IMIMOL, A COMPUTER PROGRAM FOR MOLECULAR GEOMETRY SPECIFICATION AND COMPUTATION

DANIEL B. DIX

Abstract. An easy to use and mathematically attractive method for specifying the three-dimensional geometry of molecules is made available in a new computer program: IMIMOL. The program allows the user to interact with the molecule through a natural two-dimensional interface, which also facilitates the building of large molecular structures. Furthermore, the tedium is now removed from certain molecular geometry calculations. We explain all the capabilities of IMIMOL, with detailed instructions for each procedure, and illustrate the power of the program with a biochemical example.

Contents

Date: June 21, 2004.

1. Introduction

The topology of molecules has been a natural source of examples of simple graphs [10]. Each atom determines a vertex of the molecular graph $\mathcal{G}$ and each covalent bond determines an edge of $\mathcal{G}$. Recently the three dimensional geometry of molecules has been shown to be conveniently specified using a graph theoretical structure called a Z-system [3]. In simple terms an (unoriented) Z-system consists of three tree graphs (τ_1, τ_2, τ_3), where τ_1 is a spanning tree in $\mathcal{G}$, τ_2 is a spanning tree in the line graph $L(\tau_1)$ of τ_1, and τ_3 is a spanning tree in $L(\tau_2)$. We label the edges of τ_1 with bond lengths $L > 0$, the edges of τ_2 with bond angles $0 < \theta < 180$, and edges of τ_3 with pairs (d^*, φ), where d^* is an orientation of the tetrahedron d associated to that edge (defined precisely below) and $-180 < \varphi \leq 180$ is a *wedge angle*, i.e. a signed angle between two half-planes sharing the same boundary line. When the Z-system $\Gamma = (\tau_1, \tau_2, \tau_3)$ is labeled in this manner the three dimensional shape of the molecule is uniquely determined [3]. Thus labeled Z-systems are 'mathematical legos', i.e. structures with prescribed three-dimensional geometry which can be combined and manipulated like building blocks. There are two natural joining operations on labeled Z-systems, *tethering* and *gluing* [3]. Both of these result in a labeled Z-system, and also both require the specification of certain additional geometrical data.

Although calculations and constructions with small Z-systems can be done by hand, it quickly becomes very convenient to have a computer program for this sort of thing; hence the program IMIMOL. The graph theoretical nature of Z-systems lends itself nicely to display on a two-dimensional canvas, but it should be emphasized that the appearance of the trees (τ_1, τ_2, τ_3) on this two dimensional canvas has nothing to do with the three-dimensional shape of the associated molecule; that is determined by the numerical labels of the edges of these trees as we explained above. IMIMOL does not directly produce a three-dimensional image of this geometry, but it can export all the information needed to do this easily with other (free) computer programs, such as RASMOL. In the remainder of this paper we will assume the reader is using RASMOL.

The author was inspired to create Z-systems and IMIMOL by the study of molecular biology. There are many beautiful and intricate three-dimensional molecular shapes which interact and transform themselves in that field (c.f. [6]), and there was (until IMIMOL) no free, completely flexible, and easy-to-use software system available for building and manipulating these structures (c.f. WebMO [11]). Take for example the problem of building a model of the double helix of DNA. Plastic models can be purchased (they are expensive) for this but they sag under their own weight, and one needs very many pieces (even more expensive) if one is to build other things besides DNA. Clearly a computer building environment facilitates the construction of molecular systems of arbitrary size and complexity. Furthermore, IMIMOL is designed to allow the mathematical questions of molecular geometry to be easily addressed, by exporting geometric data in the form of a MAPLE procedure file. The computer algebra system MAPLE can read this file and solve the mathematical equations whose solutions represent geometric structures of particular interest. This approach was used to classify shapes of hexagons with two-fold symmetry in [3], and it has been used to find the shape of the sugar-phosphate backbone of the double-helix structure of DNA in [4]. In this paper we will not assume the reader knows anything about molecular biology. Z-systems can be viewed as determining an embedding of the molecular graph $\mathcal{G}$ into $\mathbb{R}^3$ up to rigid motions, and so can be considered as a contribution to the subject of three-dimensional graph embedding and drawing.

The latest Windows version of the IMIMOL executable can be obtained at

http://www.math.sc.edu/˜dix/imimol.exe.

IMIMOL uses some graphics library utilities which should also be downloaded at

http://www.math.sc.edu/˜dix/glut32.dll

and kept in the same directory as the IMIMOL executable. A Unix executable can be obtained from the author upon request via e-mail. Online documentation of IMIMOL can be found at

http://www.math.sc.edu/˜dix/imimolhelp.html.

IMIMOL was written in C by Scott Johnson, with financial support from the Industrial Mathematics Institute (IMI) at the University of South Carolina.

The plan for the rest of this paper is as follows. Section 2 will discuss the basics of building and editing a molecule, atom by atom. At the same time the Z-system concepts will be explained in more detail. In section 3 we discuss how to produce three dimensional pictures of our molecules using RASMOL. Also one can produce readable printouts of a molecule or 2D images of canvas representations of molecules. In Section 4 we discuss the building up of large Z-systems by gluing together smaller ones. In section 5 we discuss sites, poses and coordinate transformation matrices, and how to transfer this information directly to the computer algebra system MAPLE. We also apply this idea to compute the shape of an alpha helix.

2. Building and Labeling a Z-system

The Windows version of IMIMOL produces two user interfaces, the main window and a console window. The console window prints certain error and other messages, and currently is for information only. In the Unix version of IMIMOL the console window is the shell from which IMIMOL was run. The main window has a title bar, a pulldown menu bar, the canvas, a list of actions, a list of modes, an atomic element box, a properties box, and some buttons. The default action is *Add*, the default mode is *Atom*, and the default element type is *H*, i.e. hydrogen. The default viewing filter is to show all atoms, bonds, labels, and roots, but to hide all angles and wedges. The default atom label level is to show all suffixes (see below). The default canvas background color is black. See Figure 1.

2.1. Atoms and Atom Labels.

To create an atom, select the action *Add* and the mode *Atom*. Edit the atomic element box to contain the atomic element symbol of the atom you wish to create. The edits become effective without the *Enter* key. Then left click on the canvas to create that sort of atom at that position. The color of the atom depends on the element type. The atom will not appear if the viewing filter is hiding the atoms. To change the viewing filter, go to and choose the *View → Hide/Show → Show Atoms* pulldown menu option. The atom label (or name) is automatically generated, and starts with the element symbol followed by a number, to keep that atom name separate from the others previously created.

To move an atom choose the action *Select* and the mode *Atom*. Then depress and hold the left mouse button over the atom you wish to move. Drag the atom to

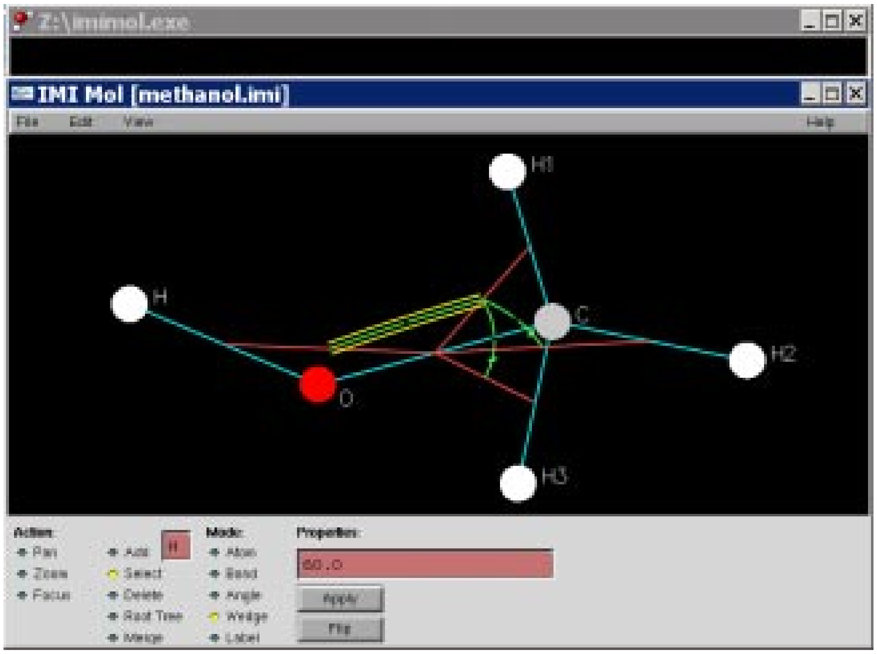

FIGURE 1. The main window of IMIMOL, with the console window above and behind it.

the position you want it and release the mouse button. The atom label will move with the atom. The atom moved will remain selected.

To delete an atom choose the action *Delete* and the mode *Atom*. Then left click on the atom you wish to delete. It should then disappear from the canvas along with its label.

To change the atom name, first select the atom as above. The selected atom will be highlighted with a yellow circle around it. To unselect all atoms left click on the background away from any atoms. The atom name of the selected atom should appear in the properties box. Edit this name as desired and press *Enter* to make the edits effective. The name should change on the canvas. Atom names are assumed to begin with an element symbol. Upper and lower case letters are distinguished. Editing this initial character of the name will change the color of the atom accordingly. Atom names are structured as strings of nonblank characters separated by periods. The first string is the name of the atom within the context of its local group. Each subsequent string, separated by a period from those to the left, is the name of a larger collection of atoms including as a subset all the subcollections named to the left. For example, the name *CA.ALA34.A.PC3.LH4* refers to the C^{α} atom (Calcium has the atom symbol *Ca*) which is part of the amino acid Alanine, which is number 34 on the A chain of the subunit PC3 of the fourth LH complex. As this example shows these names can become lengthy, and IMIMOL allows the atom names to be abbreviated by truncating a variable number of the suffixes, i.e. the rightmost strings. The abbreviation level can be adjusted via the *View* → *Atom Label Level* pulldown menu. Obviously the setting of this abbreviation level can affect the visibilty of any atom label edits.

To reposition an atom name relative to the associated atom choose the action *Select* and the mode *Label*. Each atom label will then have a white dot on its lower left corner. Depress and hold the left mouse button over the white dot of the atom label to be moved and drag it to the desired position, releasing the mouse button when the position is correct.

2.2. **Bonds and Bond Lengths.** To create a bond between two existing atoms select the action *Add* and the mode *Bond*. Then depress and hold the left mouse button over one of the atoms and drag toward the other atom of the bond. Release the mouse button when a blue line connects the two atoms which you desire to be bonded. Bonds move with their endpoint atoms, and are usually unadorned on the IMIMOL canvas. A bond may fail to appear when created if the viewing filter is set to *Hide Bonds*. This can be remedied by going to and choosing the *View* → *Hide/Show* → *Show Bonds* pulldown menu option.

To delete a bond, select the action *Delete* and the mode *Bond*. Then left click on the blank dot at the midpoint of the bond you wish to delete. The bond should disappear.

Often bonds are created between any two atoms which are covalently bonded in a molecule. But IMIMOL knows almost no chemistry, and will allow arbitrary bonding patterns; there may be bonds in IMIMOL which have no chemical meaning. Bonds are allowed to cross on the canvas. Thus arbitrary simple graphs $\mathcal{G}$ can appear on the canvas, and the appearance can be adjusted by moving atoms (or groups of atoms, see below) at the discretion of the user. No aspect of this adjustment affects the three dimensional geometry of the molecule, although the user might choose to try to portray some two dimensional projection of this geometry.

To assign bond lengths, one first chooses the action *Select* and the mode *Bond*, and then left clicks on the midpoint of the desired bond. This selects this bond, and the bond will be highlighted with two yellow lines on either side of the blue bond line. To unselect all bonds left click on the background away from any bonds. The bond length of the selected bond will appear in the properties box. Default bond lengths are assigned by a very primitive algorithm, so they mostly need to be edited. The label of a bond (i.e. its length) is actually any string, which includes numerical values (positive, in angstroms) but can also be variable names. Simply edit the contents of the property box and press *Enter* to make the new label effective.

To specify the *atom/bond tree* τ_1 we must choose a spanning tree subgraph of the graph $\mathcal{G}$. This requires that $\mathcal{G}$ be connected. The vertices of τ_1 are all the atoms of $\mathcal{G}$, but the edges of τ_1 usually comprise a proper subset of the edges of $\mathcal{G}$. To determine this subset of *in-tree* bonds we note that by default all bonds are assumed to be in-tree. To select the *out-of-tree* bonds, which are edges of $\mathcal{G}$ but not edges of τ_1, one first chooses the action *Select* and the mode *Bond*, and then left double-clicks on the midpoints of the bonds to be *out-of-tree*. These out-of-tree bonds are only shown as ghosts. These ghosts are either visible or invisible depending on the viewing filter. To make them invisible go to and choose the *View* → *Hide/Show* → *Hide Out of Tree* pulldown menu option. Their visibility can be restored via *View* → *Hide/Show* → *Show Out of Tree*. Double-clicking on the midpoint of the ghost of an out-of-tree bond will restore its in-tree status. IMIMOL does not check to see if τ_1 is a tree graph until the user tries to write a Z-matrix (see section 3.2).

2.3. **Angles and their labels.** IMIMOL automatically computes the line graph $L(\tau_1)$, although it might be invisible. Its visibilty can be restored via *View* → *Hide/Show* → *Show Angles*. The vertices of $L(\tau_1)$ are the edges of τ_1 (i.e. *in-tree* bonds), and edges of $L(\tau_1)$ are unordered pairs $\{b_1, b_2\}$ of distinct in-tree bonds which share a common atom. IMIMOL denotes edges of $L(\tau_1)$ by red lines connecting the midpoints of the two bonds involved. They are called *angles*.

To further specify the three dimensional geometry using the Z-system formalism one must define the *bond/angle tree* τ_2, which is a spanning tree subgraph of $L(\tau_1)$. The vertices of τ_2 are the same as the vertices of $L(\tau_1)$, i.e. the edges of τ_1 (the in-tree bonds), and as such are already specified. The edges of τ_2 will be called *in-tree* angles, and the edges of $L(\tau_1)$ which are not edges of τ_2 are called *out-of-tree* angles. By default all angles are in-tree. Choose the action *Select* and the mode *Angle*, and then left double-click on the midpoints of the angles to be *out-of-tree*. They are denoted by ghosts, and might be invisible depending on the setting of *View* → *Hide/Show* pulldown menu. Double-clicking on the midpoint of the ghost of an out-of-tree angle will restore its in-tree status. IMIMOL does not check to see if τ_2 is a tree graph until the user tries to write a Z-matrix (see section 3.2).

To assign a label to an angle choose the action *Select* and the mode *Angle*, and then left click on the midpoint of the angle to be labeled. This will select this angle, and it will be highlighted with two yellow lines on either side of the red angle line. To unselect all angles left click on the background away from all angles. The label for the selected angle (the defaults are usually not correct) will be displayed in the property box. The label in the properties box can be edited; then press *Enter* to make the new label effective. The label can be any string (including a number) and is interpreted geometrically as the measure (in degrees strictly between 0 and 180) of the angle in space between the two bonds (thought of as line segments in space), whose vertex is at the common atom. One must be careful to correctly label each in-tree angle, as IMIMOL has no facility for systematically doing this.

2.4. **Wedges and their labels.** IMIMOL automatically computes the line graph $L(\tau_2)$, just as it did for $L(\tau_1)$. The vertices of $L(\tau_2)$ are the edges of τ_2 (i.e. *in-tree* angles), and edges of $L(\tau_2)$ are unordered pairs $\{a, a'\}$ of distinct in-tree angles which share a common bond. IMIMOL denotes edges of $L(\tau_2)$ by green lines connecting the midpoints of the two angles involved. They are called *wedges*.

The shape of a wedge can be continuously deformed into a parabolic arc. Choose the action *Select* and the mode *Wedge*, and then depress and hold the left mouse button over the midpoint of the wedge. Then drag that midpoint and release the mouse button when the wedge has the desired appearance.

The Z-system formalism requires the specification of the *angle/wedge tree* τ_3, which is a spanning tree subgraph of $L(\tau_2)$. The edges of τ_3, called *in-tree* wedges, are determined by excluding the *out-of-tree* wedges following the same pattern established for bonds and angles. Choose the action *Select* and the mode *Wedge*, and then left double-click on the midpoint of the wedges to be considered out-of-tree. The ghosts of out-of-tree wedges can be rendered invisible or visible by the same mechanism as with bonds and angles. The selection of in-tree wedges is also completely analogous to that of in-tree bonds and angles.

The label of a selected wedge has two parts, as mentioned in the introduction, (d^*, φ), and we must explain these parts and their interpretation here. A wedge is an unordered pair of angles $\{a, a'\}$ which share a common bond, and since each angle involves three atoms and the atoms of the common bond are in both angles, we see that a wedge always determines exactly four atoms. The set $d = \{A_0, A_1, A_2, A_3\}$ of the four atom names determined by the wedge is called the *tetrahedron* of the wedge. There are 24 permutations of these four atom names, which are divided into two disjoint subclasses consisting of 12 permutations each. These subclasses are determined by the following rule: if $(i_0, i_1, i_2, i_3), (j_0, j_1, j_2, j_3)$ are permutations of the numbers $\{0, 1, 2, 3\}$ where (j_0, j_1, j_2, j_3) is obtained from (i_0, i_1, i_2, i_3) by a single transposition of two elements (not necessarily adjacent) then $(A_{i_0}, A_{i_1}, A_{i_2}, A_{i_3})$ and $(A_{j_0}, A_{j_1}, A_{j_2}, A_{j_3})$ are in opposite subclasses. Each subclass is called an *orientation* of the tetrahedron d. Let $[A_0, A_1, A_2, A_3]$ denote the orientation containing the permutation (A_0, A_1, A_2, A_3). Thus $[A_0, A_1, A_2, A_3]$ and $[A_1, A_0, A_2, A_3]$ are the two possible orientations of d. To label a wedge we must (in part) choose one of the two orientations of the tetrahedron of the wedge.

Wedges are of two types: *dihedrals* and *impropers*, and the two types have different behavior in regard to choosing an orientation. Both angles a and a' of the wedge $\{a, a'\}$ involve a pair of bonds which share an atom. Let A (resp. A') be the atom name of the common atom of the bonds of the angle a (resp. a'). We say $\{a, a'\}$ is a *dihedral* if $A \neq A'$; We say $\{a, a'\}$ is an *improper* if $A = A'$. Suppose $d = \{A_0, A_1, A_2, A_3\}$ is the tetrahedron of the wedge $\{a, a'\}$, where $\{A_0, A_1, A_2\}$ are the atoms involved in the angle a and $\{A_1, A_2, A_3\}$ are the atoms involved in the angle a'. Thus $\{A_1, A_2\}$ are the atoms involved in the common bond of a and a'. In the dihedral case we have $\{A, A'\} = \{A_1, A_2\}$, although both $(A, A') = (A_1, A_2)$ and $(A, A') = (A_2, A_1)$ are possibilities. Define the *canonical dihedral orientation* of d to be:

$$\begin{cases} [A_0, A_1, A_2, A_3] & \text{if } (A, A') = (A_1, A_2), \\ [A_0, A_2, A_1, A_3] & \text{if } (A, A') = (A_2, A_1). \end{cases}$$

Since $[A_0, A_1, A_2, A_3] = [A_3, A_2, A_1, A_0]$ and $[A_0, A_2, A_1, A_3] = [A_3, A_1, A_2, A_0]$ this choice of orientation does not depend on an ordering of the two angles, i.e. (a, a') or (a', a). The independence of the orientation on the choice of an ordering is why we call it canonical. IMIMOL always assumes every dihedral wedge is labeled (in part) with the canonical dihedral orientation, and so no choice is required of the user to so label a dihedral wedge. In the improper case we have $\{A = A'\} \subset \{A_1, A_2\}$, although both $A = A_1$ and $A = A_2$ are possibilities. In this case a choice of orientation amounts to the same thing as a choice of ordering of the two angles a and a'. Suppose we order the two angles as (a, a'). Then the natural orientation associated to this ordering is:

$$\begin{cases} [A_0, A_1, A_2, A_3] & \text{if } A = A_1, \\ [A_0, A_2, A_1, A_3] & \text{if } A = A_2. \end{cases}$$

On the other hand if the angles are ordered as (a', a) then the same rule for the associated orientation gives:

$$\begin{cases} [A_3, A_1, A_2, A_0] & \text{if } A = A_1, \\ [A_3, A_2, A_1, A_0] & \text{if } A = A_2, \end{cases}$$

and since $[A_0, A_1, A_2, A_3] \neq [A_3, A_1, A_2, A_0]$ and $[A_0, A_2, A_1, A_3] \neq [A_3, A_2, A_1, A_0]$ we find that opposite orderings lead to opposite orientations. So in IMIMOL a choice of orientation of an improper wedge is indicated by an arrow on the green wedge arc, so that the arc points from the first angle to the second angle. To change the orientation of an improper wedge, choose the action *Select* and the mode *Wedge*, and then left click on the desired wedge. Then press the button *Flip* to reverse the orientation of that wedge. The arrow on the green wedge arc should change directions.

The numerical labeling $\varphi \in (-180, 180]$ of a wedge is handled via the properties box exactly as for bonds and angles. But we must explain how this number is interpreted geometrically in conjunction with the chosen orientation. Suppose the tetrahedron associated to the given wedge is $d = \{A_0, A_1, A_2, A_3\}$ and the assigned orientation is $d^* = [A_0, A_1, A_2, A_3]$, where $\{A_1, A_2\}$ are the atoms associated to the common bond. It is always possible to find exactly two permutations of d in the given orientation with this property; the other one is $d^* = [A_3, A_2, A_1, A_0]$. Let $\mathbf{R}_{A_j} \in \mathbb{R}^3$ be the position in space of the atom with name A_j, $j = 0, 1, 2, 3$. Consider the directed line in space through $\mathbf{R}_{A_1}$ and $\mathbf{R}_{A_2}$, where the positive orientation is in the direction $\mathbf{R}_{A_2} - \mathbf{R}_{A_1}$. Consider the half-plane H_0 (resp. H_3) in space bounded by this line and containing $\mathbf{R}_{A_0}$ (resp. $\mathbf{R}_{A_3}$). If the directed line is considered as an axis of right-handed rotation then the label φ represents the angle of rotation needed to carry H_0 into H_3. The full label of the wedge is then (d^*, φ). It follows then that $(-d^*, -\varphi)$ represents then same geometric situation as (d^*, φ), where $-d^* = [A_3, A_1, A_2, A_0]$ is the orientation opposite to d^*. Thus by a proper choice of the orientation it is possible to only use wedge angles φ in the range $0 \le \varphi \le 180$. When $d^* = [A_0, A_1, A_2, A_3]$ and $0 < \varphi < 180$ we also have that

$$\det\begin{pmatrix} 1 & 1 & 1 & 1 \\ \mathbf{R}_{A_0} & \mathbf{R}_{A_1} & \mathbf{R}_{A_2} & \mathbf{R}_{A_3} \end{pmatrix} = \det(\mathbf{R}_{A_2} - \mathbf{R}_{A_1}, \mathbf{R}_{A_0} - \mathbf{R}_{A_1}, \mathbf{R}_{A_3} - \mathbf{R}_{A_1}) > 0.$$

However for dihedrals we will never use the opposite of the canonical orientation, so the full range $-180 < \varphi \le 180$ is required. Note also that if we use the other permutation to represent the same orientation, i.e. $d^* = [A_3, A_2, A_1, A_0]$, then the axis of rotation has its positive direction reversed and the roles of H_0 and H_3 are reversed as well, so the same angle φ describes the same geometric situation as with the other permutation.

2.5. **Editing Z-systems.** The previous sections assume the atom/bond tree τ_1 is perfected first, and then the bond/angle tree τ_2, and finally the angle/wedge tree τ_3. However, one may go back and add or delete atoms and bonds, or move bonds or angles or wedges in tree or out of tree in various orders. Angles and wedges can also be deleted altogether (choose the action *Delete* and the appropriate mode and left click on the midpoint of the edge to be deleted). However any change of the atom/bond tree will inevitably affect both the bond/angle and the angle/wedge graphs. Also any change of the bond/angle tree will inevitably affect the angle/wedge graph. Because the line graphs are automatically generated and the edges of the automatically generated line graph are automatically assumed to be in tree, often the higher graphs must be pruned to produce trees.

3. Visualization of Molecules

3.1. **Sites and the Root Site.** Suppose $\mathcal{N}$ is the set of all atom names for a molecule and suppose $\Gamma = (\tau_1, \tau_2, \tau_3)$ is an unoriented Z-system for this molecule. Thus the set of vertices of τ_1 (i.e. the set of atoms) is in one-to-one correspondence with the set $\mathcal{N}$. A *site* $r = (A_0, A_1, A_2)$ *of* Γ is an ordered triple of distinct atom names from $\mathcal{N}$ where $\{A_0, A_1\}$ determines a vertex of τ_2 (i.e a bond), and

$\{A_0, A_1, A_2\}$ determines a vertex of τ_3 (i.e. an angle). The common atom of the angle could be either A_0 or A_1. If the molecule is embedded in three dimensional space, a Cartesian coordinate system can be defined whose origin is at $\mathbf{R}_{A_0}$, whose x axis is parallel to $\mathbf{R}_{A_1} - \mathbf{R}_{A_0}$, whose y axis is in the half-plane bounded by the x-axis and containing the point $\mathbf{R}_{A_2}$. The direction of the z axis is then determined by the right-hand rule. In order to visualize a molecule in a three dimensional manner it is necessary to determine an embedding into $\mathbb{R}^3$, and for this purpose we define a special site, called the *root site*.

To define the root site in IMIMOL first check the viewing filter (i.e the *View* → *Hide/Show* pulldown menu) to make sure that atoms, bonds, angles, and roots are visible. Then choose the action *Root Tree* and the mode *Atom* and left click on the atom to be the root atom, i.e. A_0. Then choose the mode *Bond* and left click on the bond to be the root bond, i.e. $\{A_0, A_1\}$. The root bond must be incident on the root atom. Then choose the mode *Angle* and left click on the angle to be the root angle, i.e. $\{A_0, A_1, A_2\}$. The root angle must be incident on the root bond. These items will all be highlighted with greater size and alternate colors.

3.2. **3D Viewing using RASMOL.** Once a Z-system has been completely labeled (with numbers, not variable names) and a root site has been defined it is possible to export the molecule in a form which can be visualized using the computer program RASMOL. Choose the *File* → *Export Mopac* pulldown menu option. The canvas will be replaced by an interface through which the used specifies the name of the file to be written. The default name ends in the .zmat file extension, but the part before the period should be edited, e.g. to *mol.zmat*. Press the *OK* button to write a file with the indicated name. A message should appear in the console window confirming the successful Mopac Z-Matrix writing operation, and the IMIMOL canvas should reappear. An error can occur during this step if the graphs (τ_1, τ_2, τ_3) or the root site are not constructed properly, and the error message in the console window may be helpful in finding the error.

The sophisticated molecular graphics program RASMOL is available in both Windows and Unix versions freely from the website:

http://www.umass.edu/microbio/rasmol/.

The following is written for version 2.7.2.1. Upon execution RASMOL produces two user interfaces, a viewing window and a console window. In the console window the user should type: *zap* and *Enter*, and then *load mopac mol.zmat* and *Enter*. On the viewing window choose the *Settings* → *Rotate All* pulldown menu option. The *Display* → *Ball and Stick* option will produce a chemically natural picture of the molecule. Note that bonds will not appear in RASMOL unless their lengths have chemically reasonable values. 3D rotation of the molecule can be accomplished by depressing and holding the left mouse button in the viewing window and dragging. Many other options are available, and we refer the user to the RASMOL website for more guidance.

If the user wishes to change the value of some bond length, bond angle, or wedge angle, and then see the resulting change in 3D, the steps are exactly as we have explained above: editing the appropriate edge label, writing the Z-matrix, zapping the old RASMOL image, loading the new Z-matrix, and adjusting the view to suit the user. The step of writing the Z-matrix, usually performed through the *File* → *Export Mopac* pulldown menu option can be skipped if the first time the Z-matrix is written the user selects the *Auto update* button before pressing *OK*. Then every time *Enter* is pressed to confirm edits in the properties box the Z-matrix file will be automatically written. The steps performed by RASMOL can also be reduced to a single keystroke by utilizing suitable RASMOL scripts.

3.3. **Labeling Atoms in 3D.** The atom labels in IMIMOL (even in abbreviated form) can be exported so as to be seen in a RASMOL image of the molecule. To export the atom labels as they are displayed in IMIMOL choose the *File* → *Export RASMOL Labels* → *All* pulldown menu option. The canvas will be replaced by a file name interface. The file name containing the RASMOL script drawing the atom names has a .lbl file extension. Edit the part of the name before the period appropriately, e.g. to *mol.lbl*, and press *Enter*. Then at the RASMOL command line type *script mol.lbl* and *Enter*. It is important that the RASMOL label script be exported immediately after the Mopac Z-matrix is exported, because this will insure that the labels will be assigned to the correct atoms. If anything changes the order of the atoms in the exported Z-matrix, the label script must be rewritten.

Sometimes the labeling of every atom produces too much clutter in a 3D image. To export the atom labels of a select subset of atoms IMIMOL allows the user to mark those atoms whose labels are to be exported. Choose the action *Select* and the mode *Label*. Each atom label will have a white dot on its lower left corner. Left double click on each label to be marked for exportation. Marked atom labels should be highlighted in white rectangles. Then choose the *File* → *Export RASMOL Labels* → *Marked* pulldown menu option and export the label script file as before. If in RASMOL the molecule is reloaded and the label script executed as above then only the desired labels should appear in the 3D image.

3.4. **Writing Gaussian Z-Matrices.** The numerical labels of bonds, angles, and wedges are not visible in IMIMOL except one at a time. However it is possible to write a Gaussian Z-matrix for the molecule in which all these labels will be visible at once. This can be useful to inspect for errors in labeling or for unusual features, such as long or short bonds, deformed angles, cis peptide bonds or chirality flips (detected in the signs of improper wedge angles). To obtain this view of the molecule first define the root site. Then choose the *File* → *Export Gaussian* pulldown menu option. The exported file will have a .gzmat file extension. The file name of the exported file (e.g. *mol.gzmat*) is determined as before and then the *OK* button is pressed. Success or errors in the Z-matrix writing is handled as with Mopac Z-matrices.

The structure of a Gaussian Z-matrix is discussed in [5]; an example is given in section 5.4. The atoms of the root site determine the first three lines of the Z-matrix but the order of the atoms thereafter depends on a choice of order of picking leaves off the tree τ_3, and this can be done many different ways. IMIMOL keeps all the atoms in a local group together as much as possible, producing a so-called *lexicographical Z-matrix*, based on the concatenated structure of the atom names. This produces a very readable result.

3.5. **Exporting Canvas Images.** IMIMOL produces user adjustable 2D images of molecules and their Z-systems and annotations (see 4.6). These images can be exported as encapsulated Postscript files for inclusion as figures in papers or presentations. The canvas background color is black by default, but this can be changed to white for exporting. Choose the *View* → *Background Color* → *White* pulldown menu option to make this change. Once the canvas has the desired appearance choose the *File* → *Export Postscript* pulldown menu option. The canvas changes to a filename selection interface. The file extension for the encapsulated Postscript file is .eps. Edit the filename before the period (e.g. to *mol.eps*) and press *OK*. The user is assumed to be able to make use of the file so written.

4. Building Larger Systems

4.1. **Canvas Viewing.** Larger systems may not fit entirely on the canvas at once, so IMIMOL provides various features for adjusting the canvas view. To translate all the atoms on the canvas choose the action *Pan*. Then depress and hold the left mouse button on the canvas and drag it in the direction the atoms are supposed to move. Release the mouse button when they are moved far enough. This dragging operation can be repeated as many times as necessary. To enlarge or shrink the molecule (around the center of the canvas) choose the action *Zoom*. Then depress and hold the left mouse button on the canvas and drag it up or to the right (resp. down or to the left) to shrink (resp. enlarge) the molecule. Release the mouse button when the desired scale is achieved. When many atoms are visible at once it is sometimes desirable to focus in on a particular part of the molecule. Although this can be done with a combination of panning and zooming, there is a faster way. First one chooses the action *Focus*. Then depress and hold the left mouse button at the upper left corner and drag to and release the mouse button at the lower right corner of a rectangle on the canvas containing the atoms of interest. This rectangle should be visible in dotted lines before the mouse button is released. Upon release the canvas view should adjust to show (at least) all the atoms within this rectangle as enlarged as possible. To return to a view in which all the atoms are visible choose the *View* → *Recenter Molecule* pulldown menu option.

4.2. **Saving .imi Files.** IMIMOL contains more than just a Z-system and its labels. It contains a particular 2D view of this Z-system together with any root site (see 3.1) and annotations (see 4.6) the user has defined. All this information can be saved for use in building future molecules. To save this information about the current molecule choose the *File* → *Save* pulldown menu option. If there is no name displayed (enclosed in brackets) in the title bar of the IMIMOL main window, then the canvas will change into a file name selection interface. The file extension is .imi for the file containing all this information. Edit the file name before the period (so that the result is e.g. *mol.imi*) and press *OK*. If there aready is a file name displayed in the title bar then the *File* → *Save* operation will assume that the current molecule should be saved under that file name, and hence will not bring up the file name selection interface. To save under a file name different from the current one, use the *File* → *Save As* pulldown menu option and choose the file name as described above. The file *mol.imi* is an ASCII (ordinary text) file, and can be read using a text editor like Emacs (Unix) or Wordpad (Windows). However it is mostly to be imported by IMIMOL and so we will not discuss exactly how all the information is arrayed in the file. The knowledgeable user can debug various IMIMOL difficulties with this file.

4.3. **Importing .imi Files.** The user can create a library of basic molecules, each saved in its own .imi file, which can then be assembled into larger molecules. To read a .imi file to begin a molecule building session choose the action *Select* and the mode *Atom*. Then choose the *File* → *Open* pulldown menu option. Choose the appropriate file name and press *OK*. To add a molecule stored in a .imi file to the current construction choose the *File* → *Add From File* pulldown menu option and choose the file name to be read as before. The atoms added will all be selected, as indicated by yellow circles around them. They may be on top of atoms already present (those atoms will not be selected). To translate all these selected atoms against the background of the unselected atoms depress and hold the left mouse button over one of the selected atoms and drag all of them to the desired location and release the mouse button. The yellow circles around the selected atoms will move as the mouse is dragged, but the atoms themselves will not move until the mouse button is released. The selected atoms will remain selected. To rotate the selected group of atoms (about their geometric center) depress and hold the right mouse button and drag. The yellow circles will rotate clockwise (resp. counterclockwise) if the mouse is dragged downward (resp. upward). When the mouse button is released the atoms themselves will move. The atoms will remain selected. To reflect the group of selected atoms about a horizontal line through the geometric center of the group press the *Flip* button. Reflections about a vertical line can be accomplished via a 90° rotation, a horizontal reflection, followed by $-90°$ rotation. Using these operations the user can position the added atoms wherever desired. The selected atoms cannot be scaled independently of the other atoms.

These translation, rotation, and reflection operations can be applied to any selected group of atoms, not just those resulting from a *File* → *Add From File* operation. First choose the action *Select* and the mode *Atom*. Then depress and hold the left mouse button at the upper left corner and drag to and release the mouse button at the lower right corner of a rectangle on the canvas containing the atoms to be selected. This rectangle should be visible in dotted lines before the mouse button is released. Upon release all the selected atoms will be indicated with surrounding yellow circles. This multiple selection mechanism is called *rubber banding*. Then translations, rotations, and reflections can be accomplished as described earlier. If these atoms are connected to any unselected atoms the bridging bonds (and angles and wedges) will be distorted in the process. To unselect all the atoms left click on the canvas background away from all the atoms; the yellow circles should disappear.

4.4. **Editing Multiple Atom Names.** It is not uncommon to import several copies of the same .imi file, and this can result in several atoms with the same name. Since we want the set of atom names to be in one-to-one correspondence with the set of atoms, IMIMOL provides a means to edit many atom names at once. Suppose a group of atoms is selected (this implies the chosen action is *Select* and the mode is *Atom*). IMIMOL displays a comparison of the atom labels of all the selected atoms in the properties box. If all the atom names in the selected group have the same number of parts, i.e. strings separated by periods, then IMIMOL denotes the nonidentical parts by asterisks separated by periods, whereas the parts which are identical for all the atom labels are displayed in detail. The identical parts may be edited. Additional parts can be added to all the atom names even if there are no parts which are identical. Typically this capability allows the user to add to all atom names in a group a name for the group separated by a period from the previous parts of the atom names (i.e a suffix). Pressing *Enter* causes the edits to become effective.

4.5. **Gluing Molecules Together.** Suppose the IMIMOL canvas contains two separate molecules and no two atom names are the same for the collection of atoms from both molecules. We assume both molecules have at least three atoms. Mathematically in order to glue the Z-systems for these molecules together one must specify two sites: (A_0, A_1, A_2) from one molecule and (B_0, B_1, B_2) from the other. It is also required that the only bond incident on the atom A_0 is $\{A_0, A_1\}$, and that the only bond incident on the atom B_0 is $\{B_0, B_1\}$. In IMIMOL we position the two molecules so that the two bonds $\{A_0, A_1\}$ and $\{B_0, B_1\}$ are nearby and anti-parallel, i.e. A_0 is close to B_1 and A_1 is close to B_0. We remark that the atoms A_0 and B_0 will be lost in the gluing process, and they are frequently given a nonchemical element type, such as R_1 or X_2, and the color gives a visual cue that these atoms are at a gluing site, and do not represent a chemically accurate state before gluing. To glue in IMIMOL first check the viewing filter (the *View* → *Hide/Show* pulldown menu) to make sure atoms, bonds, angles, wedges, and Out of Tree are all visible. Choose the action *Merge* and the mode *Atom*. Depress and hold the left mouse button over the atom A_1 and drag it over onto the atom B_0 and release the mouse button. Then depress and hold the left mouse button over the atom B_1 and drag it over onto the atom A_0 and release the mouse button. A new bond connecting A_1 to B_1 should be visible. Its label will be the same as that of $\{B_0, B_1\}$ before the gluing operation, but the label should be edited if necessary. To finish the glued Z-system choose the action *Select* and the mode *Wedge*. Double left click on the midpoint of the ghost wedge connecting the angle $\{B_1, A_1, A_2\}$ to the angle $\{A_1, B_1, B_2\}$. (This is how the atoms A_2 and B_2 enter the gluing operation.) This new wedge will always be a dihedral, and its numerical label (the value must be supplied by the user) is assigned as described in 2.4.

4.6. **Annotations.** When finding one's way around in large molecules it is helpful to have notes or labels describing larger subunits within the molecule. We call these *annotations*. To define an annotation choose the action *Add* and the mode *Label*. Check the *View* → *Hide/Show* pulldown menu to make sure labels are visible. Then left click on the canvas at the lower left corner of a rectangle which will enclose the annotation. A vertical linear text cursor (yellow) will appear surrounded by a yellow dotted rectangle. The height of the cursor will be the height of the text. It can be increased or decreased with the *arrow up* or the *arrow down* keys. The thickness of the cursor will be the thickness of the lines in the text. It can be increased or decreased with the *Page Up* or the *Page Dn* keys. Type the text of the annotation and press *Enter* to finish defining the annotation. To edit or reposition an annotation choose the action *Select* and the mode *Label*. Each label will be drawn with a white dot at its lower left corner. Double click with the left mouse button over the white dot of the annotation you wish to edit. The yellow dotted surrounding box will appear with the cursor positioned at the end of the text. The text can be edited as usual, or its size or weight can be adjusted. The *Enter* key completes the edits. To reposition, depress and hold the left mouse button over the white dot of the annotation to be moved and drag it to its new position, releasing the mouse button when done.

Annotations do not scale up in size when one zooms away from the molecule. Also, the default size and weight of an annotation is either that of atom labels or that of the previous annotation defined. So when dealing with large molecules it is best to create annotations when zoomed in close to the molecule, and then increase its size and weight and refine its position after zooming out. Annotations can be deleted by choosing the action *Delete* and the mode *Label* and then single left clicking on the white dot of the annotation to be deleted. Atom labels cannot be deleted or adjusted in size or weight.

4.7. **Imposing Structures from PDB Files.** Many large molecules are flexible and it is very rare that the dihedral wedge angles governing interesting shapes of the molecule are published in the literature or in online databases. Rather the main database [8] contains *pdb files*, which contain one line for each atom in the molecule. Each line contains information relevant to the atom name as well as the three Cartesian coordinates of the atom in a particular embedding of the molecule. Another file format (not widely used) is the *xyz file*, which includes an atom name and three three Cartesian coordinates per line, with one line per atom in the molecule. It is clearly possible to completely label a Z-system if all this information is available.

One problem with many pdb files is that they often do not contain the coordinates for many of the hydrogen atoms of the molecule. Fortunately these atoms can be added more or less automatically, and a program MOLPROBITY exists for this [9]. The output of this program is a new pdb file with all the hydrogens added.

The main problem with pdb files is that they use a different system of atom naming than IMIMOL does. To address this problem IMIMOL provides a simple conversion from pdb to xyz, where approximate atom names are constructed by concatenating atom name information available in the pdb file. To perform this conversion choose the *File → Convert PDB to XYZ* pulldown menu option. The canvas is replaced by a file name interface. The pdb file to be converted has a name which ends with the .pdb file extension. Edit the part before the period (to get e.g. *mol.pdb*) and press *OK* to read the pdb file and write the corresponding xyz file (i.e. *mol.xyz*). The user may need to edit (using a text editor) the resulting xyz file to correct the file names.

Once we have a xyz file giving coordinates for atoms with the same names as in the the current Z-system we can get IMIMOL to compute all the numerical labels consistent with the molecular shape given in the xyz file. Choose the *File → Impose XYZ on Current* pulldown menu option. The canvas converts to a file name interface prompting the user for the file name (e.g. *mol.xyz*) to be read. Edit the part before the period and press *OK*. The console window will contain any messages about unmatched atoms or errors. These can result from errors in the Z-system or in the underlying pdb file. If the algorithm succeeds then the labels will now reflect the structure in the given xyz file.

5. Molecular Geometry Computations

5.1. **Converting Between Z-Systems.** There is not always a unique Z-system for every molecular system. For example, when two molecules undergo a chemical reaction there will be one Z-system which is suitable for describing the reactants and another Z-system suitable for the products. During the reaction certain covalent bonds break and others form and this process is accompanied by geometric transformations which can be viewed using either of the two Z-systems. This is a natural context where we might wish to convert from one Z-system to another, retaining the same underlying geometrical shape of the atoms involved.

IMIMOL accomplishes such conversions via a mediating xyz file. Suppose the current Z-system is completely labeled and a root site has been defined (see 3.1). To produce the xyz file choose the *File* → *Export XYZ* pulldown menu option. On the file name interface choose the name (e.g. *mol.xyz*) of the xyz file to be written and press *OK*. Any error messages will appear in the console window.

Now suppose one loads a new Z-system which describes the same collection of atoms or a larger collection. For example, choose the *File* → *New* pulldown menu option to clear the old Z-system. Then use the *File* → *Open* pulldown menu option to import a imi file as described in 4.3. Alternately one can edit the old Z-system to produce the new Z-system (see 2.5); this approach insures the same atom names are the basis of both Z-systems. To find the numerical labels of the edges of the new Z-system which correspond to the molecular shape described by the xyz file, choose the *File* → *Impose XYZ on Current* pulldown menu option, choose the file name of the xyz file, and press *OK*. Check the console window for error messages. If the new Z-system was for a larger collection of atoms than the old Z-system then some of the labels of the new Z-system will not be determined.

5.2. **Sites, Poses, and Coordinate Transformation Matrices.** In section 3.1 a site $r = (A_0, A_1, A_2)$ belonging to a Z-system was defined. It was motivated by the fact that if each of the atoms $A \in \mathcal{N}$ is assigned a position $\mathbf{R}_A \in \mathbb{R}^3$ then the site r determines a Cartesian coordinate system, or *pose* $E_r(\mathbf{R}) = (\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$. The origin of this coordinate system is $\mathbf{e}_0 = \mathbf{R}_{A_0}$. The x-axis of this coordinate system is determined by the unit vector $\mathbf{e}_1$, which is parallel to $\mathbf{R}_{A_1} - \mathbf{R}_{A_0}$. The y-axis of this coordinate system is determined by the unit vector $\mathbf{e}_2$, which lies in the half-plane bounded by the x-axis and containing $\mathbf{R}_{A_2}$. The unit vector $\mathbf{e}_3$ in the direction of the z-axis is then determined by the right-hand rule. In formulae we have

$$\mathbf{e}_1 = \frac{\mathbf{R}_{A_1} - \mathbf{R}_{A_0}}{\|\mathbf{R}_{A_1} - \mathbf{R}_{A_0}\|}, \qquad \mathbf{e}_2 = \frac{(\mathbf{1} - \mathbf{e}_1\mathbf{e}_1^T)(\mathbf{R}_{A_2} - \mathbf{R}_{A_0})}{\|(\mathbf{1} - \mathbf{e}_1\mathbf{e}_1^T)(\mathbf{R}_{A_2} - \mathbf{R}_{A_0})\|}, \qquad \mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2,$$

where $\mathbf{1}$ denotes the 3×3 identity matrix and $\mathbf{e}_1^T$ denotes the 1×3 row vector obtained by transposing the 3×1 column vector $\mathbf{e}_1$. We must exclude certain singular embeddings $\mathbf{R}\colon \mathcal{N} \to \mathbb{R}^3$ so that $E_r(\mathbf{R})$ is well-defined.

In general a *pose* is a list $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, where $\mathbf{e}_0 \in \mathbb{R}^3$ and $X = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is a right-handed orthonormal basis of $\mathbb{R}^3$. If $\mathbf{x}$ is a point in space it can be described by the coordinate vector $\mathbf{c} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ in the pose E, i.e. $\mathbf{x} = \mathbf{e}_0 + \mathbf{e}_1 x + \mathbf{e}_2 y + \mathbf{e}_3 z = \mathbf{e}_0 + X\mathbf{c} = (\mathbf{e}_0, X)\begin{pmatrix} 1 \\ \mathbf{c} \end{pmatrix}$. If $E = (\mathbf{e}_0, X)$ and $E' = (\mathbf{e}_0', X')$ are any two poses then there is a unique 4×4 matrix $\mathcal{A}$, whose first row is $(1, 0, 0, 0)$, such that $E' = E\mathcal{A}$. In fact we have

$$(\mathbf{e}_0', X') = (\mathbf{e}_0, X)\begin{pmatrix} 1 & (0,0,0) \\ \mathbf{b} & A \end{pmatrix} = (\mathbf{e}_0 + X\mathbf{b}, XA),$$

so that $A = X^{-1}X'$ and $\mathbf{b} = X^{-1}(\mathbf{e}_0' - \mathbf{e}_0)$. The matrix $\mathcal{A} = \begin{pmatrix} 1 & (0,0,0) \\ \mathbf{b} & A \end{pmatrix}$ is called a *coordinate transformation matrix*. This is because if the space point $\mathbf{x}$ is represented using two coordinate vectors $\mathbf{c}, \mathbf{c}'$ with respect to the poses E, E' then

$\begin{pmatrix}1\\ \mathbf{c}\end{pmatrix} = \begin{pmatrix}1 & (0,0,0)\\ \mathbf{b} & A\end{pmatrix}\begin{pmatrix}1\\ \mathbf{c}'\end{pmatrix}$, as can be seen by multiplying both sides of $E' = E\mathcal{A}$ on the right by $\begin{pmatrix}1\\ \mathbf{c}'\end{pmatrix}$.

If r, r' are two sites belonging to the Z-system Γ then for every three dimensional embedding $\mathbf{R}$ of the molecule consistent with the numerical labels of Γ the coordinate transformation matrix $\mathcal{A}$ such that $E_{r'}(\mathbf{R}) = E_r(\mathbf{R})\mathcal{A}$ can be computed in terms of the numerical labels of Γ and is independent of the particular embedding $\mathbf{R}$ used to determine the poses. It was proved in [3] that there exist a sequence $r = r_0, r_1, \ldots, r_m = r'$ of distinct sites belonging to Γ such that each successive pair (r_{j-1}, r_j), where $1 \leq j \leq m$, is one of the following three types:

(1) r_j is obtained from r_{j-1} by interchanging the first two atom names A_0 and A_1. $\{A_0, A_1\}$ is a bond of Γ and suppose it is labeled with $L > 0$. Then $E_{r_j}(\mathbf{R}) = E_{r_{j-1}}(\mathbf{R})T_1(L)$, where

$$T_1(L) = \begin{pmatrix}1 & 0 & 0 & 0\\ L & -1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{pmatrix}.$$

(2) r_j is obtained from r_{j-1} by interchanging the last two atom names A_1 and A_2. $\{A_0, A_1, A_2\}$ is an angle of Γ and suppose it is labeled with $\theta > 0$. Then $E_{r_j}(\mathbf{R}) = E_{r_{j-1}}(\mathbf{R})T_2(\theta)$, where

$$T_2(\theta) = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & \sin\theta & 0\\ 0 & \sin\theta & -\cos\theta & 0\\ 0 & 0 & 0 & -1\end{pmatrix}.$$

(3) $r_{j-1} = (A_0, A_1, A)$ and $r_j = (A_0, A_1, A')$, where there is a wedge of Γ connecting the angles $\{A_0, A_1, A\}$ and $\{A_0, A_1, A'\}$. Suppose the wedge is labeled with $(\pm[A_0, A_1, A, A'], \varphi)$. Then $E_{r_j}(\mathbf{R}) = E_{r_{j-1}}(\mathbf{R})T_3(\pm\varphi)$, where

$$T_3(\varphi) = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \cos\varphi & -\sin\varphi\\ 0 & 0 & \sin\varphi & \cos\varphi\end{pmatrix}.$$

It follows that $\mathcal{A}$ can be written as a product of m matrices, each of which is one of the above three types.

5.3. **Exporting Maple Procedures.** IMIMOL provides a way of computing this matrix $\mathcal{A}$ symbolically. First, the root site must be defined to be r, as discussed in 3.1. Next, and alternate site r' must be defined. This is done exactly as for the root site except when the alternate root atom, bond, or angle is being selected the *Shift* must be held down while left clicking on the desired atom, bond, or angle. The alternate root vertices will be highlighted in different colors from the usual root site. Then choose the *File* → *Export Maple Procedure* → *No conversion* pulldown menu option if the angular labels are already in radians. Choose the *File* → *Export Maple Procedure* → *Convert to radians* pulldown menu option if the angular labels are in degrees (the usual situation). The canvas converts to a file name interface.

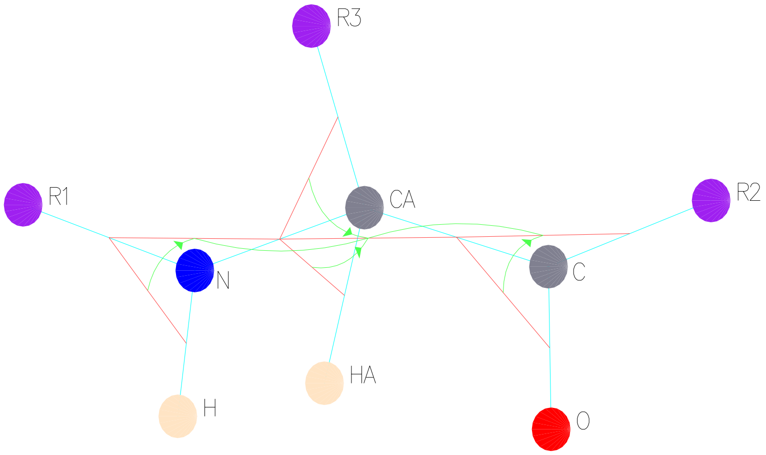

Figure 2. A Model Amino Acid. The dummy atoms $R1$ and $R2$ are to be involved in peptide bond gluing sites, and $R3$ is for attaching the side chain. Bonds are light blue lines, angles red lines, and wedges are green arcs. Improper wedges are shown as arrows.

The Maple procedure file to be exported has a .mp file extension. Edit the part of the file name before the period (so that the result is e.g. *mol.mp*) and press *OK*.

The file *mol.mp* should be edited with a text editor to change the name of the procedure and to add any variables to the list of arguments. This procedure assumes that procedures $T1(L)$, $T2(\theta)$, and $T3(\varphi)$ defining the matrices of the previous subsection have been defined. The uses of these procedures in MAPLE are best illustrated in an example (see below).

5.4. **Application: Computing the Alpha Helix.** We will now illustrate the use of IMIMOL for finding an idealized *alpha helix*. First we build a model amino acid (see Figure 2) and save it in the file *aa.imi*. The labels can be seen in the Gaussian Z-matrix for this same molecule:

	$j=0$	$j=1$	L	$j=2$	θ	$j=3$	φ
$i=1$	$R1$						
$i=2$	N	$R1$	1.33				
$i=3$	CA	N	1.45	$R1$	120		
$i=4$	C	CA	1.52	N	109.47	$R1$	ϕ
$i=5$	$R2$	C	1.33	CA	120	N	ψ
$i=6$	$R3$	CA	1.5	N	109.47	C	-120
$i=7$	HA	CA	1.0	N	109.47	C	120
$i=8$	H	N	1.0	$R1$	120	CA	180
$i=9$	O	C	1.23	CA	120	$R2$	180

The bond lengths L have been taken from [2]. The bond angles θ and the wedge angles φ are either idealized or variable. The labels of the wedges $[C, CA, N, R1]$ and $[R2, C, CA, N]$ are the strings "phi" and "psi" respectively.

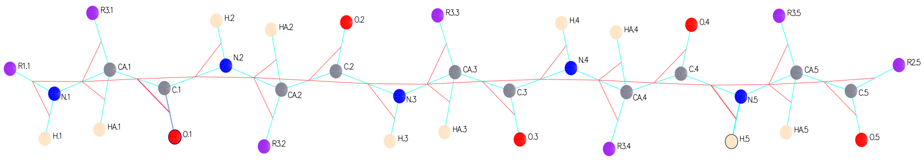

FIGURE 3. A Chain of Five Amino Acids. The sites $r = (H.5, N.5, C.4)$ and $r' = (O.1, C.1, CA.1)$ are highlighted. The wedges are not shown.

Next we glue two copies of this amino acid together, where the copy on the right is reflected about a horizontal line. The atom names of the left-hand (resp. right-hand) copy are edited to end in the suffix .1 (resp. .2). The dihedral wedge added during gluing $[CA.1, C.1, N.2, CA.2]$ is assigned the label 180 (to yield a *trans* peptide unit). This molecule is then saved in the file *2aa.imi*. Then we glue two copies of *2aa* together (no flipping this time) in the same manner with the appropriate suffixes on the atom names of the right-hand copy. This molecule is saved in the file *4aa.imi*. Then we glue *aa* to *4aa*, giving the atoms of the molecule *aa* the suffix .5. This five amino acid molecule is then saved in the file *5aa.imi*. The root site is then defined as $r = (H.5, N.5, C.4)$, and an alternate site is defined to be $r' = (O.1, C.1, CA.1)$. This is shown in Figure 3.

To determine an alpha helix geometrically we wish to find angles ϕ, ψ such that the atoms $N.5, H.5$, and $O.1$ are collinear, which is the preferred H-bonding geometry. For this purpose we export a Maple procedure file (converting to radians), called *5aa.mp*, and edit its first line to say: *helix:=proc(phi,psi)*. It is also a good idea to add comment lines explaining what the procedure does (see the MAPLE manual [7]). This procedure computes symbolically the 4×4 matrix $\mathcal{A}(\phi, \psi)$ such that $E_{r'}(\mathbf{R}) = E_r(\mathbf{R})\mathcal{A}(\phi, \psi)$ for every embedding $\mathbf{R}$ of the molecule *5aa* consistent with all the labels, including ϕ, ψ. $\mathcal{A}(\phi, \psi)$ is a product of 36 matrices. The atoms $N.5, H.5$, and $O.1$ will be collinear if $\mathbf{R}_{O.1}$ lies on the x-axis of $E_r(\mathbf{R})$. Let $\mathbf{c}' = (0, 0, 0)^T$ (resp. $\mathbf{c} = (x, y, z)^T$) be the coordinates of $\mathbf{R}_{O.1}$ with respect to $E_{r'}(\mathbf{R})$ (resp. $E_r(\mathbf{R})$), i.e.

$$\mathbf{R}_{O.1} = E_{r'}(\mathbf{R}) \begin{pmatrix} 1 \\ \mathbf{c}' \end{pmatrix} = E_r(\mathbf{R})\mathcal{A}(\phi, \psi) \begin{pmatrix} 1 \\ \mathbf{c}' \end{pmatrix} = E_r(\mathbf{R}) \begin{pmatrix} 1 \\ \mathbf{c} \end{pmatrix}.$$

Thus $\begin{pmatrix} 1 \\ \mathbf{c} \end{pmatrix} = \mathcal{A}(\phi, \psi) \begin{pmatrix} 1 \\ \mathbf{c}' \end{pmatrix}$ and hence $y = \mathcal{A}(\phi, \psi)_{3,1}$ and $z = \mathcal{A}(\phi, \psi)_{4,1}$. We want $(y, z) = (0, 0)$ so this give two equations: $\mathcal{A}(\phi, \psi)_{3,1} = 0, \mathcal{A}(\phi, \psi)_{4,1} = 0$, in the two unknowns ϕ, ψ. Both equations involve complicated polynomial equations in sines and cosines of ϕ, ψ, and it is hopeless to find solutions in closed form.

Start the MAPLE 8 program and type at the MAPLE prompt:

```
with(LinearAlgebra);
T1:=proc(L) Matrix(4,4,[[1,0,0,0],[L,-1,0,0],[0,0,1,0],[0,0,0,-1]]) end proc;
T2:=proc(theta) Matrix(4,4,[[1,0,0,0],[0,cos(theta),sin(theta),0],
[0,sin(theta),-cos(theta),0], [0,0,0,-1]]) end proc;
T3:=proc(phi) Matrix(4,4,[[1,0,0,0],[0,1,0,0],[0,0,cos(phi),-sin(phi)],
[0,0,sin(phi),cos(phi)]]) end proc;
read ”5aa.mp”;
fsolve({helix(phi,psi)[3,1]=0,helix(phi,psi)[4,1]=0}, {phi=-60,psi=-50});
```

The starting values of $(\phi, \psi) = (-60, -50)$ are based on published estimates of these parameters for alpha helices. MAPLE returns the solution $\phi = -56.50, \psi = -50.46$. At this solution the z-coordinate of $O.1$ is $z = -1.76$, which is pretty close to an optimal H-bonding distance of about 1.9. The point of all this is that by solving a simple geometric problem with a chain of idealized amino acids we obtain the essential geometry of the right-handed alpha helix, which in actuality is the result of a complicated balance of many chemical forces. Note that we also obtain another solution to our system of equations, namely $\phi = 56.50, \psi = 50.46$, which determines a left-handed alpha helix. There is also an approximate solution (i.e. where $y^2 + z^2$ is small but positive) at $\phi = 0, \psi = 180$. Neither of these other two solutions are observed in practice because they contain atoms colliding with each other.

Acknowledgments: Thanks to the Industrial Mathematics Institute (IMI) for summer support for the author as well as providing the programming services of Scott Johnson, who collaborated with the author on the design of IMIMOL and wrote the C code.

References

[1] C. Brandon, J. Tooze, *Introduction to Protein Structure*, Garland Publishing, Inc., New York, 1991.

[2] Creighton, T.E., *Proteins, Structures and Molecular Principles* W.H. Freeman and Co., New York, 1983.

[3] D. Dix, *Polyspherical Coordinates on Orbit Spaces with Applications to Biomolecular Shape*, preprint, 2004.

[4] D. Dix, *Building the DNA Double Helix, A Mathematical Approach*, in preparation, 2004?

[5] J.B. Foresman, A. Frisch, *Exploring Chemistry with Electronic Structure Methods*, second edition, Gaussian, Inc., Pittsburgh, 1996.

[6] H. Katayama, *Building Geometric Models of Biological Molecules*, Master's thesis in mathematics, University of South Carolina, 2003. c.f. http://www.math.sc.edu/~dix/katayama.

[7] B.W. Char, K.O. Geddes, G.H. Gonnet, B. Leong, M.B. Monagan, S.M. Watt, *Maple V Language Reference Manual*, Springer-Verlag, New York, 1991.

[8] See the website, http://www.rcsb.org/pdb

[9] See the website, http://www.kinemage.biochem.duke.edu/

[10] N. Trinajstić, *Chemical Graph Theory*, Vol. I and II, CRC Press, Boca Raton, 1983.

[11] See the website, http://www.webmo.net.

Department of Mathematics, University of South Carolina
E-mail address: dix@math.sc.edu