Abstract of “Interpretation of Molecule Conformations from Drawn Diagrams” by Dana Tenneson, Ph.D., Brown University, May 2008.

In chemistry, molecules are drawn on paper and chalkboards as diagrams consisting of lines, letters, and symbols which represent not only the atoms and bonds in the molecules but concisely encode cues to the 3D geometry of the molecules. Recent efforts into pen-based input methods for chemistry software have made progress at allowing chemists to input 2D diagrams of molecules into a computer simply by drawing them on a digitizer tablet. However, the task of interpreting these parsed sketches into proper 3D models has been largely unsolved due to the difficulty in making the models satisfy both the natural properties of molecule structure and the geometric cues made explicit in the drawing.

This dissertation presents a set of techniques developed to solve this model construction problem within the context of an educational application for chemistry students. Our primary contribution is a framework for combining molecular structure knowledge and molecule diagram understanding via augmenting molecular mechanics equations to include drawing-based penalty terms. Additionally, we present an algorithm for generating molecule models from drawn diagrams which leverages domain-specific and diagram-driven heuristics. These heuristics make our process fast and accurate enough for molecule diagram drawing to be used as an interactive technique for model construction on modern Tablet PC computers.

Interpretation of Molecule Conformations from Drawn Diagrams

by
Dana Tenneson
A. B., Vassar College, 2000
Sc. M., Brown University, 2003

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2008

This dissertation by Dana Tenneson is accepted in its present form by
the Department of Computer Science as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date ______________ ____________________________

Andries van Dam, Director

Recommended to the Graduate Council

Date ______________ ____________________________

Pascal Van Hentenryck, Reader

Date ______________ ____________________________

Matthew Zimmt, Reader
(Department of Chemistry)

Approved by the Graduate Council

Date ______________ ____________________________

Sheila Bonde
Dean of the Graduate School

Vita

Dana Tenneson was born in September of 1977 in the rural town of Sedro-Woolley, Washington; a town best known for its annual Loggerodeo festival. Dana spent thirteen years in the Sedro-Woolley public school system graduating at the top of the class of 1996. Attending Vassar College afterwards, Dana completed an A.B. in computer science and mathematics in 2000, winning that year's Holdeen-Adams prize for excellence in computer science as well as department and general honors. Dana applied to four Ph.D. programs in computer science for the next fall, but after receiving rejection letters from all of them, opted to attend Brown University despite the rejection and moved to Providence to find a way in. After working a year for the Genetic Anomalies studio of game publisher THQ, and two years completing an Sc.M. at Brown (including a year of part time work for pharmaceutical software studio LeapFrog**R**), Dana was able to transfer to the Ph.D. program in 2003. As a Brown University graduate student, Dana pursued educational applications of computers which lead to two major projects: The Graphics Teaching Tool for design students and ChemPad for introductory organic chemistry students.

Dana's proudest achievements outside academia include winning the national 4-H dairy quiz bowl in 1992, earning a league honorable mention for high school varsity soccer in 1996, and earning the Wohelo Award from Camp Fire in 1998.

Dana first learned to program computers in elementary school by reading the instruction manuals for the Apple IIc and the Texas Instruments 99/4A.

Acknowledgements

There are many I need to thank for helping me to get this far. Most directly, my advisor Andy van Dam and chemistry collaborator Matt Zimmt have encouraged and supported me throughout this work. Without the time they somehow managed to make for me, none of this would have been possible. Similarly, my third committee member, Pascal Van Hentenryck, pointed me towards algorithms and techniques late in my work which made the difference between good and great performance. While not on my committee, I additionally owe a great debt to Anne Spalter who originally brought me into collaboration with the Brown Computer Graphics Group and helped me get to know all its great people.

I have had many collaborators on this work over the years ranging from close partnerships with Ben (Sasha) Shine and Chris Maloney, to undergraduate researchers Peter Goldstein and Dimitar (Dimo) Bounov, and computer graphics research staff Loring Holden, Bob Zeleznik, and Tim Miller. I particularly owe Ben for positing in 2004 that it would be great if organic chemistry students could draw diagrams and see the corresponding models.

As a culmination of my formal education, I would like to thank the teachers who have been particularly inspirational and given me extra educational opportunities throughout my life. From earliest to latest these would be Marlis Kusella who got me out of class a few afternoons a week to learn something more interesting, Doug Walker who spent Saturdays with me at the the ESD letting me play with the latest technology, Kenlynn Nelson who made four years of French class about more important things than learning French, Wade Webber who showed me that attitude and determination will get you farther than ability, Micheal Joyce who convinced me that computers have so much more potential for education, literature, and society than have been explored, and finally Louis Voerman whose career path is everything I want mine to be.

Moreover, I thank my parents Dale and Shelley for being loving and supportive, my sister KT for reminding me of my shortcomings, and the rest of my family for providing an anchor of reality throughout the graduate school experience. Extra thanks to my uncle Randy for first introducing me to computers by playing Choplifter with me on his Apple][and building me electrical gadget toys when I was little. I also have to thank Sue, my Sweetie, for her tolerance of my graduate student schedule and for patiently awaiting her shiny.

Finally, I thank Willow, Neko, Spot, Poke, Fuddy, and Junior for leading by furry example and teaching me the only thing in life worth stressing about is the time dinner will be served.

Sponsorship

A number of people and organizations have made this work financially possible. First, many thanks to Atlantic Philanthropies which provided the original funding to develop and deploy ChemPad and to Brown University President, Ruth Simmons, who was the deciding factor in receiving that funding. Similarly, my thanks to the Andries van Dam Fellowship donors John Crawford, Andrew Marcuvitz, William O'Farrell, Evan Schrier, Brad Silverberg, David Simons, Andy van Dam, and Jeffrey Louis Vogel who provided the majority of my financial support thereafter. Beyond my own funding, HP helped us acquire the Tablet PCs for our Tablet Lab and undergraduate researchers over the years. These tablets have been an excellent resource and have all received much use. Furthermore, Brown University's Undergraduate Teaching and Research Assistantships have funded the work of three such undergraduate student researchers who spent summers improving ChemPad. Emil Sarpa and Sun Microsystems have my thanks for helping to support another graduate school research project of mine thereby helping to support this work indirectly. Finally, it has been a pleasure to be part of the Microsoft Center for Research on Pen-Centric Computing for the last few years and receive the *promotional* support of Paul Oka, Jane Prey, and Sashi Raghupathy.

Contents

⋆ Parts of this thesis appeared have appeared previously in papers [79, 81].

List of Tables

List of Figures

Chapter 1

Introduction

The keyboard... How quaint.

Scotty in *Star Trek IV: The Voyage Home*

The above quote comes from a classic science fiction movie where Montgomery Scott, an engineer from the future, must use computer input devices of the mid-1980's to input the molecular structure of "transparent aluminum" into a computer. Although many advances have been made in computational chemistry and molecular modeling systems since the film's release, methods for computer input of molecular structure today still largely depend on the same keyboard and mouse Capt. Scott was dismayed to use in 1986. This is unsurprising given the continued dominance of the keyboard and mouse as computer input devices. However, while the keyboard is ideal for input of linear text and the mouse is ideal for certain forms of two-dimensional input, molecules, alternatively, are three-dimensional in nature and new input mechanisms will be needed to manipulate them in the sci-fi future.

This dissertation explores one such alternative to the mouse and keyboard for molecular structure input: the stylus. In particular, we focus on the task of interpreting drawn molecule diagrams into 3D structure as in the example in Figure 1.1. While the stylus is still a two dimensional input tool, the same as a mouse, the stylus allows for a fluid form of interaction more alike to writing on paper. Molecule diagrams are already ubiquitously used tools in chemistry settings for tasks not involving computers. As molecules are three-dimensional structures that chemists need to depict on two-dimensional media, such as paper and blackboards, for purposes of communication, there are standard practices for drawing molecule diagrams which concisely convey a great deal of 3D information. To a trained chemist, a molecule diagram can indicate a specific 3D structure, that is, a specific arrangement of atoms and bonds, or the diagram can intentionally be more ambiguous and indicate a number of such possibilities. Since chemists are already trained in the production of molecule diagrams, and the diagrams encode the necessary structure information, drawing molecule

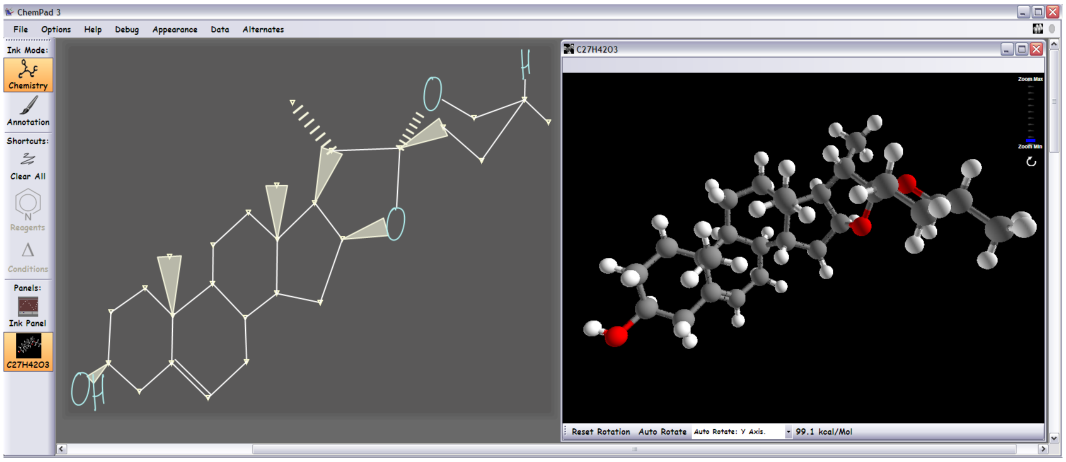


Figure 1.1: *Interpreting molecule diagrams (left) into interactive 3D models (right). The Diosgenin steroid shown here contains 18 drawn cues (6 wedge bonds, 2 dashed bonds, and 10 bonds in perspective) as to the intended structure of the molecule. These cues must be rectified with hundreds of chemical feasibility measures to generate the appropriate model.*

diagrams makes for an appealing means of input for 3D models of molecules.

1.1 Molecule Diagrams

At their simplest, molecule diagrams, such as Lewis structure and line structure diagrams, consist of letters, lines, and symbols representing the atoms of the molecule and the bonds between the atoms. An atom is depicted by the letters of its chemical symbol found on the periodic table, such as an “O” for oxygen or “Br” for bromine. Two atoms held together through the sharing of a pair of electrons, also known as a covalent bond, are drawn with a line between them. Similarly, two lines between atoms represent the sharing of two pairs of electrons (a double bond), and three lines represent three pairs of electrons (a triple bond). Figure 1.2 shows a number of examples of molecule diagrams one would find in a chemistry course. A couple abbreviations are commonly made in these diagrams to facilitate the quick drawing of molecules. First, the large number of hydrogen atoms and the bonds to them can be omitted since their presence can be determined based on the number of bonds other atoms are forming in the molecule. For instance, a carbon atom typically forms four bonds, so a diagram containing a carbon with two drawn bonds is assumed to have two more bonds to hydrogen atoms. In cases where the presence or absence of a hydrogen atom is ambiguous, the chemist can explicitly note this in the diagram. Second, since organic molecules are typically built around frameworks of connected carbon atoms, the carbon’s letter “C” is usually omitted leaving only intersecting bond lines to indicate its presence. In this way, the 14 atom butane molecule in Figure 1.3 can be depicted with only three lines.

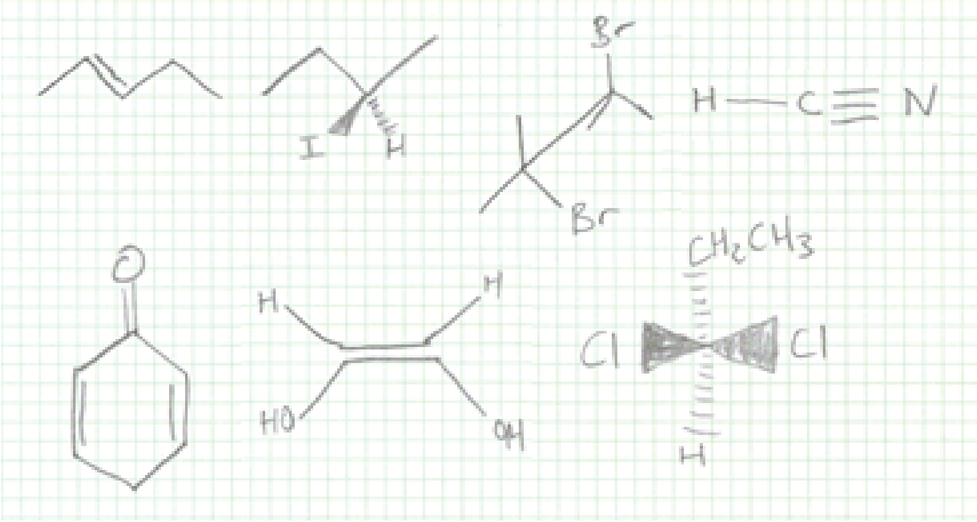


Figure 1.2: *Samples of molecule diagrams. Lines represent bonds and letters represent atoms.*

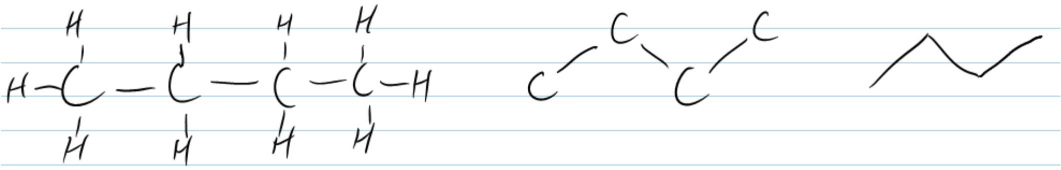


Figure 1.3: *Three ways to draw a butane molecule on paper. The drawing on the left explicitly indicates each of the atoms in the molecule. The drawing in the middle treats hydrogen atoms as implicit and begins to draw the carbon atoms in an approximation to the angles formed in an actual 3D model. The drawing on the right leaves carbons implicit at the intersection of bonds. This last drawing is the type one would most often see used in an organic chemistry course.*

Beyond these simplest means of drawing atoms and bonds, molecule diagrams can contain a variety of notations. Bonds can be drawn with triangles instead of lines to indicate local 3D perspective. Pluses and minuses can indicate the presence of charge. Angles between bonds can indicate 3D relationships when the molecule is viewed from a specific angle. The presence of electrons can be depicted by dots. Common groupings of atoms, or functional groups, in the molecule can be abbreviated such as "Bu" for an 13 atom butyl group. The full molecule diagram vocabulary is vast as there are a great number of different 3D structures to represent accurately and concisely.

This dissertation will focus on interpreting molecule diagrams for organic molecules. We specifically concentrate on the diagrams one would expect to find in an introductory organic chemistry classroom[1]. The techniques presented here are computationally effective within this domain and solutions are given to the dominant problems with interpretation of diagrams in this domain. However, these solutions are suitable to handle many of the molecule diagrams one would expect to find in more advanced organic chemistry texts[2]. Moreover, our overall framework for satisfying the drawn constraints (those made clear by the way the diagram is drawn) and rectifying these constraints with the structural constraints (those imposed by the rules of molecular structure) is generic. Details are given on extending this framework for interpretation of other classes of molecule diagrams.

1.2 Molecule Models

In contrast to molecule diagrams, molecule models, or conformations, are full 3D representations of molecules with precise coordinates for each atom in the molecule. These models are most commonly depicted as the classic "ball and stick" rendering where atoms are colored balls held together by sticks indicating bonds. Actual (physical) plastic "ball and stick" modeling kits are ubiquitous educational tools in chemistry classrooms. In this depiction the atom colors represent different elements and ball sizes can indicate the relative size of the atom, but otherwise there is no encoding of structure information which must be deciphered by the user such as one would find in a molecule diagram. Alternate renderings for molecule models can show the entire electron cloud of each atom, but the underlying model data is the same.

These models are not simple extrusions of the drawn diagram. While diagrams may have regions which each show approximations of 2D projections within the 3D model, the full 3D structure is far from planar. In actuality, molecule structures are far from static as well. As molecules collide constantly, their atoms are pushed into different positions relative to each other. It is nonetheless useful to consider molecule models as having a single location for each atom since interatomic forces cause molecules to be in some conformations much more frequently than others. These likely

[1] Here we are speaking of molecule diagrams containing up to roughly 100 atoms, a half dozen fused rings, and a dozen drawn cues to the 3D structure.

[2] For example, the diagrams in Corey and Cheng's *The Logic of Chemical Synthesis* [23], an advanced chemical synthesis text, are of the scale for which our techniques are computationally effective.

conformations are the ones that minimize the strain of the interatomic forces, or energy, of the molecule.

1.3 Molecule Diagram Interpretation

Computer interpretation of 3D model structure from molecule diagrams requires an understanding of both the physical properties of molecular structure and the notations that chemists use in diagrams to specify that structure. An understanding of interatomic forces is used to divine the great deal of structure information left implicit in a diagram. For instance, the bonds between carbon atoms in butane should have lengths of approximately 1.535 angstroms, but one would not typically state that in a diagram. In conjunction with this domain knowledge, the structure cues may clarify between options that would otherwise remain ambiguous or even contradict said knowledge. Again with butane, a diagram can specifically show the molecule at its global energy minimum, or a local minima with an energy that is about 4 kJ/Mol higher than the global minimum. To successfully interpret molecule diagrams, an algorithm must have an understanding of both diagram cues and physical structure properties and be able to solve for both categories in a computationally efficient manner.

Molecule diagram interpretation could have industrial and educational applications. Besides providing a drawing-based molecule modeling tool for desktop computers, diagram interpretation could enable electronic laboratory notebooks to have a better understanding of the chemistry notes kept within them. According to a recent review of electronic laboratory notebooks in pharmaceutical research and development, “Current drawing tools require the use of a tool kit of structural elements, such as atoms and bonds, an experience that can be improved dramatically if hand-drawn chemical structures can be interpreted into a computer representation.” [26]. In education, diagram interpretation can be used to give students a tool to visualize their molecules in 3D without needing complex molecular modeling packages. While these commercial modeling packages can be used efficiently by experts, the learning overhead makes them prohibitive for students. Diagram interpretation makes an appealing alternative as students learn to draw molecule diagrams early in their studies and a drawing-based interface would have little to no overhead for them.

1.4 Thesis Statement

Combining molecular structure knowledge with diagram drawing cue knowledge to produce correct molecule models forms the core thesis of this dissertation:

Molecule diagram drawing cues can be represented in molecular mechanics force fields as appended energy penalty terms. Use of force fields containing these terms in conformation generation algorithms yields molecule conformations which match the drawn diagram as well as being energetically feasible.

1.5 ChemPad

Molecule diagram interpretation is presented here in the context of the creation and continued development of the ChemPad application. ChemPad is a Tablet PC application for chemistry classrooms designed to help introductory organic chemistry students learning about 3D molecule structure and the correspondence between molecule diagrams and their models. While ChemPad contains a number of pedagogically-driven visualizations and tools, the core functionality of ChemPad is creating 3D molecule models from the diagrams students draw with a stylus on the Tablet PC. ChemPad was developed through collaboration with faculty and students of the Brown University Chemistry Department to ensure its usability and pedagogical value.

Over the past four years we have had hundreds of student users of ChemPad. Their feedback has driven both application feature development and the advancement of diagram interpretation technology. Additionally, ChemPad has been available as a free download from the project website throughout development and had had more than 350 downloads in the last six months[3]. User feedback on ChemPad has been consistently positive and using ChemPad has yielded a statistically significant improvement in problem-solving abilities requiring 3D thinking for students who otherwise have difficulty [81].

1.6 Overview and Contributions

This dissertation presents the molecule diagram interpretation and molecule model conformation generation technology present in ChemPad with an emphasis on the following main contributions:

- A framework for encoding molecule diagram drawing constraints such that they are compatible with existing molecule structure constraints and the algorithms which use these structure constraints.
- Implementation of four common molecule diagram drawing constraints within this framework.
- An algorithm for constructing molecule models based on this framework.

In Chapter 2 we review work related to this dissertation. We give an overview of foundational technologies used in this work such as molecular mechanics and FluidInking. We discuss traditional systems for modeling molecules on the computer and other domains for which pen-based computing have been applied. Finally, we cover other current research projects in creating pen and voice-based interfaces for chemistry.

In Chapter 3 we present algorithms for the task of building 3D models from parsed molecule diagrams. We review the history of conformation generation algorithms in the ChemPad application

[3] September 2007 through February 2008

with an emphasis on what types of diagrams each version of the software could not handle, and how the following algorithm overcame the problems of the version before. We give a final algorithm that leverages expert information present in input diagrams, yet can gracefully recover from mistakes in interpretation of this information.

In Chapter 4 we present Ink-Modified Molecular Mechanics as a means to create force fields which can simultaneously understand the mathematical properties of molecule structure as well as constraints on this structure that are indicated explicitly in drawn diagrams. We give four example terms for formulating diagram understanding into a force field along with the partial derivatives necessary for implementation.

In Chapter 5 we give evaluations of the algorithms from the previous chapters. Additionally, we give evaluations of the overall ChemPad application and its educational value.

In Chapter 6 we summarize the contributions of this dissertation and review future research directions for this work.

The appendices provide a computer scientist's introduction to molecular mechanics, gradient formulae and their derivations useful for replicating this work, a detailed listing of the test cases used in the algorithm evaluation of Chapter 5, and pseudocode for additional algorithms present in ChemPad.

Chapter 2

Related Work

Automatic interpretation of molecule diagrams into models brings together two very different areas of research. The user's first point of contact with the system is the pen-based interface, an old, but active research area in computer science. Understanding general primitives made with a pen, as well as higher level objects and interactions within the chemistry domain remain open research problems. For the task of constructing the 3D models, we revisit fundamental research into computational chemistry, as the process requires a detailed understanding of the shapes that molecules take. Without this work and its precursor, mathematical modeling of intramolecular forces, we would have no foundation for making models that are chemically accurate. We present here an overview of research in these areas that have been instrumental or inspirational to our work.

2.1 Pen-Based Computing

Starting with Sutherland's Sketchpad [76], many forays into pen-based interfaces for computers have been made over the last forty years. A digital stylus is an obvious input device for situations where bulkier input devices such as keyboards and mice are unfavorable due to their size. The stylus is therefore quite useful in situations where the entire computer needs to be lightweight and portable such as cell phones, PDA's, or the new ultra mobile PC's. Alternatively, large, marker-like styli are used for digital whiteboard environments where people want to leverage the power of a computer with the large-scale writing form factor of a chalkboard. The stylus is also obvious for input of anything people are accustomed to writing out on pen and paper. In particular for influencing this dissertation, much work has focused on pen-based interfaces for input of information in domains which have standard 2D notations. For example, circuit diagrams [28, 32], mathematics [54, 52], and music [30] are domains which fall into this category as one typically does not write their notations on paper in a strictly linear fashion. In such domains, computer users of pen-less systems need to become experts in using the mouse and keyboard interfaces to accomplish that which is straightforward with a pen. Chemistry molecule diagrams is another such domain. While chemists who on a regular basis use computers for entering molecule diagrams can perform this task very quickly, it requires learning non-obvious macros and shortcuts.

These domain-specific pen interfaces often do not directly use the stream of stylus input data, but use an intermediate processing layer which makes sense of the raw data and passes along higher level objects. A usual first step in processing ink is the detection of cusps in ink strokes. These cusps segment a single stroke into parts which may (or may not) be logically different. For instance, a rectangle may be drawn as a single stroke of the pen, but cusp detection would note the places where the stroke sharply changes direction thereby indicating the four sides of the rectangle. Cusp detection algorithms are usually based on creating polynomial approximations of the stroke point data and noting the speed of the pen at each point [58, 72, 50]. Once cusp detection is accomplished, it can be useful to then combine these segmented ink strokes into higher-level components, particularly for diagram recognition tasks. For instance, the example rectangle may have been drawn as a single stroke, or as four different strokes. As a human is able to recognize both cases as the same object, we need the computer to accomplish the same. As the goal of this process is to allow differences in the way users prefer to draw their symbols, Hammond and Davis [39] created a system for creating general definitions of diagram objects. Alternatively, some systems like FluidInking [94] require using single strokes to distinguish different components of the diagram. While the former of such systems are more user friendly, detecting diagram objects generally is made difficult by the great deal of difference between the ways users draw diagrams and segmentation by restriction to single strokes removes this ambiguity. Alvarado and Lazzereschi [5] found that in practice, users drawing diagrams without such restrictions do not use a single way of drawing objects and change stroke order, the number of strokes used, and the amount of time that passes between intra- and extra-object strokes.

Besides object primitives such as lines and curves, text is a common component in diagrams. For inked text, a variety of techniques exist to understand the ink as the text it represents. Optical character recognition (OCR), the comparison of the image of the digital ink to images of known characters, has been a popular and well developed technique for this task[1]. OCR is popular in many contexts where a simple recognizer with little computational power is needed. However, the state of the art for text recognition on Tablet PC's is the neural network and language model approach of the Microsoft Handwriting Recognizer [66]. This recognizer performs well with first time users as it has been trained over a large set of handwriting data. Furthermore, it improves its recognition by adapting to the user's writing style as the user continues to input more data into the system [18, 38]. However, it can not easily be expanded to understand new characters that might appear in diagram text. Nonetheless, it is a popular system for pen computing applications and research projects.

Once a penned diagram has been segmented into its logical components, a domain-specific parsing can be constructed. For math, this would be equivalent to a typeset or LaTeX equation with

[1] An extensive overview of OCR is given in Cole's book, *Survey of the State of the Art in Human Language Technology* [22].

subscripts, superscripts, and the scope of operators understood. For circuit diagrams, this would take the recognized switches, resistors, and other electrical components and understand the drawn wires to know which components are connected to each other. In our case of molecule diagrams, this parsing represents the atoms of the molecule and the bonds that connect them as well as any structure indicating cues that are present in the diagram.

2.1.1 Sketch-Based Modeling

Beyond diagram interpretation, there have been a number of explorations into pen-based interfaces to create 3D models. Zeleznik's Sketch system and its followers allowed users to build models such as one would make with a CAD (Computer Aided Design) system using penned (and/or moused) gestures [97, 10, 51]. A gesture-based system differs from a general diagram interpretation systems in that no diagram is made. Instead the gestures provide an intuitive interface for directly constructing and manipulating the 3D scene. For instance, a rectangular solid would be constructed by drawing three intersecting lines at a corner, rather than drawing all the lines of the solid. Alternatively, Grimstead and Varley created systems for interpreting solid polyhedral objects from line drawings [35, 85]. These systems take precise drawings of entire objects, such as one would draft for a machine part, and create the appropriate 3D geometry. Most similar to our work in this domain is that of Lipson and Shpitalni [56] who define "image regularities" for implied relationships that can be detected in freehand CAD drawings. These image regularities are then expressed as mathematical terms in a compliance function which is optimized to reconstruct a rough approximation of the desired 3D shape. While their specific formulation is for solid objects, the general concept of representing compliance with a diagram in terms of a mathematical function is close to what we apply in Chapter 4 to the task of enforcing molecule drawing cues.

While CAD-like systems are concerned with creating exact geometry, sketch-based modeling techniques can also be used to create geometry from rough sketches. Igarashi's Teddy and its successor projects give users the ability to create stuffed animal-like models through freeform drawing [45, 44]. With Teddy one can make 3D models quickly which need only to look correct, but have no exact geometric requirements. Similarly, Nealen *et al.* created a sketch interface for editing existing 3D meshes where the mesh is deformed to the sketched line [60, 59]. This system starts with exact geometry as created by a 3D modeler or 3D object scans, and uses the sketch interface to deform the model to the sketched, inexact curve. With this system, an artist can quickly change the pose of a model or features of the model such as the shape of a nose on a face. In both of these domains, the created objects are solid and object contours provide the major cues to the 3D structure intended. To advance this sort of sketching, Karpenko's SmoothSketch [49, 48] provided more general techniques for modeling shapes to match the contours and for dealing with the inherent 3D ambiguity in this type of sketch.

In contrast to these sketch-based modeling systems, molecules are not solid objects, but their diagrams similarly encode structure cues through the positioning and drawing style of bonds in place of contours. If drawn properly, a molecule diagram can either be ambiguous or unambiguous as to the 3D shape that is intended. The bond lines are not contours for 3D shapes, but connectors as wires in circuit diagrams. The task of building a 3D molecule model from a diagram does not attempt to make a shape which when viewed from the correct angle looks like the diagram. It does, however, occasionally use the spatial information as approximations to the desired shape under perspective viewing. As such, existing techniques in sketch-based modeling do not directly apply to our molecule modeling task.

2.1.2 Pen-Based Tools For Chemistry

While ChemPad was the first pen-based interface for molecule diagrams we are aware of, since its initial release there has been new interest in academia in the development of pen-based input systems for chemistry. The project most similar to our work is Ouyang's molecule drawing system which parses freeform digital ink drawings of molecules to produce ChemDraw compatible files [63, 64]. Ouyang's focus has been to make a molecule diagram interpretation system which handles diagrams exactly as chemists are accustomed to drawing them. This system parses the entire diagram structure upon completion of the diagram and therefore does not give incremental feedback as the diagram is being drawn. In contrast, ChemPad and Bryfczynski's OrganicPad [14, 65], parse individually drawn diagram components to provide immediate feedback in the form of prettified handwriting or typeset text as the user works. This immediate feedback avoids the need for the user to manually search the entire molecule for recognition errors, some of which may not be obvious to a student chemist copying a complex structure from a textbook or chalkboard. In the domain of mathematics interpretation, research suggests there is value for both immediate and delayed feedback systems [96, 33]. We suspect this to be true of chemistry as well, but that is currently an untested hypothesis.

The aforementioned OrganicPad [14, 65], like ChemPad and many of these new research initiatives in pen-based chemistry input, is focused on use by chemistry students rather than professionals. OrganicPad's novelty lies in its architecture as a system to allow communication between student and teacher in the chemistry classroom. OrganicPad allows students to input molecule diagrams on tablets in response to the teacher's posed questions. The answers can both be automatically evaluated via comparison to the diagram drawn by the teacher and solicit personal responses by the teacher. Another classroom based tool is the ChemTeach system by Jiang *et al.* [47] which allows users to input chemistry equations, as opposed to diagrams, using pen and speech. The understood equations are then prepared for presentation in PowerPoint-like applications in a classroom. While chemistry equations are present in many molecule diagrams, the understanding of them has been limited in pen-based molecule diagram systems to date. For middle school chemistry classrooms, the Chemnation tool allows students to create flipbook style animations of molecules on handheld computers [71]. As chemnation runs on low-power handhelds, its understanding of chemistry is quite limited and there is no system-based requirement that the molecule diagrams are spatially correct. The impetus is on the student to create the correct animation.

In comparison to the contributions of this dissertation, these systems are focused on tasks other than generating 3D models. Only OrganicPad contains a rudimentary 3D model conversion system. Instead, the contributions of this dissertation are complimentary to these varied projects. The algorithms for model generation central to this dissertation are compatible with any of the aforementioned diagram drawing systems.

2.2 Molecular Modeling Packages

Commercial software packages for creating molecule diagrams and models on computers have existed for more than two decades and are a well established industry. Mouse-based molecule structure programs such as CambridgeSoft's ChemDraw [17], MDL/Draw [78], ACD Labs' ChemSketch [3], and Accelrys Insight II's Sketcher [2], are used to create typeset molecule diagrams suitable for publication as well as providing an interface for molecule database searches. Many of these programs are part of full chemistry suites including 3D molecule modeling programs. As such, some have some means to convert 2D diagrams to 3D models. These conversion tools are fragile as the user is expected to need to apply their chemistry knowledge to fix conversion mistakes [36]. For example, drawing the molecule diagram from Figure 1.1 into ChemDraw and asking for a Chem3D conversion resulted in a highly unlikely model that violated drawing cues[2].

The work presented in this dissertation differs from systems like ChemOffice in that it includes a richer understanding of chemistry structure and diagram interpretation knowledge into the conversion process. The resulting models are therefore more robust and ideally require no correction from the user. This is important for situations where the user may not know the difference between the structures, as is the case with student chemists first learning about molecule structure. This ability is also useful in situations where the user may not have the time to spend correcting the model, as is the case with electronic lab notebook initiatives (ELN) such as CombeChem and SmartTea [31, 57, 16]. While it is usually not the goal to create 3D models from lab notes, doing so automatically would allow the automatic storing of structure data of performed experiments. This database structure data could then be searched outside of the laboratory [20].

Underlying contemporary 3D molecular modeling systems is an algorithmic and mathematical understanding of the 3D shapes molecules assume. The mathematical modeling of molecule forces to determine chemical feasibility, or molecular mechanics, is a well developed field of study. Formulations for bond length strain, angle strain, steric strain, and torsional strain were developed in the

[2] Even after energy minimization, the resulting structure was 16 kcal/Mol higher in energy than the structure generated by ChemPad. Furthermore, it ignored the explicit chair conformation cue in the upper right ring and placed a wedged methyl group as if it were dashed, which changed the stereochemistry of the molecule.

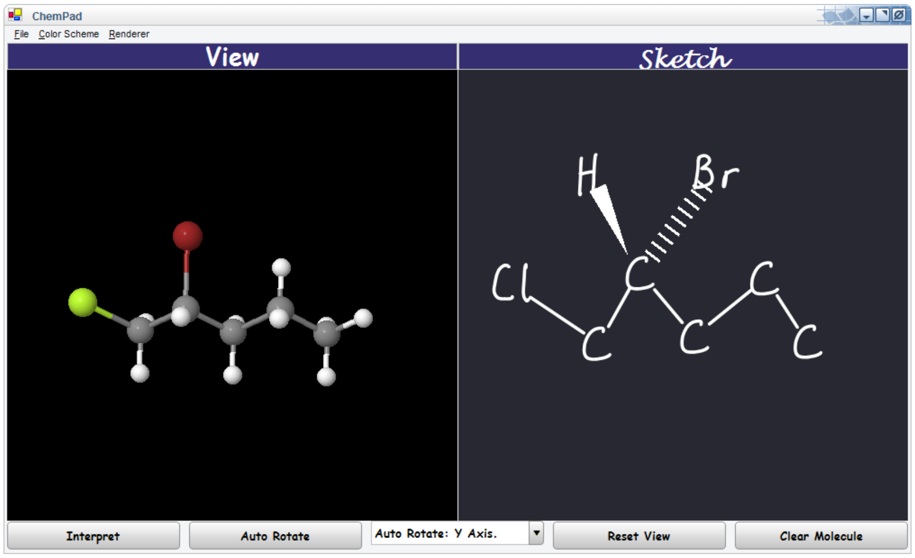


Figure 2.1: *The interface for ChemPad 2004-2006.*

1940's and 1950's and some mathematical models date back to the mid-1800's [69]. Updating these formulas to create robust computer models continues to this day with different formulations specialized for different types of molecules. In particular, our work leverages the GAFF model [90] which is specialized for organic molecules such as we use in our organic chemistry classroom. Similarly the AMBER model [92] is specialized for proteins and Allinger's MM2 model [4] for hydrocarbons[3].

As an understanding of molecular mechanics is critical to the understanding of the techniques in this dissertation, a primer on the topic is given in Appendix A. Some molecular mechanics concepts are also covered in brief along with the techniques of our system which use these concepts. More detailed information can be found in Rappé's [69] and Allinger's [15] books on the subject.

2.3 The ChemPad Input System

The techniques and algorithms which form the foci of this dissertation take as input the parsed molecule diagram described earlier in Section 2.1. This parsing details the atoms involved in the molecule and the bonds which connect the atoms. While these algorithms are generic as far as the diagram input system is concerned, we have developed our own input system for ChemPad. We give a brief overview for completeness.

The first version of ChemPad (circa 2004) is shown in Figure 2.1 [82, 80][4]. It used the Fluid Inking [94, 95] ink gesture library to define single-stroke gestures for components of chemistry sketches

[3] Gundertofte periodically publishes a performance comparison of the different force fields [37]

[4] This interface was developed in collaboration with Sascha Becker, Bob Zeleznik, and Loring Holden

such as atoms and bonds. These single-stroke gestures made it simple to distinguish between logically different diagram components. The gestures were chosen to be close to the standard notations to reduce the learning required on the part of the user. For instance, drawing a “Cl” for Chlorine consisted of drawing a cursive “C” and “l” without lifting the pen. The exception to the single stroke rule is that higher order bonds were drawn with multiple strokes by drawing additional bond lines between atoms. As the components were interpreted, the user was given immediate feedback as to the recognized gesture and overall parsing. Because of this, we knew the parsing of the diagram was correct at each step of the input process.

Later, we developed the interface shown earlier in Figure 1.1 to overcome several shortcomings of the original interface[5]. The new input system is still based on the Fluid Inking system, but greedily gathers strokes together to allow multi-stroke input. The open drawing space allows multiple molecule drawings and 3D windows to be open simultaneously. Perhaps most importantly to users, this system can account for implicit carbon atoms at bond junctions, thereby greatly reducing the time needed to input the diagrams and reducing the amount of learning required to use the system.

One critical piece of information missing from the parsed output of these systems is the force field-specific atom type (or atom symbol) for each atom in the molecule. The atom symbol characterizes the different types of shapes that atoms can take in different parts of molecules. For instance, while all carbon atoms are the same in terms of physics and chemistry, for the GAFF force field, there are 17 different atom symbols for carbon. These symbols can be determined based on the topology of the molecule using Wang’s Atomtype algorithm [89]. We perform this typing as a post-processing step of the above diagram input. With the atoms typed, the parsing contains all the necessary information for the model construction techniques in the next chapter.

[5] This interface was developed primarily by Christopher Maloney.

Chapter 3

Conformation Generation

Before discussing methods for interpreting molecule models from drawn diagrams, we must first clarify what molecule models are. A molecule model is specified predominantly by the types of atoms in the molecule, their bonding topology, and the 3D coordinates of each atom. As such, these three pieces of information are common to many well-used molecule file formats such as mol2, alc, pdb, and mdl. Some file formats, such as txyz and c3d2, omit the bonding information as it can be reasonably deduced at load time from domain knowledge about distances between bonded atoms. However, in general, these are the three pieces of information required to completely define a molecule model so that it can be displayed with a 3D viewer.[1] The viewer then creates a scene where atoms, usually depicted as spheres colored according to the atom's type, are centered at the given coordinates. Bonds, if shown, can be included as cylinders between the spheres of the involved atoms. With these items in place, the viewer presents the completed 3D scene to the user.

In terms of generating molecule models from drawn diagrams, once the atoms and their topology have been determined from a molecule diagram[2], the only remaining information needed to create the interactive model are the atoms' 3D coordinates. **Conformation generation** is this choosing of a 3D location for every atom in a molecule. In this dissertation, the term **conformation** is used to indicate a specific location of each atom in a molecule[3]. More generally in chemistry, the term conformation is often used to make a distinction between one structure and another under a simple change, such as a rotation around a single bond. Vollhardt and Schore's *Organic Chemistry: Structure and Function* [86], for instance, introduces conformations in the context of rotation about carbon-carbon single bonds and distinguishes between staggered conformations (where neighboring atoms do not overlap as you look down the rotated bond) and eclipsed conformations (where neighboring atoms do overlap). The generality of the definition of conformation we use can encompass

[1] Additional information on common molecule file formats can be found in the documentation for the OpenBabel molecule model converter application [43].

[2] This is the parsing previously described in Chapter 2.1 generated by the ChemPad input system (Chapter 2.3).

[3] This definition for conformation is drawn from computational chemistry literature.

molecule structural differences, such as stereoisomerism, which to chemists do not normally fall under the category of conformation.

One might believe the entire task of conformation generation is questionable when considering the actual physical properties of molecules. Indeed, molecules are not rigid 3D structures, but atoms in motion held in certain shapes by the forces between electrons and protons. Therefore choosing a single location for each atom in the molecule can only represent the shape of a molecule at an instant in time. Small molecules may transition between only a few preferred states, but the number of possibilities grows exponentially as the molecule size increases. Not only can an actual molecule be found in any of these states at a moment, but it can be found between these discrete states or somewhere close to one of these states or transitions. Therefore, it may seem strange to select one specific coordinate for each atom. Nonetheless, this structure selection is useful as some conformations are much lower in energy than others, and therefore more likely to exist at any particular moment. Chemists usually prefer to think of molecules as assuming the conformations which minimize the molecule's energy since the molecules are close to those structures most of the time.

3.1 Mathematical Formulation

For the sake of clarity, the conformation generation task is to create an algorithm[4] $F : D \to C$ which maps a (parsed) diagram D to a conformation C. Informally, we are interested in finding an algorithm F in the set of all such algorithms $\mathfrak{F}$ which gives the best conversions as judged by the chemist user. While this informal metric does not easily convert to a single function to judge the algorithms we propose, we explore different dimensions of such algorithm performance later in Chapter 5.

A diagram, D consists of a set of atoms $a_1, a_2, ..., a_n \in \vec{a}$ and a set of bonds $b_1, b_2, ..., b_m \in \vec{b}$. Each atom in the diagram consists of an X ($a_i^X \in \mathbb{R}$) and Y ($a_i^Y \in \mathbb{R}$) coordinate for its location in the diagram, as well as the atom type[5] T ($a_i^T \in \mathfrak{T}$). Each bond consists of the two atoms it connects a_j, a_k ($b_i^{a_j} \in \vec{a}$, $b_i^{a_k} \in \vec{a}$), its order[6] O ($b_i^O \in$ [single, double, triple]), and the way it was drawn ∇ ($b_i^\nabla \in$ [normal, wedge, dashed]).

The generated conformation C consists of the transformed atoms $\bar{a}_1, \bar{a}_2, ..., \bar{a}_n \in \vec{\bar{a}}$ where we now have a 3D coordinate for each atom ($\bar{a}_i^{X'} \in \mathbb{R}$, $\bar{a}_i^{Y'} \in \mathbb{R}$, $\bar{a}_i^{Z'} \in \mathbb{R}$) and bonds with references to the transformed atom (If $b_i^{a_j}$ in D, then $b_i^{\bar{a}_j}$ in C).

[4] This algorithm may or may not be deterministic and therefore may not be a function.

[5] The set of all possible atom types $\mathfrak{T}$ are force field-specific "types" of atoms based on atomic number and connectivity. See the end of Chapter 2.3 for details on atom types.

[6] A bond's order indicates how many electron pairs are shared in the bond. A single bond has order 1, a double bond order 2, etc.

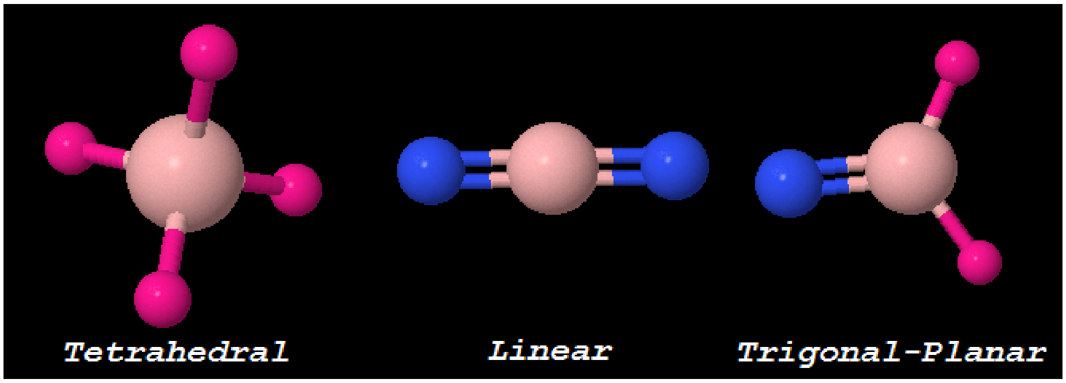


Figure 3.1: *Examples of common atom templates. The pink spheres represent the center of the template. Magenta spheres are single-bonded neighbors and blue spheres are double-bonded neighbors.*

3.2 The Plastic Kit Approach to Conformation Generation

A ubiquitous tool in chemistry education is the ball-and-stick modeling kit. Balls are used to represent atoms in a molecule where each ball's color indicates the element. The balls are held together by sticks, springs, or plastic plugs representing the bonds between the atoms. A student with a plastic ball-and-stick molecule modeling kit can easily generate a conformation for a given molecule by plugging the pieces together. Modeling kits, with their predefined locations for where bond sticks can be placed on each atom, help students accurately depict many molecules' preferred conformations. These predefined locations help because atoms tend to prefer to bond in specific 3D shapes which the pieces are modeled to recreate. These shapes are dependant on the atom in question, the number of neighboring atoms, and the order of the bonds attached to those atoms. The kit pieces are manufactured to give close approximations to most of these constraints. For the constraints that cannot simply be manufactured into the pieces, the student is then responsible for using their knowledge of chemistry to assemble the pieces into reasonable conformations.

In a similar fashion, the first version of ChemPad combined digital kit pieces, or templates, together to match the molecule's topology. While the intricacies of the algorithm are covered in detail in Tenneson 2005 [79], the basic concepts are presented here to better illustrate the advances made in later work. Figure 3.1 shows examples of atom templates used in this early version. The templates are a set of direction vectors where neighboring atoms can be attached. The proper template is found by table lookup keyed by the atom's element, the order of the incident bonds, and the orders of bonds incident on neighboring atoms. For instance, a carbon atom with four single bonded neighbors would get the tetrahedral template, while a carbon with two single-bonded neighbors and a double-bonded neighbor would get a trigonal-planar template. The proper bond lengths between the template pieces would then be looked up in another table by the element of each atom in the bond and the order of the bond between them. In the example, the single-bonded carbon, when attached to an oxygen atom, would have a longer bond length than if the bond between the two were of a higher order, i.e. double or triple bonded.

This approach simplifies the conformation generation problem for both us and the modeling kit user to that of selecting torsional angles for the bonds of the molecule[7], a common practice in conformation determining systems [29, 73]. In brief, a torsional angle is the angle of rotation around a bond as shown in Figure 3.2. Single (sigma) bonds in molecules can usually be thus rotated about without breaking the bond. Therefore, we can think of the bonds as freely rotating in the templates. Just as a kit user would assemble a model, we need only add atom templates to a model one at a time and pick the appropriate torsion for the newly bonded template. The reduction of the dimensionality of the problem has made the approach appealing to computational chemists. Where the initial problem involved $3 * a$ dimensions, where a is is the number of atoms in the molecule, this approach produced a problem with fewer than a dimensions.

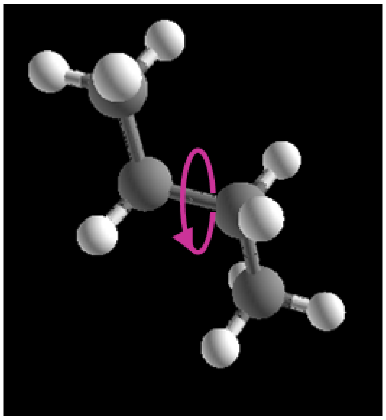


Figure 3.2: *Torsional angles in molecules. The four grey carbon atoms in butane define a torsion (drawn in purple) between the middle two bonded atoms. If one were to turn this molecule so that one of the middle two atoms overlapped the other in the viewer's perspective, the torsion angle is the angle formed by that one point and the two outer carbons.*

In the mathematical formulation, we can think of this system as abstracting the algorithm F into two subalgorithms F_1 and F_2 where $F(D) = F_2(D, F_1(D))$. $F_1 : D \rightarrow \vec{\theta}$ takes the diagram and produces a set of torsional angles $\vec{\theta}$ for each bond torsion. $F_2 : D, \vec{\theta} \rightarrow C$ takes the torsional angles and the diagram and produces the conformation. F_2 is straightforward to implement as the chemistry literature provides the ideal bond lengths and ideal bond angles formed at atoms. Determining F_1 is therefore what we explore in this section.

Figure 3.3 shows our initial boiler plate (F_2) algorithm based on this modeling kit-based (or torsion based) model of the conformation generation problem. It begins by placing an atom at the origin. Then, one at a time, it finds an atom that can be attached to the existing structure and adds it. It picks an unused vector of the parent (neighbor atom) template to determine where it should place the new atom. It also rotates the child (new atom) template so that the child's primary vector is pointing towards the parent. In this simple algorithm, conformations generated will have correct bond lengths and correct angles formed by sets of three atoms[8], but torsional angles are random and there is nothing to prevent interpenetration of atoms.

[7] More detail on torsional angles and their use in Molecular Mechanics can be found in Appendix A.4.

[8] This is true for non-cyclic molecules. Since this algorithm has no concept of rings, the bond length and angles formed at the last ring atom to be added is likely to be very wrong

```
public GenerateConformation(Molecule m)
{
    // Place the first atom at the origin
    m.Atoms[0].Location = (0,0,0);
    m.Atoms[0].Template = TemplateLookup(m.Atoms[0].AtomicNumber,
        m.Atoms[0].NeighborCount, m.Atoms[0].NeighborBondOrders);
    m.Atoms[0].Connected = true;
    // Loop until all atoms have a 3D coordinate.
    while(!AllConnected(m))
    {
        foreach(Atom a in m.Atoms)
        {
            // If we do not have a 3D coordinate for this atom,
            // but a neighbor does, give this
            // atom a coordinate and connect it.
            if(!a.Connected && HasConnectedNeighbor(a,m))
            {
                Atom parent = ConnectedNeighbor(a);
                Connect(a, parent, BondBetween(a,parent));
            }
        }
    }
}


protected Connect(Atom child, Atom parent, Bond b)
{
    Vector direction = UnusedTemplateVector(parent);
    child.Location = parent.Location +
        direction * BondLength(child, parent, b.Order);
    child.Template = TemplateLookup(child.AtomicNumber,
        child.NeighborCount, child.NeighborBondOrders);
    RotateForAlignment(child.Template, -direction);
    child.Connected = true;
}
```

Figure 3.3: *Initial algorithm for the modeling kit-based approach to conformation generation.*

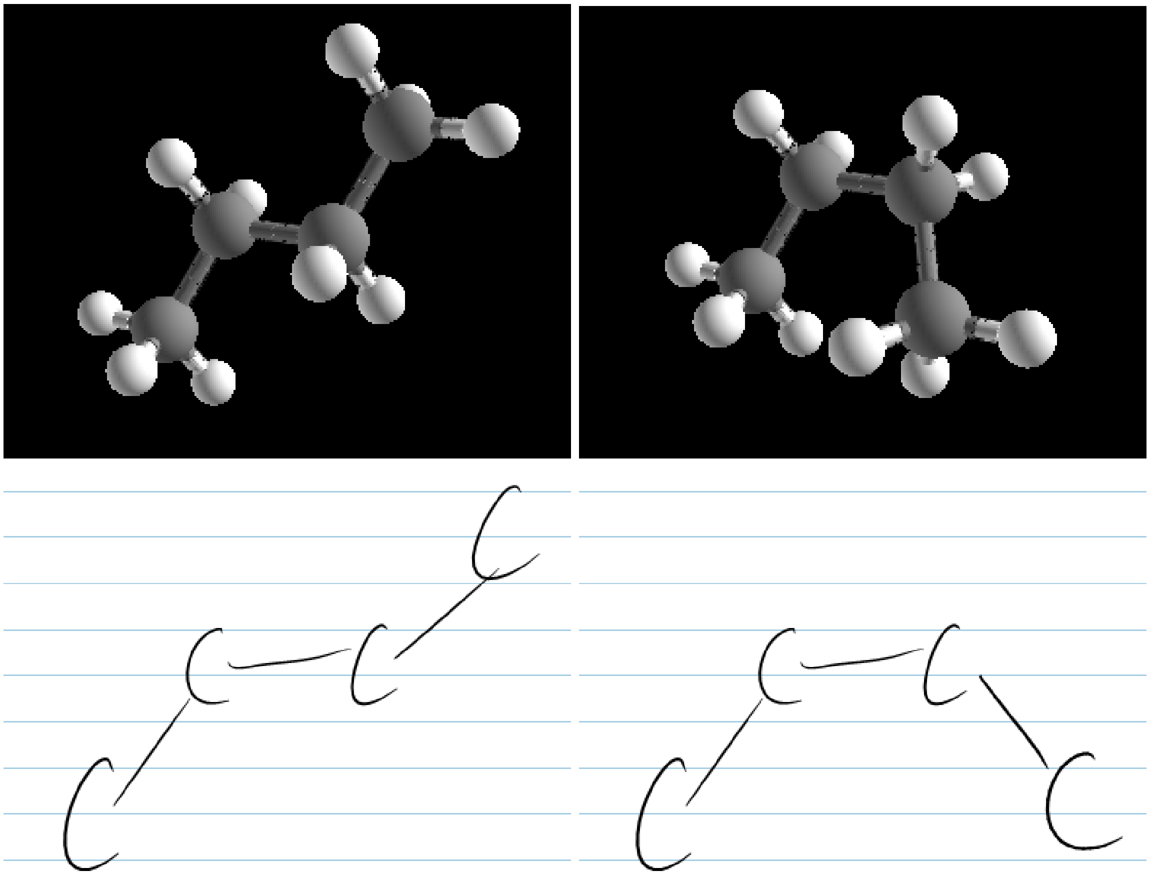


Figure 3.4: *Anti and syn-periplanar drawings of butane. The anti-periplanar drawing and conformation on the left places the 1 and 4 carbons as far apart as possible. The syn-periplanar (eclipsed) drawing and conformation on the right places the 1 and 4 carbons as close together as possible.*

3.2.1 Following Drawn Cues

For simple molecules, we can make good decisions about what the torsional angles should be based strictly on the drawing cues present in the user's diagram. This is because the user presumably understands enough about chemistry to draw the molecule in a way that represents a likely (or desired) set of torsional angles. Given a bond for which we need to set a torsional angle, we look at the drawing to see if the bond and two neighboring bonds (one adjacent to each end) have been explicitly drawn. If so, then we can tell from the turn directions in the drawn angles whether the user is indicating an anti-periplanar or syn-periplanar torsion. Anti-periplanar refers to a torsion of 180° which is indicated by opposed turn directions consecutively in the diagram and syn-periplanar refers to a torsion of 0° indicated by the same turn direction consecutively in the diagram. Because larger atoms involved in a torsion generally want to stay as far apart as possible, the former torsion is very common in molecules. Figure 3.4 shows the difference between the two. Any bond where there is not enough information to define one of these torsional angles is left random. Although this will

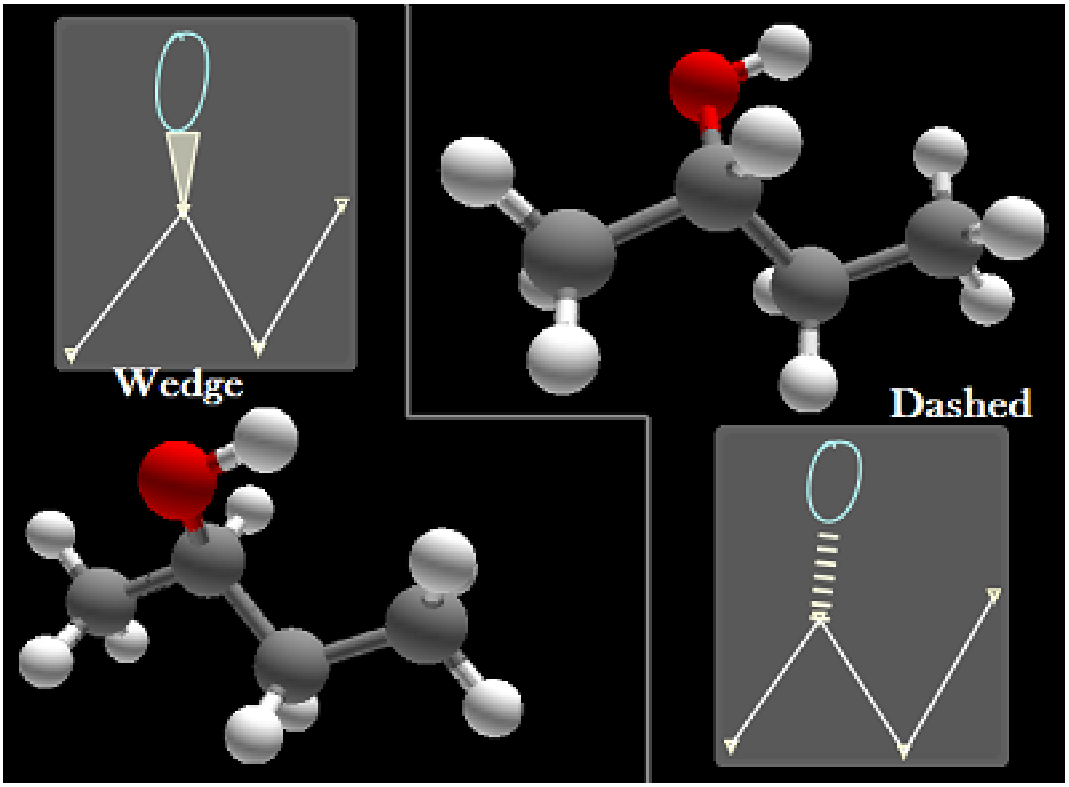


Figure 3.5: *Wedge and dash bonds in 2-butanol. Seeing the carbons as defining a plane, the oxygen connected by a wedge bond comes towards the viewer. Conversely, the oxygen with the dashed bond goes away from the viewer.*

be the majority of the bonds in the molecule, it will not be the most important bonds. The bonds which define the central structure of the molecule will have been made explicit in the drawing. By adhering to the user's drawn indications of these two torsional angles, we are leveraging the user's knowledge of chemistry rather than encoding complex chemistry knowledge into the system.

Wedge and dash bonds are notations used to indicate other common torsional angles in organic molecules. The wedge and dash indicate standard single bonds, but specify the bond's 3D perspective in the diagram. Usually, we additionally have two straight-line (or normal) bonds incident at a wedge or dash bond and can therefore see the perspective of the wedge or dash bond as being up or down from the plane defined by the normal bonds. A wedge bond indicates the atom is coming up from the plane while the dash bond is down behind the plane. Figure 3.5 shows the wedge and dash bonds as they are drawn and the 3D perspectives they represent.

In the context of setting torsional angles, when adding a wedge or dash bond to the conformation, we first check if there are two normal bonded atoms available at the wedge or dash to define the plane. We make sure to add the normal bonds to the conformation first. Thereafter, we have enough information to make a good choice of which template vector to use for adding the wedge or dash bonded atom. We expect that in a tetrahedral template, such as is typically used in cases involving wedge and dashed bonds, after the plane bonds are added there will be two unused template vectors. One will come forward from the plane for a wedge bond and one will go back from the plane for a dash bond. In this approach, the order the atoms are added greatly affects the quality of the conformation generated. If we were to look at the remaining template vectors after adding only one of the normal bonds at the central atom, it would be difficult to tell which vector of the three vectors to use for the wedge or dash. There may be two vectors in the correct direction, or even if there is only one, the entire template could be later rotated to make the syn-periplanar or anti-periplanar torsion with the addition of a normal bond. This could rotate the wedge or dash bond into the wrong location.

While we need to add the normal bonds which are adjacent to a wedge or dash bond before the wedge or dash itself, we shouldn't save all wedge and dash bonds for last. Since wedge and dash bonds are indicative of parts of a molecule which have significant impact on the overall structure, it is useful to add these molecule parts as early as possible to the conformation. In contrast, implicit hydrogen atoms do relatively little to determine the overall structure of the molecule[9], and these atoms should be added to the conformation last. While the quality of conformations generated in any scheme which adds atoms one at a time is greatly influenced by the order atoms are added, it is unclear which order would be optimal. As the complexity of the molecules increases, additional factors such as the length of carbon chains and the size of rings become important to this ordering. Generally, we know that some attachment ordering must apply to the problem because real molecules are built up from forming bonds in some order. We will cover the prioritizing of atoms for larger molecules in more detail in Section 3.4.6, but for this simplified algorithm, we seek atoms with wedge and dash bonds to add first, add their normal bond neighbors before the wedge and dash bond neighbors, and keep implicit hydrogen atoms for last.

3.3 Improving Generated Conformations with Molecular Mechanics

While the conformations generated by approximating a modeling kit are appropriate for many small organic molecules comprised of a carbon backbone and one or two small functional groups, the technique breaks down as molecules get larger and the functional groups become more complex. A fundamental reason for the breakdown is the aforementioned lack of chemistry knowledge on the part of the system. When chemists draw a molecule diagram for another chemist, there is a level of chemical knowledge assumed to be present in a chemist reading the diagram. The reading

[9] The presence of hydrogen atoms actually has a great deal to do with the structure of organic molecules in terms of the forces between atoms. However, with respect to interpreting a molecule diagram, they are relatively unimportant and are therefore omitted from most drawings. Hence the undrawn hydrogens are termed “implicit”.

chemist is assumed to know how to deal with steric strain, the force which repels non-bonded atoms when they attempt to interpenetrate. Additionally, the reading chemist is assumed to know more about torsional angles than is made explicit in a typical molecule drawing. Without this kind of information, the system makes mistakes that are obvious to the trained chemist. Moreover, even a trained chemist may not know immediately the preferred conformation for a larger molecule diagram. A hypothetical chemist using a molecule modeling package to perform the conformation generation task would first start with a general shape based on their understanding, then run an optimizer with an even greater level of chemistry domain knowledge to put the model into the best conformation. These optimizers are based on molecular mechanical or quantum mechanical models of the interactions of atoms in a molecule and can correct shape relationships which would otherwise be invisible or not apparent to the chemist. They can not only twist bond torsional angles, but deform bond lengths and angles away from their equilibria just as would occur in a real molecule.

While ab initio quantum mechanics uses first principles to create highly realistic models of intramolecular forces [69], the models are too computationally intensive to use as part of an interactive technique. Alternatively, molecular mechanics uses computationally efficient spring-like force approximations which are suitable for our system. We can therefore attempt to replicate the technique of our hypothetical chemist modeling the drawn conformation. By post-processing the conformation from the initial algorithm with a molecular mechanics optimizer, the conformation's chemical feasibility is improved. Figure 3.6 shows this new approach.

In terms of our mathematical formulation from Section 3.1, molecular mechanics can act as an approximation to our metric for the performance of $F \in \mathfrak{F}$. We can say that F, which produces conformations $\mathfrak{C}$, is better than F', which produces conformations $\mathfrak{C}'$, if the members of $\mathfrak{C}$ have lower energy than members of $\mathfrak{C}'$ as measured by molecular mechanics. This approximation does not fully encompass what chemist users expect from the system, but it is a beginning to understanding their expectations. As molecular mechanics optimizers are readily available, we take one such optimizer $F_3 : C \to C$ and include it in our overall conformation generation algorithm $F(D) = F_3(F_2(D, F_1(D)))$.

Using the molecular mechanics optimizer prevents a number of types of mistakes that the basic modeling kit algorithm can make. For instance, as was previously mentioned, the basic algorithm has no check to prevent interpenetration of atom nuclei. A simple example of this is the cyclohexane molecule and the difference between how it is drawn and the shape it takes. Cyclohexane consists of six carbons bonded to each other in a ring with each carbon additionally attached to two hydrogen atoms. A usual way to draw cyclohexane is as a hexagon of normal straight-line bonds – a shape which (to our system) implies that the carbons are coplanar. However, when six tetrahedral carbons (with angles between bonds of about 109 degrees) are placed together in a planar ring, two of the carbons interpenetrate as can be seen in Figure 3.7. While atoms are not solid objects per se, this interpenetration is highly improbable because of the natural repulsion of positively charged

```
public GenerateConformation(Molecule m)
{
    // Place the first atom at the origin
    Atom a = HighestPriorityAtom(m.Atoms);
    a.Location = (0,0,0);
    a.Connected = true;
    // Loop until all atoms have a 3D coordinate.
    while(!AllConnected(m))
    {
        a = HighestPriorityAtom(
           UnconnectedAtomsWithConnectedNeighbor(m.Atoms));
        Atom parent = ConnectedNeighbor(a);
        Connect(a, parent, BondBetween(a,parent));
    }

    Optimizer.Optimize(m,GAFF.Instance);
}

protected Connect(Atom child, Atom parent, Bond b)
{
    RotateToMatchDrawing(parent.Template, b.2DFeatures);
    Vector direction = BestUnusedTemplateVector(parent);
    child.Location = parent.Location +
        direction * BondLength(child, parent, b.Order);
    child.Template = TemplateLookup(child.AtomicNumber,
        child.NeighborCount, child.NeighborBondOrders);
    RotateForAlignment(child.Template, -direction);
    child.Connected = true;
}
```

Figure 3.6: *Algorithm for the modeling kit-based approach to conformation generation with drawn torsion interpretation and molecular mechanics post-processing.*

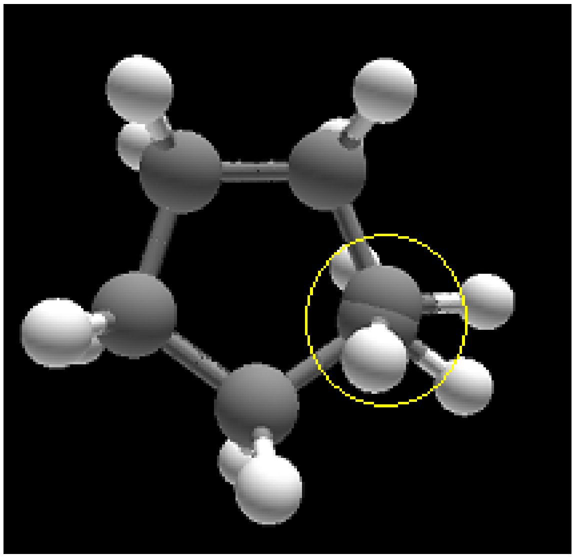


Figure 3.7: *A crude interpretation of cyclohexane taking the "straight" bonds in the diagram as being coplanar. The yellow circle indicates the two carbon atoms interpenetrating.*

nuclei. Having a standard interpenetration detection algorithm wouldn't solve the problem either since nuclei can be pushed together a little by other structure constraints. The molecular mechanics optimizer models these forces between the interpenetrating atoms and adjusts all of the atoms in the cyclohexane ring to assume one of its two typical forms, both of which are non-planar.

A second mistake inducing deficiency of the basic algorithm is that the algorithm can only deal with torsional angles where they are defined in the diagram. Other torsional angles between atoms are simply ignored. While a well-drawn molecule diagram does a good job of showing the anti-periplanar nature of a carbon backbone, side branches lack information about this constraint. A simple example here is a t-butyl functional group (a carbon attached to three other carbons). In this example, the hydrogen and carbon atoms in the t-butyl group should align themselves in such a way that maximizes parallelity and symmetry amongst themselves[10] and the other atoms in the molecule. This ideal arrangement is shown in Figure 3.8. However, without drawn cues to direct conformation construction, all torsional angles look similarly good and the ideal arrangement is unlikely to occur. This problem is inherent in actual plastic model kits as well. Accordingly, chemistry students are taught to build their models to minimize torsional strain. For our algorithm, adding the post processing step rotates the torsions until the ideal is achieved.

A third mistake the basic algorithm can make is that the above mentioned rules for choosing an atom template have exceptions. Foremost in the exceptions is dealing with template changes

[10] In actuality, the t-butyl group should maximize staggered torsional angles. This proper arrangement is most visibly distinguishable as three bonds to hydrogen becoming parallel to the bond to the group.

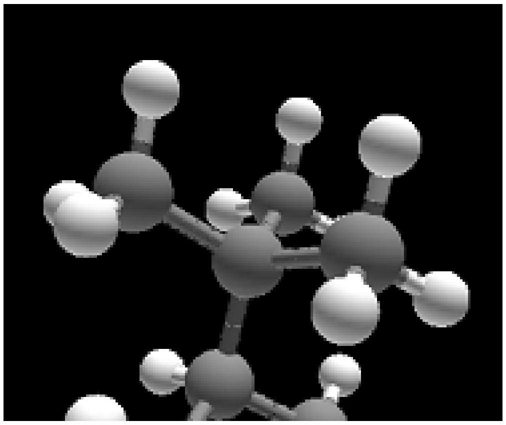


Figure 3.8: *A t-butyl functional group with the carbons properly aligned.*

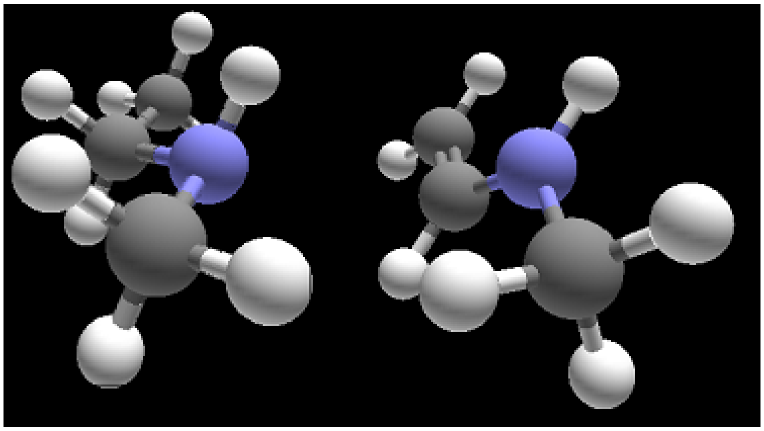


Figure 3.9: *Hybridizations of nitrogen. The nitrogen in both molecules are connected to two carbons and a hydrogen. The one on the left is in a pyramidal shape, while the one on the right has a planar shape due to the resonance with the double-bonded carbon neighbor.*

due to resonance. For example, a nitrogen atom bonded to three other atoms with single bonds normally takes a pyramidal shape. However, if the nitrogen is adjacent to a carbon participating in a double bond to a different atom, as shown in Figure 3.9, electrons will be shared between the double bond and the nitrogen thereby changing the nitrogen's shape to trigonal-planar. While creating a system to account for any one of these template rule exceptions is not difficult, determining and compensating for all such rules exceptions is a separate research task. In particular, it's the task of creating a well defined molecular mechanics equation and optimizer.

3.3.1 Molecular Mechanics at a Glance

Further improvements can be made to our algorithm by exploring the components of a molecular mechanics optimizer system. These systems consist of two parts which can be separated, improved separately for our task, and recombined. The first part is the model for the forces between atoms in the molecule. This model gives an explicit, differentiable formula for the calculation of the current energy of the molecule given a conformation. For a molecule with a given set of atoms and bonds, the energy function is defined over the domain $\mathbb{R}^{3N}$ where N is the number of atoms in the molecule and the input is the x, y, z coordinates of each atom. The second part is the optimization algorithm to modify the conformation into the one with the lowest energy. Due to the size of the domain, global minimization is a very difficult problem. Since the energy formula is differentiable and the gradient is analytically defined, annealing and gradient-based optimization techniques, such as gradient descent and conjugate-gradient, have been popular for this class of problems [62, 69]. Additionally, Wang [88] has given an interesting alternative using Branch and Bound after discretizing the solution space[11].

Because molecular mechanics energy formulas are approximations, there are different equations, or force fields, used for approximating different types of molecules. For instance the AMBER force field [67, 92] is designed for approximating energies of proteins, while the GAFF force field [90] modifies AMBER to approximate energies of simple organic compounds. Similarly, the CHARMM [13] and (Allinger's) MM4 [61] force fields each define the force field equation differently to focus on approximating observed forces in molecules. As we used an implementation of the GAFF force field in ChemPad, this dissertation makes reference to terms and properties of GAFF. However the techniques described generalize to the other force fields.

Force fields approximate energy as a sum over many terms, each of which can deviate from their equilibrium state. The greater the deviation, the higher the energy reflected in the term sum. First, the length of each bond is checked against the experimentally determined equilibrium for two such bonded atoms. Next, the angles formed by each set of three adjacent atoms is checked against the recorded equilibrium. Afterwards comes torsional angles. Finally, each pair of atoms which are more than three bonds away have their distances checked for steric strain. Simple molecules where all of

[11] An overview of optimization techniques for different molecular mechanics problems is provided by Leach [55]

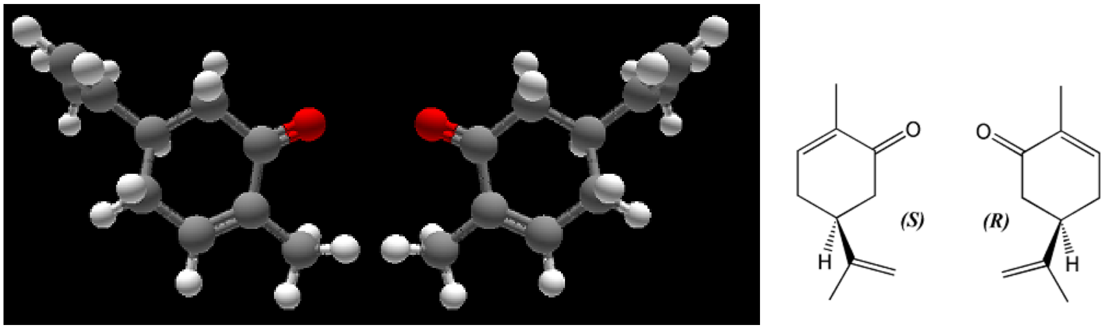


Figure 3.10: *Two versions of the carvone molecule. S-carvone is the smell and flavor of caraway. R-carvone is the smell and flavor of spearmint.*

these constraints can be satisfied simultaneously will have a very small energy[12]. Larger molecules tend towards terms in conflict with each other and non-zero energies. These four terms form the basic set of terms found in all force fields, but different force field implementations contain additional terms, such as hydrogen bonds and electrostatic effects, to replicate phenomena observed in their target set of molecules. Greater detail on force field equations are given in Appendix A.

3.3.2 Optimizations Which Violate Drawn Cues

Using a molecular mechanics optimizer as a post-processing step fixes the aforementioned problems associated with the modeling kit approach. Appropriate torsional angles for side-chain functional groups and other underdefined structures can be applied. Structures which are drawn as planar, but which are inherently not, such as cyclohexane and other rings, correct themselves. Shapes based on resonance and other "exception" rules correct themselves. Unfortunately, a number of new problems can arise during energy minimization. Because molecular mechanics has no concept of handwritten cues, certain structure requirements detected by the handwriting heuristics can be lost during optimization. While the molecular mechanics optimizer may generate a conformation that follows the rules of chemistry, it may very well no longer match the way the molecule was drawn.

The first, and most catastrophic of these structural information losses can actually result in the wrong molecule being generated by the system. The two molecules pictured in Figure 3.10 are carvone molecules which have the same formula and the same connectivity, but are different 3D structures. Indeed, they are mirror images of each other. In chemistry parlance, the molecules are *stereoisomers*, the atom defining the shape is the *stereocenter*, and carvone is *chiral*. Because distances and angles are identical in mirror images, molecular mechanics considers these molecules to be equally optimal and either solution is acceptable to a minimization task. The 3D difference is important though. Because one cannot be converted into the other without breaking a bond, the

[12] Note that steric strain is defined for many pairs of atoms and that there will be weak repulsion or attractions between these distant atoms. Therefore, even with all other terms in perfect equilibrium, the calculated energy will not be zero for molecules larger than ethane.

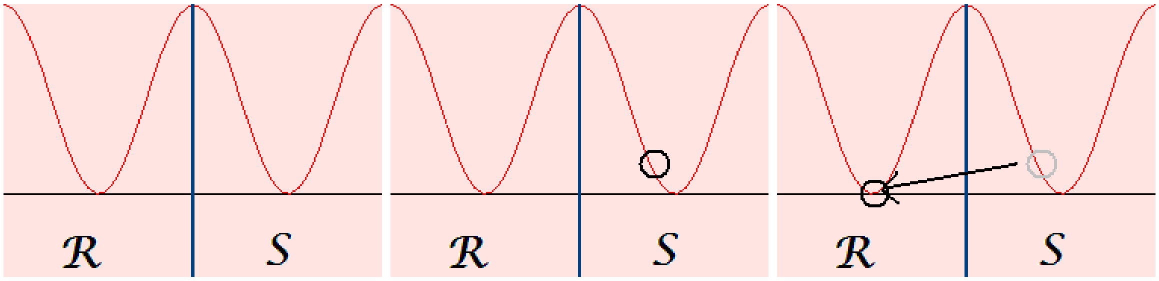


Figure 3.11: *A molecular mechanics optimization over a cross section of the energy surface. Both the R and S minima are equally optimal numerically, but they represent different molecules. In this cross section, we see a hypothetical conformation created close to the S minimum, presumably because the drawn diagram indicates the S relationship. Unfortunately, the optimizer can accept either minimum as valid.*

two molecules have different chemical properties. The one on the right is the molecule responsible for the taste of spearmint while the molecule on the left is the one responsible for the taste of caraway. Since the differences in molecules such as these are not detectable in the connectivity of the molecule, chemists use notations, such as wedge and dash bonds, in their sketches to convey the correct 3D shape.

The underlying problem with the optimization is illustrated in Figure 3.11. Here we are generating a conformation for a hypothetical chiral molecule. The first image shows a simplified one-dimensional cross section of part of the force field for the chiral molecule. On one half of the graph, the conformation makes the stereocenter rectus (R) and the other half makes the stereocenter sinister (S). The second picture shows the starting location on the graph for a conformation hypothetically generated by the basic modeling kit algorithm. The input drawing indicated an S stereocenter and the current solution is very close to the S minimum. If the optimization were only occuring on this one dimension, the correct answer would be reached with a steepest descent, conjugate-gradient, or other similar algorithm. Unfortunately, the real minimization is multi-dimensional, the optimizer may employ hill-climbing techniques to find better minima, and the two minima on this dimension are equally good to the optimizer. So, the optimizer can legitimately change the chirality to R to generate a minimum. Depending on the starting conformation and the optimizer implementation, this sort of change does occur in practice resulting in the generated conformation for the wrong stereoisomer (R).

The second class of problems that arises after optimizing conformations is that of handwritten notations which purposely depict molecules in non-optimal conformations. Since molecules are constantly changing, it is often useful to draw a molecule in a higher energy conformation. This could be a conformation which is chemically relevant but is not the energetically optimal one. For example, a carbon backbone can be drawn with an eclipsed, syn-periplanar conformation to show a transition state necessary for a reaction to occur. Here there would be a good deal of torsional

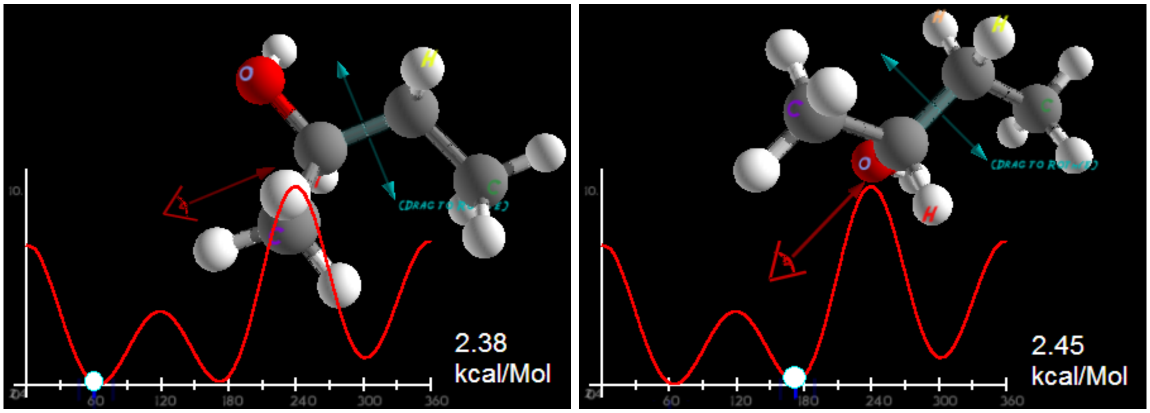


Figure 3.12: *(R)-2-butanol and its energy curve for rotation of the bond between carbon 2 and carbon 3. A GAFF calculation of the energy of the molecule indicates the energy is slightly lower on the left where the oxygen is coplanar with three carbons than on the right where the four carbons are coplanar. However, chemists may prefer to think of this molecule with the entire carbon backbone coplanar and draw it as such.*

and possibly steric strain. Since the optimizer is oblivious to the way the molecule was drawn, a molecule drawn as such would be forced into the lower energy anti-periplanar conformation despite the chemist's intention in the diagram.

Similarly, chemists sometimes think of molecules in non-optimal conformations when there are alternate conformations which are close to the optimum and fit more readily into existing mental frameworks. For example, in GAFF, a 2-butanol molecule, such as shown in Figure 3.12, has an optimal energy when the oxygen is anti-periplanar to the ethyl end of the carbon backbone. However, chemists largely prefer to think of the methyl end assuming the anti-periplanar position which makes all the carbons planar. A molecule sketch disambiguates the intention of the chemist, but molecular mechanics optimization will favor whichever conformation has the lower energy.

As opposed to these first two problem classes, a third class of problems where the optimizer cannot take drawn information into account is one which the basic modeling-kit algorithm could not handle either. The modeling kit algorithm assigns torsional angles based on the assumption that the molecule has been drawn entirely from a "top down" perspective. However, sometimes a molecule sketch will be made from an alternate view angle or even multiple view angles as shown in the pictures of cineole and diosgenin in Figure 3.13. These sketches use perspective cues less explicit than wedge and dash bonds to indicate 3D structure. Instead of assuming the entire drawing is from the "top down", the chemist is expected to be able to see the 3D in the line drawing and interpret the conformation correctly. Once again, the molecular mechanics optimization will ignore these drawn cues and produce conformations which simply lower the overall energy in the molecule.

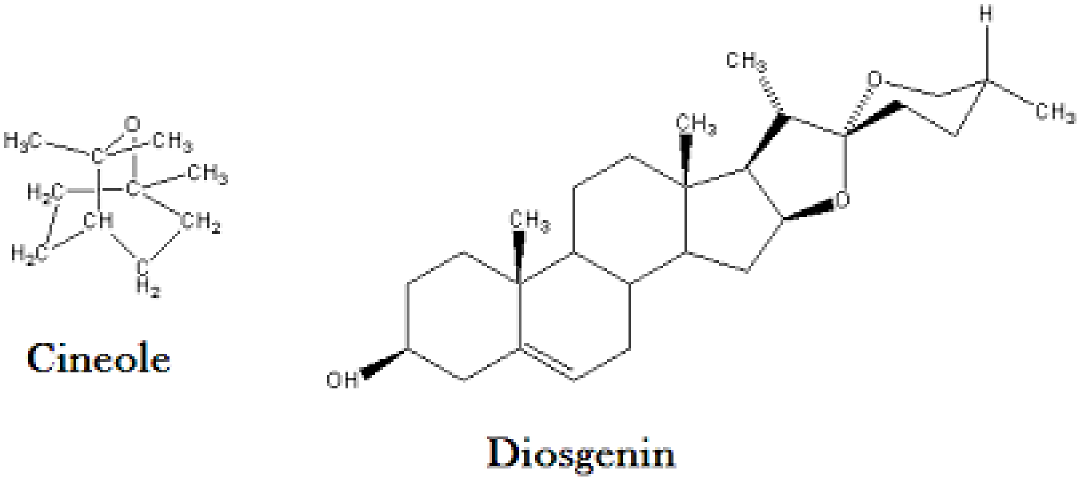


Figure 3.13: *Typeset diagrams of the cineole and diosgenin molecules. The cineole molecule is drawn from the side showing an explicit "boat" structure to the ring. The diosgenin molecule contains several rings drawn from the "top-down" view and a ring at the right drawn from the side indicating its relationship with the neighboring ring and the equatorial position of its methyl group.*

3.3.3 Multi-pass Optimization

For stereochemistry, the first class of problems, it is reasonably straightforward to detect when the conformation output by the optimizer no longer matches the drawing. Once detected, the problem can be potentially solved by the simple heuristic of switching the locations of two of the involved atoms and reoptimizing the solution. Therefore, we can modify our modeling-kit algorithm into the multi-pass system in Figure 3.14. The first pass was the same as our original algorithm and produced an energetically good conformation which may or may not actually match the drawing. Thereafter, we check for errors, apply the error-fixing (swap) heuristic and reoptimize until no such errors are found. Unfortunately, there is no guarantee that the optimization step will not simply return the conformation to a state we have already visited thereby causing an infinite loop. At some point, the algorithm needs to give up and inform the user that a viable conformation could not be found.

In general, while using a multi-pass system could fix many conformation errors, it could never guarantee that both parts of the system would agree on a conformation. The two knowledge systems could potentially be in conflict with each other. Indeed, this was certain in the second class of problems where the diagrams depicted non-optimal conformations. To overcome this, a single system with a sophisticated understanding of both molecule diagrams and molecule forces is needed to satisfy both types of constraints simultaneously.

```
public GenerateConformation(Molecule m)
{
    // Place the first atom at the origin
    Atom a = HighestPriorityAtom(m.Atoms);
    a.Location = (0,0,0);
    a.Connected = true;
    // Loop until all atoms have a 3D coordinate.
    while(!AllConnected(m))
    {
        a = HighestPriorityAtom(
           UnconnectedAtomsWithConnectedNeighbor(m.Atoms));
        Atom parent = ConnectedNeighbor(a);
        Connect(a, parent, BondBetween(a,parent));
    }

    Optimizer.Optimize(m,GAFF.Instance);
    // Check for stereochemistry errors.
    while(ErrorDetector.Errors(m) > 0)
    {
        // Fix each error
        foreach(Error e in ErrorDetector.Errors(m))
        {
            // Find the atom where the neighboring atoms are
            // incorrectly placed relative to each other
            Atom errorSpot = e.Location;
            // Trade their locations.
            m.SwapSubbranchLocations(m.Neighbors(errorSpot)[0],
                                     m.Neighbors(errorSpot)[1]);
        }
        // Reoptimize.
        // Note, this may cause new stereochemistry errors.
        Optimizer.Optimize(m,GAFF.Instance);
    }
}
```

Figure 3.14: *Adding multi-pass optimization to the algorithm from Figure 3.6. Stereochemistry errors are detected and changed with simple heuristics.*

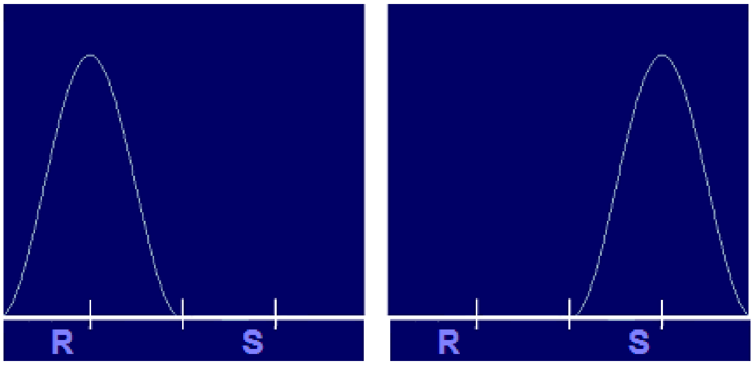


Figure 3.15: *Different forms of an example IM3 term. Here a term for distinguishing R and S stereocenters has two forms. Each form is a continuous function which is added to the force field. The R form discourages S conformations by adding energy only to the S side of the energy surface. The S form adds energy only to the R side of the energy surface. Combining one of these forms with the energy function in Figure 3.11 disambiguates the desired energy minimum.*

3.4 Ink-Modified Molecular Mechanics: Combining Diagram and Chemistry Requirements

Rather than creating an entirely new system from scratch to contain both drawing and chemistry structure knowledge, we can add the drawing knowledge directly into a molecular mechanics force field. This technique, which we call Ink-Modified Molecular Mechanics (IM3), formulates molecule diagram constraints into terms that are compatible with force field terms and can therefore be appended to the standard energy equation. An IM3 force field calculates energy for a given conformation *and* its drawing. It returns a low energy when the conformation matches the diagram and is naturally low in energy. A high energy represents something improbable in the interpretation of the diagram, or the arrangement of the atoms. Therefore, a molecular mechanics optimizer, given an IM3 force field, solves for both classifications of constraints at the same time.

As they are an important contribution of this dissertation, the added terms of an IM3 force field are described fully in Chapter 4. In brief, penalty terms exist to solve the three types of problems noted in Section 3.3.2. Energy is penalized when the wrong stereoisomer is generated, a drawn suboptimal torsion is ignored, or when perspective cues are violated. These penalties are in the same form as real molecular mechanics energy penalties for compatibility with standard optimizers, but the IM3 penalty terms are tracked separately so that the system can tell which term is being violated.

In contrast to the regular force field terms which have single forms, these drawing cue-based terms typically have two or three different forms, one of which is "on" at a given time. That is to say, while a regular force field term, such as bond length, may have different constants in the equation based on the involved atoms, the energy graph of all bond length terms look the same. These IM3 terms have different forms based on discrete, detectable conditions in the diagram. For instance, going back to the example in Figure 3.11, the associated term determines from the drawing whether the R or S form is indicated. If the R form is desired, the IM3 term applies the form which adds energy only on the S side of the energy surface. Conversely, if the S form is desired, the term applies the other form which adds energy only to the R side of the energy surface. While the entire energy surface we optimize over remains a continuous function, IM3 makes discrete changes to that function to ensure the optimization produces answers which match the diagram.

3.4.1 Conformation Generation using an IM3 Engine

With an IM3 optimizer, we could simply replace the old optimizer from Figure 3.14 and get an optimized result which no longer ignores drawing cues. However, much greater overall improvement can be made be reexamining the conformation generation technique from the beginning. While the old system with IM3 can handle slightly more difficult test cases, there is still a low cap on the size and complexity of the diagrams that can be accurately handled. This is due to the lack of chemistry knowledge in the initial modeling kit algorithm which provides the optimization starting location. The optimization techniques employed here are generally good at finding local minima close to the starting state as they are based on gradients. Universally reaching the global optimum with these algorithms is unlikely and algorithms which find the global optimum are too computationally intensive for our interactive purposes. Therefore, to prevent these early errors due to lack of understanding, it would be preferable to use all the information in the IM3 force field instead of templates when constructing the conformation. In other words, we would like the force field to drive the conformation generation process.

This change removes our abstraction of comprising F as the direct combination of subalgorithms F_1, F_2, and F_3. Instead we will continue to use our modified molecule mechanics optimizer F_3^{IM3} iteratively in F. At each iterative step we produce a partial conformation C_i with i atoms placed. Given a new subalgorithm $F_4 : a_i, C \to C$ which simply gives the atom a_i a somewhat arbitrary 3D location in the conformation, we then move the atom to the correct location using $C_i = F_3^{\text{IM3}}(F_4(a_i, C_{i-1}))$. This is repeated until we have the final conformation C_n.

Previously, when we attached a new atom during conformation generation, its starting location was determined by using drawn cues to pick an attachment direction vector from the parent atom's template. With IM3 acting as our template replacement in the conformation generation system, the process becomes simpler. We know that neighbor atoms are generally going to distribute themselves about a parent atom so that they are not close together. If they were close together, their electron shells would overlap and repulse each other[13]. When adding a new atom, we therefore place the new atom generally "away" from its siblings, and then run a local or "limited" optimization on the atom

[13] An exception to this would be if the sibling atoms are bonded as in cyclopropane.

```
public GenerateConformation(Molecule m)
{
    // Place the first atom at the origin
    Atom a = HighestPriorityAtom(m.Atoms);
    a.Location = (0,0,0);
    a.Connected = true;
    // Loop until all atoms have a 3D coordinate.
    while(!AllConnected(m))
    {
        a = HighestPriorityAtom(
           UnconnectedAtomsWithConnectedNeighbor(m.Atoms));
        Atom parent = ConnectedNeighbor(a);
        // Direction which maximizes the distance to the nearest neighbor
        Vector direction = NextThomsonLocation(parent);
        // Note the energy before the addition.
        double energyBefore = GAFF.Instance.Energy(m);
        // Connect the child atom -- note the new Connect method
        Connect(a, parent, BondBetween(a,parent),direction);
        // Optimize just the location of this atom
        Optimizer.LimitedOptimization(m,IM3FF.Instance);
        // Compare the energy now to that from before. If there
        // is too much increase, optimize the whole molecule.
        double energyAfter = GAFF.Instance.Energy(m);
        if(energyAfter - energyBefore > THRESHOLD)
        {
            Optimizer.Optimize(m,IM3FF.Instance);
        }
    }
}


protected Connect(Atom child, Atom parent, Bond b, Vector direction)
{
    // Place the child atom in that direction.
    child.Location = parent.Location +
        direction * BondLength(child, parent, b.Order);
    child.Connected = true;
}
```

Figure 3.16: *Using IM3 as a replacement for templates in previous algorithms. Here the new atom is placed not based on any template, but only away from the atoms it neighbors topologically. A limited optimization (allowing only the new atom to move) using the modified force field then moves the atom to the best position. If the addition increases energy above a threshold, a full optimization is used.*

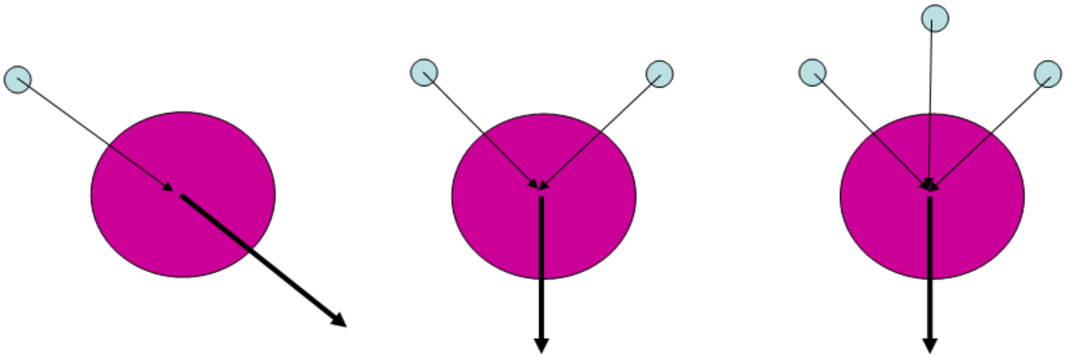


Figure 3.17: *The Thomson problem's solutions for starting positions in atoms of low valence. The direction which maximizes the minimum distance is usually the sum of the normalized input vectors from existing atoms. For only one neighbor, we place the new vector directly away. For two neighbors the new vector is in the same plane splitting the large angle. For three neighbors forming a tripod, the new direction is the top of the tripod. Degenerate cases where this sum is small (such as two atoms directly across from each other), have their solution on the cross product of the input vectors.*

added to bring it to a close local energy minimum. The pseudocode for this is shown in Figure 3.16. In a simple molecule lacking confounding factors and an appropriate choice of the "away" vector, this process will bring the atom to rest on one of the previously defined template vectors.

We are defining the "away" vector here as being in a direction which maximizes the minimum angle formed with the parent atom and a sibling atom. This starting configuration roughly corresponds to the Thomson Problem and the related Plum Pudding Model of the atom [83, 93] where electrons were believed to be distributed throughout an atom and were trying to distribute themselves away from each other to minimize the repulsion of the electrons. While the problem's general form for distribution of points on a sphere is as of yet unsolved and the plum pudding model was disproved, for our purpose of creating starting positions for atoms with low valence, this approach is straightforward and effective. Figure 3.17 illustrates finding solutions for the Thomson problem where the number of neighboring atoms is four or less as is the case with our generator [6].

The "limited optimization" mentioned above refers to running an optimization where only a subset of atoms in the molecule are allowed to move. Most often, we only allow a single atom, the newest one, to move in the optimization. This can be accomplished by artificially setting the force field's gradient to 0 for the position of atoms not allowed to move before giving the result to the optimizer. Usually being able to move only the most recent addition is sufficient to create a low energy partial conformation and this technique causes the optimizer to converge quickly as each atom is added. However, the optimizer is not always able to produce a low energy result using only the most recent atom. Ring closure is a good example of this. With the addition of the last atom, the system can only produce a low energy conformation if all of the other ring atoms have been properly positioned to form the final shape of the ring. When this, or some other local conformation conflict cannot be resolved with the limited optimization, we can detect the problem by comparing the energy of the conformation after the atom was added to the energy from before the atom was added. When this energy difference, or delta, is above a threshold[14], the algorithm invests the time required to perform a full optimization with all atoms allowed to move in hopes of overcoming the local conflict.

3.4.2 A Search Framework for Conformation Generation

One problem that can arise while using the IM3-based conformation generation algorithm in Figure 3.16 is that as each atom is added, the limited optimizations can only optimize over relationships defined in the current state of the molecule. A low energy partial molecule can be constructed from which we cannot add an atom without greatly increasing the energy. For instance, consider the process for assembling the carvone conformation from Figure 3.10 with the slight modification that the hydrogen at the stereocenter be implicit instead of explicit. The IM3 term which handles this form of stereochemistry is calculable when the three carbon neighbors of the stereocenter are present in the molecule and is left as zero before then. However, the actual stereochemistry is determined as soon as *any* three neighbors are present. This is because the third atom adjacent to a stereocenter[15] to be added has two low energy regions it can occupy, but there is only one remaining location when the final atom is placed. If there is no IM3 term encouraging the correct stereochemistry at the time the third atom is placed, both locations will look favorable. Therefore, if the three carbon atoms which define the IM3 term are not added before the hydrogen, the stereocenter's chirality is random.[16]

Our solution to this problem is to change our approach from linear construction of a conformation to a tree search of the space of conformations. We still add atoms one at a time to the conformation, but create multiple execution paths for other choices we could have made for that addition. Figure 3.18 illustrates the basics of the algorithm with a best-first search. As we are placing atoms into a continuous space, one could argue that there are an infinite number of possible choices over which to search. Therefore, we must choose a means to discretize these placements into a small number of possible choices from which we define our search tree. The discrete, and therefore obvious, first choice upon which we can form a branch is based on which atom is placed at each step. We draw from the top p choices of the atom prioritization scheme at each step. For the continuous space of places to put atoms, the second choice for search branching is based on the different local minima the added atom can come to rest upon after limited optimization. There are usually multiple possibilities for this branch and we ideally would like to find all of them. To accomplish this, we change the location where we start the atom before minimization. While we still start the new atom "away"

[14] We use a threshold of 10kcal/Mol.

[15] Which is by definition sp^3 hybridized.

[16] Note that in our atom prioritization system, implicit hydrogens always come last and therefore this example never actually has random output in the given algorithm. However, the problem is present in the less straightforward Diosgenin diagram in Figure 3.13. Here the fused ring atoms have the highest priorities and can determine actual stereochemistry before the atoms on wedge bonds are connected.

```
protected double CostFxn(PartialConformation m)
{
    return GAFF.Instance.Energy(m) + m.AtomsNotConnected * USER_CONST1 +
          m.AreNonHydrogensNotConnected * USER_CONST2;
}
public GenerateConformation(Molecule m, int atomBranches)
{
    // Create a priority queue of partial conformations sorted by
    // our cost function
    PriorityQueue<PartialConformation,CostFxn> queue;
    // Create the root nodes
    List<Atom> highAtoms = HighestPriorityAtoms(m.Atoms, atomBranches);
    foreach(Atom a in highAtoms)
    {
        PartialConformation node = new PartialConformation(null);
        node.Atoms[a].Location = (0,0,0);
        node.Atoms[a].Connected = true;
        queue.Push(node);
    }
    // Get the high priority node and add an atom
    while(queue.Size > 0)
    {
        PartialConformation node = queue.Pop();
        // Check if we're done
        if(AllConnected(node.Molecule))
            return node.molecule;
        // Branch based on best atoms to use
        List<Atom> highAtoms = HighestPriorityAtom(
           UnconnectedAtomsWithConnectedNeighbor(node.Atoms));
        foreach(Atom a in highAtoms)
        {
            Vector idealDirection = NextThomsonLocation(parent);
            // Branch based on vectors around the ideal vector
            List<Vector> directions = VectorsAround(idealDirection, 0.5);
            foreach(Vector d in directions)
            {
                Atom parent = ConnectedNeighbor(node.Atoms[a]);
                Connect(node.Atoms[a], parent,
                    BondBetween(node.Atoms[a],parent),d);
                ...limited and full optimization code goes here...
                queue.Push(newNode);
}   }   }   }
```

Figure 3.18: *Using search in the conformation generation algorithm. Here a priority queue sorts partial conformations based on their current energy and the amount of work they have left to do. When a partial conformation is popped from the queue, it is expanded into a number of new queue items using different atoms to add and different starting positions. The starting positions are described in Figure 3.19*

from its neighbors before limited optimization, we expect the ideal "away" vector to be at a local energy maximum around which the minima are distributed. Therefore we change over to starting the atom in four[17] different locations close to the original "away" vector and equally spaced around it as shown in Figure 3.19. As the atom falls into the closest minimum, we note the different local minima generated by each starting location and store these options as potential search branches.

For the purposes of a best-first search, which expands the search node considered to be the most promising, we need an easy to calculate metric of the quality of each node in the search space. We utilize a quality function (depicted in the pseudocode in Figure 3.18) based on the sum of the energy of the partial conformation and the number of atoms remaining to be added. The energy of the partial conformation discourages expansion of nodes which are irreparably in high energy states. The penalty for the number of atoms remaining discourages a slow, breadth-first style search. Initially, we thought to attempt to find an optimistic cost function so that we could use A^* search. However, the addition of most atoms to the conformation add almost no energy to the molecule thereby making it difficult to find a tight, optimistic cost function.

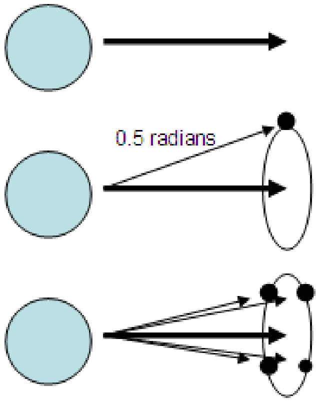


Figure 3.19: *Starting positions for an atom. At the top, we have the parent atom and the "away" vector for the starting position of the child atom. To create multiple starting positions, we rotate the vector up 0.5 radians to get the first position and then create additional positions equally spaced around the initial vector.*

Besides improving the results of conformation generation, using a search strategy has the side effect of providing multiple conformations (in multiple minima) for a given diagram. This more accurately represents the actual state of the molecule than having only a single conformation and is therefore quite useful to achieving our pedagogical goals for ChemPad. For instance, in a cyclohexane molecule, using a search scheme is likely to generate both of the classic "chair" and "boat" conformations. Since execution time is a critical issue for anything used in an interactive system, we need not wait for the search space to be exhausted in pursuit of these alternate conformations before returning an answer to the user. With a search, we can return the first completed conformation to the user immediately and continue the search as a low-priority process. As additional conformations are found, they are added to the user's result set.

[17] The number four here is chosen based on common electron hybridizations found in the types of molecules in which we are interested. *sp*, sp^2, and sp^3 hybridizations would have no more than three local minima for the atom to fall into (assuming no other interfering forces) and they are equally spaced around the atom. Using four equally-spaced starting points therefore results in finding all of the local minima.

3.4.3 Comparing Nodes

An interesting implementation challenge when creating this search system was trying to minimize duplicated computation. In the conformation generation search space, there are numerous search nodes which are effectively the same. When an atom is added, it is tried in enough starting positions to be likely to find all the local minima. However, multiple starting positions could yield the same minimum, thereby creating duplicate conformations. Atoms which do not interact much could be added in opposite orders in different branches of a tree, thereby creating duplicate conformations. When rings are completed and other high energy additions cause the optimization phase to optimize over multiple atoms, otherwise different search paths all yield the same conformation, possibly the only one of low energy remaining. With all this duplication, there is significant gain to be made in the speed of search by recognizing a duplicate path and not traversing it again.

The first kind of duplication, one where an atom is placed in the same position from multiple starting locations, is easy to detect. Each search node keeps a list of its sibling nodes. Here, "sibling" nodes are all the offspring of a parent conformation trying to add the same atom, just with different starting locations[18]. If limited optimization succeeded in optimizing by only allowing the newest atom to move, we can simply check the distance between the location of the newest atom in the two nodes. If the distance is less than some epsilon value[19], the nodes are equivalent as we know no other atom was moved under limited optimization.

Detecting duplicates is more complicated in the other cases where we need to detect rotationally equivalent conformations and therefore cannot simply compare 3D coordinates of atoms on a small set of nodes. It becomes necessary to keep a dictionary of traversed nodes to compare against instead of just a sibling list. As we know identical partial conformations contain the same atoms by definition, we make this dictionary keyed on the atoms present in the conformation. In particular, we use as key a list of boolean values for each atom in the specification. Since the length of this list could potentially be larger than the register size of the computer, we cannot store the booleans as bits in an integer, but must use an object of dynamic length such as a string (of the boolean values). Once we know two partial conformations have the same atoms, we need to perform some structural comparison to tell if they are equal. We could perform the structure comparisons using general techniques for comparing 3D objects [42, 12]. However, for the case of molecules, we can exploit molecular mechanics again to create a simpler and more useful comparison. Two equal conformations will have equal energy and equal values for each molecular mechanics term. Therefore, we can quickly compare the precomputed energies of the two nodes to see if they have the possibility of being duplicates. If so, we proceed to go through the slower process of comparing each term in their force field. This molecular mechanics-based comparison has an important advantage over general 3D object comparison techniques in that the parameters which set the maximum amount

[18] These sibling nodes come are used again in Limited Discrepancy Search in Section 3.4.4

[19] We use an epsilon of 0.05 angstroms.

of allowed difference are based specifically on properties we understand with respect to molecules such as bond length and torsional strain. As they are based on molecule properties, setting these parameters to get the desired results is intuitive. The disadvantage of this system with respect to general techniques is that the comparison requires an alignment of the atoms between the two conformations. In other words, we need to know for each atom in the first conformation which is the corresponding atom in the second conformation. We can track this alignment during conformation generation by keeping track of the diagram atom that generated each 3D conformation atom i.e. each $\bar{a}_i$ keeps a pointer to a_i. For general cases of comparing molecules without known alignments, we give further details in Chapter 5.3.1.

One final implementation detail to note is that in all of these steps for comparing search nodes, there is a requirement that the node have undergone limited optimization before the comparison is made. Otherwise the 3D data upon which the comparison is based would still have the new atom in its high energy, meaningless starting position. Unfortunately, optimization, even limited optimization, is computationally expensive and it is worthwhile to minimize the number of optimizations performed in the search. Therefore, instead of optimizing each child node as it is created, we flag them as unoptimized and do not perform the optimization until the first time the node is examined in the search. This complicates the algorithm in Figure 3.18 in that each time a node is explored, it must be optimized first and then have its status as a duplicate checked on the fly, rather than performing the limited optimization at the time of the node's creation. Moreover, it becomes valuable to have the cost function penalize unoptimized nodes to avoid breadth-first search.

3.4.4 Search Techniques

We can get around this complication of the cost function and make a big improvement in the speed of the search by rethinking our cost function. As previously mentioned, the best-first search algorithm put forward in Figure 3.18 suffers from the quality of its cost function. If we based the heuristic function only on the current energy of the partial conformation, the small energy increases typically found from one partial conformation to the next makes the search breadth-first in practice. This is too slow and requires too much memory for our Tablet PC-based input technique. Therefore we applied a penalty based on the number of unconnected atoms. If we weight the constant factor multiplied by the number of unconnected atoms too heavily, the algorithm performs as a depth-first search and then starts exploring the search space from the bottom up. While this is computationally efficient, the performance is mostly the same as that of the no search algorithm back in Figure 3.16. Only the most severe of mistakes in the search will cause backtracking and it will usually take a great deal of time searching the bottom of the search space to find an alternate conformation once the first is found.

A better heuristic function can be conceived through analyzing our search strategies and making the observation that the depth-first algorithm does a good job at most of the steps of the search.

```
protected double CostFxn(PartialConformation m)
{
    return (m.DiscrepancyCount * (TotalAtoms.Count + 1)) +
       m.ConnectedAtoms.Count;
}
```

Figure 3.20: *Limited Discrepancy Search ordering. Here the energy of the local node is not considered, but only how the number of discrepancies used followed by the number of atoms left to add. A discrepancy is whenever we do not use the heuristically chosen "best" path.*

Usually, the new atom is placed in a position which can lead to a conformation at a global minimum. In practice, there are only a small number of key decision points in the search where we want to explore a path other than the first one we choose. Therefore, we can reformulate the evaluation heuristic to sort primarily by the number of alternate paths, or *discrepancies*, needed[20]. Figure 3.20 shows the new node ordering heuristic. This sorting scheme is known as limited discrepancy search as described by Harvey and Ginsberg [40].

While up to now we were treating all children nodes of the same parent search node as equal, it now behooves us to make a judicious decision as to which one is the primary, discrepancy-less, child. For the children based on the different atoms to be added, we already have a sorting scheme. That would be the atom prioritization heuristic. However, we do not have a way to sort the different starting positions for each of those atoms. Once again we look at the drawing cues and torsional angles to help us. If the atom is involved in a torsion explicitly drawn in the diagram[21], we can look at the turn directions of the torsion in that diagram. If they are both the same direction (two left turns or two right turns), we would expect a starting position which has two identical turns in 3D (two clockwise turns or two counter-clockwise turns) to be more probable. Similarly, if the drawing has a turn in each direction, we would expect a starting position with the same to be more probable[22]. While turn direction does not necessarily distinguish a single starting position as being better than all the others, it is enough to identify some starting positions as better than others. We can simply take one of the better ones as the primary child.

Partially because of this ambiguity regarding which search node child is the best one, we can speed up the search by using one of the proposed modifications to limited discrepancy search put forth by Harvey and Ginsberg. This modification is to not count local, correctable mistakes against the number of discrepancies counted. When a node fails one of the feasibility checks that determines if it is expanded or rejected, the failure is frequently due to having selected the wrong starting position

[20] How we determine the number of alternate paths needed is later detailed in Section 3.4.5.

[21] If the atom is an implicit hydrogen, there is no cue and a primary child is chosen randomly. Explicitly drawn atoms will be involved in a torsion once the fourth explicit atom is added to the conformation.

[22] Identifying the turn directions in 2D and 3D is explained in Appendix D.3.

for this newest atom, rather than being a structural problem higher in the search. If we therefore allow the search to try the sibling search nodes which were created with different starting positions for the atom at no discrepancy penalty[23], we are likely to recover from the failure.

As noted before, when the last atom is added to a ring structure, a number of atoms in the conformation need to be moved to make the conformation viable. In these cases, the optimization may have difficulty creating a viable ring because of earlier structure decisions. Unfortunately, in our sorting of discrepancies, there is no way to prevent the choosing of a different atom to be placed rather than placing this *bridging atom* which closes the ring. Because of this, the algorithm can expend a great deal of wasted computation searching the lower part of a doomed search branch by repeatedly avoiding the inevitable placement of this atom. In these cases, we want the search to fully expand the search tree above the bridging atom to find the combination that allows the ring to close. To account for this, whenever we notice that the highest priority atom is a bridging atom, we remove other choices by not generating children for the other high priority atoms.

3.4.5 Reality Checks and Enforcement Policies

Using limited discrepancy search requires us to run a reality check at each node and detect whether we have made a mistake or not. Previously, we were sorting based on energy and the best energy solutions would be expanded upon. However, for this algorithm we need to be able to make a hard decision at each node whether to accept it and expand its children or reject it. We accomplish this determination by looking at the energy contributed to the partial conformation's sum recorded energy by each source in that sum. Our first source for that sum is the standard force field energy. While we expect the energy from this source to grow with the molecule (usually in spurts), very large jumps in energy, such that would cause an actual molecule to break apart, indicate a failure on the part of the search. So, when the delta between an interim conformation's energy and that of its parent is above a high threshold, the node is pruned.

We also use the energy of the IM3 terms for detecting errors, but in a much different fashion. If there is any energy present in one of the terms, we know the built conformation is violating the constraints of the diagram. However, whether we reject the conformation depends on the desires of the user. For instance, having a stereochemistry violation is a serious error since we have generated a molecule different from the one that was drawn. In such a case, the user would almost certainly not want such a conformation returned. However, in the case of the perspective cues not being perfectly followed, the user may have unknowingly drawn the molecule with non optimal perspective cues. Especially with a target user base of students learning chemistry, the potential of this kind of mistake is high. Therefore we need different policies on how to enforce the IM3 terms. An expert might expect each term to be fully enforced and reject any conformation which violates an IM3 term. A

[23] In implementation, the discrepancy count for these siblings is decremented when this occurs.

novice would want the search to tend towards the way the diagram was drawn, but may be willing to accept some differences for educational purposes.

To allow this sort of customization, we allow the user to set a separate enforcement policy for each of the IM3 terms. A *strict* enforcement policy requires that the term be adhered to always. Any energy present in a strictly enforced term at a search node causes the partial conformation to be rejected. Alternatively, a *loose* enforcement policy does not cause energy present in the term to reject the conformation. However, since there is still energy present, the optimizer will try to move the conformation away from violating the term. Finally, a *none* enforcement policy causes the calculation of the term to be skipped. Even if the conformation would violate the term, there is no penalty against it in any part of the conformation generation system.

One caveat with providing this kind of customization is that the user is unlikely to understand the options or care to set them up. As our target audience is unsophisticated students, we could simply provide a default set of enforcement policies which we expect would work well for this group. However, as we can detect the IM3 term drawing cues directly from the diagram, it is also possible to poll the user at the beginning conformation generation as to whether or not certain cues were intended and how strongly to emphasize those cues.

3.4.6 Prioritizing Atoms

As mentioned in section 3.2.1, the order in which we add atoms to the conformation can have a great effect on the generated result. For instance, implicit hydrogens have little effect on the overall structure of the molecule and placing them early can make it difficult to add structure defining atoms afterwards. Figure 3.21 displays the priority ordering for atoms we can legally attach to the conformation. This ordering system was developed through intuitions about which atoms in a molecule most define the structure of that molecule. In general, we believe the atoms which have greater restrictions upon them should come before atoms with fewer restrictions and that atoms which are part of larger substructures should come before those in smaller substructures. This prioritizing is not infallible, but does act as a useful heuristic.

1. Explicit atoms.
2. Ring Size.
3. Chain Length.
4. Wedge and dash bonds.
5. Most Neighbors.

Figure 3.21: *Priority order for atoms to be added to the conformation.*

When two atoms are compared, the first check is whether both atoms are explicitly drawn in the diagram. If one is not, it gets a lower priority than the explicit one. This first step greatly reduces the complexity of the problem by ordering the large number of implicit hydrogens present in organic molecules last where they have little impact on the structure of the molecule. Since atoms' types have already been assigned and these types account for hybridization, we expect these implicit hydrogens should have very little impact on the overall conformation as opposed to the larger, explicit atoms[24]. The second check is to look for the atoms' membership in rings. Rings determine more of the structure of the molecule than straight chains of atoms because of the constraints of ring closure, so an atom with membership in a ring will get a higher priority than one which is not. Furthermore, for distinguishing between two atoms in rings, the one with the larger ring gets priority. The third check is to look for the atoms' membership in long chains such as an organic molecule's carbon backbone. The longer chain atoms get a higher priority than shorter chain atoms. The fourth check is for atoms with wedge and dash bonds as they have more explicit structure than atoms without. The atom with more of these perspective cues gets the higher priority. Finally, if two atoms are equal in all of the above ways, the number of neighbors the atom has acts as a final tie breaker. If that too is identical, the atoms are left as equal.

3.5 Future Work

In this chapter we presented a series of increasingly sophisticated approaches to the task of conformation generation. Our initial approach, which treated atoms as the plastic pieces one finds in a molecule modeling kit, didn't have enough understanding of molecule structures to produce conformations for molecules with more than a couple small functional groups. Applying molecular mechanics optimizers to the task somewhat helped alleviate this limitation but introduced the problem of optimizing to structures not represented by the drawing. Modifying molecular mechanics to understand drawn diagram cues fixed this problem and enabled us to expand the domain of molecule diagrams we could handle by switching from the plastic-like templates to a force field driven model for conformation generation. This last algorithm's performance was improved by treating conformation generation as a search task and applying the Limited Discrepancy Search algorithm. By determining the primary path of the search based on cues present in the diagram, we exploit the chemist's knowledge of molecule structure akin to a human-guided search technique.

While our technique for a force field driven conformation generation is useful as an interactive technique for molecules with less than 80 atoms, there is still room for improvement in the algorithm. The algorithm does not guarantee it finds a global minimum solution. Indeed, as the size of the molecules grow, finding a global minimum becomes akin to computational chemistry's tertiary structure prediction (protein folding) problem [21] which is a task for farms of computers, not a single Tablet PC. Advances into that area of research may prove useful to future attempts of generating conformations for molecule diagrams on a much larger scale.

[24] An alternative technique for simplifying conformation generation based on the same principle is the "extended-atom representation" used in CHARMM during conformation optimization [13]. In this system, atoms are usually considered to contain their neighboring hydrogens.

Similarly, there is currently nothing in the algorithm to handle functional groups. Functional groups are sets of atoms which behave in chemically distinctive ways. As such, they (and other common structures) could have “macros” for adding them into a molecule diagram. Since our algorithm understands only individual atoms, there is no way to add a group of atoms as a predefined shape. The user must define the structure of the group explicitly. Redefining the algorithms to handle the addition of groups would additionally allow us to rethink the prioritizing of atoms. The algorithms presented here only consider adding atoms to the conformation which are directly adjacent to the current conformation. Instead, we could consider forming any bond in the conformation at each step thus potentially resulting in multiple subconformations which are then combined together. Potentially, we could improve our results by ordering the bonds in the same order in which real atoms would combine together to form the final molecule.

We had expected, initially, that temporal information from the act of drawing a structure would be a useful part of the prioritization system. One would expect that the order atoms were drawn in a molecule diagram would indicate their relative importance in determining the molecule’s structure. In practice, we found that this was not strictly true as the users’ streams of consciousness were not dictated by structure. As such, temporal information is not part of our final prioritization system. Nonetheless, the order atoms are drawn is not random and could provide useful information for reordering atom priorities.

During the course of conformation generation with IM3 terms, it may become apparent that a strict enforcement of a particular term is problematic. In particular, if the user drew one of the perspective cues poorly or without thinking of the underlying structure, it may not actually be possible to satisfy the cue. In such a case, our algorithm simply returns an empty result set to the user. Discovering these problems during conformation generation and providing more useful feedback to the user about the nature of the problem would allow the user to either redraw the portion of the diagram properly or change the enforcement policy for the term.

Finally, one would expect that our algorithms would perform even better using a system that considers the constraints of all the atoms simultaneously, rather than adding atoms one at a time. Crippen and Havel [24, 25] used Distance Geometry, which does consider all locations simultaneously, to determine conformations for cyclohexane and other small molecules in the late 70’s. We therefore attempted applying a distance geometry computational package to find conformations for molecule diagrams. In practice, we found this algorithm to be too slow for an interactive technique and it produced conformations of higher energy than those produced by our basic search algorithm. Similarly, Wang’s branch and bound algorithm [88] would be interesting to explore for this task, but since an implementation was not readily available, it was outside of the scope of this dissertation. Although we eventually focused on using our understanding of chemistry to create quick, sufficient conformations, conformation generation from such a non-linear approach remains appealing.

Chapter 4

Ink-Modified Molecular Mechanics

In Chapter 3, we noted that a key problem with building a molecule model which matches a given (parsed) diagram is the task of simultaneously making the model chemically feasible while satisfying the constraints of the drawing. However, half of that task, namely satisfying chemical feasibility, alone has received a good deal of attention and research. Specifically, molecular mechanics methods provide mathematical equations, or force fields, for the energy of a given molecule model. Of two models for the same molecule, the one with the lower energy is the more feasible one. As a model's energy is defined entirely by the coordinates of its atoms, finding a chemically feasible model consists of selecting a 3D coordinate for each atom in the model such that the model's energy is minimized. From an optimization standpoint, the task is to find near-global minima in the $3 * a$ dimensional energy surface where a is the number of atoms in the molecule. In the last chapter we described techniques specific to molecule models which facilitate finding these minima. In this chapter, we describe a modification to the standard force field's energy surface which removes minima in locations which do not match the drawing. The remaining minima, as found by conformation generation techniques yields, are both chemically feasible (because they're at minima in the energy surface) and satisfy the constraints of the drawing (since minima which do not satisfy these constraints have been removed from the energy surface.)

$$Energy = \sum_{\text{Bonds}} + \sum_{\text{Angles}} + \sum_{\text{Torsional angles}} + \sum_{\text{Steric}}$$

Figure 4.1: *The basic form of a molecular mechanics force field. Four terms (bond length, bond angles, torsional angles, and steric repulsion) contribute energy when a molecule model violates their ideal state. Models which have a low energy are considered more chemically feasible than higher energy ones. Depending on the type of molecules to be analyzed by the force field, additional terms (such as electrostatic and hydrogen bonding) can be added as well.*

$$Energy = \sum_{\text{Bonds}} + \sum_{\text{Angles}} + \sum_{\text{Torsional angles}} + \sum_{\text{Steric}} + \sum_{\text{Drawing Cue 1}} + \ldots \sum_{\text{Drawing Cue } n}$$

Figure 4.2: *The basic form of an ink-modified molecular mechanics force field. For each of n drawing cues, we create a non-negative term which models adherence to the drawing constraint as energy. These terms are appended to the regular force field to create a composite function which is low in energy only when the conformation adheres to both the chemical structure properties in the regular force field and the defined drawing constraints.*

4.0.1 Existing Force Fields

Figure 4.1 shows the generic form of a typical force field such as is typically used today in computational chemistry. This form is a sum of terms which each penalize specific unlikely relationships. For instance, an unlikely bond length is quantified in the bond term. For each unlikely relationship, the total energy is increased. Ideally, a molecule will have a shape where all of the terms will be feasible and the sum energy will be close to zero. Practically, this is not always the case as terms can be in conflict in larger molecules. By seeking the global minimum energy, one minimizes the total unlikelihood of all such relationships in the molecule.

As an understanding of molecular mechanics is critical to understanding this dissertation, but molecular mechanics itself is not the focus of the dissertation, greater detail on the topic and how to calculate its terms are provided in Appendix A.

As previously mentioned, using molecular mechanics alone as a basis for the diagram interpretation problem does not necessarily produce the desired results. Since molecular mechanics is based strictly on 3D coordinates and is oblivious to the way a molecule is drawn in 2D, generating a model at a minimum in the force field is potentially at the expense of the 3D detail specifically expressed in the drawing. Indeed, all 3D detail from the diagram is completely ignored in this optimization problem. In the worst case, a fundamentally different molecule from what was drawn can be produced. This was the case in the Carvone example of Figure 3.10 in the last chapter where there are two equally optimal energy minima in the force field, but the energy between the two states is very high. The high energy between the minima prevents the molecule from moving from one minimum to the other without breaking a bond. This makes the molecules in the two different states chemically different. While a drawing indicates one of these minima as being the one the chemist desires, unmodified molecular mechanics sees the two as equal.

4.1 Modifying a Force Field

To account for cues in a diagram that indicate 3D relationships, we propose adding energy to the force field output wherever the conformation does not adhere to the drawing cues. In this way, we can remove these “false” minima from the energy surface. For each drawing cue we want our system to understand, we formulate the cue's requirements as an energy term, that is to say, a function which, given a molecule conformation, returns a real number indicating the energy for that conformation. We then add the energy term to the force field, thereby making all conformations which do not adhere to the cue have a higher energy than they did in the unmodified force field. We refer to including these additional drawing-based terms into a force field as Ink-Modified Molecular Mechanics (IM3). Figure 4.2 shows the generic form of such a force field where the standard force field term sums are followed by term sums for each drawing cue. This augmented force field will make it possible for an conformation generation algorithm to (i) reject conformations which are chemically incorrect and (ii) accept conformations which are chemically suboptimal, but which are the intention of the user and explicitly marked as such in the drawing.

These additional drawing cue-based terms must adhere to a set of constraints which ensure that the techniques from Chapter 3, as well as existing gradient-based algorithms for conformation generation and optimization, can be applied to the resulting force field. First, the energy function of the entire force field and its gradient must remain continuous. Normal force fields are continuous functions with continuous gradients. Gradient-based optimization techniques, such as we use in our work, require this continuous gradient as the gradient is used to direct how the algorithm should change the conformation to minimize the energy. If an IM3 term were to simply add a flat energy penalty to all points which violate the cue, the resulting force field would not be continuous. Therefore, we design the terms to have zero energy penalty and zero gradient at the boundaries between the region where the cue is adhered to and where it is not. As the conformation moves further into the "wrong" region, the energy increases. Mathematically, since the continuity of functions and their derivatives is maintained under summation, we know that the overall force field and gradient will be continuous if each term is continuous with continuous gradients.

Second, the gradient of the augmentation terms must have a known analytical solution. This allows gradient-based optimization algorithms to be performed quickly[1]. While numerical methods for calculating a gradient will still produce approximately the same output, the performance drops significantly. The direct, non-sampling numerical method for measuring the gradient requires the force field to be evaluated three times for each atom. As our algorithm for conformation generation typically calculates the gradient hundreds of times over the course of execution, this additional order of magnitude becomes very costly. Therefore the analytical solution is needed to make conformation generation feasible for use in an interactive system. While analytically taking partial derivatives of vectors is generally a straightforward (and time consuming) task, finding usable analytical gradients of force field terms is not always so. Blondel and Karplus [11] noted this when they found a derivative free of singularities for the standard torsion term 25 years after the original, flawed formulation was used by Warshel and Lifson [91]. Therefore, we keep this constraint in mind when picking our formulations. Overall, this constraint acts as a guideline that while some energy formulations may

[1] The derivatives for the example terms in this chapter are given in Appendix B.

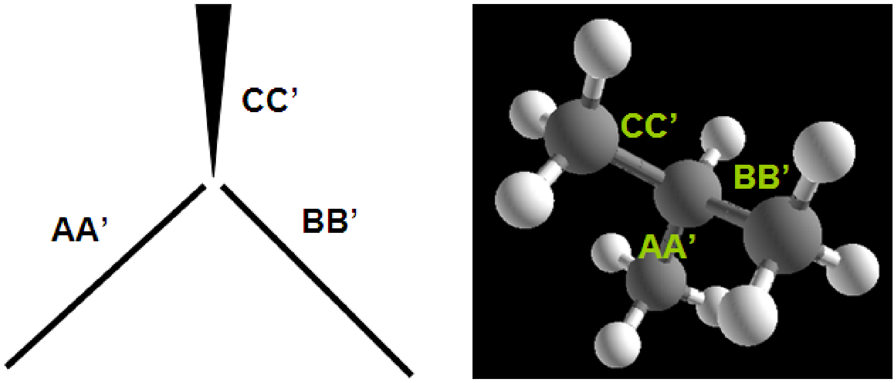


Figure 4.3: *The three bonds defined in the R/S stereochemistry term. We use the notation $\overrightarrow{AA'}$ to indicate the bond between the atom at location A and the atom at location A'. $\overrightarrow{CC'}$ is the bond drawn with a wedge or dashed bond. $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$ are the closest two straight-line bonds to C and define a plane which divides 3D space into two halves. One half is correct places where C' can be placed, the other is incorrect places whereC' can be placed.*

be obvious for a drawing cue, non-obvious formulations which leverage known derivatives are more useful.

Third, a term should behave like a standard force field energy term and only penalize the energy when the drawing cue is being violated. This means the term should be non-negative, have a value of zero wherever the drawing cue is adhered to, and have an increasingly positive value the more the cue is violated. This requirement is different from the other two in that it is not a required property for conformation generation algorithms to be able to use the force field. Instead this requirement gives the force field the desired property of evaluating to the real energy whenever a conformation violates no drawing cues.

Given these requirements, we next present four IM3 terms to serve as examples of how to formulate drawing cues as energy penalties. Each term handles one of the most common drawing cues used in an organic chemistry course. This set is by no means exhaustive of all drawing cues in chemistry, even those used in an introductory organic chemistry course. This set does serve to illustrate that the general technique of Ink-Modified Molecular Mechanics is sound for handling different kinds of drawing cues, including ones indicated by explicit symbols and marks as well as more perceptually based cues where the relative angles of bonds indicate the 3D relationship. Moreover, we intend this set to provide templates for the future development of additional terms to handle other drawing cues.

$$\overrightarrow{CROSS} = (\frac{\overrightarrow{AA'}}{\|AA'\|} \times \frac{\overrightarrow{BB'}}{\|BB'\|})$$

$$Energy_{RS} = \begin{cases} 0 & (\overrightarrow{Cross} \cdot \overrightarrow{CC'})\text{'s sign matches the bond} \\ \Sigma_{\text{W,D Bonds}} K_{RS} * (\overrightarrow{CROSS} \cdot \frac{\overrightarrow{CC'}}{\|CC'\|})^2 & \text{otherwise} \end{cases}$$

Figure 4.4: *The R/S stereochemistry term defined over the planar bonds $\overrightarrow{AA'}$,$\overrightarrow{BB'}$, and wedge or dashed bond $\overrightarrow{CC'}$. $\overrightarrow{CROSS}$ is the normal to the dividing plane (which contains $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$). K_{RS} is a user-defined constant determining the strength of the adherence requirement. The dot product of $\overrightarrow{CC'}$ with the normal of the dividing plane grows in magnitude the more the location of C' violates the drawing cue.*

Condition	Angle Checked	Direction Indicating Up
$A = B$	$\angle A'AB'$	Right
$A = B'$	$\angle A'AB$	Left
$A' = B$	$\angle AA'B'$	Left
$A' = B'$	$\angle AA'B$	Right

Table 4.1: *Determining if $\overrightarrow{Cross}$ points up or down. The bonds $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$ are adjacent, so they have one point in common. Once this point in common is known, the turn direction of the listed angle's 2D diagram points will indicate $\overrightarrow{Cross}$'s pointing direction.*

4.2 The R/S Stereochemistry Term: Wedge and Dashed Bonds

The first IM3 term prevents errors of the type in the Carvone example from the last chapter (Section 3.3.2). In a molecule diagram, a triangular wedge or a series of dashed parallel lines is used to indicate a bond with a specific 3D perspective. The wedge bond is coming up towards the viewer while the dashed bond is going away[2]. Looking at the diagrams for Carvone again (Figure 3.10), the details in the diagrams that distinguish spearmint from caraway are the sides upon which the oxygen atom resides and the wedge and dashed bonds at the bottom of the diagram rings. The carbon at the bottom of the ring in each diagram assumes a tetrahedral shape and has four different neighbors. These neighbors are different due to the location of the oxygen. As there are two different ways to connect four neighbors to a tetrahedron (invariant to rotation), this atom's 3D structure defines two chemically different molecules. Chemists call this type of atom a stereocenter. As they are in the Carvone diagrams, wedge and dashed bonds are often used to distinguish such R and S stereocenters because the wedge or dash indicates a specific 3D arrangement of the bond. Therefore, we call the term to handle wedge and dashed bonds the R/S stereochemistry term.

[2] Whereas the wide end of the wedge indicates which end of the bond is coming up from the local plane of the drawing, the dashed bond is inherently ambiguous [46]. ChemPad makes a best guess as to which end of a dashed bond is which by checking each end for enough normal bonds to define a plane. If a single bond cannot be determined, the diagram is rejected as ambiguous.

In calculating the entire term, we take the sum of individual R/S stereochemistry term instances in the set of all wedge and dashed bonds in the diagram. For each of these non-normal bonds, we create a plane which separates our perspective's "front" from "back". This allows us to distinguish if the bond is adhering to the drawn perspective in 3D. However, we cannot simply define this plane as being derived from some global constant viewing direction such as looking down the positive y-axis and using the plane $z = 0$. While this could work for some very small molecules, in the case of larger molecules, the chemists viewing direction can be different for different regions of the diagram. This is because molecules assume fully 3D shapes do not necessarily stay close to a single "viewing" plane. Therefore in creating a molecule diagram, chemists create perspective cues relative to local features. So, we must use only local features to define the separating plane. In particular, normal (not wedge or dashed) bonds in the same vicinity of the perspective bond should be not in perspective from the chemist's local view. Given two normal bonds adjacent to the perspective bond, we can define the dividing plane as the plane going through the three distinct points of the two normal bonds.

Figure 4.3 shows the formulation of the R/S stereochemistry term which is defined with respect to three bonds. We use the capital letters A, A', B, B', C, and C' to refer to the 3D locations of the involved atoms. $\overrightarrow{CC'}$ is the bond in perspective while $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$ are the normal bonds forming the plane from which $\overrightarrow{CC'}$ is in perspective. While we are defining this term over three bonds, and therefore six atoms, there are only actually four real atoms. We give this formulation over six atoms to be generic with respect to the connectivity of the three bonds. In the example of Figure 4.3, A, B, and C are locations for the same atom ($A = B = C$). However, this particular overlap will only be the case when there are two normal bonds directly adjacent at C. If there is only one normal bond at C, but it has a normal neighbor, we can still use these two bonds to define the plane. In this case, we could have $A = C$ and $A' = B$. Once again, there are only four actual atoms involved, but it is no longer the case that $A = B = C$. To satisfy this type of alternative, and others that may later be developed, the formulation is made over the six atoms of the three bonds with no constraint as to which atoms are equivalent.

Given $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$, the topologically closest normal bonds to atom C, we form the dividing plane A^*OB^*. The star notation here indicates that A^* is either A or A'. Similarly, O is a placeholder for the atom in both $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$. To make the term able to increase the energy penalty the more the term is violated, we want to be able to calculate how far C' is on the wrong side of the dividing plane. We calculate $\overrightarrow{Cross}$, the "up" pointing normal to the plane, from the cross product of $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$. The magnitude of the dot product $\overrightarrow{Cross} \cdot \overrightarrow{CC'}$ gives us a measure of how far C' has moved to one side of the plane. This dot product forms the core of the R/S stereochemistry term's energy penalty which is shown in Figure 4.4. If C' is on the dividing plane, both the dot product and the full term value are zero. The dot product is squared to make the gradient zero at the boundary plane, and then multiplied by a user-defined constant K_{RS} which determines the importance of the term[3]. The larger the constant used, the more local minima on the wrong side of the dividing plane will be discouraged. We will use these K_* constants in each of the IM3 terms we present as a means for setting the user's importance of different terms.

Of course, this formulation only adds the energy penalty when C' is on the wrong side of the plane A^*OB^*. Before we do this, we need to determine if C' is indeed on the incorrect side. Given that $\overrightarrow{Cross}$ is indeed pointing in the direction towards the viewer, the sign of the dot product indicates on which side of A^*OB^* we can find C'. If the dot product is positive, $\overrightarrow{CC'}$ is pointing up (as a wedge bond should) and if the dot product is negative, $\overrightarrow{CC'}$ is pointing down (as a dashed bond should). Unfortunately, the direction of $\overrightarrow{Cross}$ depends on which of the two normal bonds we chose to be $\overrightarrow{AA'}$ and which was chosen to be $\overrightarrow{BB'}$. This decision was made arbitrarily. Moreover, which end of those bonds is which was also an arbitrary decision which affects the direction of $\overrightarrow{Cross}$.

We therefore need to look at the 2D diagram again to decide if $\overrightarrow{Cross}$ is pointing up and invert it if it is not. By looking at the turn direction in 2D of $\angle A^*OB^*$, we can know which way $\overrightarrow{Cross}$ points in 3D. For the purposes of this chapter, we are defining here the concept of "turn direction" for an arbitrary angle $\angle IJK$ in 2D as whether one turns left (clockwise) or right (counter-clockwise) moving from I to J to K. In 3D this same relationship can also be measured given a view vector[4]. Using turn directions allows us to determine if $\overrightarrow{Cross}$ is pointing up because we know the direction that is "up" from the 2D page, mainly that it is coming up out of the page. For instance, consider a 2D diagram on the $z = 0$ plane with $A = (-1, 0, 0)$ and $A' = B = (0, 0, 0)$. The z coordinate of the cross product $\overrightarrow{AA'} \times \overrightarrow{BB'} = 1 * B'^{Y}$. Therefore, if B' has a positive y coordinate (a left turn for $\angle A^*OB^*$), $\overrightarrow{Cross}$ points in the positive z direction which is the "up" we desire. This checking of turn direction is generally applicable and Table 4.1 enumerates the combinations for selection of A^*,O, and B^* and which angle turn direction to check to determine if $\overrightarrow{Cross}$ needs to be inverted to make it point "up".

Revisiting the formula, it may seem strange to a chemist that this term is allowed to have an energy of zero when C' is coplanar with $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$. After all, C' should be in perspective from that plane, not on it. However, such an end result conformation does not occur in practice because of the interactions with the rest of the regular force field terms. For instance, in sp^3 hybridized atoms, which is where we typically find wedge and dash bonds, the angle strain term is at its maximum in the coplanar conformation. The R/S stereochemistry term need only discourage the conformation generation algorithms from exploring one side of that maximum. The gradient-based techniques used by the algorithms will then find the actual energy minimum on the correct side of the plane.

[3] We set $K_{RS} = 100$ by default.

[4] Algorithms for determining turn direction is covered in Appendix D.3.

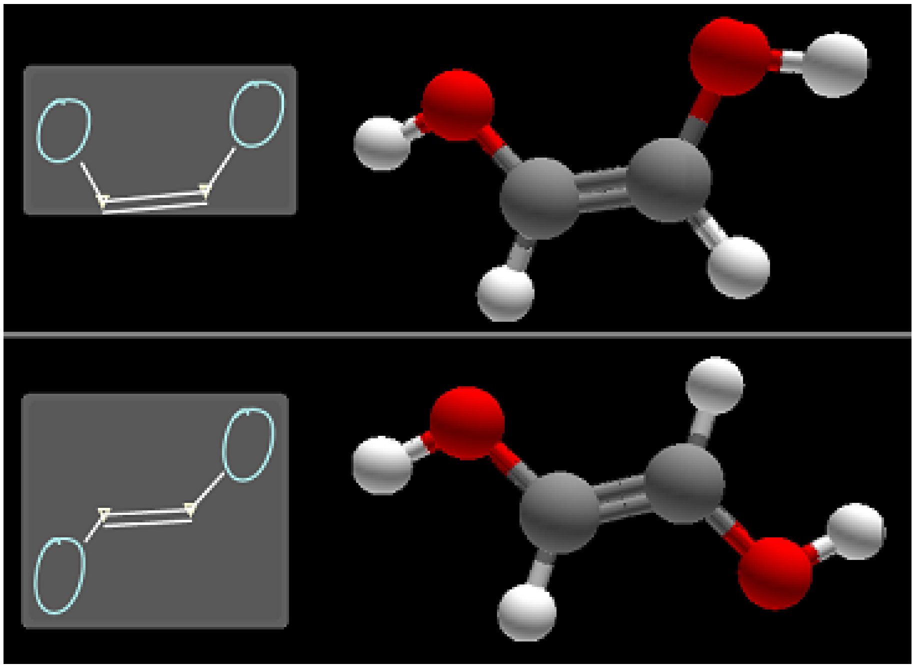


Figure 4.5: *Z and E configurations of alkenes. Double bonds do not freely rotate, so the two molecules shown here are chemically different. The Z/E stereochemistry term in these models is defined over the grey carbon and red oxygen atoms.*

4.3 The Z/E Stereochemistry Term: Rigid Double Bonds

The second IM3 term prevents errors which arise in building models containing double bonds. The atoms involved in a double bond share more electrons than atoms involved in a single bond. Whereas atoms can typically rotate about single bonds, these extra shared electrons prevent such free rotation in the double bond. Therefore, the molecules shown in Figure 4.5, which differ by rotation around a double bond, are chemically different. We can distinguish between these molecules by looking at the atoms which neighbor each of the atoms in the double bond. In the figure, there are oxygen and hydrogen atoms adjacent to each of the carbons in the bond. By CIP ordering[5], a system chemists use to rank the priority of atoms, these oxygen atoms are given a higher priority than the hydrogen atoms. We refer to the top molecule where the higher priority oxygens are both on the same side of a line through the double bond as the Z stereoisomer. The other form, where the high priority oxygens are on different sides of a line through the double bond, as is the case in the Figure's bottom molecule, is the E stereoisomer. Thus, we name the term for handling this distinction the Z/E stereochemistry term.

We could expand this 2D concept of a line through the double bond into a plane in 3D which divides the correct locations from incorrect locations. This would be akin to the dividing plane we

[5] A CIP-determining algorithm appears in Appendix D.1.

$$\overrightarrow{CROSSA} = (\frac{\overrightarrow{AB}}{\|AB\|} \times \frac{\overrightarrow{AA'}}{\|AA'\|})$$

$$\overrightarrow{CROSSB} = (\frac{\overrightarrow{BA}}{\|BA\|} \times \frac{\overrightarrow{BB'}}{\|BB'\|})$$

$$Energy_{ZE} = \begin{cases} 0 & \text{the 2D and 3D turn directions match} \\ \Sigma_{\pi \text{ - bonds}} K_{ZE} * (\overrightarrow{CROSSA} \cdot \overrightarrow{CROSSB})^2 & \text{otherwise} \end{cases}$$

Figure 4.6: *Formulation of the Z/E stereochemistry term. The term is defined for each double (π) bonded pair of atoms A and B and their respective neighbors A' and B'.*

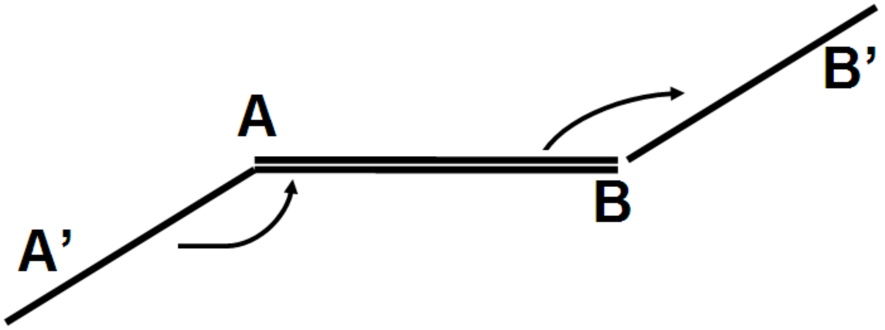


Figure 4.7: *The Z/E stereochemistry term is defined over the drawn double bond $\overrightarrow{AB}$ and its adjacent bonds $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$. The turn directions in 2D of $A'AB$ and ABB' are noted as being the same (left-left,right-right) or different (left-right,right-left). This is then checked against the turn directions in 3D being the same or different as the 2D. This is easily calculated in the formulation by $\overrightarrow{CROSSA} \cdot \overrightarrow{CROSSB}$.*

used for the R/S stereochemistry term. However, as we have only two atoms on the plane available, this formulation is not readily apparent. Instead, we use turn direction as the foundation for the Z/E term. As shown in Figure 4.7, we define the double bond to be $\overrightarrow{AB}$ and the bonds to the high priority atoms on each side of the bond as $\overrightarrow{AA'}$ and $\overrightarrow{BB'}$. Then, in the 2D diagram, either the turn directions of the angles $\angle A'AB$ and $\angle ABB'$ are the same (left-left or right-right) which indicates the E stereoisomer, or different (left-right or right-left) which indicates the Z stereoisomer. Given this knowledge of what the turn directions are in 2D, we then need to find the turn directions in 3D for comparison.

Instead of using a standard algorithm for determining turn direction in 3D, we can perform a simpler check for this term since all we are checking is if the consecutive turn directions are the same or different, not the specific direction of each term. Assuming for a moment that A, A', B, and B' are coplanar, which is the standard force field's typical energy minimum for these cases, the cross products $\overrightarrow{A'A} \times \overrightarrow{AB}$ and $\overrightarrow{AB} \times \overrightarrow{BB'}$ indicate relative turn directions in 3D. If both cross products, which we'll call $\overrightarrow{CROSSA}$ and $\overrightarrow{CROSSB}$ respectively, are in the same direction, then $\angle A'AB$ and $\angle ABB'$ both turned in the same direction. If $\overrightarrow{CROSSA}$ and $\overrightarrow{CROSSB}$ are in opposite directions, then $\angle A'AB$ and $\angle ABB'$ turned in different directions. We can check whether or cross product vectors are in the same direction by looking at the dot product $\overrightarrow{CROSSA} \cdot \overrightarrow{CROSSB}$. This dot product will be negative if and only if the turns are opposing in 3D.

This brings us to the formulation of the Z/E stereochemistry term given in Figure 4.6. We use the sign of $\overrightarrow{CROSSA} \cdot \overrightarrow{CROSSB}$ to determine if the 3D is matching the drawing cue. If it is not, this dot product's value is again useful for determining the magnitude of the penalty. Intuitively, $\overrightarrow{CROSSA}$ and $\overrightarrow{CROSSB}$ are normals to the planes formed by the atoms in the double bond and the points A' and B' respectively. Since $\overrightarrow{CROSSA}$ and $\overrightarrow{CROSSB}$ are normalized, their dot product is the cosine of the angle between the vectors. When a given atom such as B' is on the plane dividing the locations where B' is supposed to be from the locations where B' is not supposed to be, the normals are perpendicular and their dot product is zero. As B' moves to one side or the other of the plane, the dot product increases until A, A', B, and B' are co planar and the dot product is one. As we are only counting energy on the side where B' is not supposed to be, the energy is at a maximum when B' is coplanar with the other three points and B' is on the wrong side of the dividing plane.

We previously mentioned that the minimum energy in the standard force field for cases such as these occurs when A, A', B, and B' are coplanar. This term does not fix these points to the plane though. If other forces in the molecule push B' off the plane, there will be no additional energy from the term until the turn direction of $\angle ABB'$ changes. Instead the standard force field terms alone are responsible for finding the energy minimum which may be somewhat non-planar.

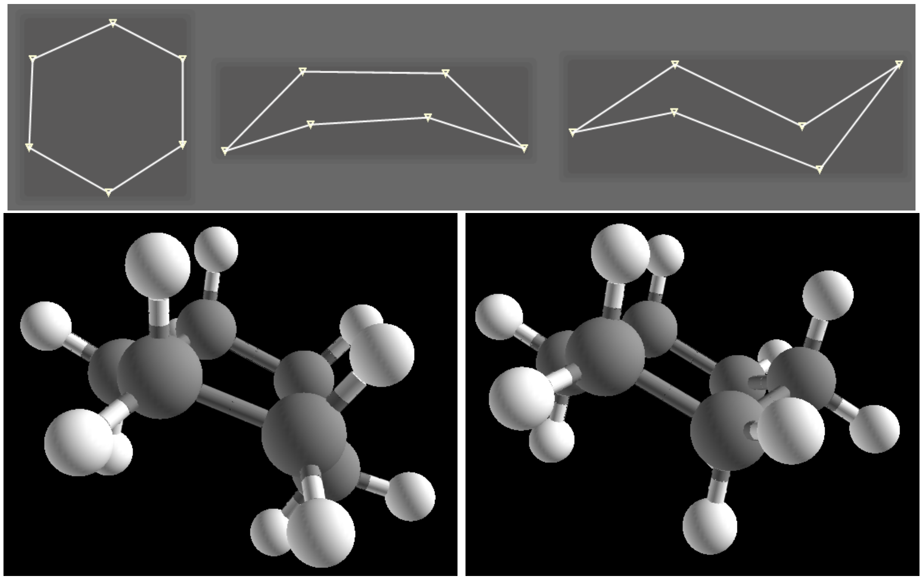


Figure 4.8: *The cyclohexane molecule is commonly found in two shapes: the boat (lower left) and chair (lower right). When drawing a diagram of cyclohexane, the chemist may not desire a specific structure and make a generic "top-down" diagram as in the upper left. If the chemist does wish to indicate a specific conformation, a perspective drawing of the bonds in the ring is made. This is shown in the upper center and upper right where the desired central structure is drawn in perspective. This type of perspective for rings is handled by the Planar Perspective term.*

4.4 The Planar Perspective Term: Rings Drawn From the Side

The third IM3 term distinguishes not between chemically different molecules, but between the shapes a given molecule can take. While many molecule diagrams are made from a "top down" perspective, it can be useful to make some diagrams, or parts of diagrams, from a "sideways" perspective to show a specific 3D shape. Figure 4.8 shows three drawings of cyclohexane and two 3D models. The two models are not chemically different. Indeed, cyclohexane molecules switch between the two shapes more than 100,000 times per second [70]. However, the "sideways" drawings of cyclohexane do specifically indicate one of the models, while the "top down" view remains ambiguous. Furthermore, the boat conformation on the lower left has a noticeably higher energy than the chair conformation on the lower right. While the previous terms most often distinguish between models of similar energy, this one will more often need to overcome the global energy minimum which does not match the diagram in favor of a non-global, local minimum which does match the diagram. We refer to this term as the Planar Perspective term because the diagrams which invoke it contain a ring drawn in such a way as to evoke a sense of perspective relative to a plane perpendicular through the ring.

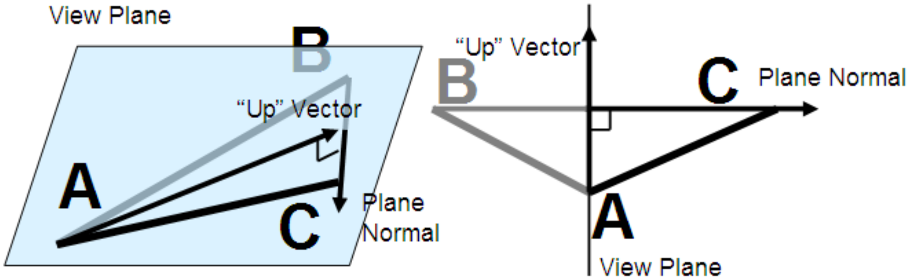


Figure 4.9: *A tightly drawn angle between ring bonds, as drawn on the left, indicates that the ring is in perspective. We set the view plane as being perpendicular to $\overrightarrow{BC}$ going through the point A. The "up" vector, which provides a reference direction for all angles that we form in 2D and 3D, is starts at A and ends at the midpoint between B and C ($\overrightarrow{A\frac{B+C}{2}}$).*

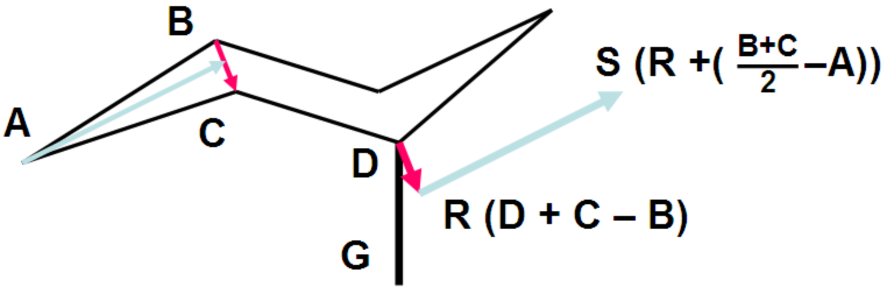


Figure 4.10: *The planar perspective term is over the bond $\overrightarrow{GD}$ in the ring with the tight angle $\angle BAC$. To compare the orientation of $\overrightarrow{GD}$ with its 2D diagram counterpart, we calculate the torsional angle $\angle^{(3D)}GDRS$ where $R = D+$ the view plane normal, and $S = R+$ the "up" vector. The 2D angle calculation can be taken more directly as the angle $\angle^{(2D)}GD(D + \overrightarrow{Up})$*

$$\theta = |\angle^{(3D)}GDRS - \angle^{(2D)}GD(D + \overrightarrow{Up})|$$

$$\Delta_\theta = \frac{\pi(\theta - \theta_{Eq})}{2(\pi - \theta_{Eq})}$$

$$Energy_{PP} = \begin{cases} 0 & \theta \le \theta_{Eq} \\ \sum_{\text{Perspective Ring Atoms}} K_{PP} * (1 - \cos^2(\Delta_\theta)) & \text{otherwise} \end{cases}$$

Figure 4.11: *The planar perspective term formulation. For a torsion $\angle^{(3D)}GDRS$ we find the 2D to 3D difference θ which is checked against the user-defined threshold θ_{Eq}. If $\theta > \theta_{Eq}$ we apply the penalty. This term is formulated in this fashion so that we can use the standard molecular mechanics torsion derivative when calculating the force field gradient. See Appendix B for the gradient.*

Before attempting to measure if the 3D structure matches this perspective, we must first be able to tell the difference between rings that have perspective and those which do not. The key to detecting a "sideways" portion of the drawing is to look for tight angles formed between bonds in a ring. In a "top down" diagram, the tightest angle one would expect to see would be about 60° in the cyclopropane molecule. Conversely, a "sideways diagram" would want to put the two bonds very close to each other – about 15° to 30° to indicate that in perspective, these bond lines should overlap. The "sideways" drawing in Figure 4.8 have these tight angles at the left and right sides of the diagrams. As these bonds are supposed to be overlapping in 3D, these vertices with the tight angle between drawn bonds can be used to define a view plane for the ring. This plane passes through the vertex atom and has a normal through the two connected atoms. This relationship is shown in Figure 4.9.

Ideally, this perspective plane should match the perspective part of the diagram. That is to say that if we project the 3D points in and adjacent to the ring onto this plane, this projection would look much like the drawn diagram. However, as drawn bond lengths may be distorted, which would defeat image-based attempts to match the 3D to the 2D, we rely upon only comparing angles in the projection to the angles in the diagram. For example, with the cyclohexane shown in Figure 4.10, the drawing contains the axial point G which is to say that $\overrightarrow{DG}$, G's bond to the ring, points down in the diagram. This axial positioning forms the roughly 90° angle $\angle GDC$. If G had instead been drawn as equatorial, which is to say $\overrightarrow{DG}$ would point rightish in the diagram, the angle would be roughly 145°. If we discover that the 3D angle $\angle GDC$ when projected onto the perspective plane form an angle close to 145°, we know the 3D is not matching the axial drawing cues.

In practice, this comparing of angles formed with neighbors is not sufficient to ensure that all angles match. Without a sense of which direction is "up" *in* the plane, the example 90° angle $\angle GDC$ could be satisfied by G being either up or down in the plane[6]. Therefore, we want to select a single vector in both 2D and 3D, which is in the respective perspective plane, to call the "up" direction. This vector can then form the reference vector for comparing angles rather than using neighbor atoms. In particular, we base the "up" vector on the points in the angle $\angle ABC$ at the tight vertex. We know that in perspective, B and C are supposed to overlap and that the indicated distance between them is provided for clarity in the drawing. Taking the average of B and C, we get the point where they both would be if the drawing been directly "sideways". We use the vector from A to this midpoint as the reference "up" direction which is shown by the blue line in Figure 4.10. For the rest of the ring, we know the intended perspective is being adhered to when the difference in the angles formed by the "up" vector with a bond originating at the ring is below θ_{Eq}, a user-defined threshold[7].

[6] If the example in Figure 4.10 were instead cyclopentane by not containing the atom furthest from A, this "up/down" problem would manifest itself.

[7] By default, we set $\theta_{Eq} = 45°$

We can simplify the expression and calculation of this angle comparison by putting it into a format already used in molecular mechanics systems. By redefining “the angle formed in the projection on the plane” as “the torsional angle about the plane normal”, we have a calculation in terms of torsional angles which is already defined in regular force fields[8]. This redefined calculation is therefore simple to compute with existing force field tools. To create the torsion $\angle GDRS$ shown in Figure 4.10, we add the plane normal to D to make the point R. Using the plane normal makes the torsion formulation match the projection formulation. Then to create the final point in the torsion, we add the “up” vector to R to make the point S.

As users do not exactly draw the perspective angles in their diagrams, the term needs to give the user a little leeway in the angles they draw. We define θ_{Eq} as this user-defined constant which is the maximum amount of angle difference allowed before the energy penalty begins to apply. If the absolute difference θ between the 3D $\angle GDRS$ and the 2D $\angle GD(D + \overrightarrow{Up})$ is greater than θ_{Eq}, the term is violated and the penalty defined in Figure 4.11 is applied. As we would like the energy to be zero at $\theta = \theta_{Eq}$ and one at $\theta = \pi$, the maximum angle, this penalty uses Δ_θ as an “adjusted” angle taking the scale of the leeway θ_{Eq} into account. $1 - \cos^2(\Delta_\theta)$ then ranges from zero to one as the difference between the angles under perspective increases beyond the threshold.

The amount of leeway given can cause problems with diagrams that were drawn poorly, as can easily be the case with those drawn by students. Using the example from Figure 4.10 again, the point G could have been drawn as being somewhere between axial and equatorial. If the leeway is large enough to incorporate both of these likely 3D positions, there will be no energy penalty for either of them. This is perhaps acceptable for a “garbage in, garbage out” case where the student drew G as being in the middle of these two states. It is less acceptable when G was drawn in such a way to indicate to a viewer that it is axial, but the drawing is sloppy enough to make it fall into the range of having no energy penalty when equatorial. For this reason, a user would want to set θ_{Eq} to a small value. Alternatively, setting a small θ_{Eq} could cause neither axial nor equatorial to fall into the range of acceptable angles formed. As opposed to the stereochemistry terms, this term is finicky in the exactness it demands from the diagram and adjusting θ_{Eq} is not enough to compensate for imprecise diagrams.

4.5 The Dihedral Perspective Term: Specific Dihedral Angles

Like the planar perspective term, the fourth IM3 term does not distinguish between chemically different molecules, but between conformations taken by a given molecule. Unlike all the previous terms, this one can be in true contradiction with the underlying force field since it may interpret

[8] The calculation of torsional angles, particularly with respect to molecules, is reviewed in Appendix D.2. Perhaps, more importantly, this calculation has known derivatives.

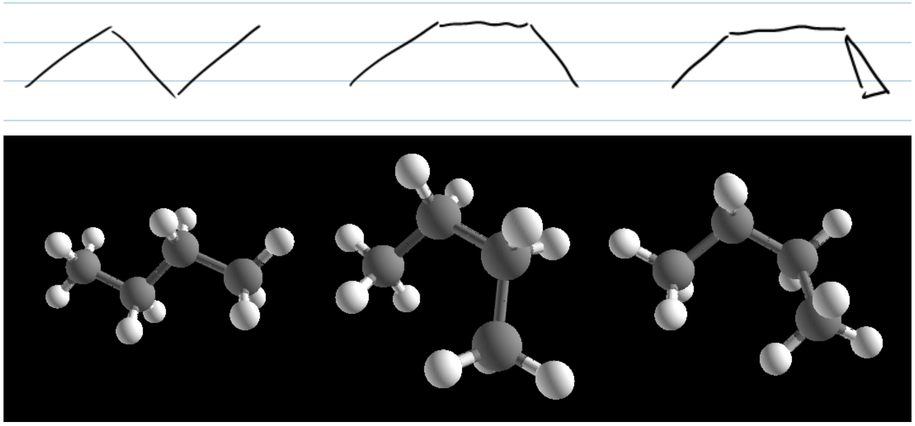


Figure 4.12: *Ways of drawing an eclipsed or gauche torsion. The drawing on the left shows butane as anti-periplanar with the outside carbons as far apart as possible which is the energy minimum for the torsion. The center drawing shows butane as being syn-periplanar with the outside carbons as close together as possible which is the energy maximum. The drawing on the right shows butane with the outside carbons close, but not coplanar with the inside carbons. This conformation is a local energy minimum.*

the drawing as indicating the conformation is at an energy maxima, rather than a minimum. In particular, the dihedral perspective term is concerned with torsional angles formed in non-ring atom chains.

Under most conditions, a chemist would draw a chain of singly-bonded atoms in a zig-zag fashion as shown on the left in Figure 4.12. Looking at any individual torsion in the chain, one would find the dihedral angle to be roughly 180° which places the opposing chain atoms as far apart as possible. This is the lowest energy conformation as the non-bonded atoms are far from each other and there is minimal overlap of the electron shells. Alternatively, the chemist could draw one of the other torsion relationships from Figure 4.12 to show a specific conformation other than the anti-periplanar, zig-zag one. While not the global minimum, or in some cases not even a local minimum, real molecules pass through these conformations often[9] and thus chemists need to be able to indicate these conformations in diagrams. As this term concerns itself with the showing of dihedral angle approximations in perspective, we refer to it as the Dihedral Perspective Term.

This term also differs from the previous terms in that our target audience of student chemists are not necessarily expected to know the difference between the different ways to draw these cues and their relative energies. A first pass at making sure we give the expected behavior is to check the angles formed within the drawing. If the angles are mostly right angles, the user may be drawing a simplistic Lewis line structure where there are no 3D cues. Beyond this, we leave it up to the user

[9] This is subject to other constraints in the molecule. Generally speaking, single bonds rotate freely.

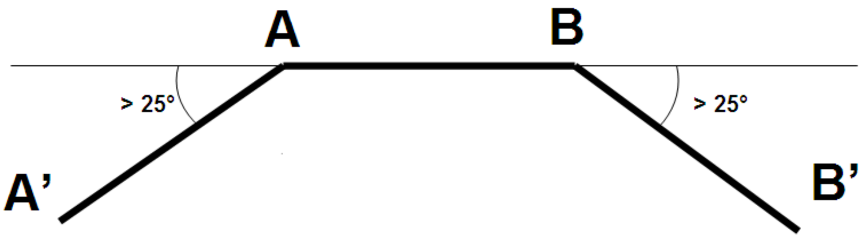


Figure 4.13: *The dihedral perspective term is over the bond $\overrightarrow{AB}$ and the neighboring bonds $\overrightarrow{A'A}$ and $\overrightarrow{BB'}$. These bonds must be on the same side of the bond $\overrightarrow{AB}$ (i.e. the turn direction $\angle A'AB$ must be the same as $\angle ABB'$) and the angles must be greater than the threshold of $25°$. If $\overrightarrow{A'A}$ and $\overrightarrow{BB'}$ are of the same bond type as shown here, the eclipsed relationship is used. If they were different, as would be the case if one were a wedge, the gauche relationship would be used.*

to properly set the enforcement policies (Section 3.4.5) for the software to use this term if the user is advanced enough.

Additionally contrasting the previous terms, this term comes in two flavors. Depending on the symbols used, the term can be indicating the energy maximum, or a non-optimal minima. These two flavors require different calculations to enforce. Truthfully, this could be considered two different terms, but they are detected by the same set of drawing cues so we present them together.

4.5.1 Eclipsed Relationship

Figure 4.13 shows the three bonds over which the Dihedral Perspective Term is defined. $\overrightarrow{AB}$ is the center bond which formes the dihedral angle in question with $\overrightarrow{A'A}$ and $\overrightarrow{BB'}$. This term is defined for two torsion relationships: eclipsed and gauche. Eclipsed relationships occur when the dihedral angle is 0° and A' and B' are as close to each other as possible. If one were to look down the torsion in 3D, the two atoms would be overlapping, or eclipsing, each other. Drawing the bonds as in the middle example of Figure 4.12 with $\overrightarrow{A'A}$ and $\overrightarrow{BB'}$ staying on the same side of the $\overrightarrow{AB}$ line indicates this relationship. Detecting the relationship in the diagram is accomplished using the tools from the Z/E stereochemistry term. We can detect the relationship by checking the 2D turn directions $\angle A'AB$ and $\angle ABB'$. If (i) both turn directions are the same (right-right or left-left), (ii) the bonds were drawn the same (normal, wedge, or dash), (iii) and the angles are above a threshold[10], we know the chemist intended an eclipsed conformation.

[10] This threshold prevents diagrams drawn without care of angles, such as leftmost drawing in Figure 1.3 from invoking the term. We use a default threshold of 25°.

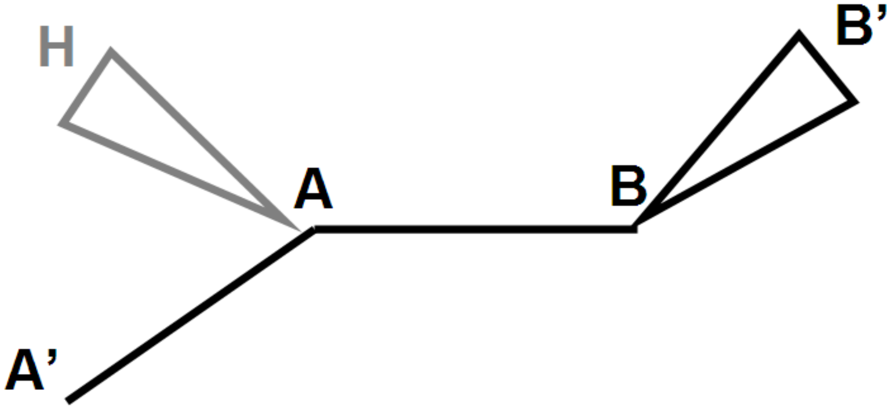


Figure 4.14: *Diagram of the eclipsed implicit hydrogen special case of the dihedral perspective term. Here the implicit hydrogen and its bond to A are drawn in gray. Even though the drawn bonds have opposite turn directions, we still need to consider the eclipsing of B' with the implicit hydrogen H.*

$$\phi = \angle^{(3D)} A'ABB'$$

$$\gamma = \begin{cases} \pi * \frac{5}{3} & \text{Implicit hydrogen case and } \phi > \pi \\ \pi * \frac{7}{3} & \text{Implicit hydrogen case and } \phi < \pi \\ \pi & \text{The regular, not implicit hydrogen case} \end{cases}$$

$$Energy_{DP} = \sum_{\text{Perspective Non-ring Bonds}} K_{DP} * (1 + \cos(\phi - \gamma))$$

Figure 4.15: *The Dihedral Perspective Term formulation for eclipsed cases. This term is identical to the standard molecular mechanics torsion term with $n = 1$. We compare the measured torsion $A'ABB'$ against the angle we expect to find. Normally, the expected angle is π, but when we're eclipsing with an implicit hydrogen, the angle is either $\pi * \frac{5}{3}$ or $\pi * \frac{7}{3}$. Since we don't know which, we measure to the closest.*

For an eclipsed relationship, we want the dihedral angle $\angle A'ABB'$ to be 0. By measuring the difference between the current torsional angle ϕ and the ideal one, we can use the same formula as the standard molecular mechanics torsion formulation. This formulation is given in Figure 4.15. Here γ is set to π instead of 0 because we would like $1 + \cos(\phi - \gamma)$ to be 0 when $\phi = 0$. As $\phi = 0$ increases, the total energy increases to twice the user-defined constant K_{DP}.

While the other terms have had large regions of the energy surface where no energy is added, using this formulation, we add energy to the force field for all but the exact energy maximum. When considering a strict enforcement policy (Section 3.4.5) for an eclipsed relationship, there is only one exact dihedral angle which satisfies the policy. Whereas the other terms mark a violation of a strict enforcement policy if the contributed energy is above a small delta, doing so for this term would prevent reasonable energetically probable interpretations of the energy. In practice, we find that setting the cutoff threshold to about 15° of torsion difference before the term is violated still yields the desired results.

As an exception to the term detection scheme, an eclipsed relationship can also be defined involving an implicit bond. If (i) A and B are sp^3 hybridized, (ii) have no explicit neighbors other than A' and B', (iii) $\angle A'AB$ and $\angle ABB'$ are opposite turn directions, and (iv) exactly one of $\overrightarrow{A'A}$ and $\overrightarrow{BB'}$ is a wedge or dash bond, we have the atom at the end of the wedge or dashed bond eclipsing an implicit hydrogen. Figure 4.14 shows this relationship. Here B' at the end of the wedged bond $\overrightarrow{BB'}$ is eclipsing the implicit hydrogen H.

Practically, when we are detecting this exception, we do not want to switch over to performing a different calculation on $\angle HABB'$. So, during term calculation we track that this exception form was used, and we change γ in the calculation to reflect that we are not eclipsing the atom A', but a sibling hydrogen assumed to be present. When we are dealing with the eclipsing of implicit (or nonexistant) hydrogens, we set $\gamma = 300°$ or $60°$ whichever is closer to the current ϕ. This will ensure that B' is eclipsing one of the implicit hydrogens at A. Using two different angles like this creates two potential minima. Which hydrogen is actually eclipsed is determined by the R/S stereochemistry term which is active for wedge and dash bonds. Looking again at Figure 4.14, note here that $\overrightarrow{BB'}$ is a wedge or dash and that $\overrightarrow{A'A}$ and $\overrightarrow{AB}$ are both normal bonds. These were the requirements for the R/S stereochemistry term to be applied. Therefore, with the R/S stereochemistry term and the dihedral perspective term active, B' will eclipse the correct hydrogen without any additional instrumentation in the dihedral perspective term.

4.5.2 Gauche Relationship

In contrast to the eclipsed relationship, the gauche relationship is at a local energy minimum, but not the global minimum. A gauche relationship occurs when (i) the 2D turn directions $\angle A'AB$ and

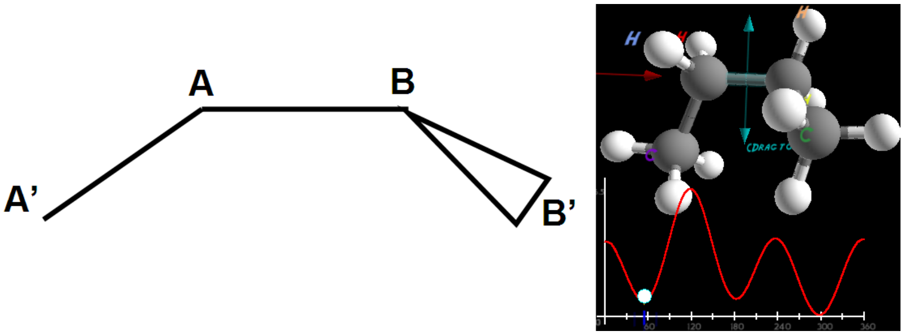


Figure 4.16: *The gauche relationship in the dihedral perspective term. Here $\overrightarrow{BB'}$ is coming forward from the plane $A'AB$. This places the dihedral angle at a local energy minimum which is shown in the energy curve at the bottom of the 3D image. The global minimum occurs when the dihedral angle is* 180°.

$\angle ABB'$ are the same and (ii) the bonds $\overrightarrow{A'A}$ and $\overrightarrow{BB'}$ are drawn differently[11] as shown in Figure 4.16. Just like the standard form of the eclipsed relationship, the turn directions are the same. The difference between being eclipsed at the energy maximum and being gauche at a local minimum is that the wedge or dashed bond allows that atom to move off of the maximum. Alternatively, if (i) the bonds are drawn the same, but (ii) A' or B' has more than two incident normal bonds, we use the gauche term as well instead of the eclipsed term which would otherwise be used. Having more than two normal bonds present is a cue that the drawing is being loose with the drawing cues and an exact eclipsed relationship was probably not intended. The lower energy gauche relationship is more likely in this case. Finally, we also apply the gauche term to the edge case of (i) the 2D turn directions $\angle A'AB$ and $\angle ABB'$ are the different and (ii) the bonds $\overrightarrow{A'A}$ and $\overrightarrow{BB'}$ are both wedges or both dashed. This is to account for butane molecules drawn with wedges or dashes at both ends and similar substructures in larger molecules where there are no other local, explicit cues to determine the 3D structure.

Once we have decided to apply the gauche relationship, we treat the bond $\overrightarrow{AB}$ as the double-bond in the Z/E Stereochemisty Term of Section 4.3 and use the Z/E equation. While the Z/E Stereochemistry term is intended for sp^2 hybridized (planar) relationships, its penalty is only defined when the turn directions differ from that in the drawing. Here, when the term is applied to sp^3 hybridized atoms, the two local gauche minima become the best choices. For the cases where we have a wedge or dashed bond in the relationship, as we did with the implicit hydrogens for the eclipsed relationship, we can then let the still active R/S stereochemistry term determine which of the minima is the correct minimum to use.

[11] This is differently out of the options normal, wedge, and dashed.

$$Energy' = \sum_{\text{Bonds}} + \sum_{\text{Angles}} + \sum_{\text{Torsional angles}} + \sum_{\text{Steric}} + \\ \sum_{\text{R/S}} + \sum_{\text{Z/E}} + \sum_{\text{Planar Perspective}} + \sum_{\text{Dihedral Perspective}}$$

Figure 4.17: *The basic form of the force field shown in Figure 4.1 with IM3 terms appended. Energy′ accounts for chemical feasibility and adherence to the four types of drawn cues (R/S stereochemistry, Z/E stereochemistry, rings with perspective view planes, and torsions in perspective).*

4.6 Combined Force Field & Future Work

A simplified form of our force field equation using the example IM3 terms is shown in Figure 4.17. The sum energy here accounts for all of the chemical feasibility and drawing adherence measures. A low value indicates that both are in good standing. In practice, it is useful to keep a separate value for each IM3 term penalty in addition to the sum force field energy. This allows the enforcement policy system (Section 3.4.5) to know the difference between feasibility problems and drawing adherence problems.

These four terms do not account for all of the notations used in molecule diagram drawing, but do handle a large number of diagrams one would expect to find in an undergraduate organic chemistry course. Within the context of organic molecule diagrams, we do not yet account for the cues of charge, lone pairs, crossing bonds (where one is in front in perspective), dots, and heavy bond lines. Beyond of our organic molecule context, the number of notations and cues found in molecule diagrams is vast and the four IM3 terms presented here provide only a beginning to the full problem of understanding 3D structure in molecule diagrams. We hope that this template for defining diagram constraints in terms of molecular mechanics will lead to the development of terms for new notations and eventually a system which can interpret the full range of molecule diagrams.

Additionally, while these terms are based on the parsing of the diagram, the digital ink data which comprises the diagram could still be useful for improving these terms or creating new ones. We defined atoms as having a single point and bonds as having two points. This is a lossy simplification of the actual data. With this simplification, we have no indication of the relative sizes of drawn symbols, the pressure applied to the stylus in each stroke, the time taken to draw each stroke, the number of strokes used, nor a measure of the confidence of our parsing. At the simplest, this richer data could be used to indicate confidence in our use of IM3 terms and act as switches to turn on and off terms. More interestingly, this data could be fully integrated into existing and new terms to define the relative importance of different areas of the diagram. We could then adjust the weight constants of the terms based on this importance measure.

Chapter 5

Evaluation

For the purposes of evaluation of the work in this dissertation, there are various high level questions one could pose. These questions fit into categories of questions about the application (such as "Is ChemPad useable by student chemists?" and "Does ChemPad help students learn to visualize molecules in 3D?") and questions about the conformation generation system (such as "How often is the algorithm successful at building the right conformation?" and "How fast is the algorithm?"). We present here the results of our investigations into both categories of questions.

5.1 Use By Students

To evaluate the ChemPad application as a whole, we consider how well it satisfies its primary purpose: to help introductory organic chemistry students learn to visualize molecules in 3D. To this end, twice a week throughout the middle of the semester, we have made ChemPad available to Introductory Organic Chemistry students at Brown University by opening labs of Tablet PCs with the software installed. At these labs, the students are provided with worksheets to help them learn the software and to give them some chemistry problems to solve using ChemPad. While many students stick to the content of the worksheets, we have had a number of students who bring their own problems to work on, or simply use the software to explore structure and see what can be made from various diagrams of their own design. Over the last three years, we have had more than 250 student users in our lab sessions. Additionally, ChemPad has been used as a lecture tool in the classroom by connecting the professor's Tablet PC to the lecture hall projector to show molecules to the entire class. In this form, ChemPad acts as a quick modeling tool for molecules both prepared in the lecture notes and spontaneously inspired by student questions.

Our evaluation of the above mentioned user experiences has both qualitative and quantitative aspects. First, we performed a qualitative analysis of the student user experience by asking for user feedback. Second, we performed a quantitative analysis of whether ChemPad achieves its pedagogical goals by tracking students' abilities to perform 3D thinking tasks in course examinations before and after using ChemPad. While the results of these analyses are more fully given in our earlier publications [81, 80], we present the highlights here.

The students who used ChemPad in the labs were generally quite positive about the experience. Positive comments mostly referred to being able to learn more about molecule visualization and the pedagogically-motivated visualizations we added to the system. Many students expressed their sincere appreciation that the tool had been developed with them in mind. Students were able to understand the interface and successfully use the software after only a few minutes of instruction. Negative comments by the users most often had to do with handwriting recognition of the characters for atoms and the accompanying frustration that a seemingly obviously drawn atom would be recognized as a different atom, or worse yet, an erase gesture. Other negative comments often critiqued specific interface design decisions. Through gathering such user comments on the software, we were able to make many usability improvements over the course of development. We treated chemistry students as our clients to ensure that ChemPad is a useful tool.

The pedagogical value of ChemPad was evaluated over a series of quizzes and exams in the first year of the project. A weekly quiz on stereochemistry, the first topic in the course that really demands 3D visualization ability, was used to identify students who were having difficulty with visualization. Amongst this subset of the class, we compared the ability of students who used ChemPad to those who did not by comparing later quiz and exam questions requiring 3D visualization skills. This comparison is shown in Figure 5.1. While we originally intended to use another lecture section of the course as a control group for the study, when students from that section asked to use the ChemPad labs, the course instructor and the author felt we could not in good conscience deny them. Therefore, we used statistical analysis to control for “student motivation and aptitude” by using their performance on other chemistry quiz and exam questions to normalize their scores on the 3D thinking questions.

After normalization, we found that there was still statistical significance to the performances of students who used ChemPad being greater than the performances of those who did not. Moreover, this improvement was more noticeable in 3D visualization tasks that were unlike the tasks students had on the worksheets they received in the ChemPad labs. We hypothesize that since the lab worksheets were made available to all students on the course website, they were probably used by many students who did not attend the labs to study for the exams. However, the students who actually used the software, and not just the worksheets, were able to transfer their knowledge to new and different problems.

5.1.1 Pedagogical Methodology

ChemPad’s educational success in this study can be attributed to the pedagogical methodologies which guided our development of the software. In particular, ChemPad is intended to help students

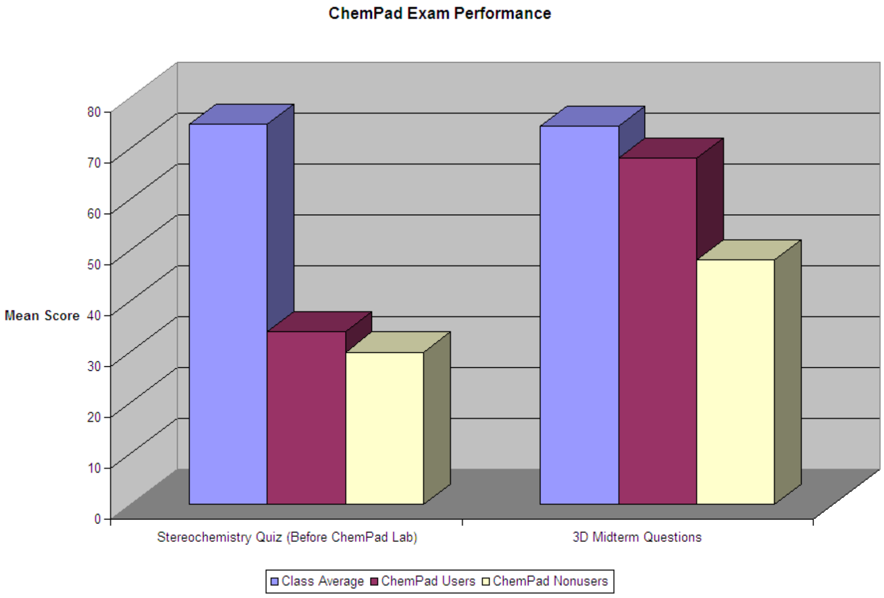


Figure 5.1: *Student performance on questions requiring 3D visualization skills. The blue bar indicates the average of the class, while the red and yellow bars are for students identified as having difficulty visualizing molecule structures with red being students who used ChemPad and yellow being students who did not. The scores on the left show users and non-users of ChemPad before the ChemPad lab opened. The scores on the right are from 3D questions on an exam after the completion of the lab.*

with a combination of exploration and scaffolding provided by its visualization assistance. Tools which allow visualization in science have been valued much in educational literature [75, 34]. For chemistry in particular, Steiff posits in an article for the *Journal of Chemical Education* that "...the use of computer-aided visualization tools dissuades rote memorization by encouraging students to actively investigate the nature of atomic structures." [74]. Others have seen this value in molecule visualization and several Organic Chemistry course web pages provide students with the opportunity to manipulate pre-built 3D molecules with web browser plugins such as Chime [77]. While most of the visualizations ChemPad provides are similar to these, the key difference is that with ChemPad, the student is not restricted to the existing visualizations. In ChemPad, the student can make modifications to the molecule and thereby experiment with and explore their molecules. The value of this exploration where students are given the tools to raise and answer their own questions, instead of following a linear curriculum, such as that found on a web page, has been espoused in Duckworth's concept of "Having Wonderful Ideas" [27].

Scaffolding is the process of helping students work through problems they cannot accomplish alone with the theory that through such experiences they develop skills to later be able to solve the problem without assistance. The technique was first formally proposed by Vygotsky in his theory on the Zone of Proximal Development [87]. Scaffolding comes both explicitly and implicitly in ChemPad. As for the tasks for which ChemPad scaffolds students, implicitly, the standard visualization capability of ChemPad is helpful for solving a number of problems one would find in an Organic Chemistry course. This was indicated by the number of students attending the lab who used ChemPad to work through problems in their textbooks. Explicitly, ChemPad contains additional visualization tools for stepping students through the process of determining the chirality of molecules and the relative energies of different conformations. With continued development of visualizations and pedagogically-oriented tools, the ChemPad platform could come to provide explicit scaffolding for a wide range of organic chemistry tasks.

5.2 Algorithm Evaluation

To evaluate the actual conformation generation algorithms and the IM3 force field which are the foci of this dissertation, we prepared a test set of 103 "hand-drawn" molecule diagrams (detailed in Appendix C.) We worked with our chemistry collaborator to choose these tests to be representative of relationships found in organic molecules, specifically relationships and drawing cues we've found to be problematic over the years of developing this work. For each drawn diagram, one to five possible "ground truth" models were assigned to which we could compare the output of the system. These models represented solutions that were plausible interpretations of the diagram, usually including the best interpretation and occasionally including incorrect interpretations, but they were not exhaustive sets of all plausible interpretations. Each "ground truth" model was given a quality score from Figure 5.2 which indicated how well the algorithm performed to produce the model from the diagram. These

- Preferred: A conformation specifically indicated by the diagram.
- Not Preferred: A conformation chemically equal to the one indicated in the diagram, but different.
- OK: A conformation which is a reasonable interpretation of the one indicated in the diagram, but not the only correct interpretation.
- Error: A conformation chemically different from the one indicated in the diagram.

Figure 5.2: *Quality scores given to each reference conformation in the test set. Tests which were unambiguous had one conformation marked as preferred, and others marked as not preferred. Tests which were ambiguous had reasonable interpretations marked as OK. Any conformation chemically different from the test diagram is marked as an error.*

scores were chosen to allow us to answer the following questions:

- Did the algorithm succeed in creating a conformation?
- How long did it take the algorithm to generate the first conformation?
- How long did it take the algorithm to generate each conformation?
- Did the algorithm find the ideal conformation?
- Did the algorithm create a chemically wrong conformation?
- Did the algorithm create multiple correct conformations for an ambiguous diagram?

When creating this test set, we derived almost all of the tests from a data set used by Wang during developing the General Amber Force Field. The data set used in Figure 4 of his paper on the force field [90] contains a number of molecules with different conformations provided. These conformations provided the "ground truth" models we needed. We selected a subset of these conformations for which to draw appropriate molecule diagrams and to score each conformation. Diagrams were made for generic and specific ways of drawing the selected conformations. Our aim was to include tests covering different molecule sizes and features, as well as the drawing cues one can make in the ChemPad inking system. While this test set may not be objectively a good benchmark for future conformation generation systems, as there are only 103 tests and they only contain diagrams that can be drawn with the ChemPad input system, it is suitable for comparing different versions of the ChemPad algorithm.

Each test diagram was run on nine different versions of the conformation generation algorithm. The first version was our full, completed algorithm. Each other version disabled one or more parts of our algorithm to measure the effect the disabled contributions make to the system. The nine different versions of the algorithm we used are:

1. The Full System

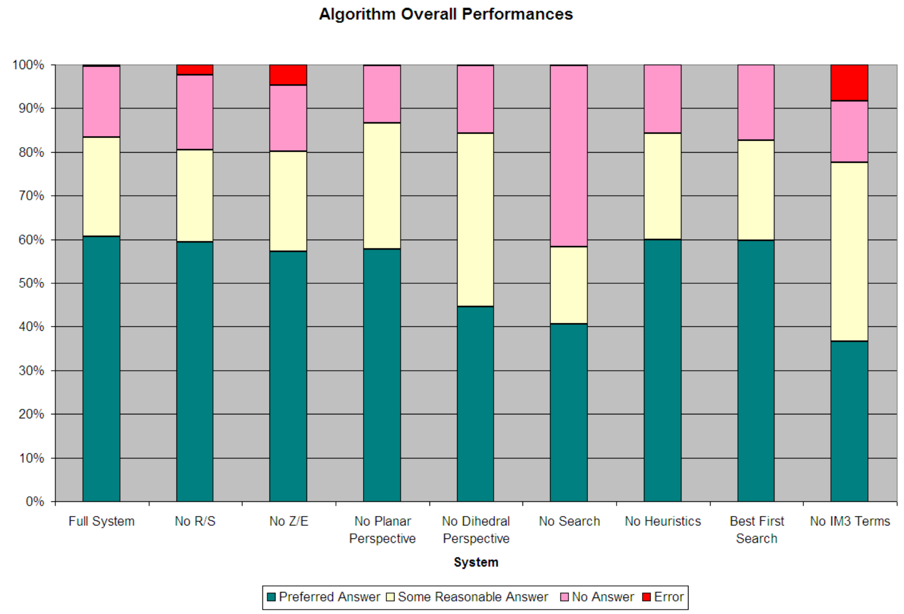


Figure 5.3: *Success and failure rates out of the entire test set. The green bar marks tests where the preferred answer was generated and no chemically incorrect conformations were generated. The yellow bar marks additional tests where some reasonable answer was given, but not the best one. This counts "OK" and "Not Preferred" conformations, but does not count conformations which do not match any of the reference conformations. The pink bar marks tests which returned no answer. The red bar marks tests which generated a chemically incorrect conformation.*

2. Without the R/S Stereochemistry Term (Section 4.2)
3. Without the Z/E Stereochemistry Term (Section 4.3)
4. Without the Planar Perspective Term (Section 4.4)
5. Without the Dihedral Perspective Term (Section 4.5)
6. With no search mechanism (Section 3.4.2)
7. With no chemistry heuristics driving the search (Section 3.2.1)
8. With Best-First Search instead of Limited Discrepancy Search (Section 3.4.4)
9. With none of the above IM3 terms

5.2.1 Test Results

Figures 5.3 - 5.7 give an overview of the results of the algorithm analysis. Each test was run six times per algorithm version on a modern Tablet PC with an Intel Core 2 Duo processor, 512M of memory, and running Windows XP Tablet Edition for an operating system. The mean of the six

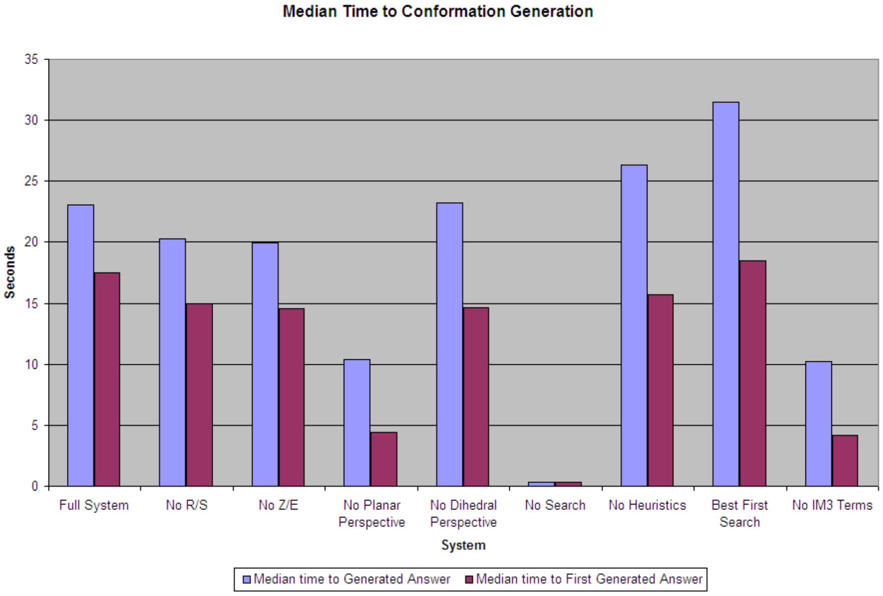


Figure 5.4: *Time to generate conformations for each algorithm variant in Section 5.2. The blue bar indicates the median time to generate all conformations. The red bar indicates the median time to generate the first conformation for each test.*

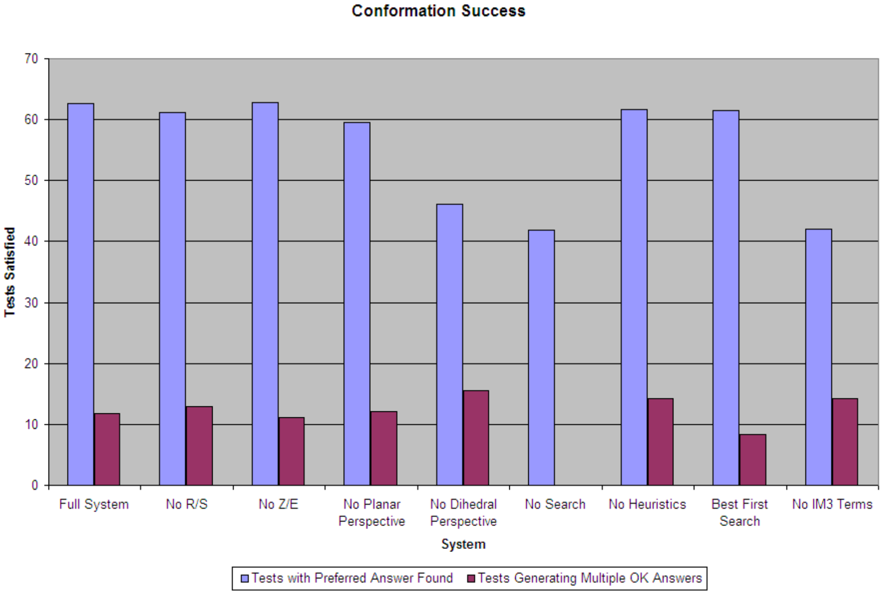


Figure 5.5: *Successful completion rates at the conformation generation task for each algorithm variant in Section 5.2. The blue bar indicates the number of tests (out of 79) which yielded the ideal conformation. The red bar indicates the number of tests (out of 24) which yielded multiple good interpretations of an ambiguous diagram.*

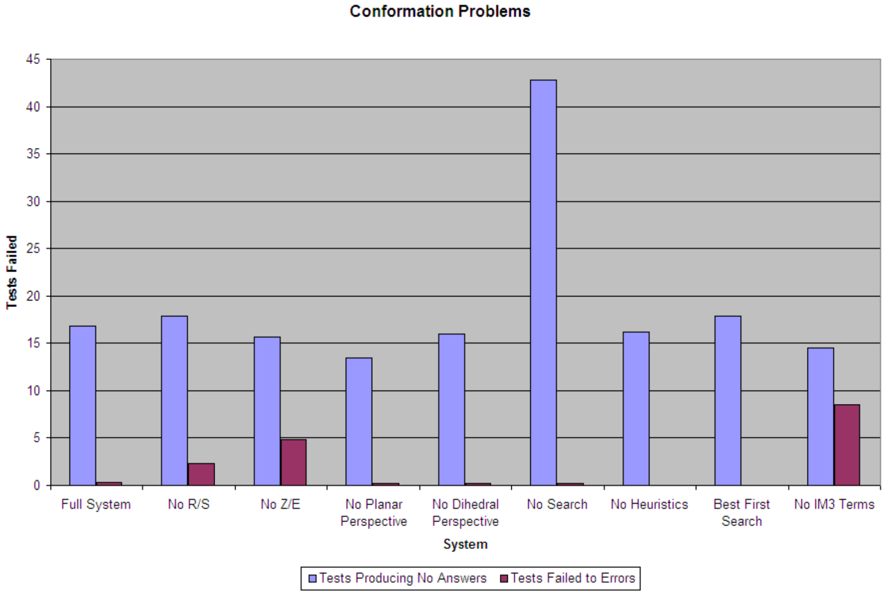


Figure 5.6: *Failed completion rates at the conformation generation task for each algorithm variant in Section 5.2. The blue bar indicates the number of tests which yielded no predicted conformations. The red bar indicates the number of tests (out of 13) which yielded chemically incorrect conformations.*

test runs are presented in the figures. Each figure shows a comparison of the nine algorithm versions. The first figures gives a high-level view of the performance and the remaining figures address two of the questions from Section 5.2.

Figure 5.3 shows the overall success rate of each algorithm. The green and yellow bars sum to the general success of the algorithm finding an expected conformation with the green being the preferred conformation. The pink and red bars show where the algorithm failed with pink being conformations which were not expected and red being conformations which are chemically wrong. Of the IM3 terms, the stereochemistry terms prevent incorrect conformations from being generated and the perspective terms increase the number of preferred conformations returned. Out of the generation algorithm changes, removing search greatly reduces the overall success rate, while removing the heuristics and using best first search has little effect on the success rate. For the latter two, we expect this as the two are meant to improve performance speed, not fundamentally change the results we generate. We also note that the results here for the planar perspective term do not show much of an increase in preferred conformations returned. We suspect this is due to our algorithm returning multiple results for these diagrams. In such cases it is very likely to get the preferred one conformation out of the multiple attempts. The term instead acts to filter out those from this set which are not the preferred conformation.

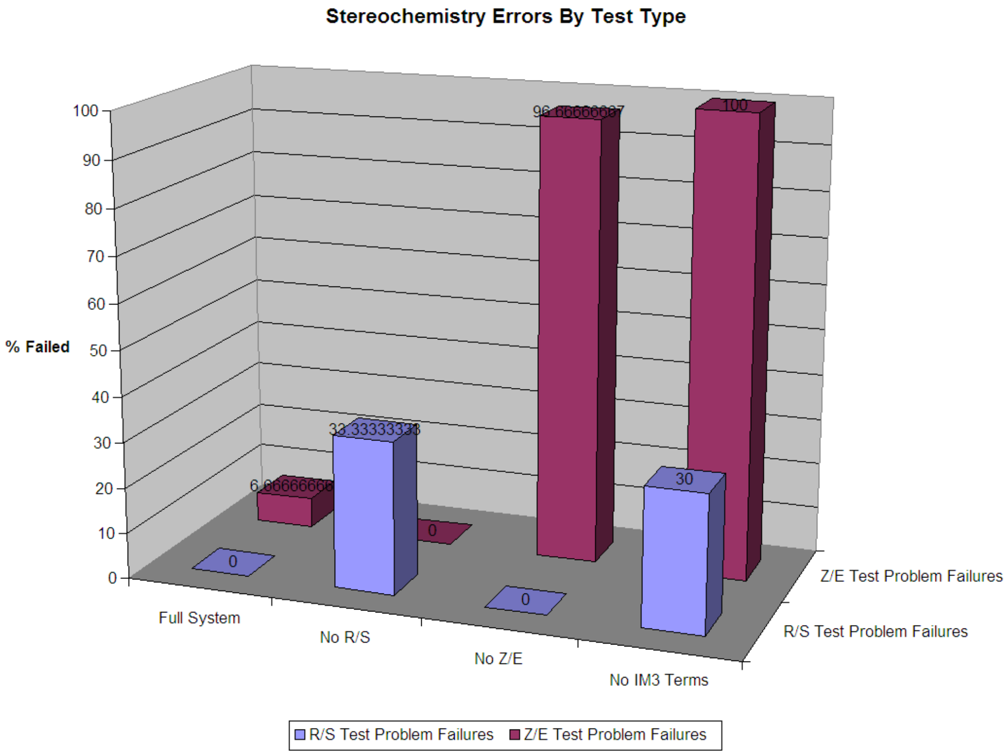


Figure 5.7: *Percentage of R/S stereochemistry and Z/E stereochemistry failures by algorithm variant. Here the algorithm variants are evaluated only over the subset of tests which could be failed due to an R/S stereochemistry (blue bar) or Z/E stereochemistry (red bar) error. The IM3 terms defined to address each problem almost completely eliminate errors of the type.*

Most critically missing from Figure 5.3 is the time required to run the algorithm. This is shown in Figure 5.4 which compares across algorithm versions the median time to construct each conformation as well as the median time to construct the first conformation. The time to construct the first conformation is the time until the user first has a model with which to interact. Disabling IM3 terms causes these times to decrease, presumably because it is easier to satisfy constraints in the diagrams. Disabling search greatly reduces the time to build the first model because no alternatives are considered. Alternatively, disabling the other two algorithm advances[1] increases the time required to build subsequent conformations indicating they both make the system run faster. The use of Limited Discrepancy Search over Best First Search provides the greater speed improvement for conformations after the first

Figure 5.5 shows the algorithm versions abilities to successfully find conformations. One measure of success is how often the algorithm was able to produce the ideal answer. There were 79 tests in the set with this possibility and the number successful is represented by the blue bar in Figure 5.5. Turning off the dihedral perspective term reduces this number as that term handles diagrams of models where the ideal state is not at an energy minimum. The algorithm would otherwise discourage these answers. Turning off search also reduced the ability to find these ideal conformations as the system only has one chance at finding the best conformation. Alternately, the other measure on this figure is how many ambiguous test diagrams yielded multiple good but different answers. There were 24 tests where this was a possibility and the number successful is indicated by the red bar. In general, turning off all of the advancements which restrict the search (the IM3 terms and the search heuristics) increased this value as they were not there to prune the search tree.

Based on the result that restricting the search decreases the likelihood of finding multiple, good answers, it may seem beneficial to remove these restrictions. However, by doing so, erroneous answers are also produced. Figure 5.6 shows failures by the different algorithms. In particular, the red bar indicates the generation of models which are chemically wrong. There were 13 tests for which this was a possible outcome. As expected, turning off the IM3 terms meant to handle these cases (R/S and Z/E) leads the algorithm to make these terrible mistakes. While the distribution of errors by term is unbalanced, the amount of errors produced is largely a function of the number of types of tests in the test set. For instance, out of the tests in Appendix C, there are five where the R/S Stereochemistry term would have effect and an error was possible. Additionally, one of those could reasonably be expected to be solved using either the R/S term or the dihedral perspective term. Therefore, in the absence of the R/S term, one would expect the generation system to get the test wrong half the time i.e. error increasing by $\frac{4}{2} = 2$, which is what we find in the results.

To reduce the scope of the tests which matter to each term, Figure 5.7 shows the performance of four of the algorithm variants on the stereochemistry problems in the test set. The red bar shows the

[1] These would be the the heuristics and the limited discrepancy search.

percentage of R/S stereochemistry tests failed by the algorithm and the blue bar does the same for the Z/E stereochemistry tests. Here the algorithms performed largely as expected. The algorithms with terms designed to handle the tests almost universally succeeded. In the case of the one R/S test which was failed by the full algorithm, that test produced the incorrect stereoisomer only after returning the ideal answer. In comparing the failure rates of algorithms without terms to handle the tests, the Z/E tests were failed more often than the R/S tests. As previously mentioned, the Z/E tests were much more likely to produce multiple answers. They were thereby able to "succeed" at finding the wrong answer more often.

Another type of failure is the inability to produce any of the expected answers which is indicated by the blue bar in Figure 5.6. Disabling search greatly increases this value as the system then only has one chance at producing a good answer. All of the algorithms had a failure rate much higher than we experienced in informal testing of the system. A closer inspection of the raw result data indicated that some tests are much more likely to be missed than others. Indeed, there were 11 tests that were missed by most of the system configurations. We explored why this is and out of those, several of them are being marked as failures incorrectly when the produced results ideally should have produced a recognizable result. In these cases, the algorithm created reasonable answers that don't match any of the reference conformations in our test set. Additionally, some are additionally marked incorrect because the algorithm finds ways to stretch bonds and angles enough to decrease the energy over the reference conformation, thereby producing a better answer. Finally, some are indeed incorrect and for these, most of the versions of the algorithm are producing the same incorrect result.

5.3 Evaluation System

As mentioned in the last section, one major caveat for all of these results is that the evaluation system is not foolproof at determining when one conformation matches another. When using the comparison system in Section 3.4.3 for the purpose of conformation generation, a false negative only negatively affects speed as redundant work is performed. Conversely, a false positive could prevent that branch of the search space from being explored. For that reason, the system has been tuned to favor false negatives over false positives. In the case of the evaluation system, a false negative will make the system be unable to correctly identify the output conformation, thereby making the algorithm appear to perform worse than it does. Or, in the case of a false negative in comparing to a conformation marked as an error, making the algorithm appear to perform better than it does. For the purpose of this evaluation, since all algorithms were evaluated with the same evaluation system, we believe the relative abilities of each system are accurately represented if not the absolute abilities.

5.3.1 Comparing Nodes When the Atom Alignment is Unknown

Critical to the evaluation system is the ability to compare one molecule conformation to another in a way that is invariant to rotation and slight differences between the conformations. Section 3.4.3 showed how this can be done for models where we have an existing alignment of the atoms. By alignment, we mean that for any atom in one conformation, we know which is the corresponding atom in the other conformation. For the evaluation system, we don't necessarily have an alignment since the ordering of the atoms in the reference conformation may be quite different from the ordering in the constructed conformation. What we do have is the alignment between the atoms as drawn in the diagram and the atoms in the generated conformation. This is already being tracked by the conformation generation process for the purpose of detecting duplicate nodes. Therefore, what is needed is the alignment between the atoms in the reference conformation and the atoms in the test diagram. Since these are invariant at test runtime, we can therefore pre-generate these alignments at test construction time without any need to perform additional calculation on a run-by-run basis.

Practically, it may be difficult to tell what is the correct alignment between atoms in the diagram and in the reference conformation. Moreover, there is often a good deal of ambiguity as to the correct alignment as molecules contain symmetry. A methyl group (a carbon atom with three hydrogen atoms connected) has six correct alignments for the hydrogens alone. Any hydrogen atom connected to that carbon in the diagram can correctly be mapped to any of the hydrogens connected to the corresponding carbon in the reference molecule. Even in the absence of hydrogens and halogens, symmetry can exist. For instance, any alkane (carbon chain) can can be correctly aligned starting at either end of the chain.

Instead of searching for the single alignment which is definitely the correct correspondence between the diagram atoms and the reference conformation atoms, we can instead think of having a set of alignments, any one of which could be the proper one. Starting with the set of all possible alignments, a set of size $n!$ (where n is the number of atoms in the molecule), we eliminate alignments based on two easy to evaluate rules.

1. Do the atomic numbers match?

2. Do the neighbors match?

We also eliminate aligning the implicit hydrogens in the comparison to reduce the number of symmetries. For many of the tests, these two rules are enough to reduce the number of possible alignments to one or two possibilities and all but one set have less than ten. At run time, we then perform the comparison on each possible alignment of the atoms and report a match if any of the alignments cause the conformations to match.

5.4 Future Work

While sufficient for the purposes of comparing different versions of the IM3 force field and conformation generation algorithms within ChemPad, this test set is quite limited. It does not address the vast number of types of molecule diagrams that can be drawn, nor does it attempt to represent the actual distribution of molecule diagrams used by chemists. Creating a full benchmark for comparison of systems which generate conformations from diagrams would be an ambitious, but worthwhile endeavor.

Similarly, we have not yet run a rigorous comparison of the performances of the ChemPad algorithm variants to the conformation generation systems present in OrganicPad and ChemOffice. While we performed some manual comparisons of the systems, such as the example in Chapter 2.2 which compared the interpretations of the diosgenin diagram by ChemPad and ChemOffice, and were generally satisfied that our system is more sophisticated than these others, we have not had time to architect these other systems to run our benchmark test set.

Moreover, this analysis of the algorithm versions counts failures of the system, but does not explore in depth the reasons for the failures. A more detailed analysis of the reasons behind suboptimal performance, particularly for the full system with no parts disabled and cases where enabled IM3 terms should have prevented errors, would be critical to advancing this work.

Chapter 6

Conclusion

Little previous research exists on automated systems for understanding drawn molecule diagrams as 3D structures despite the potential for such systems in electronic lab notebooks, chemistry education tools, and desktop molecular modeling systems. The contributions presented in this dissertation provide only a beginning to tackling this problem whose scope is as large as both the number of different techniques chemists use to indicate structure in their diagrams and the number of molecules chemists wish to represent with diagrams. We provide here a review of these contributions.

In Chapter 3 we provided a history of our iterative process of creating an algorithm for conformation generation based on a parsed molecule diagram. This history highlighted the problems with obvious approaches to the problem, such as treating atoms as if they were the rigid plastic pieces found in modeling kits and using molecular mechanics optimizers to improve generated conformations. We then detailed the means to overcome many of the stated problems. We showed how to formulate the generation process as a search problem using domain-specific and diagram-driven heuristics to guide the search, thereby improving algorithm speed and accuracy.

In Chapter 4 we presented the framework of Ink-Modified Molecular Mechanics which adds an understanding of the underlying diagram to the force field calculating conformation energies. This allows for the conformation generation techniques of the previous chapter, as well as existing conformation optimization algorithms, to solve for chemically likely conformations which match the diagram. We went on to give IM3 terms to account for four of the most prominent 3D structure cues chemists draw in diagrams or organic molecules. These terms were R/S stereochemistry as indicated by wedge and dash bonds, Z/E stereochemistry of alkenes, rings drawn from a sideways perspective, and nonoptimal torsional angles in perspective. We also provided guidelines for creating future IM3 terms (Section 4.1) so that they can be generically compatible with our terms and a variety of force fields.

In Chapter 5 we evaluated the performance of our system in terms of both our goals for making a useful tool for users and the speed and precision of the algorithms involved. For the former, we noted that users generally approved of the use drawing molecule diagrams as a molecule modeling technique and that students having difficulty with visualizing diagrams in 3D were able to apply the visualization skill they gained using ChemPad to new problems. Amongst this group of students, those who used ChemPad had statistically better scores on 3D visualization exam problems than the students who did not use ChemPad. For the latter, we looked at the performance of our algorithmic contribution by comparing our final system to eight variants which had one or more of the improvements described in Chapters 3 and 4 disabled. The stereochemistry IM3 terms performed their task of preventing the system from generating incorrect stereoisomers particularly well while the perspective terms performances were less apparent in our measures. The use of search through the space of discrete generation steps had a major impact on the ability of the system to produce correct answers. Applying Limited Discrepancy Search with heuristics based on the user's drawing cues increased the speed of the system (median time to generation dropped by approximately 27%) while still producing conformations of the same quality as best first search. We concluded with ideas as to how to expand our evaluation framework from the current one suitable only for evaluating different versions of our program into a general benchmark for conformation generation systems.

6.1 Additional Research Directions

We have presented a number of directions for future work related to these main contributions at the end of their respective chapters. Here we recap the main future work directions already presented and give a few additional directions that derive from the work as a whole.

We presented here IM3 terms for handling four types of drawing cues present in molecule diagrams, but as molecule diagrams are a large language with each branch of chemistry having its own conventions, shorthands, and cues defined, there are many more than four drawing cues a *full* molecule diagram interpretation system should take into account. The IM3 framework should theoretically be able to handle such drawing cues given further time spent developing the terms for those additional cues. While this task is ambitious, we believe the example terms we presented here should make good templates to follow in the creation of additional terms.

Our conformation generation system attempts to solve a difficult global optimization problem efficiently through an ad hoc approximation of the bonding processes through which molecules are actually created and search over discrete decisions within that approximation. This system scales well as the size of the molecule increases[1], but fails to account for more global interactions of atoms in the molecule. Therefore, as the molecule gets bigger, the chances of finding the global energy

[1] This complexity is $O(n * O(\text{force field}))$ where $O(\text{force field})$ is $O(n)$, $O(n \log n)$, or $O(n^2)$ depending on force field implementation.

minimum decreases. We believe a more accurate model of the processes by which atoms become bonded in nature could help overcome some of this limitation. However, as the molecules become very large, the main task becomes that of the protein folding problem.

While the techniques presented here depend on an accurate parsing of a diagram as input, one could alternatively use conformation generation as part of a “language model” for the diagram parsing task itself. Given a set of likely parsing hypotheses for a diagram, the energies of the corresponding conformations could give insight into the intended parsing. Hypotheses which yield very high conformations energies would indicate that the hypothesis should be rejected for being chemically unsound.

We have created here a means for chemists to create 3D models using a stylus via molecule diagrams, but we have not explored techniques for controlling and manipulating said models with the stylus. We have primarily used standard mouse-originated interaction techniques for controlling the 3D scenes such as a camera zoom slider and a virtual trackball for rotation. However, depending on hardware implementation, the stylus has much more potential for 3D control as the driver can report the pressure, tilt, twist, yaw, pitch, and roll of the stylus in the ink data. This true 3D stylus data could be used to make the pen into a full 3D control device. Additionally, even without this data, just as standard 2D mouse-based widgets are not ideal for use by a stylus, we expect that 3D widgets could be improved by exploring the form factor further.

Besides generic 3D control with a stylus, there is also the question of how to design molecule manipulation techniques which would be intuitive to students. For instance, we currently allow rotation around single bonds in ChemPad by tapping on the bond and drawing along the perpendicular axis while in “rotation mode.” We do not have a means of performing a chair-boat interconversion, a scaling of bond lengths, stretching of angles, nor other manipulations a student might want to perform while exploring the molecule visualization. Beyond this, even with an intuitive means to change these structure components, one would need to come up with a system for making the rest of the molecule respond to the local changes. In our current version, single bond rotation has no effect on the rest of the structure, which is not an accurate representation of molecule forces. We believe that adding a simple and fast force simulator which updates the rest of the structure in response to these manipulations could give feedback that would improve student intuitions. Furthermore, one could adapt the idea of modifying molecular mechanics for diagram understanding into modifying molecular mechanics for interaction in simulation. Here the terms would not be to make the conformation adhere to the drawing cues, but for adherence to gestures or other interactions instantiated in the 3D scene.

6.2 Concluding Remarks

Organic Chemistry is a *hard* course. It is infamously known for its ability to thin the number of college students who pursue premed and chemistry majors. Interacting with these students during the course of this work has underscored the magnitude of the enrolled student population who do not have *any* background experience with 3D visualization. They struggle to attain this skill while keeping up with the rest of the course's staggering work load. We have been inspired by both the determination we've seen in students apply to learning this skill through ChemPad and the thanks we've received from already skilled students who said ChemPad adjusted and reinforced their level of understanding. We count a victory for every student who succeeds at Organic Chemistry because of this work and goes on to become a successful doctor or chemist. Even one is enough to make the world a better place.

Appendix A

Molecular Mechanics Primer

This primer is intended to give an introduction to molecular mechanics for computer scientists. Molecular mechanics concerns itself with formulating molecule structure energies efficiently on a computer. A number of different molecular mechanics systems, or force fields have been developed over the years to accommodate different types of molecules and calculations. A force field contains an equation defining the energy of a conformation and a data set of constants for the equation terms. While the formula terms are general in their definitions, the constants are used to approximate the interactions of specific types of atoms in specific environments. They are produced from data fitting techniques applied to results from lab experiments in molecule structure.

Where equations are shown in this section, the equations shown are for the AMBER [67] and GAFF [90] force fields in particular although the formulations are usually very similar for other force fields. Good explanations of molecular mechanics concepts for computer scientists wishing to implement a force field can be found in Heath, Kavraki, and Shehu [41].

A force field equation typically contains terms for bond length, angles between atoms, torsional angles of bonds, improper torsional angles for certain atoms, and Van der Waal interactions between atoms separated by at least 4 bonds. Additional terms may be present to account for more complex phenomena.

$$\begin{aligned} E_{\text{total}} = \sum_{\text{bonds}} K_r(r - r_{eq})^2 + \sum_{\text{angles}} K_\theta(\theta - \theta_{eq})^2 \\ + \sum_{\text{dihedrals}} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] \\ + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}} \right] \end{aligned}$$

A.1 Atom Types

Although atoms are defined in chemistry by their atomic numbers and atomic weights alone, for molecular mechanics, atoms are also differentiated by additional features such as the connectivity neighborhoods they inhabit. For this reason, there are 17 different versions of, or atom types for, carbon within the GAFF force field. Alternatively, there is only one atom type for fluorine. These different atom types can be thought of as the different pieces in a plastic ball-and-stick modeling kit, since they define the idealized shape the atoms take under different conditions. However, instead of defining only the idealized shape local at that atom, they define the shape as a function of nearby atom types and define the strength of those shapes or how easily they give way. In the following force field terms, whenever a constant is determined by the atoms engaged in the term, it is the atom types, rather than the atomic numbers that determine the constants.

A.2 Bond

$$\sum_{\text{bonds}} K_r(r - r_{eq})^2$$

The distance between bonded atoms achieves equilibrium where the force of the pull by electrons shared by the atoms exactly equals the force of the repulsion of the atom nuclei. The bond length term measures the amount of energy required for two bonded atoms to have a distance different from this equilibrium. In the formulation, K_r is the empirically determined force constant, r_{eq} is the empirically determined equilibrium bond length, and r is the current bond length. K_r and r_{eq} depend on the specific atoms (atom types) in the bond and the order of the bond and can be found in the parameter sets of the force field.

A.3 Angle

$$\sum_{\text{angles}} K_\theta(\theta - \theta_{eq})^2$$

or

$$\sum_{\text{a}\epsilon\text{atoms}} \sum_{\text{a1,a2}\epsilon\text{a.neighbors}} K_\theta(\theta - \theta_{eq})^2$$

Similarly, there is an equilibrium state for the angles formed by a given atom and any two of its neighboring atoms. These atom angles represent the specific hybridization of the atom electrons. The angle term represents the amount of energy required to “bend” these bonds into non-idealized conformations. Here K_θ is the empirically determined force constant and θ_{eq} is the empirically determined equilibrium constant. θ is the existing angle between the atoms. K_θ and θ_{eq} both depend on the three atoms engaged in the angle.

A.4 Torsion

$$\sum_{\text{dihedrals}} \frac{V_n}{2}[1+\cos(n\phi-\gamma)]$$

or

$$\sum_{\text{b}\in\text{bonds}} \; \sum_{\text{a}\in\text{b.atom1.neighbors}} \; \sum_{\text{c}\in\text{b.atom2.neighbors}} \frac{V_n}{2}[1+\cos(n\phi-\gamma)]$$

Torsional angles, or dihedral angles, are the angles formed by four atoms connected in a three bond chain. The angle measured ϕ is that of $\angle ABC$ in the plane perpendicular to the center bond. Here both bond atoms have the point B and the other two atoms have the points A and C. The constants are n the periodcity of the term, γ the equilibrium, and V_n the force constant.

One way to think of the torsion term is to think of this as the energy representing the repulsion of atoms that are close together but not already accounted for by the bond and angle terms. The bond term defines the interactions of atoms one bond apart and the angle term handles the atoms two bonds apart. In turn, torsional angles measure the interactions of atoms three bonds apart.

From an implementation perspective, the torsional term is the most difficult to complete correctly. First, one should check closely if the V_n term has been pre-divided by 2 in the data set as it is in AMBER and GAFF. Furthermore, a means to calculate the torsional angle itself is not intuitive, although we provide a calculation in Appendix D.2.

A.5 Improper Torsion

$$\sum_{\text{improper}} \frac{V_n}{2}[1+\cos(n\phi-\gamma)]$$

or

$$\sum_{\text{a}\in\text{atoms}} \; \sum_{\text{b,c,d}\in\text{a.neighbors}} \frac{V_n}{2}[1+\cos(n\phi-\gamma)]$$

Although the energy function of the improper torsion is the same of the torsion term, improper torsional angles are defined over three atoms all connected by one bond to a fourth. In particular, the improper torsion term is defined for sp^2 hybridized atom types and is zero for all other cases. Here the torsion angle is formed in the order $\angle BACD$ even though C and D are not bonded to each other.

A.6 Steric

$$\sum_{i<j}\left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}}\right]$$

or calculated differently

$$\sum_{i<j} 4 * \rho * (\frac{\sigma}{R_{ij}}^{12} - \frac{\sigma}{R_{ij}}^{6}) + \frac{Q_i * Q_j}{\epsilon} * Rij$$

While the other terms define the interactions of atoms connected by three or less bonds, the steric term defines the interactions of each atom on other atoms more than three bonds away. These atom pairs are attracted to each other gently, but push apart strongly as the atoms begin to invade each others' electron shells. The measured term here is R_{ij} the current distance between atoms i and j. The constants are ρ the well depth of i and j, σ the hard sphere radius of i and j (the Van der Waal radii averaged under the Lorentz-Berelot combining rule), ϵ the effective dialectric constant for i and j and Q_i and Q_j electrostatic constants for i and j respectively. While the electrostatic term at the end is not strictly steric strain, it is a relationship between distant atoms and therefore considered here.

A.6.1 Others

While this concludes the force field terms present in GAFF, additional terms are present in other force fields to represent complex relationships that occur in other classes of molecules that are not represented in this set of terms. For instance, AMBER contains a term which represents the interaction of hydrogen bonds while MM2/MM3 contains a stretch-bend term and MM4 [61] contains even more terms to compensate for spectroscopic frequencies.

Appendix B

IM3 Term Derivatives

To use gradient-based optimization techniques with an IM3 force field, it is necessary to have the analytical partial derivative for each term with respect to the x, y, and z coordinates of involved atoms. For the purpose of reproducibility, the x partial derivatives are given here since taking the derivative directly is moderately complicated for all terms. Knowledge of the torsion derivative by Blondel and Karplus [11] is required for deriving the Planar Perspective term partial derivatives and the eclipsed form the the Dihedral Perspective term.

B.1 Z/E Stereochemistry Term Partial Derivatives

$$M = 2 * K_{ZE} * (\overrightarrow{CROSSA} \cdot \overrightarrow{CROSSB})$$

$$\frac{\partial Energy_{ZE}}{\partial A'_x} = M * [\frac{\overrightarrow{AB}}{\|AB\|} \times [\frac{1}{\|AA'\|} * ((1,0,0) - (\frac{\overrightarrow{AA'}}{\|AA'\|} * \frac{A'_x - A_x}{\|AA'\|}))]] \cdot \overrightarrow{CROSSB}$$

$$\frac{\partial Energy_{ZE}}{\partial B'_x} = M * [\frac{\overrightarrow{BA}}{\|BA\|} \times [\frac{1}{\|BB'\|} * ((1,0,0) - (\frac{\overrightarrow{BB'}}{\|BB'\|} * \frac{B'_x - B_x}{\|BB'\|}))]] \cdot \overrightarrow{CROSSA}$$

$$\begin{aligned}\frac{\partial Energy_{ZE}}{\partial B_x} = M * ((\frac{\overrightarrow{BA}}{\|BA\|} \times ((\frac{1}{\|BB'\|} * ((-1,0,0) - \frac{\overrightarrow{BB'}}{\|BB'\|} * \frac{-B'_x + B_x}{\|BB'\|}))) +\\ (-1 * (\frac{1}{\|AB\|} * ((1,0,0) - \frac{\overrightarrow{AB}}{\|AB\|} * \frac{B_x - A_x}{\|AB\|}))) \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \overrightarrow{crossA} +\\ \overrightarrow{crossB} \cdot (((\frac{1}{\|AB\|} * ((1,0,0) - \frac{\overrightarrow{AB}}{\|AB\|} * \frac{B_x - A_x}{\|AB\|})) \times \frac{\overrightarrow{AA'}}{\|AA'\|}))\end{aligned}$$

$$\begin{aligned}\frac{\partial Energy_{ZE}}{\partial A_x} = M * ((\frac{\overrightarrow{AB}}{\|AB\|} \times ((\frac{1}{\|AA'\|} * ((-1,0,0) - \frac{\overrightarrow{AA'}}{\|AA'\|} * \frac{-A'_x + A_x}{\|AA'\|}))) +\\ (-1 * (\frac{1}{\|AB\|} * ((1,0,0) - \frac{\overrightarrow{BA}}{\|BA\|} * \frac{A_x - B_x}{\|AB\|}))) \times \frac{\overrightarrow{AA'}}{\|AA'\|}) \cdot \overrightarrow{crossB} +\\ \overrightarrow{crossA} \cdot (((\frac{1}{\|AB\|} * ((1,0,0) - \frac{\overrightarrow{BA}}{\|BA\|} * \frac{A_x - B_x}{\|AB\|})) \times \frac{\overrightarrow{BB'}}{\|BB'\|})))\end{aligned}$$

B.1.1 Derivation

The partial derivatives of $Energy_{ZE}$ with respect to the x, y, and z coordinates of A, A', B, and B' can be calculated directly from the formula and appear below. It is important to note that our formulation satisfies the constraint of continuity because at the points where the function itself is potentially discontinuous[1], i.e. when A' or B' is on the dividing plane, both the function and the derivative are zero. For the purposes of this analysis, we are only considering the derivative at points where the $Energy_{ZE} > 0$.

The derivative of $Energy_{ZE}$ with respect to the x,y, and z coordinates of A,B,A', and B' is as follows. Let $DP = (\overrightarrow{CROSSA} \cdot \overrightarrow{CROSSB})$. The derivative $\frac{\partial Energy_{ZE}}{\partial DP} = 2K * DP * \partial DP$ is a common multiplier for all component derivatives. Then,

$$\frac{\partial}{\partial A'} DP = [\frac{\partial}{\partial A'}(\overrightarrow{CROSSA}) \cdot \overrightarrow{CROSSB}] + [\overrightarrow{CROSSA} \cdot \frac{\partial}{\partial A'}(\overrightarrow{CROSSB})]$$

Derivatives for A' and B'

Since A' does not appear in $\overrightarrow{CROSSB}$, we know that the right bracketed term is 0 and

$$\frac{\partial}{\partial A'} DP = [\frac{\partial}{\partial A'}(\overrightarrow{CROSSA}) \cdot \overrightarrow{CROSSB}]$$

Expanding the derivative of the normalized $\overrightarrow{AA'}$ vector we get that

$$\frac{\partial}{\partial A'}(\frac{\overrightarrow{AA'}}{\|AA'\|}) = \overrightarrow{AA'} \cdot \frac{\partial}{\partial A'}(\frac{1}{\|AA'\|}) + \frac{1}{\|AA'\|}\frac{\partial}{\partial A'}(\overrightarrow{AA'})$$

$$= \overrightarrow{AA'} \cdot (\frac{-1 * \frac{\partial}{\partial A'}(\|AA'\|)}{\|AA'\|^2}) + \frac{1}{\|AA'\|}\frac{\partial}{\partial A'}(\overrightarrow{AA'})$$

$$= \frac{1}{\|AA'\|}(\frac{\partial}{\partial A'}(\overrightarrow{AA'}) - \frac{\overrightarrow{AA'}}{\|AA'\|} * \frac{\partial}{\partial A'}(\|AA'\|))$$

At this point, we need to consider the x,y,and z components individually.

$$\frac{\partial}{\partial A'_x}(\overrightarrow{AA'}) = (1,0,0)$$

[1] The points where $\overrightarrow{CROSSA}$ and $\overrightarrow{CROSSB}$ are perpendicular. The function and derivative are not actually discontinuous here.

$$\frac{\partial}{\partial A'_x}(\|AA'\|) = \frac{\partial}{\partial A'_x}((A'_x - A_x)^2 + (A'_y - A_y)^2 + (A'_z - A_z)^2)^{\frac{1}{2}}$$
$$= \frac{1}{2\|AA'\|} * 2(A'_x - A_x) = \frac{A'_x - A_x}{\|AA'\|}$$

The derivatives with respect to y and z follow naturally. In total,

$$\frac{\partial}{\partial A'_x}DP = [\frac{\overrightarrow{AB}}{\|AB\|} \times [\frac{1}{\|AA'\|}((1,0,0) - \frac{\overrightarrow{AA'}}{\|AA'\|} \cdot \frac{A'_x - A_x}{\|AA'\|})]] \cdot [\frac{\overrightarrow{BA}}{\|BA\|} \times \frac{\overrightarrow{BB'}}{\|BB'\|}]$$

Similarly, for B'

$$\frac{\partial}{\partial B'}DP = \overrightarrow{CROSSA} \cdot \frac{\partial}{\partial B'}(\overrightarrow{CROSSB})$$
$$\frac{\partial}{\partial B'}(\overrightarrow{CROSSB}) = \frac{\overrightarrow{BA}}{\|BA\|} \times \frac{\partial}{\partial B'}(\frac{\overrightarrow{BB'}}{\|BB'\|})$$
$$\frac{\partial}{\partial B'}(\frac{\overrightarrow{BB'}}{\|BB'\|}) = \frac{1}{\|BB'\|}(\frac{\partial}{\partial B'}(\overrightarrow{BB'}) - \frac{\overrightarrow{BB'}}{\|BB'\|} * \frac{\partial}{\partial B'}(\|BB'\|))$$

Derivatives for A and B

With the atoms adjacent to the π-bond, neither term of the dot product derivative is reduced to zero.

$$\frac{\partial}{\partial A}DP = [\frac{\partial}{\partial A}(\overrightarrow{CROSSA}) \cdot \overrightarrow{CROSSB}] + [\overrightarrow{CROSSA} \cdot \frac{\partial}{\partial A}(\overrightarrow{CROSSB})]$$

Following the other derivatives, we get that

$$\frac{\partial}{\partial A}(\overrightarrow{CROSSB}) = \frac{\partial}{\partial A}(\frac{\overrightarrow{BA}}{\|BA\|}) \times \frac{\overrightarrow{BB'}}{\|BB'\|}$$
$$\frac{\partial}{\partial A}(\frac{\overrightarrow{BA}}{\|BA\|}) = \frac{1}{\|BA\|}(\frac{\partial}{\partial A}(\overrightarrow{BA}) - \frac{\overrightarrow{BA}}{\|BA\|} * \frac{\partial}{\partial A}(\|BA\|))$$
$$\frac{\partial}{\partial A_x}(\overrightarrow{BA}) = (1,0,0)$$
$$\frac{\partial}{\partial A_x}(\|BA\|) = \frac{A_x - B_x}{\|BA\|}$$
$$\frac{\partial}{\partial A}(\overrightarrow{CROSSA}) = \frac{\overrightarrow{AB}}{\|AB\|} \times \frac{\partial}{\partial A}(\frac{\overrightarrow{AA'}}{\|AA'\|}) + \frac{\partial}{\partial A}(\frac{\overrightarrow{AB}}{\|AB\|}) \times \frac{\overrightarrow{AA'}}{\|AA'\|}$$
$$\frac{\partial}{\partial A}(\frac{\overrightarrow{AB}}{\|AB\|}) = \frac{1}{\|AB\|}(\frac{\partial}{\partial A}(\overrightarrow{AB}) - \frac{\overrightarrow{AB}}{\|AB\|} * \frac{\partial}{\partial A}(\|AB\|))$$
$$\frac{\partial}{\partial A_x}(\overrightarrow{AB}) = (-1,0,0)$$
$$\frac{\partial}{\partial A_x}(\|AB\|) = \frac{B_x - A_x}{\|AB\|}$$
$$\frac{\partial}{\partial A}(\frac{\overrightarrow{AA'}}{\|AA'\|}) = \frac{1}{\|AA'\|}(\frac{\partial}{\partial A}(\overrightarrow{AA'}) - \frac{\overrightarrow{AA'}}{\|AA'\|} * \frac{\partial}{\partial A}(\|AA'\|))$$

$$\frac{\partial}{\partial A_x}(\overrightarrow{AA'}) = (-1,0,0)$$

$$\frac{\partial}{\partial A_x}(\|AA'\|) = \frac{A'_x - A_x}{\|AA'\|}$$

Similarly, for B

$$\frac{\partial}{\partial B}DP = [\frac{\partial}{\partial B}(\overrightarrow{CROSSB}) \cdot \overrightarrow{CROSSA}] + [\overrightarrow{CROSSB} \cdot \frac{\partial}{\partial B}(\overrightarrow{CROSSA})]$$

$$\frac{\partial}{\partial B}(\overrightarrow{CROSSA}) = \frac{\partial}{\partial B}(\frac{\overrightarrow{AB}}{\|AB\|}) \times \frac{\overrightarrow{AA'}}{\|AA'\|}$$

$$\frac{\partial}{\partial B}(\frac{\overrightarrow{AB}}{\|AB\|}) = \frac{1}{\|AB\|}(\frac{\partial}{\partial B}(\overrightarrow{AB}) - \frac{\overrightarrow{AB}}{\|AB\|} * \frac{\partial}{\partial B}(\|AB\|))$$

$$\frac{\partial}{\partial B}(\overrightarrow{CROSSB}) = \frac{\overrightarrow{BA}}{\|BA\|} \times \frac{\partial}{\partial B}(\frac{\overrightarrow{BB'}}{\|BB'\|}) + \frac{\partial}{\partial B}(\frac{\overrightarrow{BA}}{\|BA\|}) \times \frac{\overrightarrow{BB'}}{\|BB'\|}$$

$$\frac{\partial}{\partial B}(\frac{\overrightarrow{BA}}{\|BA\|}) = \frac{1}{\|BA\|}(\frac{\partial}{\partial B}(\overrightarrow{BA}) - \frac{\overrightarrow{BA}}{\|BA\|} * \frac{\partial}{\partial B}(\|BA\|))$$

$$\frac{\partial}{\partial B}(\frac{\overrightarrow{BB'}}{\|BB'\|}) = \frac{1}{\|BB'\|}(\frac{\partial}{\partial B}(\overrightarrow{BB'}) - \frac{\overrightarrow{BB'}}{\|BB'\|} * \frac{\partial}{\partial B}(\|BB'\|))$$

$$\frac{\partial}{\partial B_x}(\overrightarrow{BB'}) = (-1,0,0)$$

$$\frac{\partial}{\partial B_x}(\|BB'\|) = \frac{-B'_x + B_x}{\|BB'\|}$$

$$\frac{\partial}{\partial B_x}(\overrightarrow{BA}) = (-1,0,0)$$

$$\frac{\partial}{\partial B_x}(\|BA\|) = \frac{A_x - B_x}{\|BA\|}$$

B.2 R/S Stereochemistry Term Partial Derivatives

$$M = 2 * K_{RS} * ((\frac{\overrightarrow{AA'}}{\|AA'\|} \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|})$$

$$\frac{\partial Energy_{RS}}{\partial A_x} = -M * ((\frac{1}{\|AA'\|} * ((1,0,0) - \frac{\overrightarrow{AA'}}{\|AA'\|} * (\frac{A'_x - A_x}{\|AA'\|})) \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|})$$

$$\frac{\partial Energy_{RS}}{\partial A'_x} = M * ((\frac{1}{\|AA'\|} * ((1,0,0) - \frac{\overrightarrow{AA'}}{\|AA'\|} * (\frac{A'_x - A_x}{\|AA'\|})) \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|})$$

$$\frac{\partial Energy_{RS}}{\partial B_x} = -M * (\frac{\overrightarrow{AA'}}{\|AA'\|} \times (\frac{1}{\|BB'\|} * ((1,0,0) - \frac{\overrightarrow{BB'}}{\|BB'\|} * (\frac{B'_x - B_x}{\|BB'\|}))) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|})$$

$$\frac{\partial Energy_{RS}}{\partial B'_x} = M * (\frac{\overrightarrow{AA'}}{\|AA'\|} \times (\frac{1}{\|BB'\|} * ((1,0,0) - \frac{\overrightarrow{BB'}}{\|BB'\|} * (\frac{B'_x - B_x}{\|BB'\|}))) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|})$$

$$\frac{\partial Energy_{RS}}{\partial C_x} = M * (\frac{\overrightarrow{AA'}}{\|AA'\|} \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot (\frac{1}{\|CC'\|} * ((-1,0,0) - \frac{\overrightarrow{CC'}}{\|CC'\|} * \frac{-C'_x + C_x}{\|CC'\|}))$$

$$\frac{\partial Energy_{RS}}{\partial C'_x} = -M * (\frac{\overrightarrow{AA'}}{\|AA'\|} \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot (\frac{1}{\|CC'\|} * ((-1,0,0) - \frac{\overrightarrow{CC'}}{\|CC'\|} * \frac{-C'_x + C_x}{\|CC'\|}))$$

B.2.1 Derivation

As the R/S stereochemistry term has many terms in common with the Z/E term, we can use much of the previous derivation again.

Once again we will define the derivatives in terms of the dot product. Let $DP = (\overrightarrow{CROSS} \cdot \frac{\overrightarrow{CC'}}{\|CC'\|})$ so that $\frac{\partial Energy_{RS}}{\partial DP} = 2K * DP * \partial DP = M$ – the common multiplier for all component derivatives which we shall call M.

Derivatives for C and C'

We will solve for C and C' first. Given that C appears only on one side of the the dot product,

$$\frac{\partial DP}{\partial C} = [\frac{\partial}{\partial C}(\overrightarrow{CROSS}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|}] + [\overrightarrow{CROSS} \cdot \frac{\partial}{\partial C}(\frac{\overrightarrow{CC'}}{\|CC'\|})]$$

$$\frac{\partial DP}{\partial C} = [\overrightarrow{CROSS} \cdot \frac{\partial}{\partial C}(\frac{\overrightarrow{CC'}}{\|CC'\|})]$$

By using the following derivative we determined in the calculation of the Z/E derivatives...

$$\frac{\partial}{\partial C}(\frac{\overrightarrow{CC'}}{\|CC'\|}) = \frac{1}{\|CC'\|}(\frac{\partial}{\partial C}(\overrightarrow{CC'}) - \frac{\overrightarrow{CC'}}{\|CC'\|} * \frac{\partial}{\partial C}(\|CC'\|))$$

$$\frac{\partial}{\partial C_x}(\overrightarrow{CC'}) = (-1,0,0)$$

$$\frac{\partial}{\partial C_x}(\|CC'\|) = \frac{-C'_x + C_x}{\|CC'\|}$$

...we can find the full derivation for C and C', namely that

$$\frac{\partial Energy_{RS}}{\partial C} = M * [\overrightarrow{CROSS} \cdot \frac{\partial}{\partial C}(\frac{\overrightarrow{CC'}}{\|CC'\|})]$$

$$\frac{\partial Energy_{RS}}{\partial C} = M * [\overrightarrow{CROSS} \cdot (\frac{1}{\|CC'\|}(\frac{\partial}{\partial C}(\overrightarrow{CC'}) - \frac{\overrightarrow{CC'}}{\|CC'\|} * \frac{\partial}{\partial C}(\|CC'\|)))]$$

$$\frac{\partial Energy_{RS}}{\partial C_x} = M * [\overrightarrow{CROSS} \cdot (\frac{1}{\|CC'\|}((-1,0,0)) - \frac{\overrightarrow{CC'}}{\|CC'\|} * \frac{-C'_x + C_x}{\|CC'\|}))]$$

C' is then just the opposite of this.

$$\frac{\partial Energy_{RS}}{\partial C'} = -\frac{\partial Energy_{RS}}{\partial C}$$

Derivatives for A and A'

Exploring the other side of the derivative of DP, we see that

$$\frac{\partial DP}{\partial A'} = [\frac{\partial}{\partial A'}(\overrightarrow{CROSS}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|}] + [\overrightarrow{CROSS} \cdot \frac{\partial}{\partial A'}(\frac{\overrightarrow{CC'}}{\|CC'\|})]$$

$$\frac{\partial DP}{\partial A'} = \frac{\partial}{\partial A'}(\overrightarrow{CROSS}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|}$$

$$\frac{\partial DP}{\partial A'} = \frac{\partial}{\partial A'}(\frac{\overrightarrow{AA'}}{\|AA'\|} \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|}$$

$$\frac{\partial DP}{\partial A'} = (\frac{\partial}{\partial A'}(\frac{\overrightarrow{AA'}}{\|AA'\|}) \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|}$$

Once again, we apply derivations we found in the Z/E derivatives:

$$\frac{\partial}{\partial A'}(\frac{\overrightarrow{AA'}}{\|AA'\|}) = \overrightarrow{AA'} \cdot \frac{\partial}{\partial A'}(\frac{1}{\|AA'\|}) + \frac{1}{\|AA'\|}\frac{\partial}{\partial A'}(\overrightarrow{AA'})$$

$$= \overrightarrow{AA'} \cdot (\frac{-1 * \frac{\partial}{\partial A'}(\|AA'\|)}{\|AA'\|^2}) + \frac{1}{\|AA'\|}\frac{\partial}{\partial A'}(\overrightarrow{AA'})$$

$$= \frac{1}{\|AA'\|}(\frac{\partial}{\partial A'}(\overrightarrow{AA'}) - \frac{\overrightarrow{AA'}}{\|AA'\|} * \frac{\partial}{\partial A'}(\|AA'\|))$$

and

$$\frac{\partial}{\partial A'_x}(\overrightarrow{AA'}) = (1,0,0)$$

$$\frac{\partial}{\partial A'_x}(\|AA'\|) = \frac{A'_x - A_x}{\|AA'\|}$$

to get

$$\frac{\partial DP}{\partial A'} = ((\frac{1}{\|AA'\|}(\frac{\partial}{\partial A'}(\overrightarrow{AA'}) - \frac{\overrightarrow{AA'}}{\|AA'\|} * \frac{\partial}{\partial A'}(\|AA'\|))) \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|}$$

$$\frac{\partial DP}{\partial A'_x} = ((\frac{1}{\|AA'\|}((1,0,0) - \frac{\overrightarrow{AA'}}{\|AA'\|} * \frac{A'_x - A_x}{\|AA'\|})) \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|}$$

$$\frac{\partial Energy_{RS}}{\partial A'_x} = M * ((\frac{1}{\|AA'\|}((1,0,0) - \frac{\overrightarrow{AA'}}{\|AA'\|} * \frac{A'_x - A_x}{\|AA'\|})) \times \frac{\overrightarrow{BB'}}{\|BB'\|}) \cdot \frac{\overrightarrow{CC'}}{\|CC'\|}$$

In the same way that the derivatives of C and C' were opposites of each other, here

$$\frac{\partial Energy_{RS}}{\partial A'} = -\frac{\partial Energy_{RS}}{\partial A}$$

Derivatives for B and B'

The derivatives for B and B' are found by performing the same derivations as that for A and A'.

B.3 Planar Perspective Term Partial Derivatives

$$\frac{\partial Energy_{PP}}{\partial \theta} = 2K_{PP} * \cos\Delta_\theta * \sin\Delta_\theta * \frac{\pi}{2*(\pi - \theta_{Eq})}$$

$$\frac{\partial Energy_{PP}}{\partial G} = \frac{\partial Energy_{PP}}{\partial \theta} * \frac{\|\overrightarrow{DR}\|}{(\overrightarrow{RS} \times \overrightarrow{DR})^2}(\overrightarrow{DG} \times \overrightarrow{DR})$$

$$\frac{\partial Energy_{PP}}{\partial D} = \begin{array}{c} \frac{\frac{\partial Energy_{PP}}{\partial \theta}}{} * (((\frac{\overrightarrow{DG}\cdot\overrightarrow{DR}}{\|\overrightarrow{DG}\times\overrightarrow{DR}\|^2 * \|\overrightarrow{DR}\|} - \frac{\|(\overrightarrow{DR})\|}{\|\overrightarrow{DG}\times\overrightarrow{DR}\|^2})* \\ (\overrightarrow{DG} \times \overrightarrow{DR})) - (\frac{\overrightarrow{RS}\cdot\overrightarrow{DR}}{\|\overrightarrow{RS}\times\overrightarrow{DR}\|^2 * \|\overrightarrow{DR}\|} * (\overrightarrow{RS} \times \overrightarrow{DR}))) \end{array}$$

$$\frac{\partial Energy_{PP}}{\partial R} = \begin{array}{c} \frac{\partial Energy_{PP}}{\partial \theta} * (((\frac{\|(\overrightarrow{DR})\|}{\|\overrightarrow{RS}\times\overrightarrow{DR}\|^2} + \frac{\overrightarrow{RS}\cdot\overrightarrow{DR}}{\|\overrightarrow{RS}\times\overrightarrow{DR}\|^2 * \|\overrightarrow{DR}\|})* \\ (\overrightarrow{RS} \times \overrightarrow{DR})) - (\frac{\overrightarrow{DG}\cdot\overrightarrow{DR}}{\|\overrightarrow{DG}\times\overrightarrow{DR}\|^2 * \|\overrightarrow{DR}\|} * (\overrightarrow{DG} \times \overrightarrow{DR}))) \end{array}$$

$$\frac{\partial R}{\partial C_x} = \begin{pmatrix} -(\|BC\|^2 - ((C_x - B_x)^2)), \\ ((C_x - B_x) * (C_y - B_y)), \\ ((C_x - B_x) * (C_z - B_z)) \end{pmatrix} * \frac{-1}{\|BC\|^3}$$

$$\frac{\partial R}{\partial B_x} = -\frac{\partial R}{\partial C_x}$$

$$\frac{\partial Energy_{PP}}{\partial S} = -\frac{\partial Energy_{PP}}{\partial \theta} * \frac{\|\overrightarrow{DR}\|}{\|(\overrightarrow{RS} \times \overrightarrow{DR})\|^2} * (\overrightarrow{RS} \times \overrightarrow{DR})$$

$$\frac{\partial S}{\partial A_x} = \begin{pmatrix} -(\|A\frac{B+C}{2}\|^2 - (((\frac{B_x+C_x}{2}) - A_x)^2)), \\ ((\frac{B_x+C_x}{2} - A_x) * (\frac{B_y+C_y}{2} - A_y)), \\ ((\frac{B_x+C_x}{2} - A_x) * (\frac{B_z+C_z}{2})) \end{pmatrix} * \frac{1}{\|A\frac{B+C}{2}\|^3}$$

$$\frac{\partial Energy_{PP}}{\partial A_x} = \frac{\partial Energy_{PP}}{\partial S} \cdot \frac{\partial S}{\partial A_x}$$

$$\frac{\partial S}{\partial C_x} = \frac{\partial R}{\partial C_x} + \frac{\partial S}{\partial A_x} * -0.5$$

$$\frac{\partial S}{\partial B_x} = \frac{\partial R}{\partial B_x} + \frac{\partial S}{\partial A_x} * -0.5$$

$$\frac{\partial Energy_{PP}}{\partial C_x} = \frac{\partial Energy_{PP}}{\partial R} \cdot \frac{\partial R}{\partial C_x} + \frac{\partial Energy_{PP}}{\partial S} \cdot \frac{\partial S}{\partial C_x}$$

$$\frac{\partial Energy_{PP}}{\partial B_x} = \frac{\partial Energy_{PP}}{\partial R} \cdot \frac{\partial R}{\partial B_x} + \frac{\partial Energy_{PP}}{\partial S} \cdot \frac{\partial S}{\partial B_x}$$

B.3.1 Derivation

The above formulation shows much of the process of the derivation. Due to the complexity of the formula for the individual terms, we found it easier to show some of the partial steps, such as calculating $\frac{\partial Energy_P}{\partial A_x}$ as a product of $\frac{\partial Energy_P}{\partial S}$ and $\frac{\partial S}{\partial A_x}$.

To find the constant multiplier for all derivatives, we take from the derivative of $\cos^2$,

$$M = \frac{\partial Energy_{PP}}{\partial \theta} = 2K_{PP} * \cos\Delta_{\theta} * \sin\Delta_{\theta} * \frac{\pi}{2 * (\pi - \theta_{Eq})}$$

Our formulation is very similar to the molecular mechanics torsion term be design. We know from molecular mechanics work [11] that the derivatives with respect to θ of the torsion $\angle GDRS$ are:

$$\frac{\partial \theta}{\partial G} = \frac{\|\overrightarrow{DR}\|}{(\overrightarrow{RS} \times \overrightarrow{DR})^2}(\overrightarrow{DG} \times \overrightarrow{DR})$$

$$\frac{\partial \theta}{\partial D} = -\frac{\|\overrightarrow{DR}\|}{(\overrightarrow{DG} \times \overrightarrow{DR})^2}(\overrightarrow{DG} \times \overrightarrow{DR}) + \frac{\overrightarrow{DG} \cdot \overrightarrow{DR}}{(\overrightarrow{DG} \times \overrightarrow{DR})^2\|\overrightarrow{DR}\|}(\overrightarrow{DG} \times \overrightarrow{DR}) - \frac{\overrightarrow{RS} \cdot \overrightarrow{DR}}{(\overrightarrow{RS} \times \overrightarrow{DR})^2\|\overrightarrow{DR}\|}(\overrightarrow{RS} \times \overrightarrow{DR})$$

$$\frac{\partial \theta}{\partial R} = (((\frac{\|(\overrightarrow{DR})\|}{\|\overrightarrow{RS} \times \overrightarrow{DR}\|^2} + \frac{\overrightarrow{RS} \cdot \overrightarrow{DR}}{\|\overrightarrow{RS} \times \overrightarrow{DR}\|^2 * \|\overrightarrow{DR}\|}) * (\overrightarrow{RS} \times \overrightarrow{DR})) - (\frac{\overrightarrow{DG} \cdot \overrightarrow{DR}}{\|\overrightarrow{DG} \times \overrightarrow{DR}\|^2 * \|\overrightarrow{DR}\|} * (\overrightarrow{DG} \times \overrightarrow{DR})))$$

$$\frac{\partial \theta}{\partial S} = -\frac{\|\overrightarrow{DR}\|}{\|(\overrightarrow{RS} \times \overrightarrow{DR})\|^2} * (\overrightarrow{RS} \times \overrightarrow{DR})$$

Which immediately give us $\frac{\partial Energy_{PP}}{\partial G}$ and $\frac{\partial Energy_{PP}}{\partial D}$. However, the pseudo points R and S are defined in terms of the real points A,B,C,G, and D. Therefore, we need to find the derivatives for A,B, and C.

We start with our definitions of the points R and S.

$$R = D + \frac{\overrightarrow{BC}}{\|BC\|}$$

$$S = D + \frac{\overrightarrow{BC}}{\|BC\|} + \frac{\overrightarrow{A\frac{B+C}{2}}}{\|A\frac{B+C}{2}\|}$$

The derivatives of the components will be the sum of the derivatives with respect to the terms they comprise. As B and C appear in both, but A only appears in the definition of S, we know that

$$\frac{\partial Energy_{PP}}{\partial A_x} = \frac{\partial Energy_{PP}}{\partial S} \cdot \frac{\partial S}{\partial A_x}$$

$$\frac{\partial Energy_{PP}}{\partial C_x} = \frac{\partial Energy_{PP}}{\partial R} \cdot \frac{\partial R}{\partial C_x} + \frac{\partial Energy_{PP}}{\partial S} \cdot \frac{\partial S}{\partial C_x}$$

$$\frac{\partial Energy_{PP}}{\partial B_x} = \frac{\partial Energy_{PP}}{\partial R} \cdot \frac{\partial R}{\partial B_x} + \frac{\partial Energy_{PP}}{\partial S} \cdot \frac{\partial S}{\partial B_x}$$

Note that here each of the multiplications is between vectors such as $\frac{\partial Energy_{PP}}{\partial R}$ and $\frac{\partial R}{\partial C_x}$. Using the generalized chain rule for taking derivatives [8], this multiplication is performed using the dot product of the vectors. Also, since these are vectors, to calculate $\frac{\partial R}{\partial C_x}$, we need to calculate $\frac{\partial R_x}{\partial C_x}$, $\frac{\partial R_y}{\partial C_x}$, and $\frac{\partial R_z}{\partial C_x}$.

For the first of these (the x component):

$$R_x = D_x + \frac{C_x - B_x}{\|BC\|}$$

$$\frac{\partial R_x}{\partial C_x} = \frac{\partial}{\partial C_x}\frac{C_x}{\|BC\|} - \frac{\partial}{\partial C_x}\frac{B_x}{\|BC\|}$$

By the quotient rule,

$$\frac{\partial R_x}{\partial C_x} = \frac{(\|BC\| - C_x * \frac{\partial}{\partial C_x}\|BC\|)}{\|BC\|^2} - \frac{(-B_x * \frac{\partial}{\partial C_x}\|BC\|)}{\|BC\|^2}$$

$$= \frac{1}{\|BC\|^2} * ((\|BC\| - C_x * \frac{\partial}{\partial C_x}\|BC\|) + (B_x * \frac{\partial}{\partial C_x}\|BC\|))$$

Going to the definition of $\|BC\|$,

$$\frac{\partial}{\partial C_x}\|BC\| = \frac{\partial}{\partial C_x}((C_x - B_x)^2 + (C_y - B_y)^2 + (C_z - B_z)^2)^{\frac{1}{2}}$$

$$= \frac{1}{2\|BC\|} * (C_x - B_x) * 2 = \frac{(C_x - B_x)}{\|BC\|}$$

Plugging this in to the previous equation, we have that

$$\frac{\partial R_x}{\partial C_x} = \frac{1}{\|BC\|^2} * ((\|BC\| - C_x * \frac{(C_x - B_x)}{\|BC\|}) + (B_x * \frac{(C_x - B_x)}{\|BC\|}))$$

Factor out $\frac{-1}{\|BC\|}$ and

$$\frac{\partial R_x}{\partial C_x} = \frac{-1}{\|BC\|^3} * -1 * (\|BC\|^2 - (C_x * (C_x - B_x) + B_x * (C_x - B_x)))$$

$$\frac{\partial R_x}{\partial C_x} = \frac{-1}{\|BC\|^3} * -(\|BC\|^2 + (C_x - B_x)(-C_x + B_x)))$$

$$\frac{\partial R_x}{\partial C_x} = \frac{-1}{\|BC\|^3} * -(\|BC\|^2 - (C_x - B_x)^2))$$

Similarly for the second (y) component $\frac{\partial R_y}{\partial C_x}$

$$\frac{\partial R_y}{\partial C_x} = \frac{\partial}{\partial C_x}(\frac{C_y}{\|BC\|}) - \frac{\partial}{\partial C_x}(\frac{B_y}{\|BC\|})$$

Apply the quotient rule to get the final result.

$$\frac{\partial R_y}{\partial C_x} = (\frac{-C_y * \frac{\partial}{\partial C_x}(\|BC\|)}{\|BC\|^2}) - (\frac{-B_y * \frac{\partial}{\partial C_x}(\|BC\|)^2}{\|BC\|})$$

$$= (\frac{-C_y * (C_x - B_x)}{\|BC\|^3}) - (-\frac{B_y * (C_x - B_x)^3}{\|BC\|})$$

$$= \frac{1}{\|BC\|^3}((C_x - B_x) * (-C_y + B_y))$$

$$= \frac{-1}{\|BC\|^3}((C_x - B_x) * (C_y - B_y))$$

Continuing to apply this set of steps will yield the remaining partial derivatives for R and S.

B.4 Dihedral Perspective Term Partial Derivatives

These term formulae are taken directly from other equations. The gauche formula's derivative can be found earlier in this appendix under the Z/E Sterochemistry Term (Section B.1). The eclipsed formula is the standard molecular mechanics torsion term with $n = 1$. See [11] for the derivation.

Appendix C

Test Set and Results

C.1 Algorithm Performance Data

Tables C.1 to C.9 present the data used to compile the algorithm evaluation figures in Chapter 5 (Figures 5.4 - 5.6). For each of the nine algorithm variants, the test set in the next section was run six times. The mean of the performance measures appears in the Chapter 5 figures.

Table C.1: *Algorithm Performance Analysis: Full Algorithm Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	23.079427	23.453125	21.625	21.601562	23.046875	26.4375	22.3125
Median time to First Generated Answer	17.502604	14.765625	21.0625	19	19.625	14.78125	15.78125
Tests with Preferred Answer Found	62.666667	61	61	66	62	64	62
Tests Generating Multiple OK Answers	11.833333	11	10	13	8	14	15
Tests Producing No Answers	16.833333	21	17	13	17	15	18
Tests Failed to Errors	0.3333333	0	1	1	0	0	0

Table C.2: *Algorithm Performance Analysis: No R/S Stereochemistry Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	20.28646	18.49219	16.89063	23.29688	19.375	21.22656	22.4375
Median time to First Generated Answer	14.94792	18.875	12.76563	13.6875	13.39063	16.07813	14.89063
Tests with Preferred Answer Found	61.16667	63	62	63	61	60	58
Tests Generating Multiple OK Answers	13	16	13	13	11	12	13
Tests Producing No Answers	17.83333	18	16	16	17	21	19
Tests Failed to Errors	2.333333	2	2	3	2	2	3

Table C.3: *Algorithm Performance Analysis: No Z/E Stereochemistry Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	19.91667	18.25	17.875	18.95313	23.60938	22.14063	18.67188
Median time to First Generated Answer	14.54688	15.21875	11.79688	14.59375	14.95313	15.15625	15.5625
Tests with Preferred Answer Found	62.83333	59	63	63	64	64	64
Tests Generating Multiple OK Answers	11.16667	11	13	11	12	11	9
Tests Producing No Answers	15.66667	17	15	15	16	15	16
Tests Failed to Errors	4.833333	5	5	5	5	5	4

Table C.4: *Algorithm Performance Analysis: No Planar Perspective Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	10.36589	9.578125	10.4375	10.11719	11.21875	9.3125	11.53125
Median time to First Generated Answer	4.398438	4.4375	4.140625	4.53125	4.625	4.109375	4.546875
Tests with Preferred Answer Found	59.5	57	62	58	62	59	59
Tests Generating Multiple OK Answers	12.16667	14	12	13	11	12	11
Tests Producing No Answers	13.5	15	11	15	12	15	13
Tests Failed to Errors	0.166667	0	1	0	0	0	0

Table C.5: *Algorithm Performance Analysis: No Dihedral Perspective Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	23.26172	21.5	24.65625	23.23438	26.875	22.91406	20.39063
Median time to First Generated Answer	14.59896	12.54688	18.70313	16.375	13.28125	13.75	12.9375
Tests with Preferred Answer Found	46.16667	49	47	45	46	45	45
Tests Generating Multiple OK Answers	15.5	16	14	14	15	14	20
Tests Producing No Answers	16	14	12	19	16	18	17
Tests Failed to Errors	0.166667	0	0	0	1	0	0

Table C.6: *Algorithm Performance Analysis: No Search Mechanism Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	0.364583	0.359375	0.359375	0.328125	0.359375	0.390625	0.390625
Median time to First Generated Answer	0.348958	0.359375	0.328125	0.328125	0.34375	0.359375	0.375
Tests with Preferred Answer Found	41.83333	40	40	43	43	41	44
Tests Generating Multiple OK Answers	0	0	0	0	0	0	0
Tests Producing No Answers	42.83333	47	42	44	39	45	40
Tests Failed to Errors	0.166667	0	0	0	1	0	0

Table C.7: *Algorithm Performance Analysis: No Heuristics Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	26.36458	25.40625	30.28906	23.94531	26.35156	24.42188	27.77344
Median time to First Generated Answer	15.72656	17.09375	15.89063	14.90625	17.29688	12.53125	16.64063
Tests with Preferred Answer Found	61.66667	60	65	59	63	62	61
Tests Generating Multiple OK Answers	14.16667	12	17	15	14	14	13
Tests Producing No Answers	16.16667	18	13	18	16	13	19
Tests Failed to Errors	0	0	0	0	0	0	0

Table C.8: *Algorithm Performance Analysis: Best First Search Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	31.44401	31.5	32.55469	31.64063	29.875	35.41406	27.67969
Median time to First Generated Answer	18.48438	15.89063	18.42188	20.29688	16.9375	20.25	19.10938
Tests with Preferred Answer Found	61.5	63	60	64	61	63	58
Tests Generating Multiple OK Answers	8.333333	8	7	8	8	12	7
Tests Producing No Answers	17.83333	18	18	18	14	18	21
Tests Failed to Errors	0	0	0	0	0	0	0

Table C.9: *Algorithm Performance Analysis: No IM3 Terms Results*

Evaluation Metric	Mean of Runs	Run#1	Run#2	Run#3	Run#4	Run#5	Run#6
Median time to Generated Answer	10.24089	9.390625	10.32813	11.71875	9.71875	9.601563	10.6875
Median time to First Generated Answer	4.169271	3.28125	4.75	4.25	3.234375	4.09375	5.40625
Tests with Preferred Answer Found	42	40	41	42	43	44	42
Tests Generating Multiple OK Answers	14.16667	13	16	16	15	12	13
Tests Producing No Answers	14.5	15	17	14	13	11	17
Tests Failed to Errors	8.5	8	9	9	9	9	7

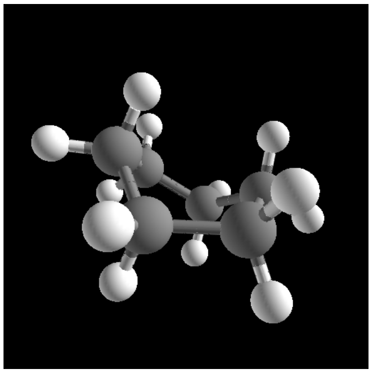
C.2 Algorithm Performance Test Set

The set of test diagrams used for the performance evaluation in Chapter 5 consisted of 103 molecule diagrams. For each diagram, there were one to five 3D reference models one would expect as likely interpretations of the diagram. For each reference model, there was an associated quality measure indicating how well the algorithm did if it produced a model matching that particular reference model. For instance, one model might be the preferred answer, while another would be an obvious error.

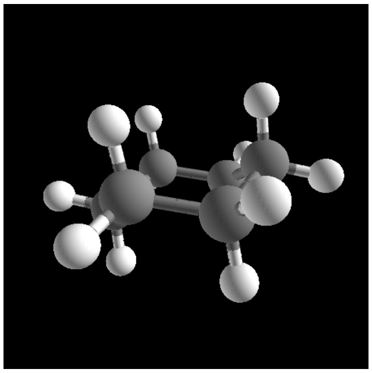
We give here the test set sorted into groups of models. Each group contains a set of diagrams and set of reference molecule models which each diagram output is compared against. This test set is derived from a set of molecule models used by Wang in the development and testing of GAFF [90]. As such, these groupings were determined by the grouping in the original set of models.

comp2

Models

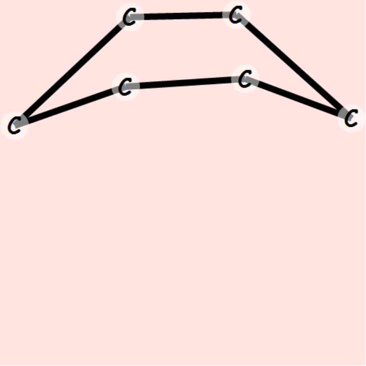


comp2_boat

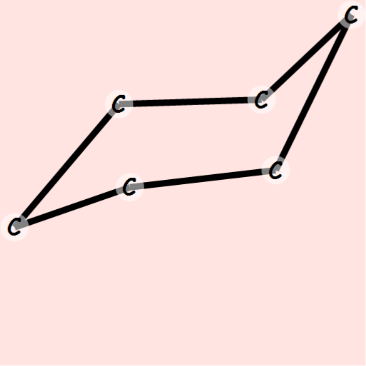


comp2_chair

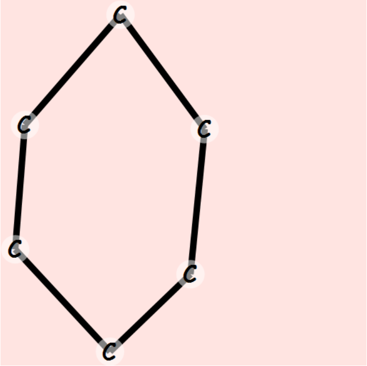
Tests



comp2_boat.cpms



comp2_chair.cpms

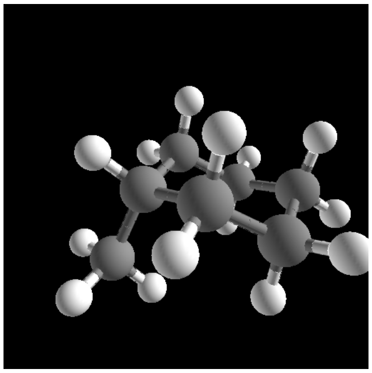


comp2_generic.cpms

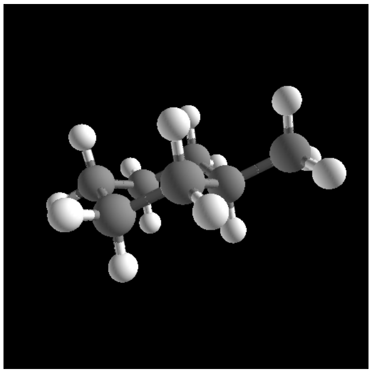
Diagram	comp2_boat	comp2_chair
comp2_boat.cpms	Preferred	Not Preferred
comp2_chair.cpms	Not Preferred	Preferred
comp2_generic.cpms	OK	OK

comp3

Models

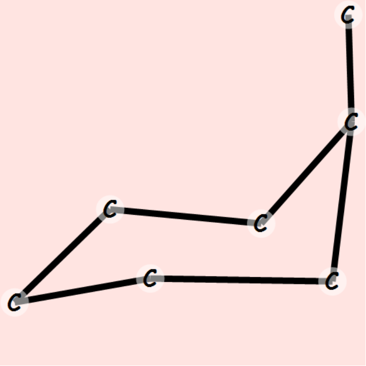


comp3_ax

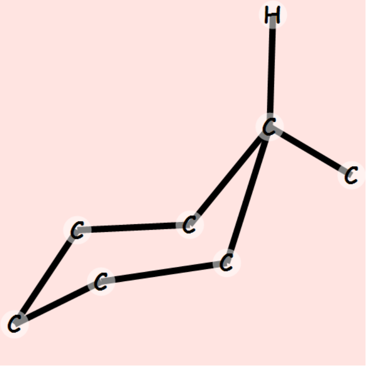


comp3_eq

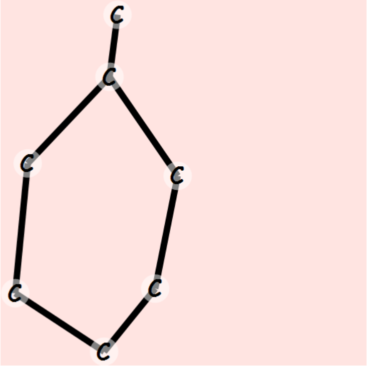
Tests



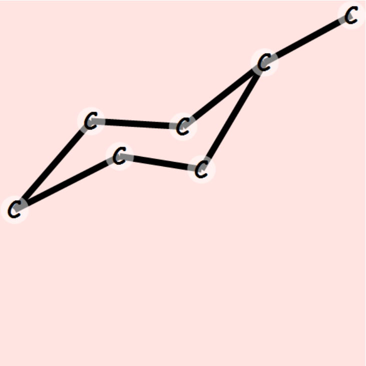
comp3_ax.cpms



comp3_eq.cpms



comp3_generic.cpms

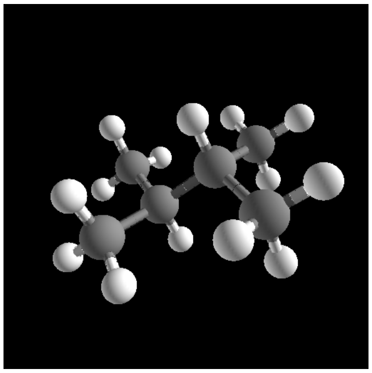


comp3_neither.cpms

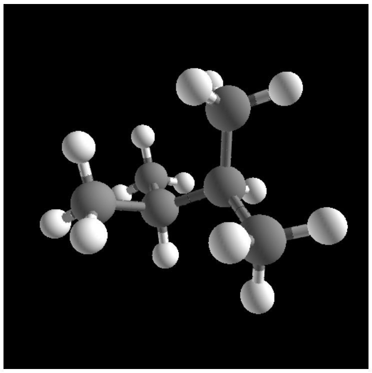
Diagram	comp3_ax	comp3_eq
comp3_ax.cpms	Preferred	Not Preferred
comp3_eq.cpms	Not Preferred	Preferred
comp3_generic.cpms	OK	OK
comp3_neither.cpms	Not Preferred	Not Preferred

comp4

Models

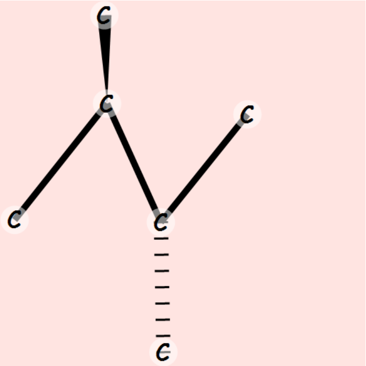


comp4_a

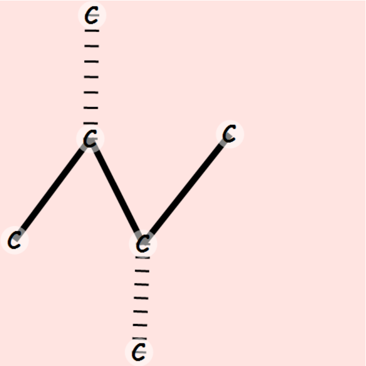


comp4_g

Tests



comp4_opposite.cpms

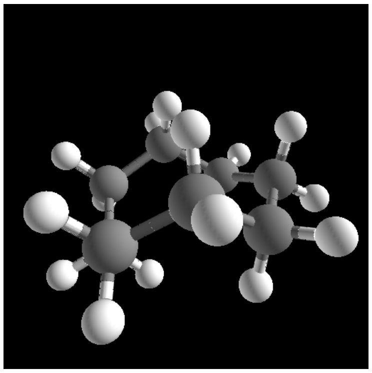


comp4_same.cpms

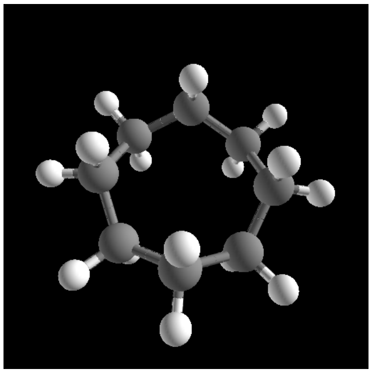
Diagram	comp4_a	comp4_g
comp4_opposite.cpms	Preferred	Not Preferred
comp4_same.cpms	Not Preferred	Preferred

comp5

Models

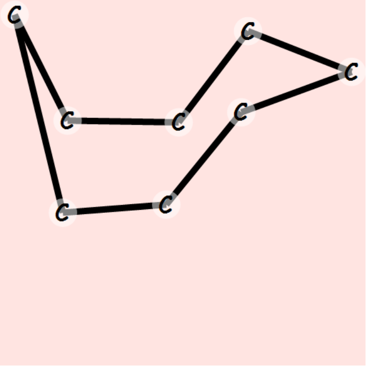


comp5_chair

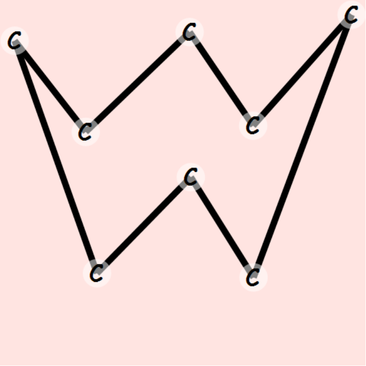


comp5_d4d

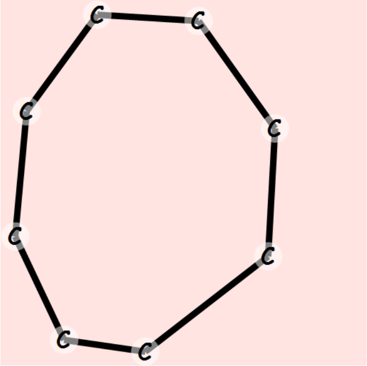
Tests



comp5_chair.cpms



comp5_d4d.cpms

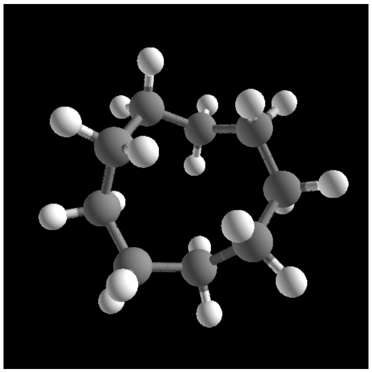


comp5_generic.cpms

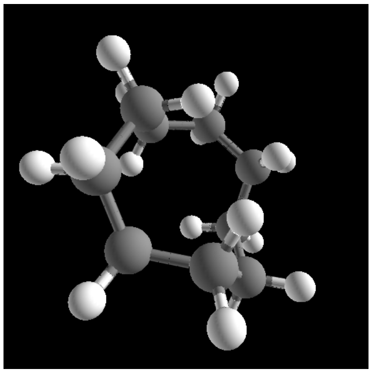
Diagram	comp5_chair	comp5_d4d
comp5_chair.cpms	Preferred	Not Preferred
comp5_d4d.cpms	Not Preferred	Preferred
comp5_generic.cpms	OK	OK

comp6

Models

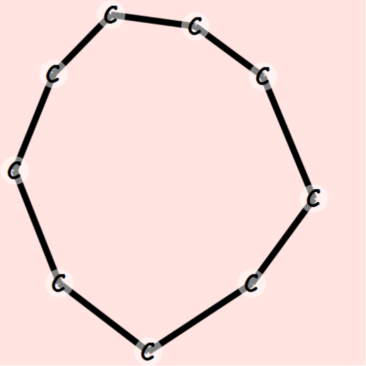


comp6_c2



comp6_d3

Tests

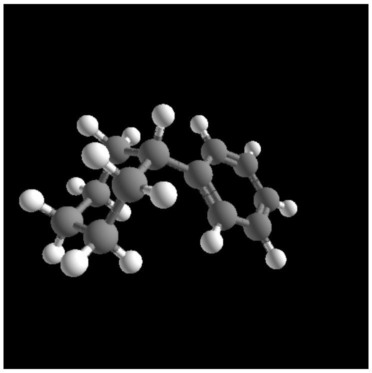


comp6_generic.cpms

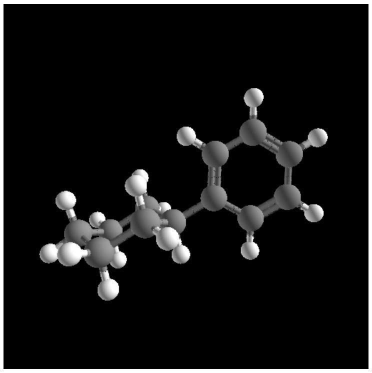
Diagram	comp6_c2	comp6_d3
comp6_generic.cpms	OK	OK

comp7

Models

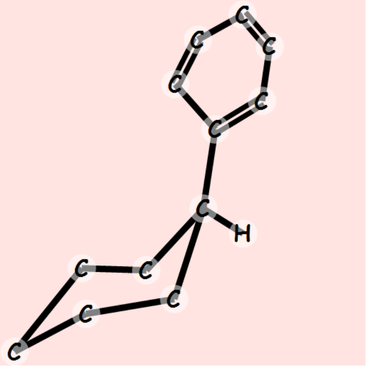


comp7_ax

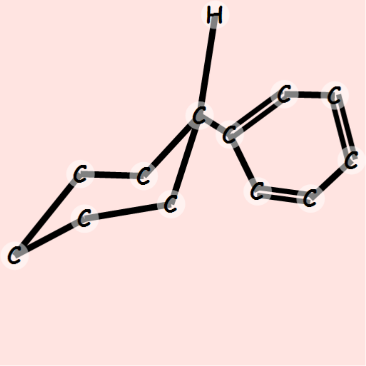


comp7_eq

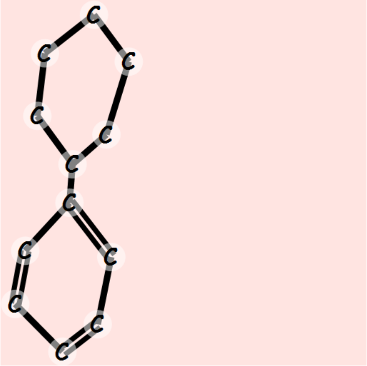
Tests



comp7_ax.cpms



comp7_eq.cpms

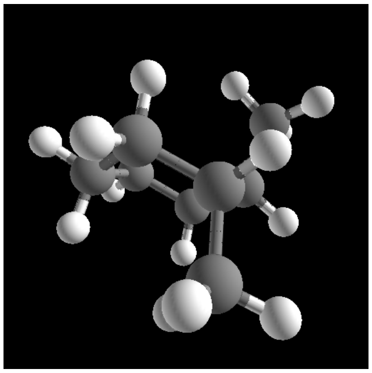


comp7_generic.cpms

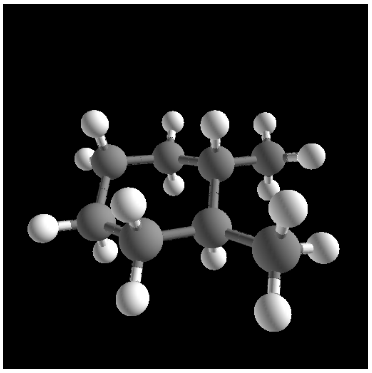
Diagram	comp7_ax	comp7_eq
comp7_ax.cpms	Preferred	Not Preferred
comp7_eq.cpms	Not Preferred	Preferred
comp7_generic.cpms	OK	OK

comp8

Models

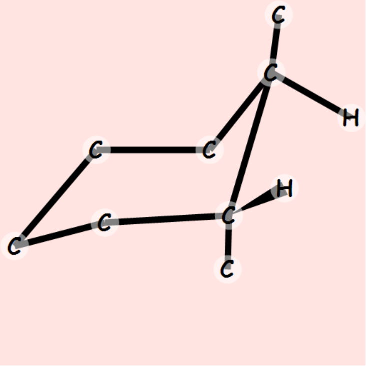


comp8_axax

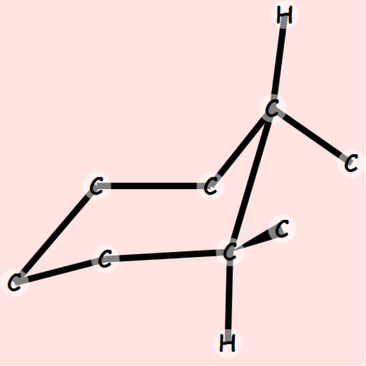


comp8_eqeq

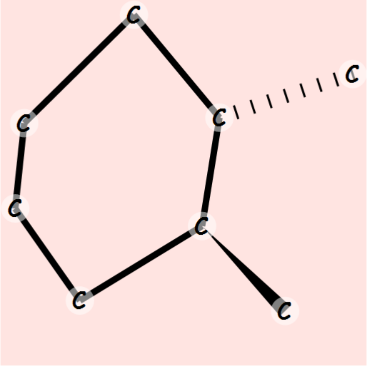
Tests



comp8_axax.cpms



comp8_eqeq.cpms

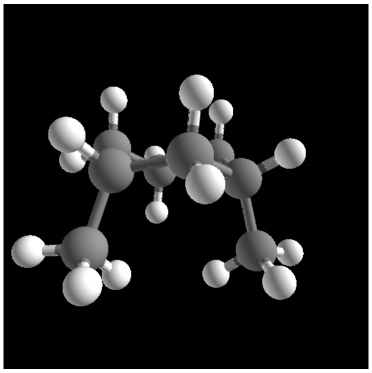


comp8_generic.cpms

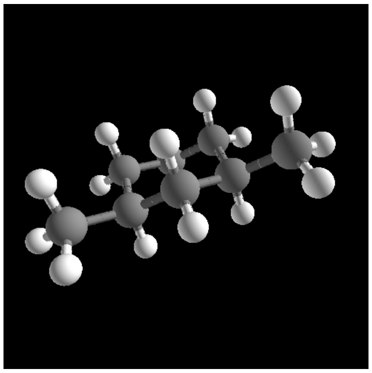
Diagram	comp8_axax	comp8_eqeq
comp8_axax.cpms	Preferred	Error
comp8_eqeq.cpms	Error	Preferred
comp8_generic.cpms	OK	Error

comp9

Models

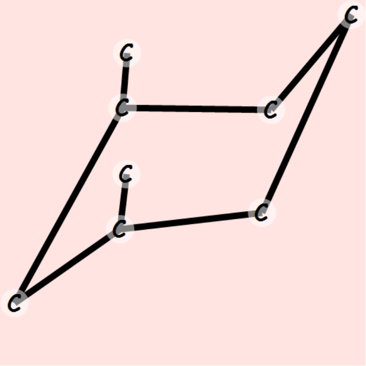


comp9_axax

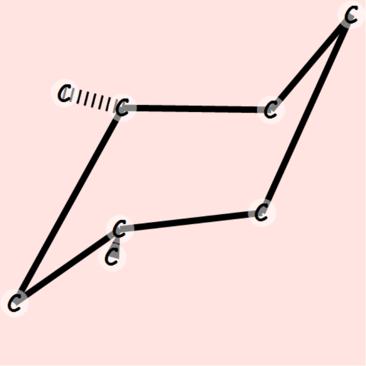


comp9_eqeq

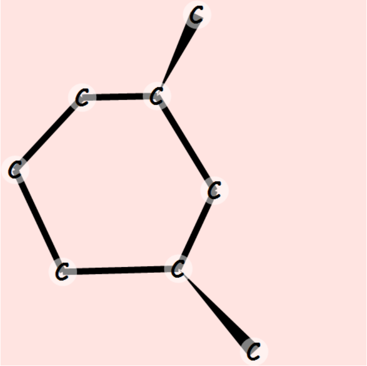
Tests



comp9_axax.cpms



comp9_eqeq.cpms

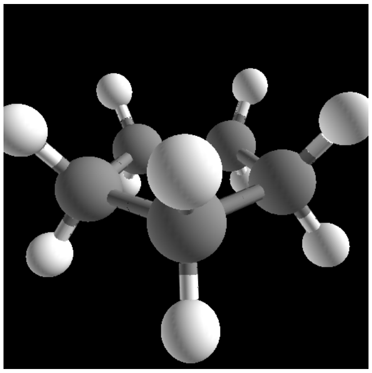


comp9_generic.cpms

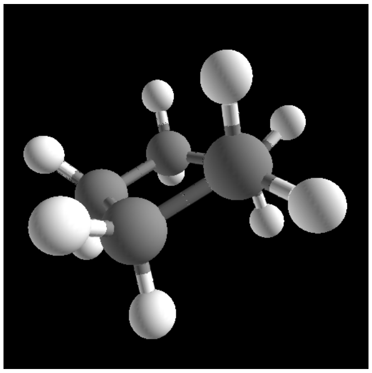
Diagram	comp9_axax	comp9_eqeq
comp9_axax.cpms	Preferred	Not Preferred
comp9_eqeq.cpms	Not Preferred	Preferred
comp9_generic.cpms	OK	OK

comp10

Models

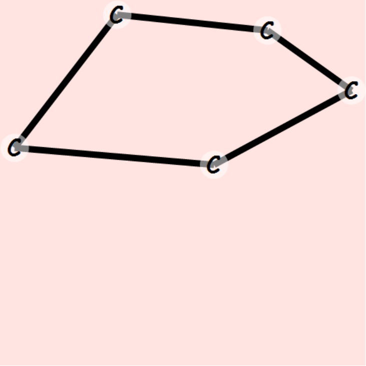


comp10_plane

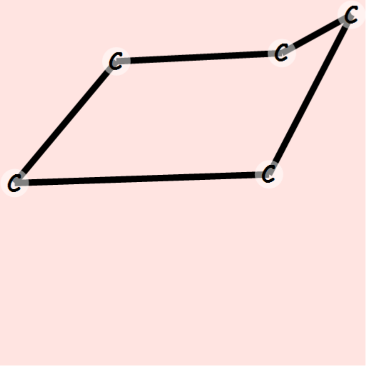


comp10_pucker

Tests



comp10_plane.cpms

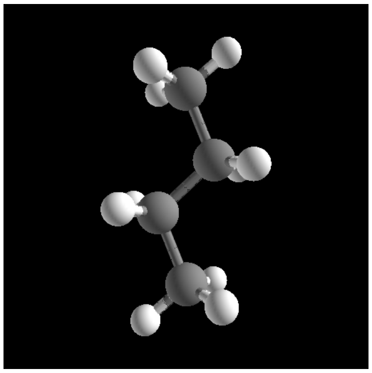


comp10_pucker.cpms

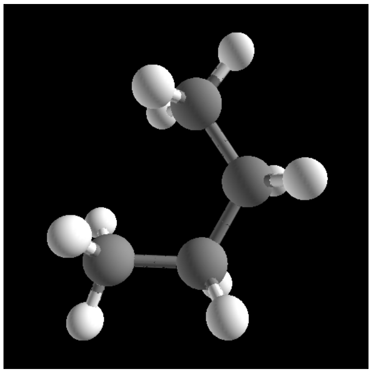
Diagram	comp10_plane	comp10_pucker
comp10_plane.cpms	Preferred	Not Preferred
comp10_pucker.cpms	Not Preferred	Preferred

comp11

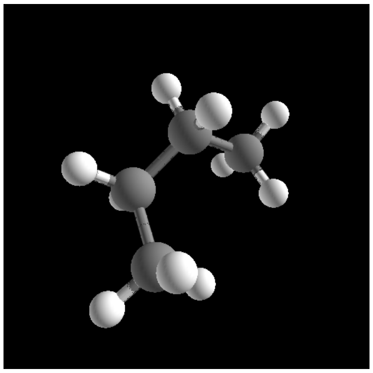
Models



comp11_a

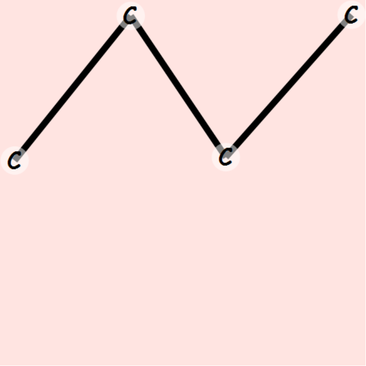


comp11_c

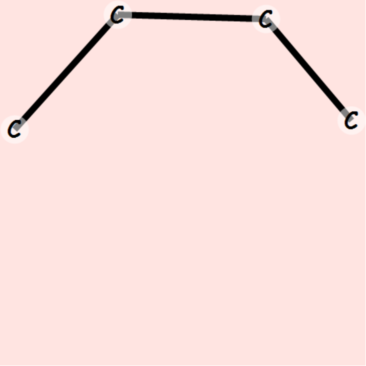


comp11_g

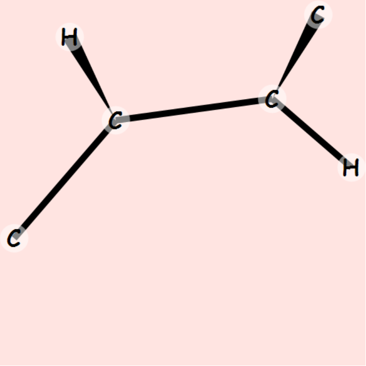
Tests



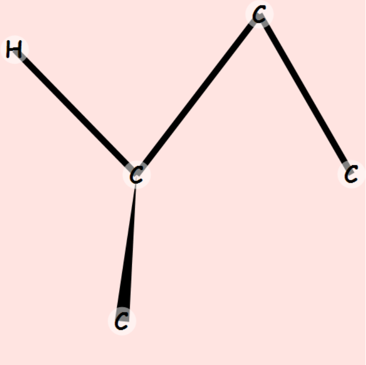
comp11_a.cpms



comp11_c.cpms



comp11_e.cpms

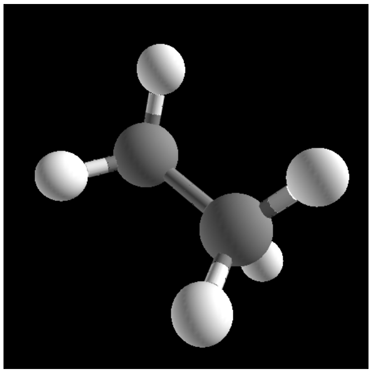


comp11_g.cpms

Diagram	comp11_a	comp11_c	comp11_g
comp11_a.cpms	Preferred	Not Preferred	Not Preferred
comp11_c.cpms	Not Preferred	Preferred	Not Preferred
comp11_e.cpms	Not Preferred	Not Preferred	Not Preferred
comp11_g.cpms	Not Preferred	Not Preferred	Preferred

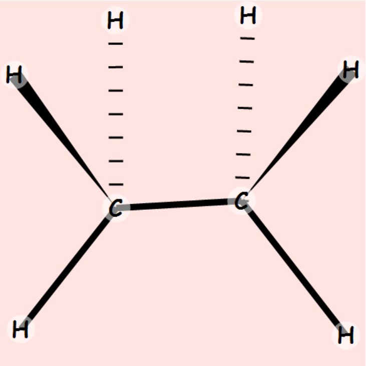
comp12

Models



comp12_eclips

Tests

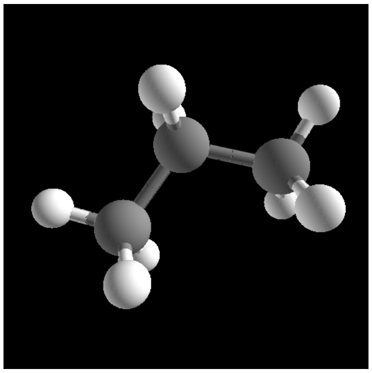


comp12_eclips.cpms

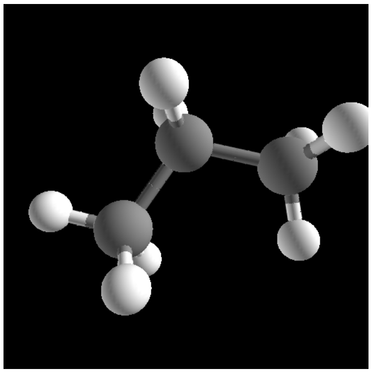
Diagram	comp12_eclips
comp12_eclips.cpms	Preferred

comp13

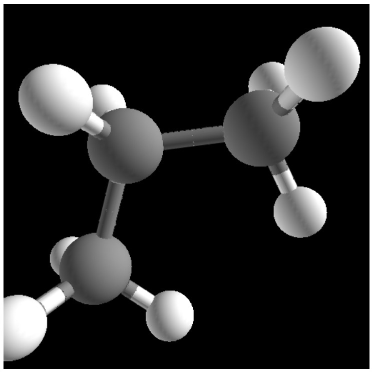
Models



comp13_conf1

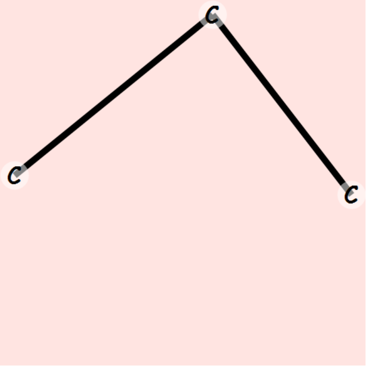


comp13_conf2

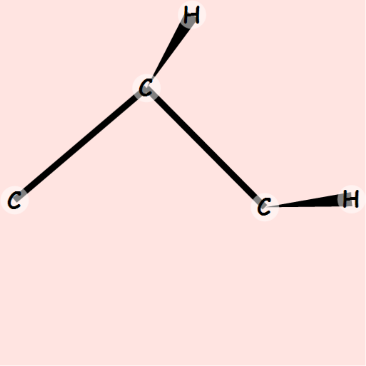


comp13_conf3

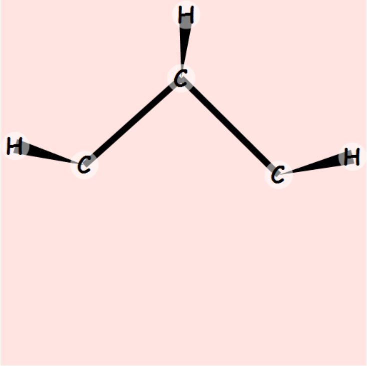
Tests



comp13_conf1.cpms



comp13_conf2.cpms

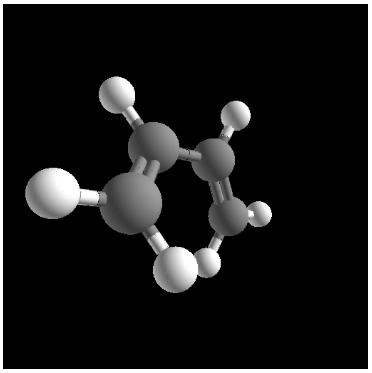


comp13_conf3.cpms

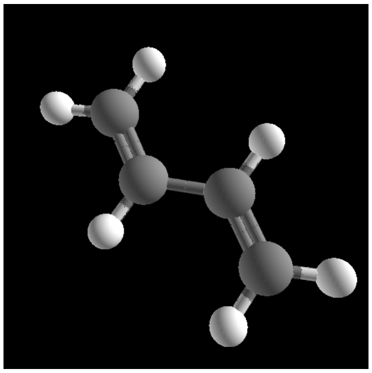
Diagram	comp13_conf1	comp13_conf2	comp13_conf3
comp13_conf1.cpms	Preferred	OK	OK
comp13_conf2.cpms	OK	Preferred	OK
comp13_conf3.cpms	OK	OK	Preferred

comp15

Models

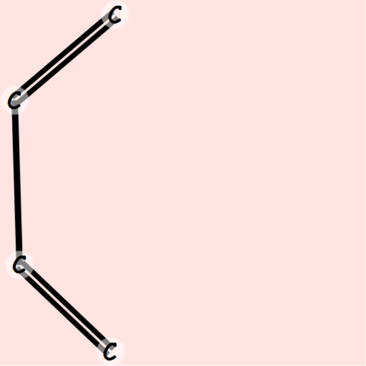


comp15_g

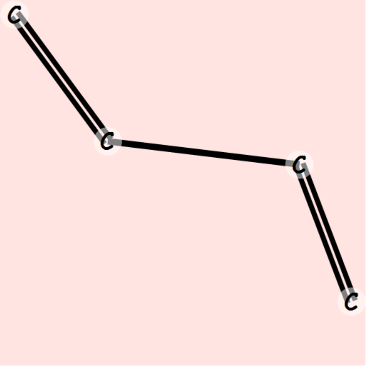


comp15_t

Tests



comp15_g.cpms

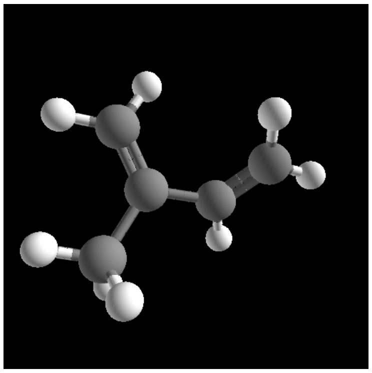


comp15_t.cpms

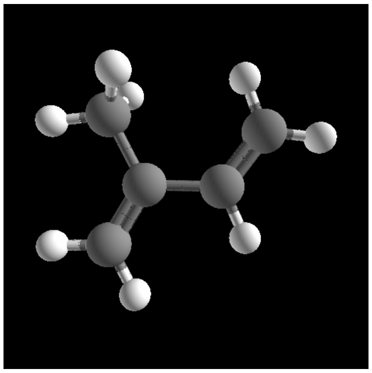
Diagram	comp15_g	comp15_t
comp15_g.cpms	Preferred	Not Preferred
comp15_t.cpms	Not Preferred	Preferred

comp16

Models

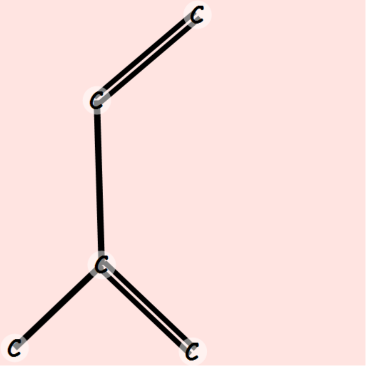


comp16_g

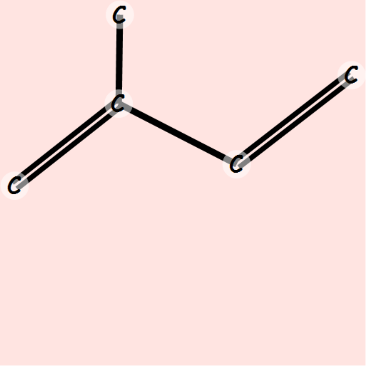


comp16_t

Tests



comp16_g.cpms

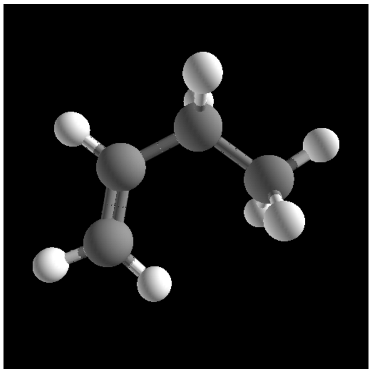


comp16_t.cpms

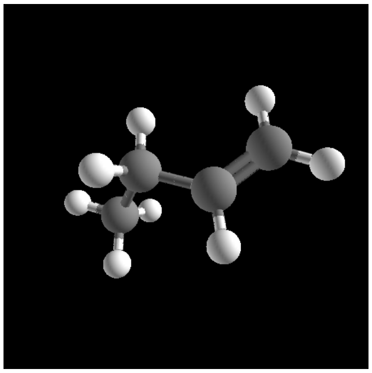
Diagram	comp16_g	comp16_t
comp16_g.cpms	Preferred	Not Preferred
comp16_t.cpms	OK	Preferred

comp17

Models

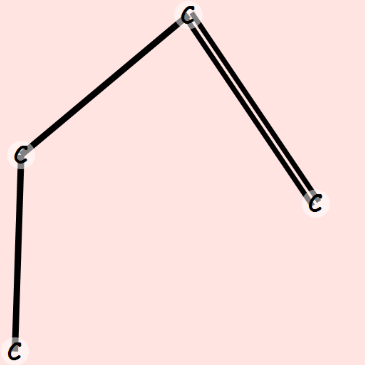


comp17_cis

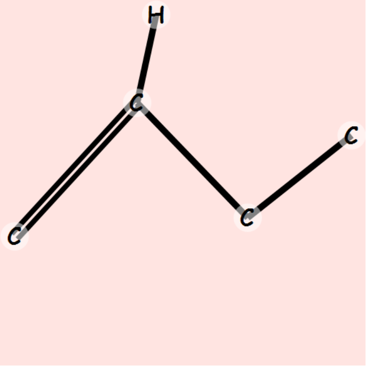


comp17_skew

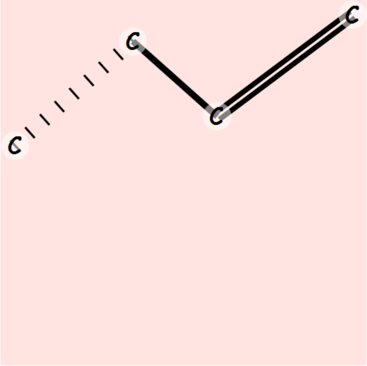
Tests



comp17_cis.cpms



comp17_neither.cpms

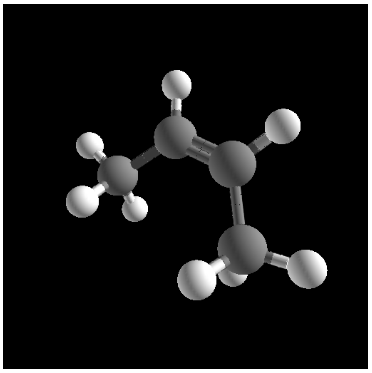


comp17_skew.cpms

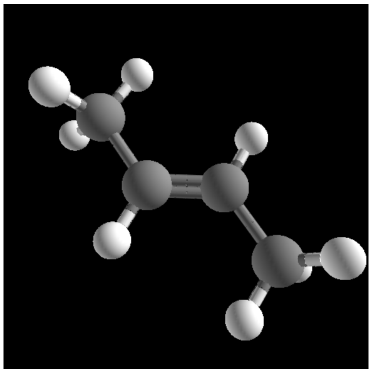
Diagram	comp17_cis	comp17_skew
comp17_cis.cpms	Preferred	Not Preferred
comp17_neither.cpms	OK	OK
comp17_skew.cpms	Not Preferred	Preferred

comp18

Models

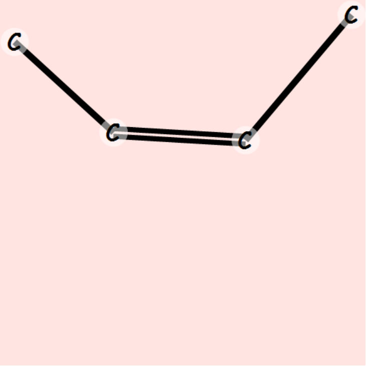


comp18_cis

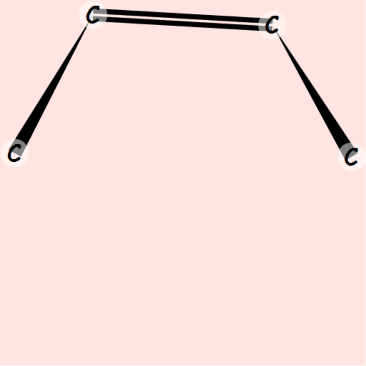


comp18_trans

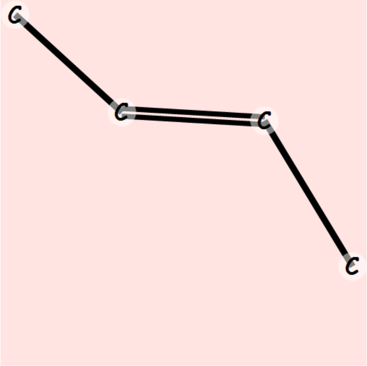
Tests



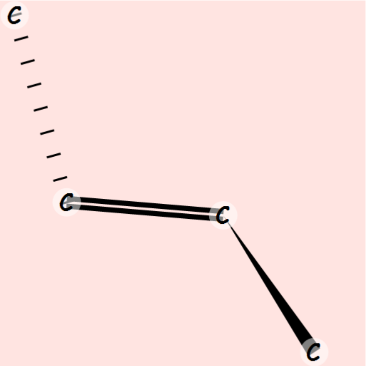
comp18_cis.cpms



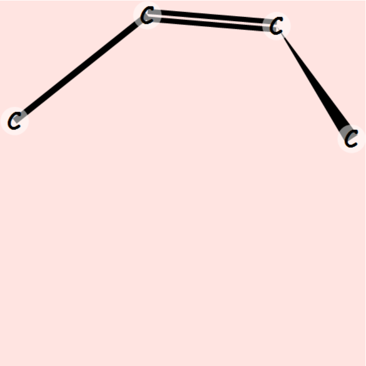
comp18_ciswedge.cpms



comp18_trans.cpms



comp18_transwedge.cpms

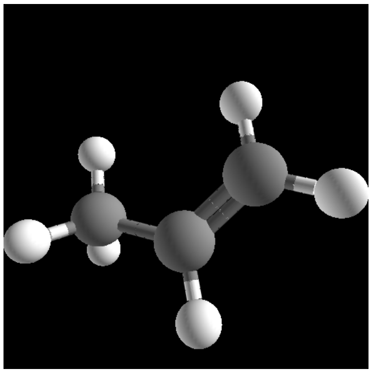


comp18_wrong.cpms

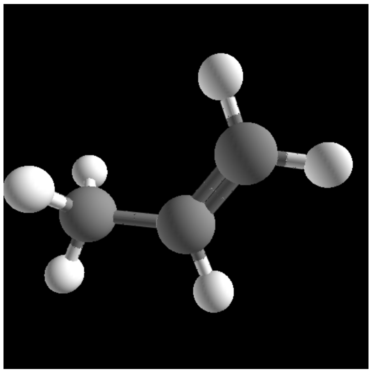
Diagram	comp18_cis	comp18_trans
comp18_cis.cpms	Preferred	Error
comp18_ciswedge.cpms	Preferred	Error
comp18_trans.cpms	Error	Preferred
comp18_transwedge.cpms	Error	Preferred
comp18_wrong.cpms	Not Preferred	Error

comp19

Models

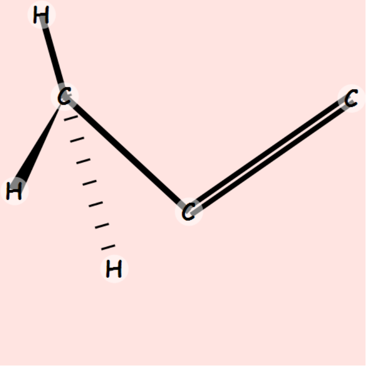


comp19_c

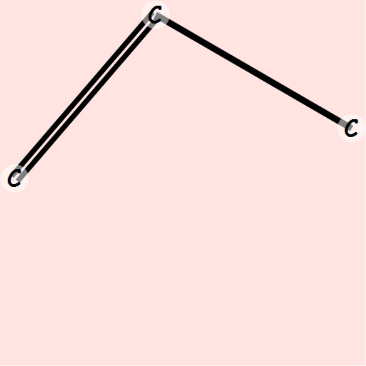


comp19_t

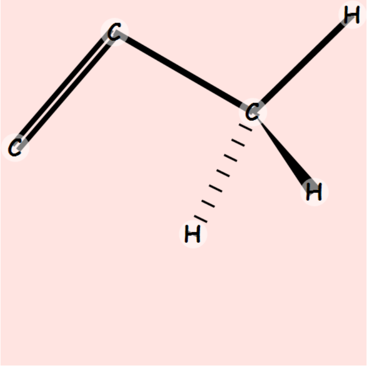
Tests



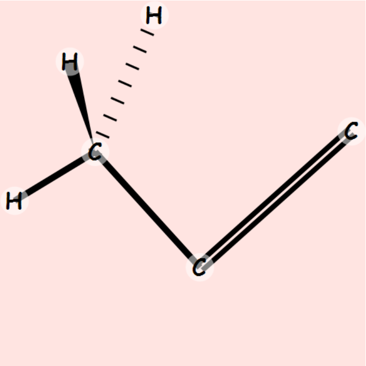
comp19_c.cpms



comp19_generic.cpms



comp19_max.cpms

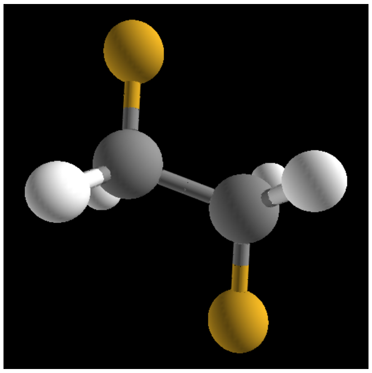


comp19_t.cpms

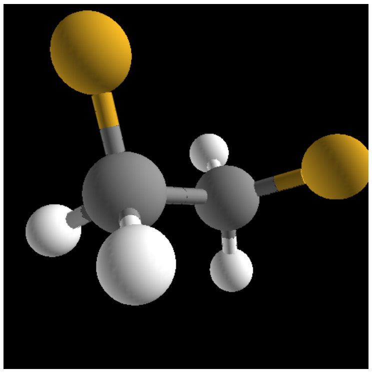
Diagram	comp19_c	comp19_t
comp19_c.cpms	OK	OK
comp19_generic.cpms	OK	OK
comp19_max.cpms	Not Preferred	Preferred
comp19_t.cpms	OK	OK

comp20

Models

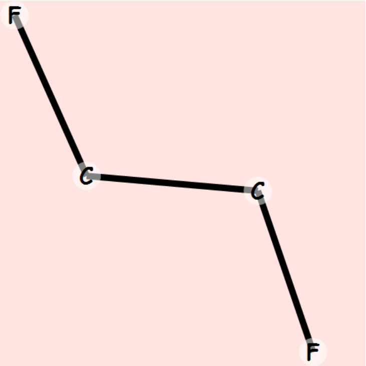


comp20_a

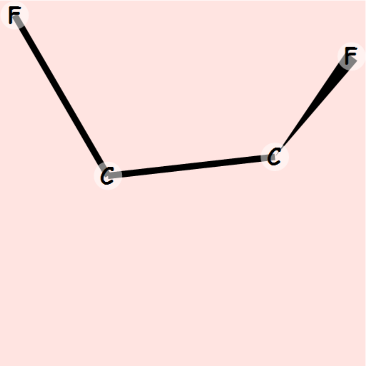


comp20_g

Tests



comp20_a.cpms

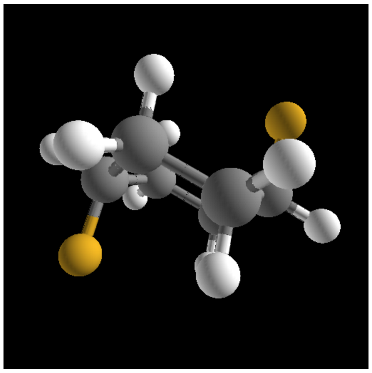


comp20_g.cpms

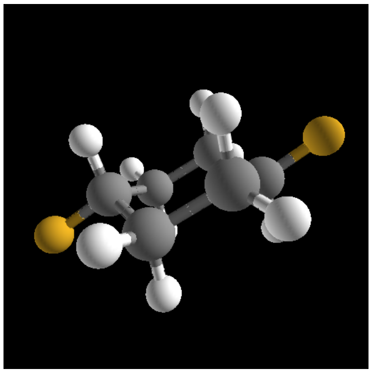
Diagram	comp20_a	comp20_g
comp20_a.cpms	Preferred	Not Preferred
comp20_g.cpms	Not Preferred	Preferred

comp24

Models

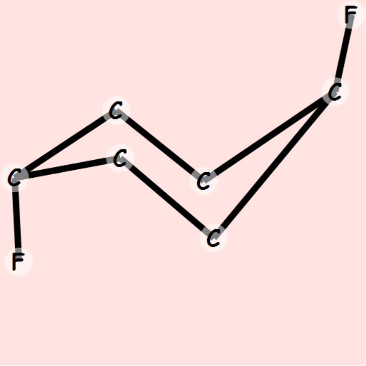


comp24_axax

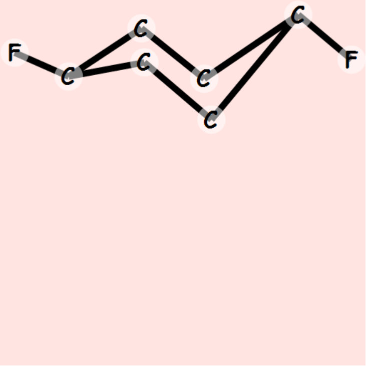


comp24_eqeq

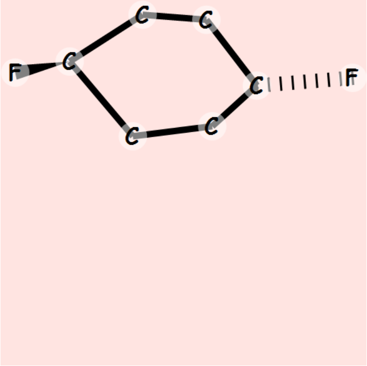
Tests



comp24_axax.cpms



comp24_eqeq.cpms

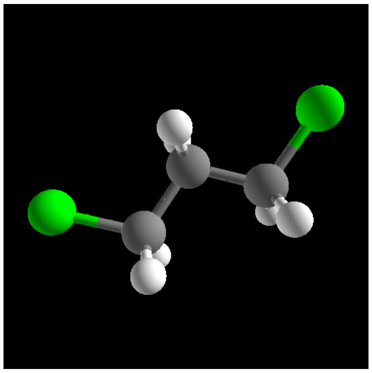


comp24_generic.cpms

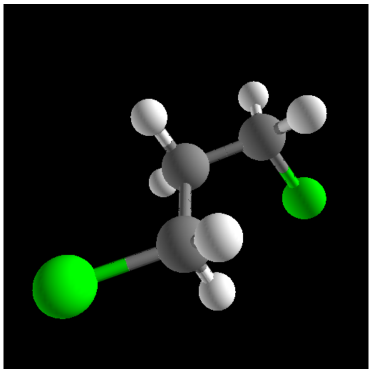
Diagram	comp24_axax	comp24_eqeq
comp24_axax.cpms	Preferred	Not Preferred
comp24_eqeq.cpms	Not Preferred	Preferred
comp24_generic.cpms	OK	OK

comp27

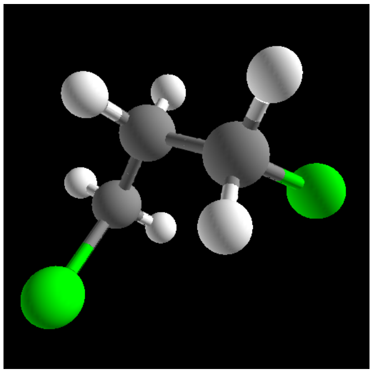
Models



comp27_aa

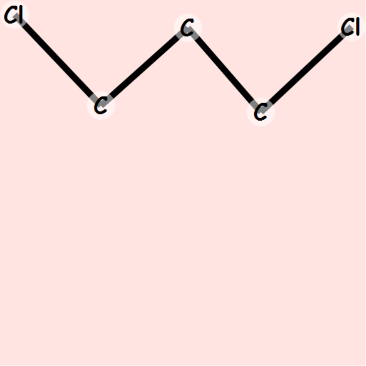


comp27_ga

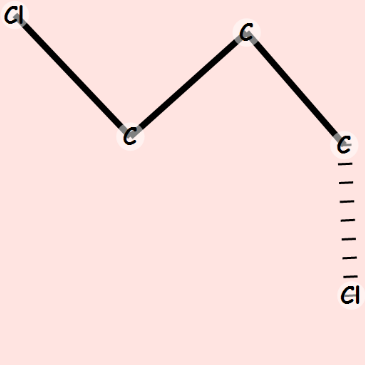


comp27_gg

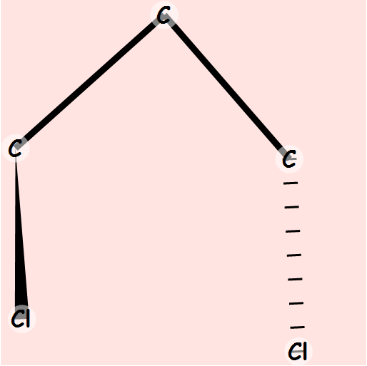
Tests



comp27_aa.cpms



comp27_ga.cpms

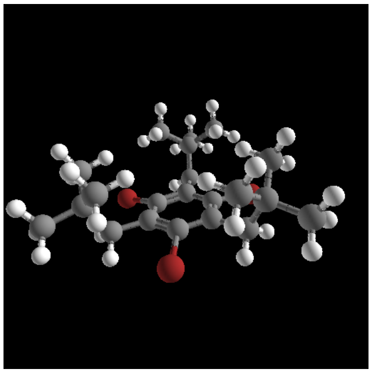


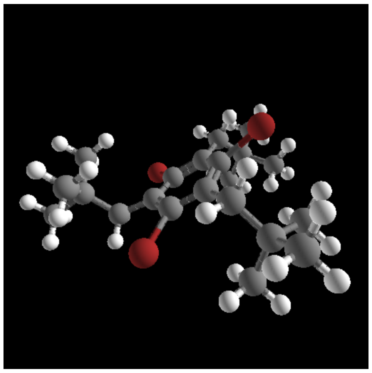
comp27_gg.cpms

Diagram	comp27_aa	comp27_ga	comp27_gg
comp27_aa.cpms	Preferred	Not Preferred	Not Preferred
comp27_ga.cpms	Not Preferred	Preferred	Not Preferred
comp27_gg.cpms	Not Preferred	Not Preferred	Preferred

comp32

Models

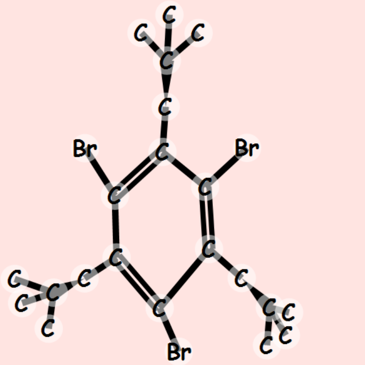


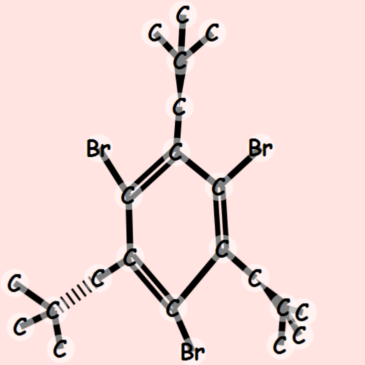


comp32_all

comp32_two

Tests





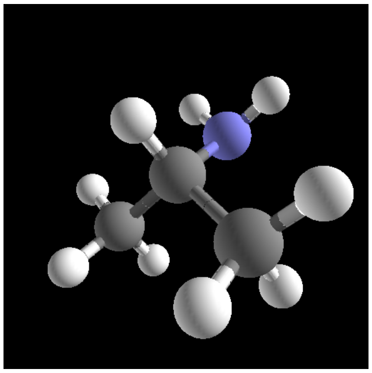
comp32_all.cpms

comp32_two.cpms

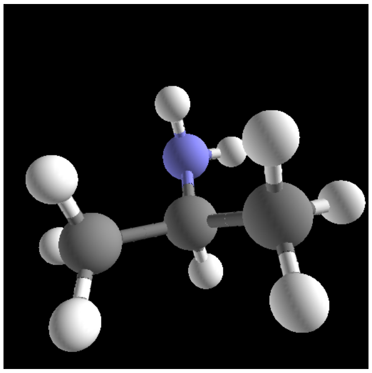
Diagram	comp32_all	comp32_two
comp32_all.cpms	Preferred	Not Preferred
comp32_two.cpms	Not Preferred	Preferred

comp36

Models

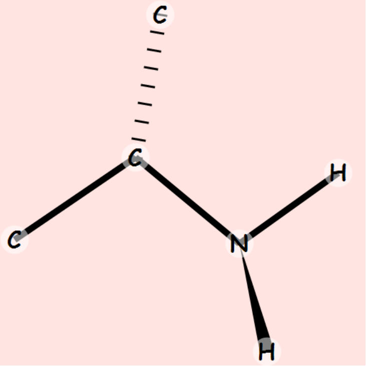


comp36_a

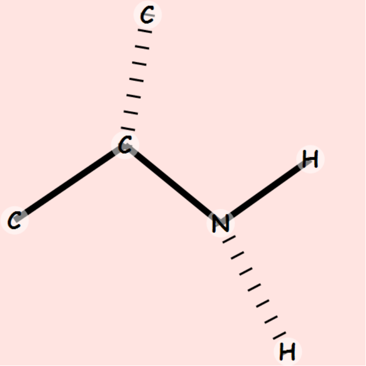


comp36_g

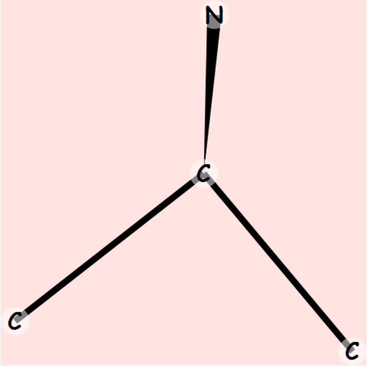
Tests



comp36_a.cpms



comp36_g.cpms

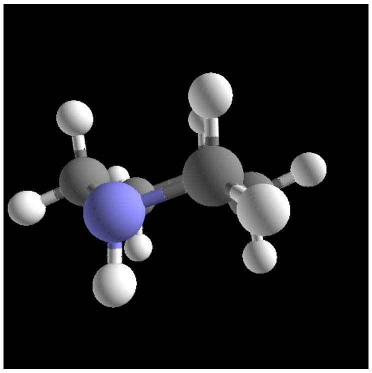


comp36_generic.cpms

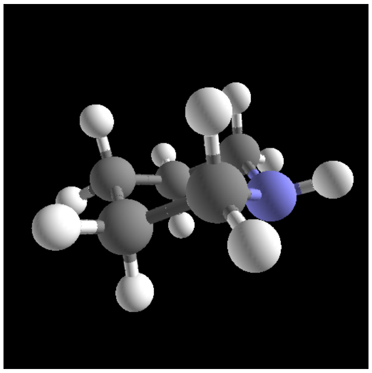
Diagram	comp36_a	comp36_g
comp36_a.cpms	Preferred	Not Preferred
comp36_g.cpms	Not Preferred	Preferred
comp36_generic.cpms	OK	OK

comp38

Models

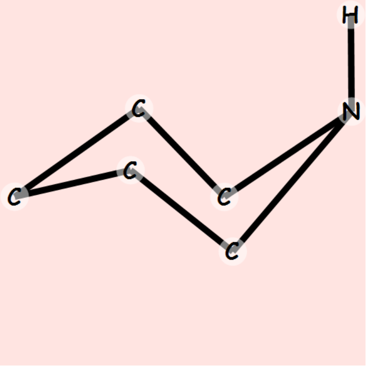


comp38_ax

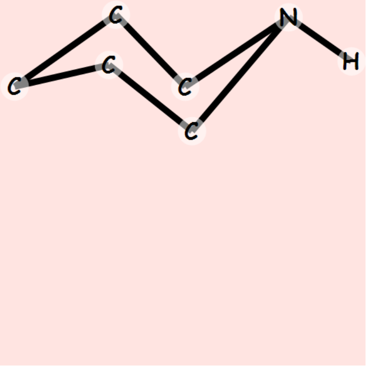


comp38_eq

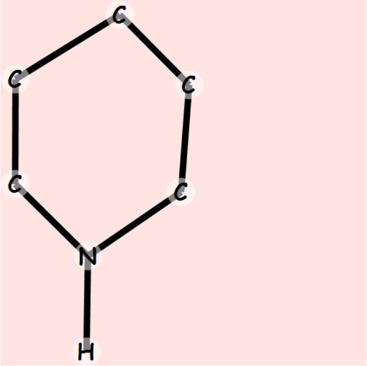
Tests



comp38_ax.cpms



comp38_eq.cpms

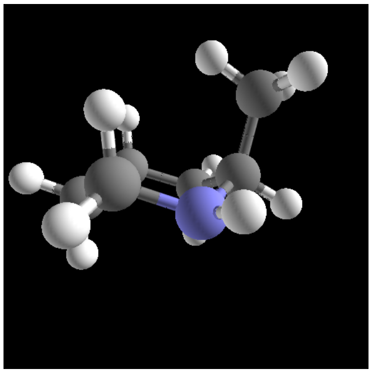


comp38_generic.cpms

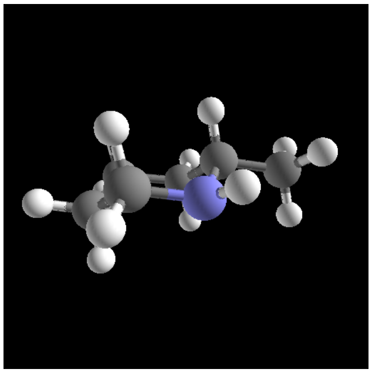
Diagram	comp38_ax	comp38_eq
comp38_ax.cpms	Preferred	Not Preferred
comp38_eq.cpms	Not Preferred	Preferred
comp38_generic.cpms	OK	OK

comp40

Models

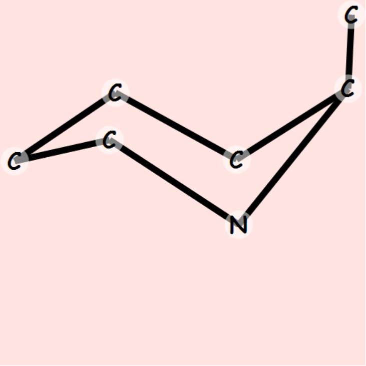


comp40_ax

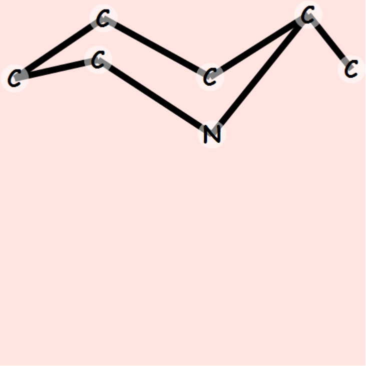


comp40_eq

Tests



comp40_ax.cpms

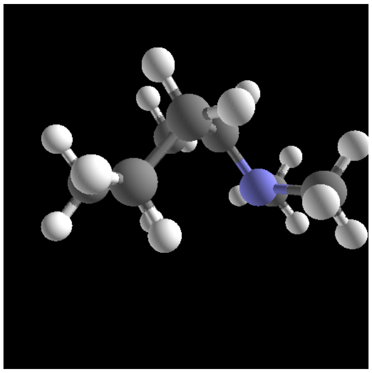


comp40_eq.cpms

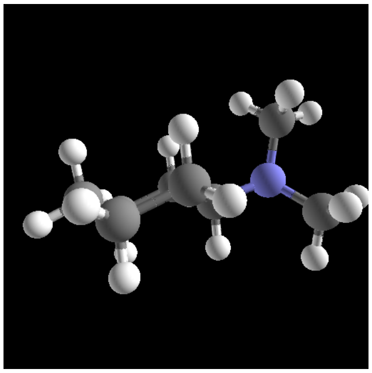
Diagram	comp40_ax	comp40_eq
comp40_ax.cpms	Preferred	Not Preferred
comp40_eq.cpms	Not Preferred	Preferred

comp43

Models

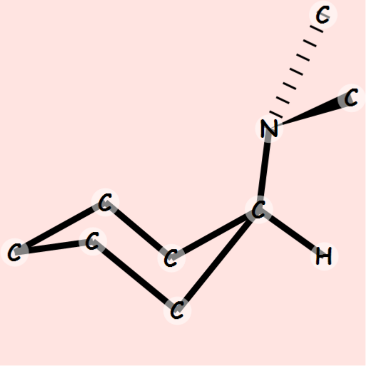


comp43_ax

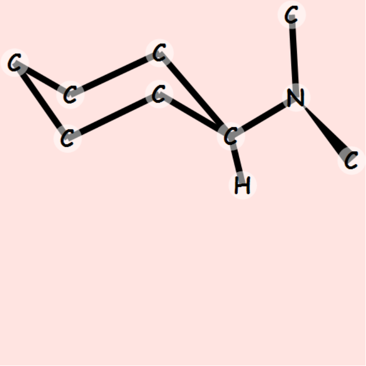


comp43_eq

Tests



comp43_ax.cpms

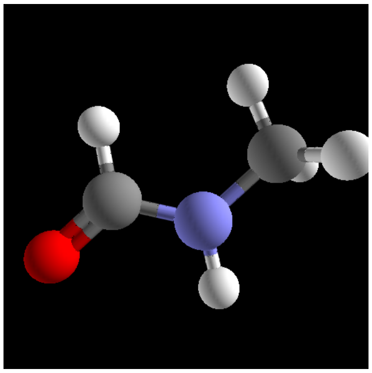


comp43_eq.cpms

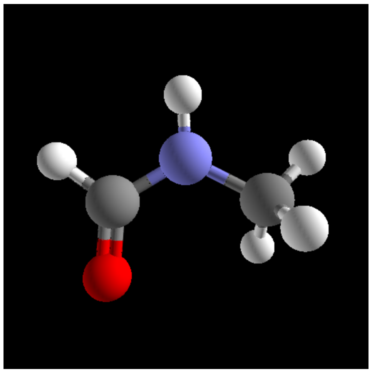
Diagram	comp43_ax	comp43_eq
comp43_ax.cpms	Preferred	Not Preferred
comp43_eq.cpms	Not Preferred	Preferred

comp44

Models

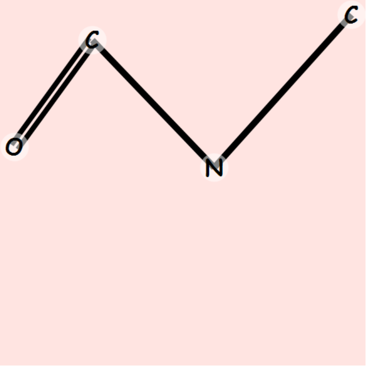


comp44_c

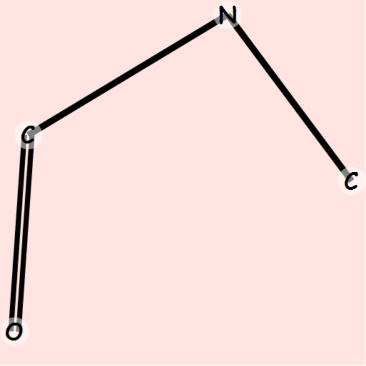


comp44_t

Tests



comp44_c.cpms

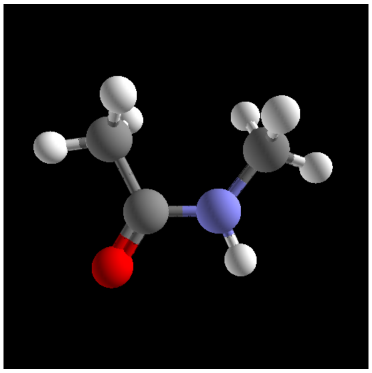


comp44_t.cpms

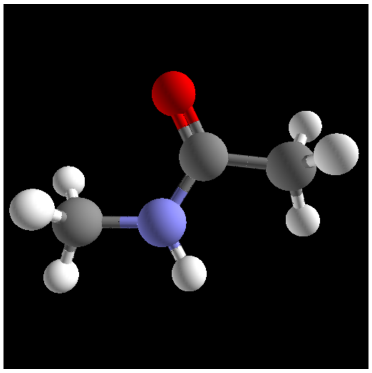
Diagram	comp44_c	comp44_t
comp44_c.cpms	Error	Preferred
comp44_t.cpms	Preferred	Error

comp45

Models

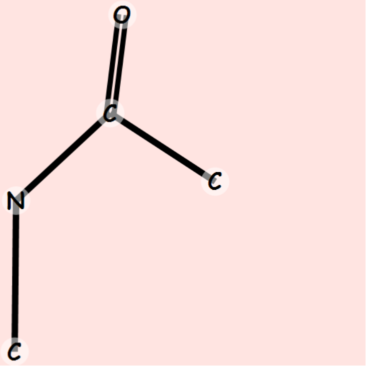


comp45_c

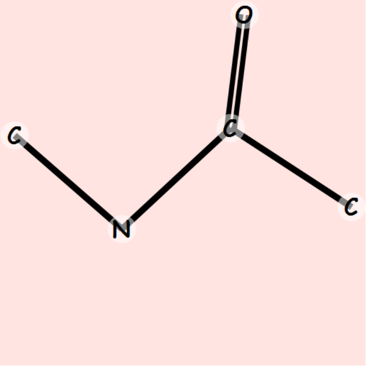


comp45_t

Tests



comp45_c.cpms

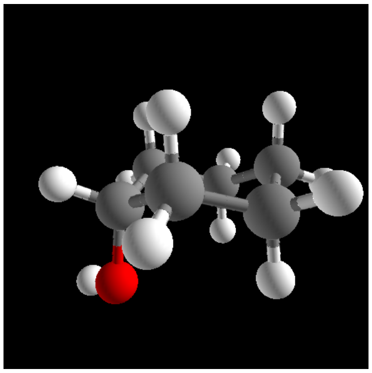


comp45_t.cpms

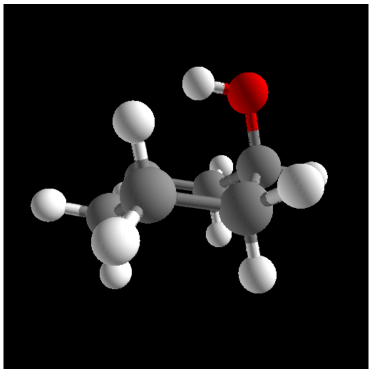
Diagram	comp45_c	comp45_t
comp45_c.cpms	Preferred	Not Preferred
comp45_t.cpms	Not Preferred	Preferred

comp50

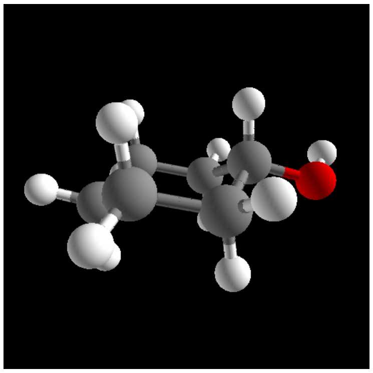
Models



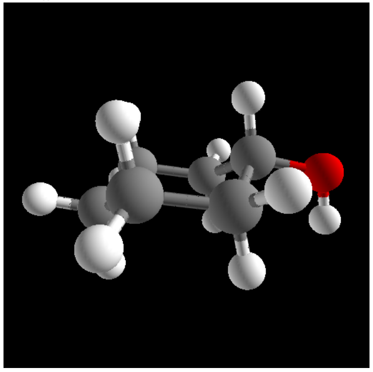
comp50_axc1



comp50_axcs

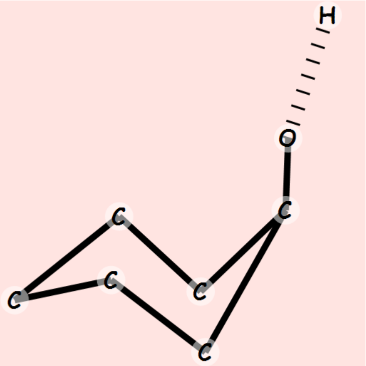


comp50_eqc1

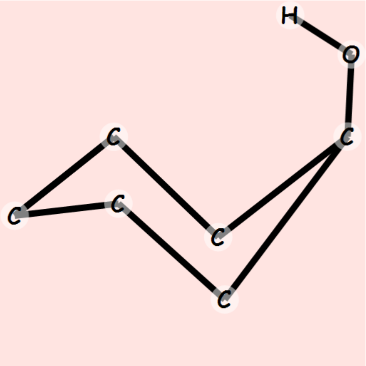


comp50_eqcs

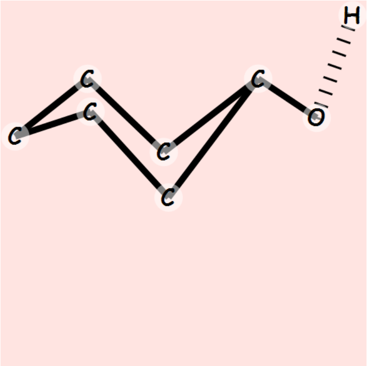
Tests



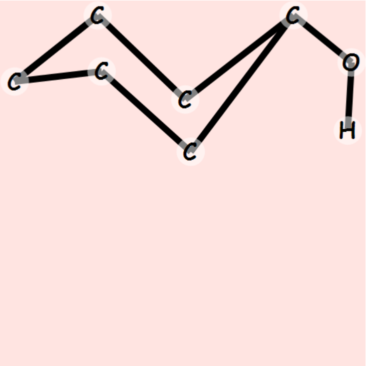
comp50_axc1.cpms



comp50_axcs.cpms



comp50_eqc1.cpms

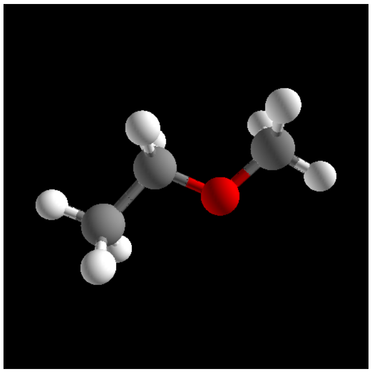


comp50_eqcs.cpms

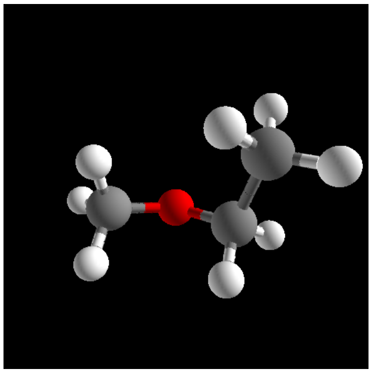
Diagram	comp50_axc1	comp50_axcs	comp50_eqc1	comp50_eqcs
comp50_axc1.cpms	Preferred	Not Preferred	Not Preferred	Not Preferred
comp50_axcs.cpms	Not Preferred	Preferred	Not Preferred	Not Preferred
comp50_eqc1.cpms	Not Preferred	Not Preferred	Preferred	Not Preferred
comp50_eqcs.cpms	Not Preferred	Not Preferred	Not Preferred	Preferred

comp57

Models

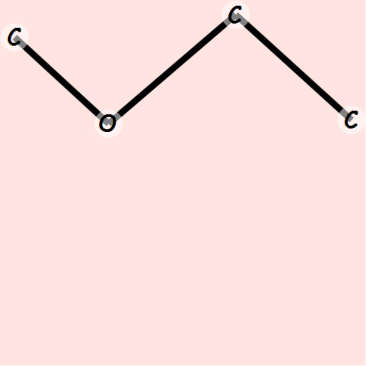


comp57_a

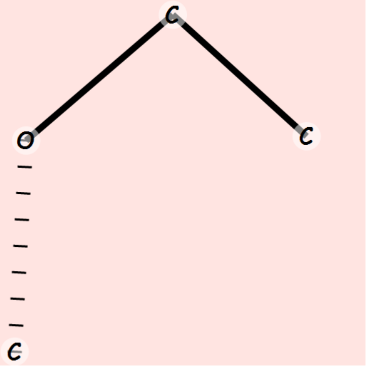


comp57_g

Tests



comp57_a.cpms

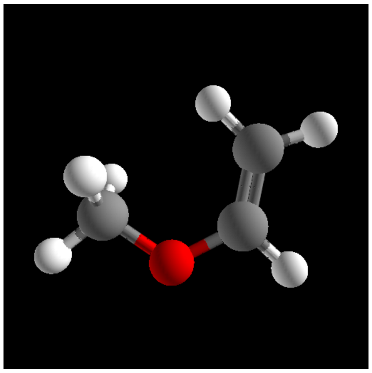


comp57_g.cpms

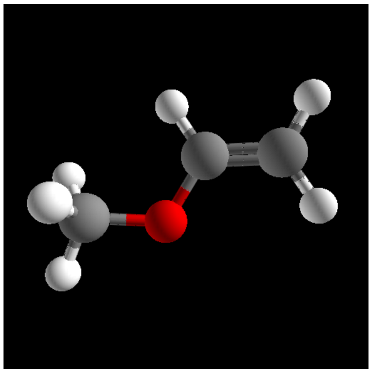
Diagram	comp57_a	comp57_g
comp57_a.cpms	Preferred	Not Preferred
comp57_g.cpms	Not Preferred	Preferred

comp58

Models

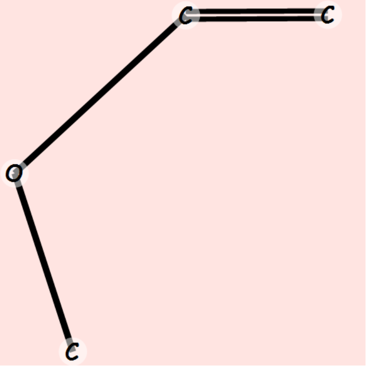


comp58_cis

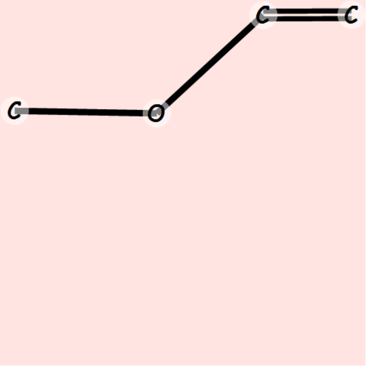


comp58_skew

Tests



comp58_cis.cpms

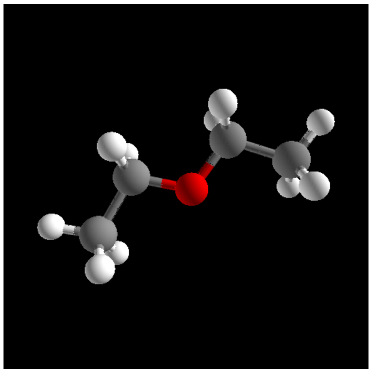


comp58_skew.cpms

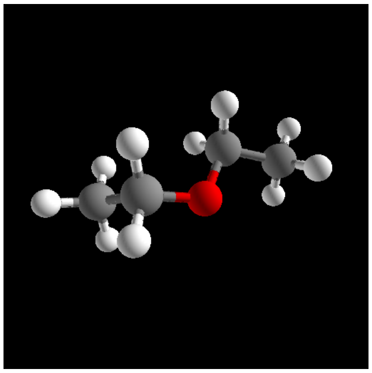
Diagram	comp58_cis	comp58_skew
comp58_cis.cpms	Preferred	Not Preferred
comp58_skew.cpms	Not Preferred	Preferred

comp59

Models

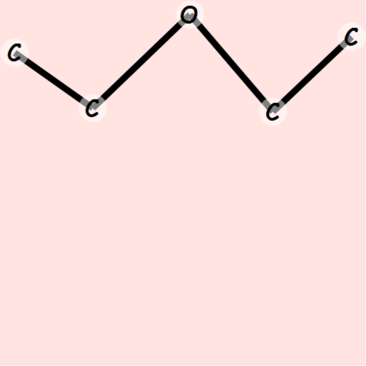


comp59_a

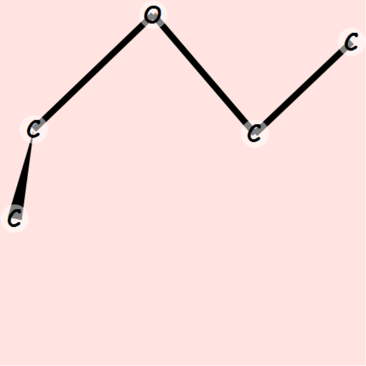


comp59_g

Tests



comp59_a.cpms

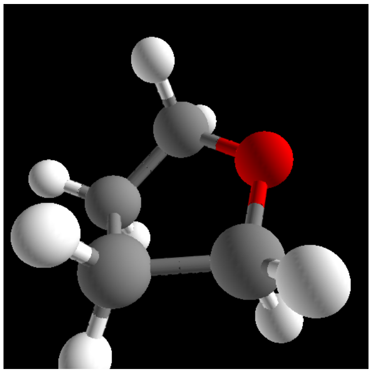


comp59_g.cpms

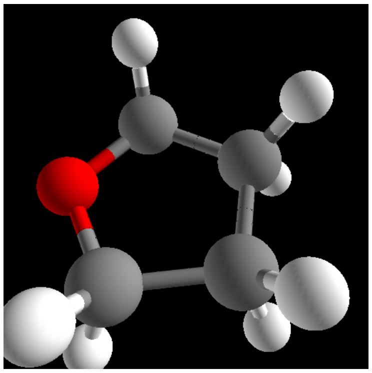
Diagram	comp59_a	comp59_g
comp59_a.cpms	Preferred	Not Preferred
comp59_g.cpms	Not Preferred	Preferred

comp61

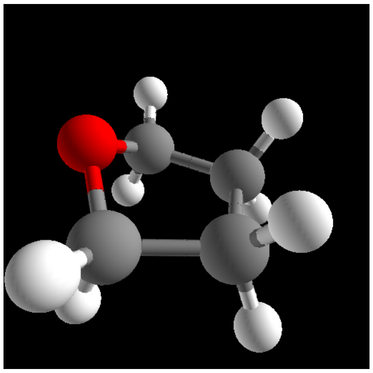
Models



comp61_c2

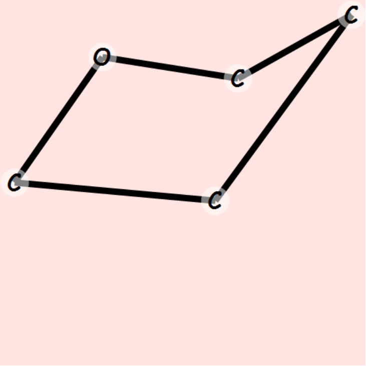


comp61_c2v

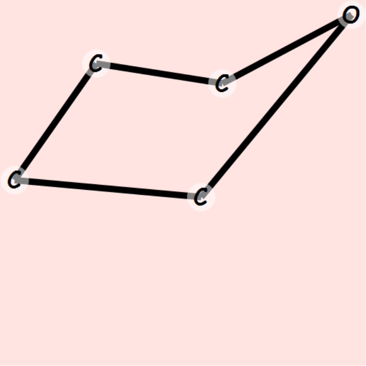


comp61_cs

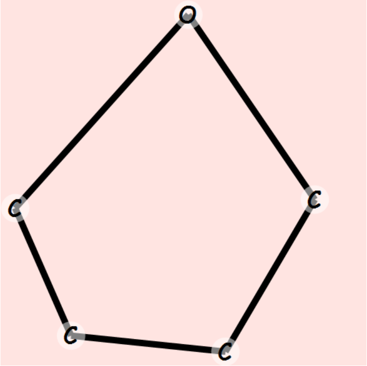
Tests



comp61_c2.cpms



comp61_cs.cpms

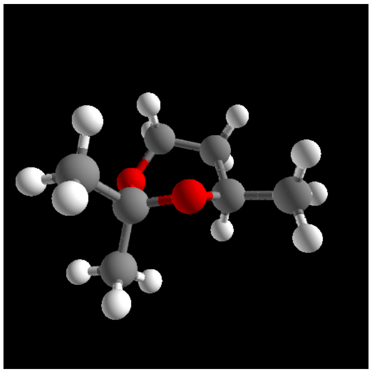


comp61_generic.cpms

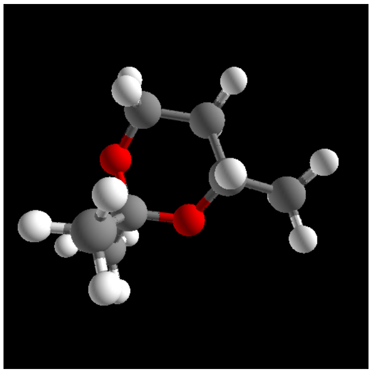
Diagram	comp61_c2	comp61_c2v	comp61_cs
comp61_c2.cpms	Preferred	Not Preferred	Not Preferred
comp61_cs.cpms	Not Preferred	Not Preferred	Preferred
comp61_generic.cpms	OK	OK	OK

comp63

Models

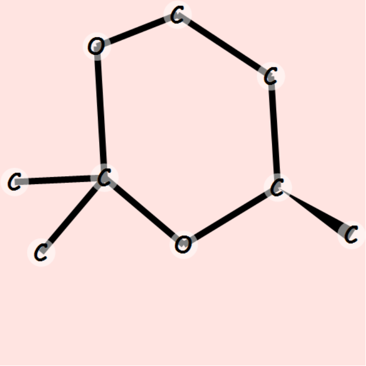


comp63_rboat

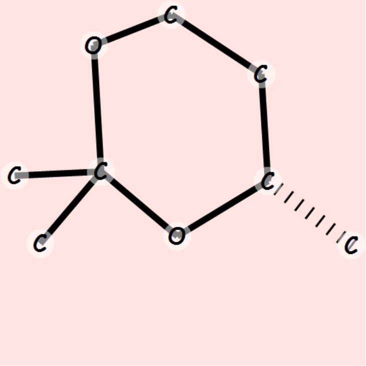


comp63_schair

Tests



comp63_r.cpms

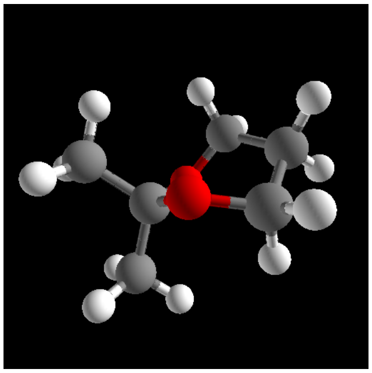


comp63_s.cpms

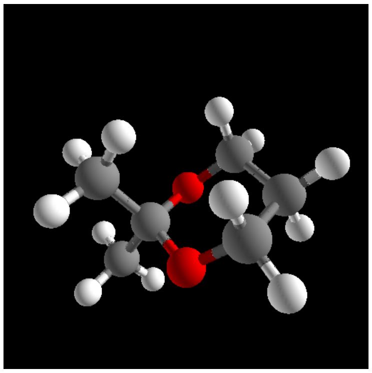
Diagram	comp63_rboat	comp63_schair
comp63_r.cpms	OK	Error
comp63_s.cpms	Error	OK

comp69

Models

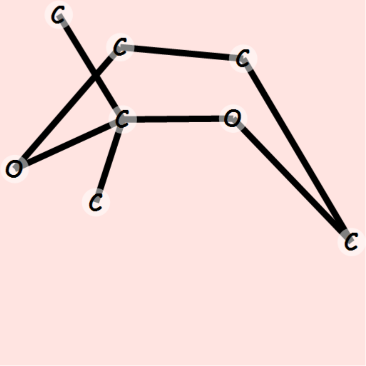


comp69_boat

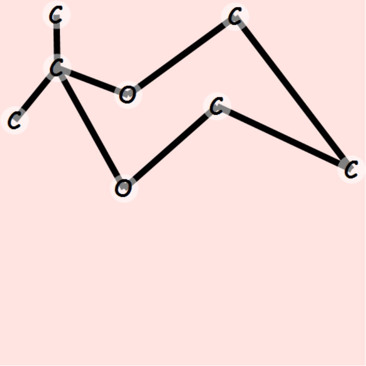


comp69_chair

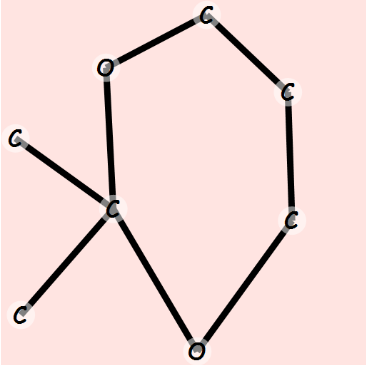
Tests



comp69_boat.cpms



comp69_chair.cpms

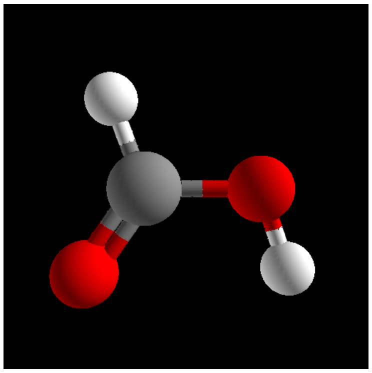


comp69_generic.cpms

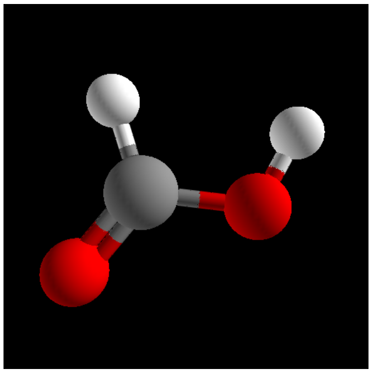
Diagram	comp69_boat	comp69_chair
comp69_boat.cpms	Not Preferred	Preferred
comp69_chair.cpms	Not Preferred	Preferred
comp69_generic.cpms	OK	OK

comp72

Models

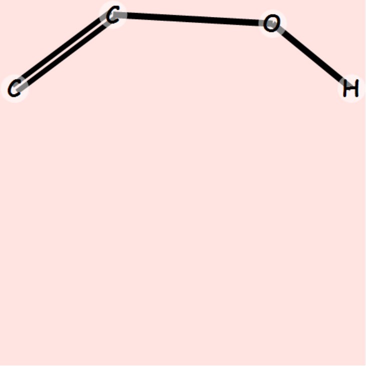


comp72_c

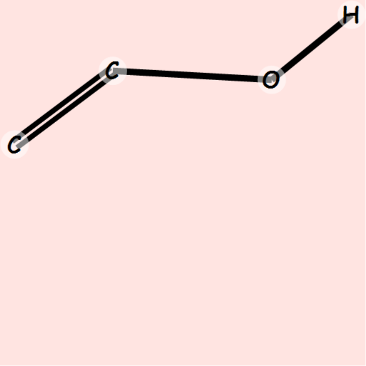


comp72_t

Tests



comp72_c.cpms

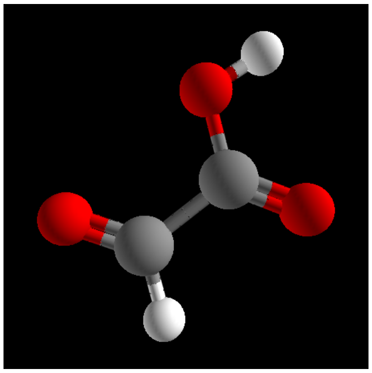


comp72_t.cpms

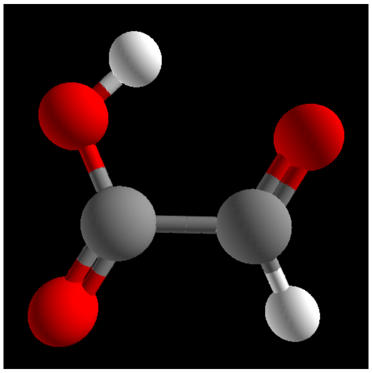
Diagram	comp72_c	comp72_t
comp72_c.cpms	Preferred	Not Preferred
comp72_t.cpms	Not Preferred	Preferred

comp73

Models

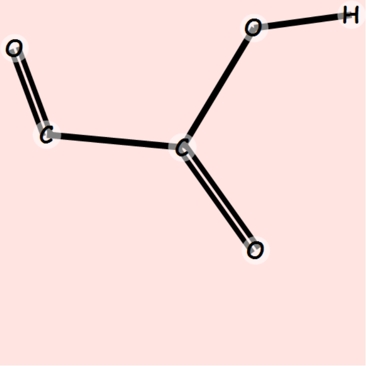


comp73_c

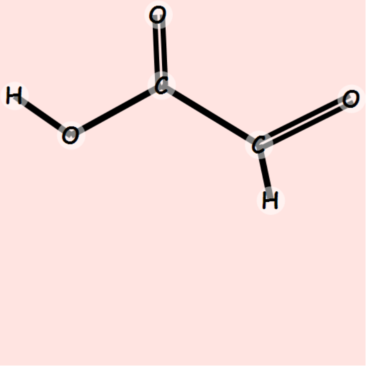


comp73_t

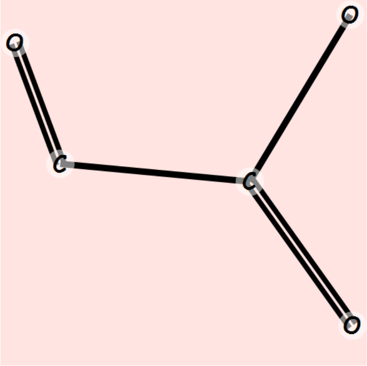
Tests



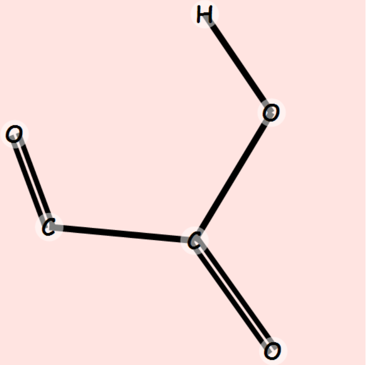
comp73_c.cpms



comp73_c2.cpms



comp73_generic.cpms

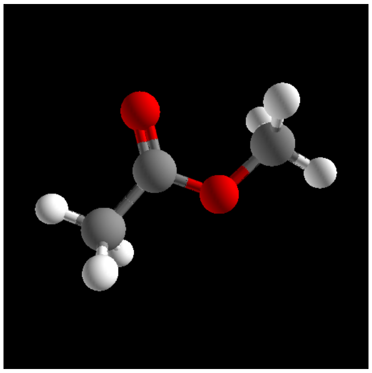


comp73_t.cpms

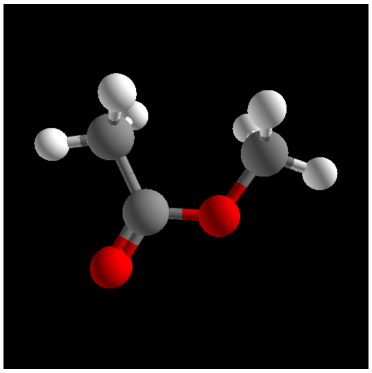
Diagram	comp73_c	comp73_t
comp73_c.cpms	Preferred	Not Preferred
comp73_c2.cpms	Not Preferred	Not Preferred
comp73_generic.cpms	OK	OK
comp73_t.cpms	Not Preferred	Preferred

comp76

Models

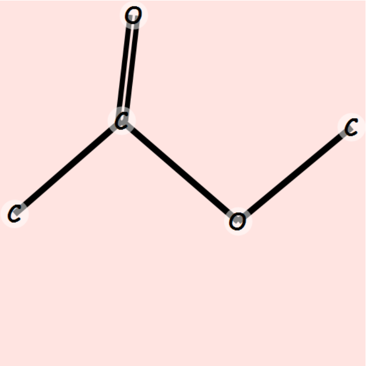


comp76_c

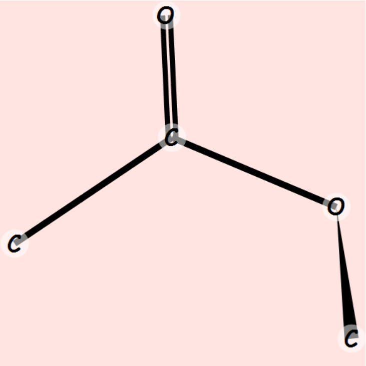


comp76_t

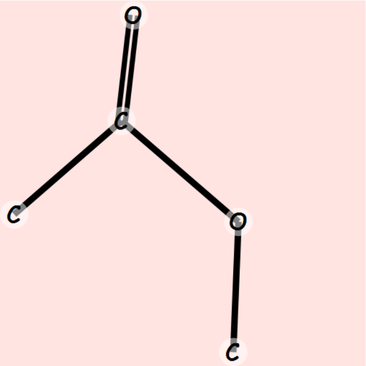
Tests



comp76_c.cpms



comp76_g.cpms

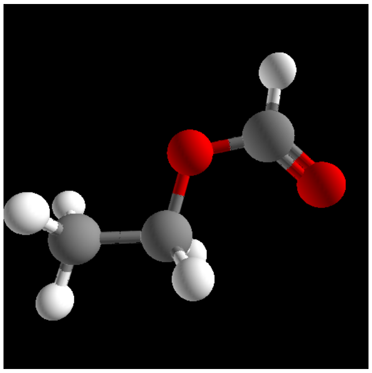


comp76_t.cpms

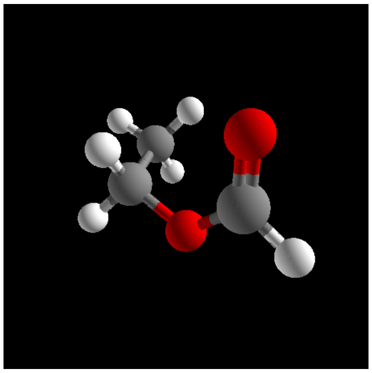
Diagram	comp76_c	comp76_t
comp76_c.cpms	Preferred	Not Preferred
comp76_g.cpms	OK	OK
comp76_t.cpms	Not Preferred	Preferred

comp77

Models

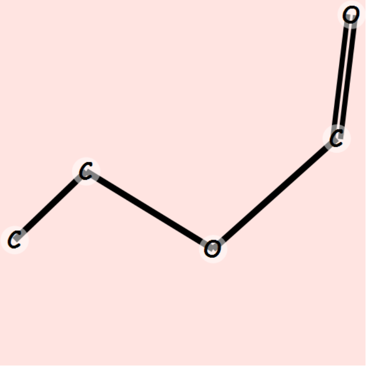


comp77_1

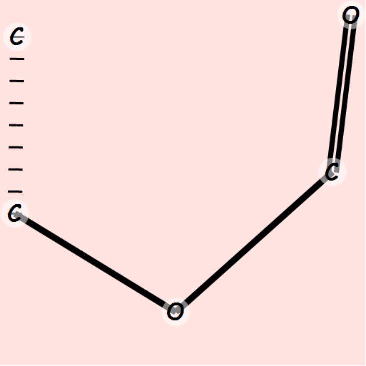


comp77_2

Tests



comp77_1.cpms

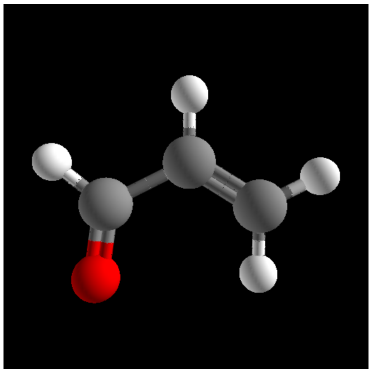


comp77_2.cpms

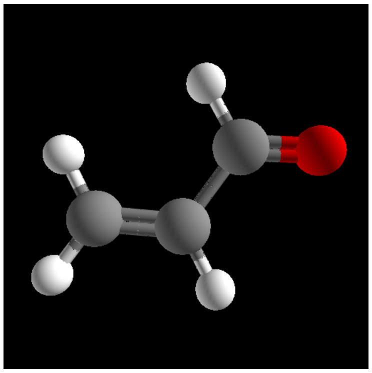
Diagram	comp77_1	comp77_2
comp77_1.cpms	Preferred	Not Preferred
comp77_2.cpms	Not Preferred	Preferred

comp80

Models

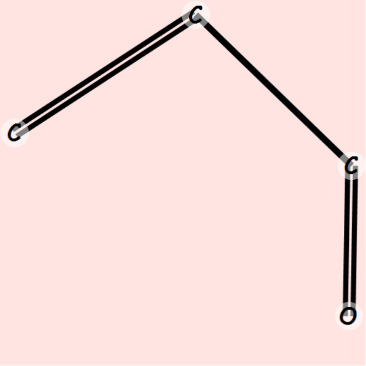


comp80_c

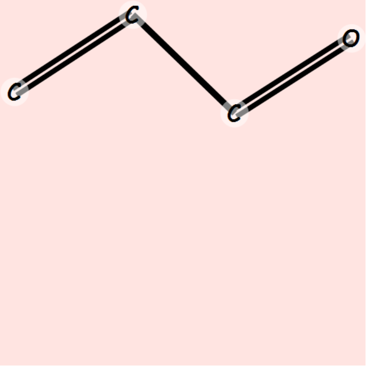


comp80_t

Tests



comp80_c.cpms



comp80_t.cpms

Diagram	comp80_c	comp80_t
comp80_c.cpms	Preferred	Not Preferred
comp80_t.cpms	Not Preferred	Preferred

Appendix D

Other Algorithms

For the purpose of reproducibility, this appendix provides pseudo and sample code for a few algorithms and functions often used in this work. The first algorithm, CIP Ordering, is provided as an easy to understand alternative to the few existing published algorithms on the topic. This algorithm also contains a property useful for educational animations of the scheme. The formula for calculating dihedral angles is documented in computational chemistry literature, but its description is often tied directly to the chemistry involved. The code sample we provide here is for the benefit of computer scientists with limited chemistry experience. Finally, the formula for determining a turn direction in 3D is critical to IM3 terms and is straightforward to calculate, but we have found no previously documented algorithm for accomplishing this task.

D.1 Determining CIP Ordering

The CIP ordering system orders constituent groups from highest to lowest when determining stereochemistry naming. Chemists describe CIP in vague terms such as those in Figure D.2 which are not fully spelled out from an algorithmic standpoint.

A human performing this operation in Organic Chemistry tends to use a little intuition to solve the problem. Given four different constituent groupings, one would try to pairwise compare the branches to find out which priorities to assign. Often two of the branches are different non-carbons (a hydrogen and an oxygen for example) which can quickly be spotted for absolute priority by only looking at the first atomic weight. When two of the branches are carbon branches, the human eye can usually note the depth of the first difference in the chain and then deduce which is the higher priority.

For the computer to determine the priorities of the branches, more rigor is required. When there is a difference one or two atoms away from a stereocenter, it is not difficult to quickly determine the priority orderings, however as the constituent branches, the algorithm needs to branch down the

```
Algorithm D.1.1: SortedPriorities(branchesList)

procedure CIPScore(constituentBranch)
  if IsAnAtom(constituentBranch)
    then string ← FormatToThreeDigits(constuituentBranch.
           AtomicNumber)
         return (string)
    else neighborValuesList = newList()
         subscoresList = newList()
         for each childBranch ∈ constituentBranch.Children
           do neighborValuesList.Add(
                FormatToThreeDigits(constuituentBranch.
                Root.AtomicNumber))
              subscoresList.Add(CIPScore(childBranch))
         neighborValuesList.SortAlphabetically()
         subscoresList.SortAlphabetically()
         score = newString()
         for each string ∈ neighborValuesString
           do score+ = string
         for each string ∈ subscoresString
           do score+ = string
         return (score)

main
  scoresList = newList()
  for each branch ∈ branchesList
    do scoresList.Add(CIPScore(branch))
  scoresList.SortAlphabetically()
  return (scoresList)
```

Figure D.1: *Assigning CIP priorities using string comparison*

1. Higher atomic number takes precedence over lower.
2. When two atoms directly attached to the stereogenic center are identical, compare the atoms attached to these two on the basis of their atomic numbers. Work outward from the point of attachment and evaluate substituent atoms one by one. Precedence is determined at the first point of difference.
3. The difference is determined by the substituent of the highest atomic number and is not additive if there is more than one substituent.
4. Where there is a double or triple bond, both atoms are considered to be duplicated or triplicated.

Figure D.2: *A chemist's description of the CIP rules [84].*

higher priority subbranch first, which the computer doesn't intuitively know how to do. Although the CIP rules have been around since the mid 60's, the importance of absolute sterochemistry determination has only been emphasized computationally for the last decade [19]. Labute's algorithm for CIP ordering [53] uses a global partitioning system to prioritize every atom in the molecule independent of the queried stereocenter. This solution allows for quick computation of the chirality of multiple stereocenters in a molecule, but is not intuitive from the perspective of the CIP rules and therefore hard to use as a basis for a pedagogically-oriented algorithm animation. A straightforward implementation of the CIP rules calls for recursion, but a recursive order in which to traverse the molecule is not apparent.

ChemPad uses an application of string comparison in a recursive algorithm to compute an individual stereocenter's constituent orders. This also notes the path traversed to the point of first difference (which could be used for an algorithm animation.) The algorithm is detailed in Figure D.1. Each constituent is assigned a numerical value that is generated recursively and stored as a string. The goal is to create strings of digits that when sorted into alphabetical order will correctly prioritize the constituent groups. Here, the recursive base case is that of a terminal atom which produces its atomic value as a 3-digit number. Oxygen is 008, for example. If the atom is non terminal, the recursive case is to generate a string with three parts. The first part is its atomic number. The second part is the ordered atomic numbers of its neighbors. The third part is the ordered recursive strings for its neighbors. Ordering here occurs by using standard string comparison. Because we are using 3-digit numbers and the periodic table contains no elements with a 4^+-digit atomic number, alphabetic ordering will sort the numbers into numeric ordering[1]. Furthermore, alphabetic ordering will give us the correct priority order because alphabetic ordering looks for the first point of difference scanning left to right and the algorithm is placing numbers in the order they are to be considered for CIP. Consider the branches in Figure D.3 and their numerical values. Here the Oxygen gets the highest priority, the 2-carbon chain the next highest priority, and the methyl group the lowest priority. The differences come up quickly in the string comparison. Note that using numerical comparison here would generate incorrect answers because the longer branches would get higher priority. In fact, the format of the strings may diverge wildly after the point of first difference with numbers no longer meaning the same thing in parallel[2].

In the more complicated example of Figure D.4 , the branching carbons make it difficult to know where the algorithm should proceed. However, by sorting the substrings as recursive generation is occurring, the longer branches are placed first in each string and the Bromine vs. Hydrogen difference comes first in the strings to be compared.

[1] An extension to the system to account for differences in isotopes is a straightforward addition of digits to each atom's value.

[2] For example, the third number in the O-H branch refers to the local atomic number of a hydrogen. The third number in the C-H branch refers to the atomic number of a hydrogen adjacent to the local carbon.

O-H 008001001

C-H 006001001001001001001

C-C-H 006006001001006001001001001001001001

Figure D.3: *Examples of constituent branches and the generated CIP score strings*

C^{-C-O-H}_{-C} 0060060060010060080010010080**01**00100100...

$C^{-C-O-Br}_{-C}$ 006006006001006008001001008**035**03500100...

Figure D.4: *Examples of branching constituent branches and the generated CIP score strings*

D.2 Calculating a Dihedral Angle

The calculation of a dihedral angle may seem difficult based on its definition of the angle formed by the the projections of the outside vectors onto the plane normal to the internal vector. However, there is a simple means of calculating this requiring no projections by using an alternate definition. Namely, that a dihedral angle is the angle between two planes defined by the points ABC and BCD. This can be calculated by taking the angle between two plane normals. Figure D.5 shows the code for this function. In the function, the subtraction of Points makes a Vector and the Angle function calculates the angle between two vectors using the standard law of cosines approach. A good explanation of this calculation can be found in the dissertations of Rainey [68] and Bekker [9].

D.3 Determining Turn Direction

Figures D.6 and D.7 give code for calculating whether turns are to the left or right in 2D and 3D. These calculations are commonly used in the IM3 terms of Chapter 4. The 3D problem is reduced to the 2D problem by converting the points to the $y = 0$ plane.

```
public double DihedralAngle(Point A, Point B, Point C, Point D)
{
    // Calculate normals between the outside vectors and the center vector.
    Vector normFront = (C - B).Cross(A - B);
    Vector normBack = (D - C).Cross(B - C);
    // Take the angle between the normals and check if this angle
    // is positive or negative.
    double angle = normFront.Angle(normBack);
    bool signNeg = (A - B).Dot(normBack) >= 0;

    // If the length of one of the vectors is 0, this is not useful info.
    // 0 is more useful data than NaN
    if (double.IsNaN(angle))
    {
        return 0;
    }
    // Check if we need to negate the angle
    if (signNeg)
    {
        angle = 2 * PI - angle;
    }
    return angle;
}
```

Figure D.5: *Algorithm for calculating the dihedral angle $\angle ABCD$. Points B and C form the center of the torsion with A adjacent to B and D adjacent to C.*

```
public bool LeftTurn2D(Point2D A, Point2D B, Point2D C)
{
    double determinant = Determinant(A.X, A.Y, 1, B.X, B.Y, 1, C.X, C.Y, 1);
    return determinant < 0.0;
}
```

Figure D.6: *Algorithm for calculating determining the turn direction of the 2D angle $\angle ABC$. If the determinant of the calculated matrix is less than zero, the turn is left. Similarly, if the determinant is greater than 0 the turn is right and the determinant equaling zero means the angle is straight. Note that this is for a screen coordinate system where the Y axis is inverted.*

```
public bool LeftTurn3D(Point3D A, Point3D B, Point3D C, Vector Look)
{
    // Projecting points onto a plane through the origin.
    Matrix projectionMatrix = new Matrix(
        Look.Y*Look.Y + Look.Z*Look.Z,-Look.X*Look.Y,-Look.X*Look.Z,0,
        -Look.Y * Look.X,Look.X*Look.X + Look.Z*Look.Z,-Look.Y*Look.Z,0,
        -Look.Z * Look.X,-Look.Z * Look.Y,Look.X*Look.X + Look.Y*Look.Y,0,
        0,0,0,1);

    A = projectionMatrix * A;
    B = projectionMatrix * B;
    C = projectionMatrix * C;
    // Now the points are projected onto the correct plane,
    // we just need to rotate that plane to be able to
    // take off the Z coordinate.
    Matrix rotationMatrix = RotationForAlignment(Look, new Vector(0, 0, -1));
    A = rotationMatrix * A;
    B = rotationMatrix * B;
    C = rotationMatrix * C;
    // Now all the points have been converted so that they
    // are on the XY plane with the look vector looking down on them.
    return LeftTurn2D(new Point2D(A.X, -A.Y),
        new Point2D(B.X, -B.Y), new Point2D(C.X, -C.Y));
}
```

Figure D.7: *Determining a left or right turn in 3D. We reduce this to the 2D problem by projecting the points onto a plane through the origin [7] and rotating that plane to that the look vector is on the negative Y axis. Here RotationForAlignment finds the rotation matrix which would make the two vectors colinear. This is the rotation matrix about the axis of their cross product through the angle between the vectors.*

Bibliography

[1]

[2] Accelrys. Insight II web site. `http://www.accelrys.com/products/insight/index.html`.

[3] ACDLabs. ACD/ChemSketch web site. `http://www.acdlabs.com/products/chem_dsn_lab/chemsketch/`.

[4] Norman L. Allinger. Conformational analysis. 130. MM2. a hydrocarbon force field utilizing V1 and V2 torsional terms. *Journal of the American Chemical Society*, 99:8127 – 8134, 1977.

[5] Christine Alvarado and Michael Lazzereschi. Properties of real-world digital logic diagrams. In *1st International Workshop on Pen-based Learning Technologies (PLT)*, 2007.

[6] N. Ashby and W. E. Brittin. Thomson's problem. *American Journal of Physics*, 54(9):776–777, 1986.

[7] Martin John Baker. Maths - plane, surface and area. `http://www.euclideanspace.com/maths/geometry/elements/plane/`.

[8] Ross Bannister. Some vector algebra and the generalized chain rule. `http://www.met.reading.ac.uk/~ross/Documents/Chain.html`.

[9] Hendrik Bekker. *Molecular dissociation induced by electron transfer to multicharged ions*. PhD thesis, University of Groningen, 1996.

[10] Oliver Bimber, L. Miguel Encarnação, and André Stork. A multi-layered architecture for sketch-based interaction within virtual environments. *Computers & Graphics*, 24(6):851–867, 2000.

[11] Arnaud Blondel and Martin Karplus. New formulation for derivatives of torsion angles and improper torsion angles in molecular mechanics: Elimination of singularities. *Journal of Computational Chemistry*, 17(9):1132–1141, 1996.

[12] Angela Brennecke and Tobias Isenberg. 3D shape matching using skeleton graphs. In *Simulation and Visualisierung (Sim Vis)*, pages 299–310, 2004.

[13] Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States, S. Swaminathan, and Martin Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4(2):187–217, 1983.

[14] Samuel Bryfczynski. OrganicPad: Organic chemistry learning tool. In *Workshop on the Impact of Pen-based Technology on Education (WIPTE) Posters*, 2007.

[15] Ulrich Burkert and Norman L. Allinger. *Molecular Mechanics.* American Chemical Society, Washington D.C., 1982.

[16] Declan Butler. Electronic notebooks: A new leaf. *Nature*, 436:20–21, July 2005.

[17] CambridgeSoft. ChemDraw web site. `http://www.cambridgesoft.com/software/ChemDraw/`.

[18] Kumar Chellapilla, Patrice Simard, and Ahmad Abdulkader. Allograph based writer adaptation for handwritten character recognition. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[19] T. Cieplak and Janusz L. Wisniewski. A new effective algorithm for the unambiguous identification of the stereochemical characteristics of compounds during their registration in databases. *Molecules*, 6:915–926, 2001.

[20] David E. Clark, Gareth Jones, and Peter Willett. Pharmacophoric pattern matching in files of three-dimensional chemical structures: Comparison of conformational-searching algorithms for flexible searching. *Journal of Chemical Information Computer Sciences*, 34:197–206, 1994.

[21] Peter Clote and Rolf Backofen. *Computational Molecular Biology, An Introduction*, chapter Structure Prediction. John Wiley and Sons, Ltd., 2000.

[22] Ronald A. Cole, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, and Victor Zue. *Survey of the State of the Art in Human Language Technology.* Studies in Natural Language Processing. Cambridge University Press, 1997.

[23] Elias James Corey and Xue-Min Cheng.

[24] Gordon M. Crippen. A novel approach to calculation of conformation: Distance geometry. *Journal of Computational Physics*, 24:96–107, 1977.

[25] Gordon M. Crippen and Timothy F. Havel. Stable calculation of coordinates from distance geometry. *Acta Crystallographica*, A34:282–284, 1978.

[26] Ping Du and Joseph A. Kofman. Electronic laboratory notebooks in pharmaceutical R&D: On the road to maturity. *Journal of the Association for Laboratory Automation*, 12(3):157–165, June 2007.

[27] Eleanor Duckworth. *"The Having of Wonderful Ideas" & Other Essays on Teaching & Learning.* Teachers College Press, 1987.

[28] B. Edwards and V. Chandran. Machine recognition of hand-drawn circuit diagrams. In *ICASSP '00. Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3618–3621, 2000.

[29] Paul W. Finn, Dan Halperin, Lydia E. Kavraki, Jean-Claude Latombe an Rajeev Motwani, Christian Sheton, and Suresh Venkatasubramanian. Geometric manipulation of flexible ligands. In *Proceedings of the First ACM Workshop on Applied Computational Geometry*, 1996.

[30] Andrew S. Forsberg, Mark Dieterich, and Robert C. Zeleznik. The music notepad. In *ACM Symposium on User Interface Software and Technology*, pages 203–210, 1998.

[31] Jeremy G. Frey. Comb-e-chem - an e-science research project. In Martyn Ford, David Livingstone, John Dearden, and Han Van der Waterbeemd, editors, *EuroQSAR 2002 Designing Drugs and Crop Protectants: processes, problems and solutions*, pages 395–398. Blackwell, Oxford, UK, 2003.

[32] Leslie Gennari, Levent Burak Kara, and Thomas F. Stahovich. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers and Graphics*, 29(4):547–562, 2005.

[33] Ray Genoe, John A. Fitzgerald, and Tahar Kechadi. A purely online approach to mathematical expression recognition. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[34] John K. Gilbert. *Visualization in Science Education*, chapter Visualization: A Metacognitive Skill in Science and Science Education, pages 1–27. Springer Netherlands, 2005.

[35] Ian J. Grimstead. *Interactive Sketch Input of Boundary Representation Solid Models.* PhD thesis, Cardiff University, 1997.

[36] Tamas E. Gunda. Chemical drawing programs - the comparison of ISIS/Draw, ChemDraw, DrawIt (ChemWindow), ACD/ChemSketch and Chemistry 4-D Draw. Review, University of Debrecen, 2007.

[37] Klaus Gundertofte, Tommy Liljefors, Per-Ola Norrby, and Ingrid Pettersson. A comparison of conformational energies calculated by several molecular mechanics methods. *Journal of Computational Chemistry*, 17(4):429–449, 1996.

[38] Patrick Haluptzok, Michael Revow, and Ahmad Abdulkader. Personalization of an online handwriting recognition system. In *Tenth International Workshop on Frontiers in Handwriting Recognition (Posters)*, 2006.

[39] Tracy Hammond and Randall Davis. Automatically transforming symbolic shape descriptions for use in sketch recognition. In *AAAI*, pages 450–456, 2004.

[40] William D. Harvey and Matthew L. Ginsberg. Limited discrepancy search. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95); Vol. 1*, pages 607–615, Montréal, Québec, Canada, August 20-25 1995. Morgan Kaufmann, 1995.

[41] Allison Heath, Lydia Kavraki, and Amarda Shehu. Energy functions. Connexions Web Module 11449 - `http://cnx.rice.edu`, 2005.

[42] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on computer graphics and interactive techniques*, pages 203–212, New York, NY, USA, 2001. ACM.

[43] Geoff Hutchison. OpenBabel file formats. `http://openbabel.sourceforge.net/wiki/Category:Formats`, 2007. Wiki entry for molecule file formats currently supported by the OpenBabel molecule file converter. (Referenced 2007).

[44] Takeo Igarashi and John F. Hughes. Smooth meshes for sketch-based freeform modeling. In *I3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 139–142, New York, NY, USA, 2003. ACM.

[45] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3D freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[46] IUPAC. Basic terminology of stereochemistry (IUPAC recommendations 1996). *Pure and Applied Chemistry*, 68(12):2193–2222, 1996.

[47] Yingying Jiang, Xugang Wang, Hongan Wang, and Guozhong Dai. Chemteach: A speech and pen-based multimodal presentation system. In *Seventh Asia-Pacific Conference on Computer-Human Interaction*, 2006.

[48] Olga Karpenko. *Algorithms and Interfaces for Sketch-Based 3D Modeling*. PhD thesis, Brown University, 2007.

[49] Olga A. Karpenko and John F. Hughes. Implementation details of SmoothSketch: 3D free-form shapes from complex sketches. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 51, New York, NY, USA, 2006. ACM Press.

[50] Dae Hyun Kim and Myoung-Jun Kim. A curvature estimation for pen input segmentation in sketch-based modeling. *Computer-Aided Design*, 38(3):238–248, 2006.

[51] Dae Hyun Kim and Myoung-Jun Kim. A new modeling interface for the pen-input displays. *Computer-Aided Design*, 38(3):210–223, 2006.

[52] George Labahn, Scott MacLean, Mirette Marzouk, Ian Rutherford, and David Tausky. A preliminary report on the mathbrush pen-math system. In *Proceedings of Maple Conference 2006*, pages 162–178. Maplesoft, 2006.

[53] P. Labute. An efficient algorithm for the determination of topological RS chirality. *Journal of the Chemical Computing Group*, November 1996.

[54] Joseph J. LaViola Jr. and Robert C. Zeleznik. MathPad^2: a system for the creation and exploration of mathematical sketches. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 432–440, New York, NY, USA, 2004. ACM Press.

[55] Andrew R. Leach. A survey of methods for searching the conformational space of small and medium-sized molecules. *Reviews in Computational Chemistry*, 2:1–55, 1991.

[56] H. Lipson and M. Shpitalni. Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-aided Design*, 28(8):651–663, 1996.

[57] m. c. schraefel, Gareth V. Hughes, Hugo R. Mills, Graham Smith, Terry R. Payne, and Jeremy Frey. Breaking the book: translating the chemistry lab book into a pervasive computing lab environment. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 25–32, New York, NY, USA, 2004. ACM Press.

[58] Dinesh Manocha and John F. Canny. Detecting cusps and inflection points in curves. *Computer Aided Geometric Design*, 9(1):1–24, 1992.

[59] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 24(3):1142–1147, 2005.

[60] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, page 42, New York, NY, USA, 2007. ACM.

[61] Neysa Nevins, Kuohsiang Chen, and Norman L. Allinger. Molecular mechanics (MM4) calculations on alkenes. *Journal of Computational Chemistry*, 17(5-6):669–694, 1996.

[62] NIH. NIH guide to molecular modeling. `http://cmm.cit.nih.gov/modeling/guide_documents/`.

[63] Tom Ouyang and Randall Davis. Recognition of hand drawn chemical diagrams. In *Second Annual CSAIL Student Workshop*, 2006.

[64] Tom Ouyang and Randall Davis. Recognition of hand drawn chemical diagrams. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 846–851, 2007.

[65] Roy Pargas, Melanie Cooper, Calvin Williams, , and Samuel Bryfczynski. OrganicPad: A tablet PC based interactivity tool for organic chemistry. In *1st International Workshop on Pen-based Learning Technologies, PLT*, 2007.

[66] James A. Pittman. Handwriting recognition: Tablet PC text input. *Computer*, 40(9):49–54, 2007.

[67] J.W. Ponder and D.A. Case. Force fields for protein simulations. *Advances in Protein Chemistry*, 66:27–85, 2003.

[68] J.K. Rainey. *Collagen structure and preferential assembly explored by parallel microscopy and bioinformatics.* PhD thesis, University of Toronto, 2003.

[69] Anthony K Rappé and Carla J. Casewit. *Molecular Mechanics across Chemistry.* University Science Books, 1997.

[70] John D. Roberts and Marjorie C. Caserio. *Basic Principles of Organic Chemistry.* W. A. Benjamin, Inc., Menlo Park, CA., second edition, 1977.

[71] Lisa Ann Scott, Robert Zimmerman, Hsin-Yi Chang, Mary Heitzman, Joseph Krajcik, Kate Lynch McNeill, Chris Quintana, and Elliot Soloway. Chemation: a handheld chemistry modeling and animation tool. In *IDC '04: Proceedings of the 2004 conference on Interaction design and children*, pages 145–146, New York, NY, USA, 2004. ACM.

[72] Tevfik Metin Sezgin. Feature point detection and curve approximation for early processing of free-hand sketches. Master's thesis, Massachusetts Institute of Technology, 2001.

[73] Andrew Smellie, Scott Kahn, and Steven Teig. Analysis of conformational coverage. 1. Validation and estimation of coverage. *Journal of Chemical Information and Computer Sciences*, 35:285–294, 1995.

[74] Mike Stieff. Connected chemistry - a novel modeling environment for the chemistry classroom. *Journal of Chemical Education*, 82(3):489–493, 2005.

[75] Mike Stieff and Michelle McCombs. Increasing representational fluency with visualization tools. In *Proceedings of the Seventh International Conference of the Learning Sciences (ICLS)*, volume 1, pages 730–736, 2006.

[76] Ivan E. Sutherland. *Sketchpad, A Man-Machine Graphical Communication System.* PhD thesis, Massachusetts Institute of Technology, 1963.

[77] Symyx. Chime molecule viewer browser plugin. `http://www.mdl.com/products/framework/chime/`.

[78] Symyx. MDL/Draw web site. `http://www.mdl.com/products/framework/mdl_draw/index.jsp`.

[79] Dana Tenneson. ChemPad: A pedagogical tool for exploring handwritten organic molecules. Master's thesis, Brown University, Providence, RI, May 2005.

[80] Dana Tenneson. Report on the development of ChemPad for teaching organic chemistry students to visualize three-dimensional molecular structures. `http://graphics.cs.brown.edu/research/chempad/report.pdf`, 2005.

[81] Dana Tenneson. ChemPad: Visualizing molecules in three dimensions. In Jane C. Prey, Robert H. Reed, and Dave A. Berque, editors, *The Impact of Tablet PCs and Pen-based Technology on Education, 2007: Beyond the Tipping Point*. Purdue University Press, 2007. Monographs from the 2007 Workshop on the Impact of Pen-Based Technologies on Education (WIPTE).

[82] Dana Tenneson and Sascha Becker. ChemPad: Generating 3D molecules from 2D sketches. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Posters*, page 87, New York, NY, USA, 2005. ACM Press.

[83] J. J. Thomson. On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. *Philosophical Magazine*, 7(39):237–265, 1904.

[84] University of Colorado at Boulder. Stereochemistry (substituted butanes, cyclopentanes, and cyclohexanes). `http://orgchem.colorado.edu/courses/ModelExStereo.pdf`, 2004.

[85] P. A. C. Varley and R. R. Martin. A system for constructing boundary representation solid models from a two-dimensional sketch. In *GMP*, pages 13–32, 2000.

[86] K. Peter C. Vollhardt and Neil E. Schore. *Organic Chemistry: Structure and Function*. W.H. Freeman and Company, third edition, 1999.

[87] L. S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1980.

[88] Cheuk-San Wang. *Determining Molecular Conformation from Distance or Density Data*. PhD thesis, Massachusetts Institute of Technology, 2000.

[89] Junmei Wang, Wei Wang, Peter A. Kollman, and David A. Case. Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics and Modelling*, 25(2):247–260, 2006.

[90] Junmei Wang, Romain M. Wolf, James W. Caldwell, Peter A. Kollman, and David A. Case. Development and testing of a general amber force field. *Journal of Computational Chemistry*, 25(9):1157–1174, 2004.

[91] Arieh Warshel and Shneior Lifson. Consistent force field calculations ii. crystal structures, sublimation energies, molecular and lattice vibrations, molecular conformations, and enthalpies of alkanes. *The Journal of Chemical Physics*, 53(2):582–594, 1970.

[92] Scott J. Weiner, Peter A. Kollman, David A. Case, U. Chandra Singh, Caterina Ghio, Guiliano Alagona, Salvatore Profeta Jr., and Paul Weiner. A new force field for molecular mechanical simulation of nucleic acids and proteins. *Journal of the American Chemical Society*, 106:765–784, 1984.

[93] L. L. White. Unique arrangements of points on a sphere. *The American Mathematical Monthly*, 59(9):606–611, 1952.

[94] Robert Zeleznik and Timothy Miller. Fluid Inking: augmenting the medium of free-form inking with gestures. In *GI '06: Proceedings of the 2006 conference on Graphics interface*, pages 155–162, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[95] Robert Zeleznik, Timothy Miller, Loring Holden, and Joseph J. LaViola, Jr. Fluid Inking: Using punctuation to allow modeless combination of marking and gesturing, 2004.

[96] Robert Zeleznik, Timothy Miller, and Chuanjun Li. Designing UI techniques for handwritten mathematics. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. Eurographics Association, 2007.

[97] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. SKETCH: An interface for sketching 3D scenes. In *SIGGRAPH 96 Conference Proceedings*, pages 163–170. Addison Wesley, 1996.