Computer Systems Research Paper

Masood Malekghassemi
Computer Systems Lab, 2007 - 2008

Abstract

Computational chemistry is no new subject – however, the idea of taking the process from including a human being in the loop between ab initio methods and general rule making for molecular mechanics calculations to excluding that same being via AI seems to be unimplemented. This project revolves around the idea that molecular mechanics is an incredibly fast and viable way of calculating the geometry of molecules and that as such, given the proper parameterizations and generalities, can be as accurate as or more accurate than ab initio methods.

Introduction

Molecular mechanics is a generally inaccurate but quick and painless way of calculating numerically the shape of an atom. It does not involve the complex numeric computations of ab initio methods – only a classical mechanics perspective of the molecule. Given the proper parameterizations, it's a highly accurate method of calculating the shape of complex molecules such as cholesterols and enzymes. An ab initio method would take months to do the same kinds of calculations on a lone desktop computer. However, individual parts of that molecule can be calculated ab initio. This information can then be generalized and stored as a rule for the way the molecule shapes itself.
The rule by itself is not very useful: however, put into the context of similar molecules of unknown shape, can be very predictive. Such a rule can be created from an ab initio run and generalized by a computer, an AI of sorts. Thus, the basic scheme of things running:

ab initio --------> AI --------> Molecular Mechanics

Background

I have been reading up on computational chemical methods, such as those for solving the multi-body Schrodinger equation and the simple ball and springs model. I have three books on the topics I'm interested in. One of them in fact has the title 'Computational Chemistry: Introduction to the Theory and Applications of Molecular and Quantum Mechanics.' The title in itself should give away that this is no novel topic – the idea of using numeric methods to solve chemical problems has been around for a long time, since the late 1930s (for computers specifically, the late 1950s or 60s). However, all of these programs appear to be written in cryptic Fortran code with non-descriptive variable names and eyesores based on the fact that Fortran 77, which all (save for one that I've found) of the programs are written in, is an incredibly ugly language due to its usage of character columns for semantic purposes. So, really, a re-write of these projects in a language that's high level and just as suited for numeric programming (optimizing C++ compilers makes the language a fair competitor with Fortran) could in itself be a whole new project, but I digress. There exists no 'state of the art' project comparable to the generality of what I'm doing. There are many computational chemistry programs out there – but there certainly aren't any programs that have an AI to generate rules for molecular mechanics calculations from runs of ab initio methods.

Procedure and Methodology

My project is split into three parts. The first of them, the molecular mechanics section, I will definitely

be done with before December (the I/O from files is being a pain in the neck, I haven't been able to decide very easily how I want my input files to be formatted). This first part is split into the Rules, the Identities, the I/O, and the System/ForceField calculations. I've finished the Rules and Identities, am almost done with the I/O, and due to my taking of the easy way out (using an iterative SCF method rather than a Lagrangian approach), the System/ForceField calculations will be a breeze (unless I hit a wall, like needing to refactor my code). The second part, the ab initio methods implementation, will probably end up using computational Hartree-Fock and/or computational Density Functional Theory for its operations (which won't need to be made generic, as it is purely a solver, not a rule maker). The final part will be the trickiest – an AI that will be able to read the data coming out of the ab initio approaches and transfer that to the generic rule based molecular mechanics part. This last part is the one part that I'm unsure of whether or not I'll be able to finish this year. If I have little schoolwork come Spring, it will be done – if I don't, well, you can say that my project was backlogged.

Testing

My program will be tested based on the functionality of the individual parts. The molecular mechanics part will be considered a limited success if I can take a methane molecule (a C with four H attached) with non-tetrahedral bond angles and get a tetrahedron out of it. It will be considered a general success if I can take a warped out of shape acetic acid molecule and get a dog (essentially what the acid is shaped like) out of it. Visualization will either be done imaginatively from looking at numbers or by cheating my graphics creativity out of a job by using GLUT to draw spheres and rods.
The ab initio methods will be tested much the same way. For both molecular mechanics and ab initio methods, I'm going to start with small molecules and work my way up (ab initio will start at H, MM will start at H_2).
Testing the AI will be a doozy. It's dooziness is why I'm unsure of being able to finish it – I won't really know if it's finished or not. I guess the best way to test it is to run the ab initio part, look at the output, interpret it myself, and then see if the AI's interpretation matches my own.

Expected Results

A program that will do what I've been saying over and over again throughout this paper.

Visualization

I'll be using the Irrlicht Library due to its ease of use and also due to the fact that senior year is turning out to be quite a doozy in its own right. Panel/Canvas 3D visualization and GUI functions are both handled by this library with very few function calls beyond the extensive set-up generally required by all 3D libraries to begin with – thus my choice.

Literature Cited

none yet, I've been devving via generic rules learned from AP Chemistry and teaching myself QC, which I haven't actually started devving yet, so... yeah... no refs