum of the energy.

The energy function can be broken up into two parts. The first part is the distance of each atom to every other atom. The program will try to maximize this distance because real atoms do push away from each other. The second part is making the distances of bonded atoms a specific distance apart. This distance is specified in a data file and depends on what two atoms are connected and what type of bond exists between them. Since the Nelder Mead function tries to minimize the energy of the system, the minimum energy will occur when the atoms are as far apart from each other, except for bonded atoms which are the specified distance away from each other.

3 Development

The first step to my project was creating the graphics for the models. The graphics for my project are a much simplified version of real molecules: spheres represent atoms and cylinders represent bonds. Nevertheless, the results are sufficient. Testing during this step was simple; all I had to do was run it and see if my program created the intended model.

The second step of my project was creating an easy and intuitive user interface. Through mainly inputs from the mouse (and some keyboard input) 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. Using empirical values of atomic radii derived by J.C. Slater and published in The Journal of Chemical Physics, the program creates atoms that are proportionally correct in size. With these features, users will be able to easily create the models. Testing this part of my project provided some difficulties that I had to overcome because I had to think of everything that a user could do and account for the inputs.

The third step of my project was creating the Nelder Mead algorithm that will correctly orient the atoms in the molecule. The dimension (N) of the problem will depend on how many atoms there are in the model. Since each atom has a x, y, and z (3 values), N = 3 * (the number of atoms) in the

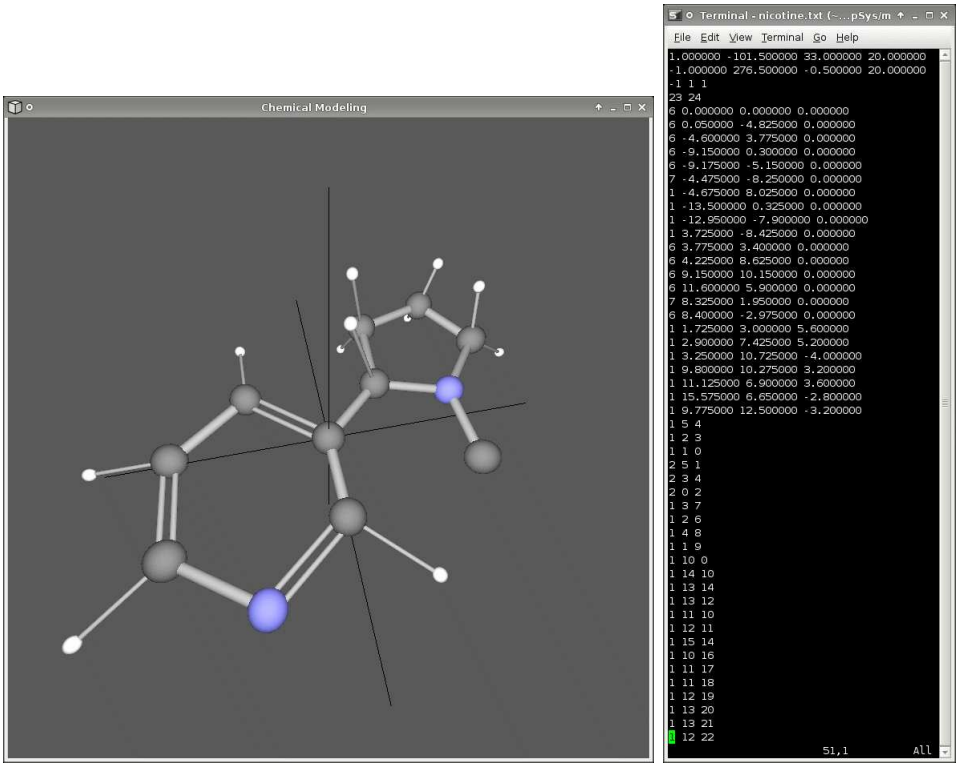

Figure 1: A model of nicotine drawn by using my project, and the file created by exporting the model.

model. The Nelder Mead algorithm uses a simplex will have N + 1 vertices, each with N dimensions. Therefore, the number of variables needed in the problem grows drastically as the number of atoms increases. In order to make my code efficient, I used pointers. Therefore, instead of having to pass a large array of N elements, I can just pass a pointer, saving time and space.

Dynamic testing will not work for my project because an atom can not bond with any random atom. Therefore, I will mostly use specific structural and functional testing and path and branch testing. In order to see if my program is constructing accurate models, I will compare my models with the actual/accepted models. By doing this, I will also be able to do error analyses. In order to thoroughly test my program, I will test many different molecules and compare my results with the actual orientations.

4 Results

The purpose of my project is to create a free program that will allow users to easily and intuitively create models of molecules. Then, after a user finished creating the molecule, the program will correctly position the atoms. My project currently allows users to create molecules. A user can also rotate the model and zoom it in and out. These functions help users get a better picture of the molecule. I have also finished programming the Nelder Mead algorithm. The focus for fourth quarter will be programming the energy function and the original simplex, and then thoroughly testing the "Auto position atoms" function.

References

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

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

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

[4] D. Shreiner,M. Woo, and J. Neider,

OpenGL Programming Guide: The Official Guide to Learning OpenGL 6th Edition, Addison-Wesley Professional, 2007.

[5] J.C. Slater, "Empirical Atomic Radius", *J. Chem. Phys. 39*, p. 3199, 1964.

[6] R.C. Weast, M.J. Astle, and W.H. Beyer,

CRC Handbook of Chemistry and Physics, CRC Press, 1984.

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