A Kinematic View of Loop Closure

EVANGELOS A. COUTSIAS,[1,*] CHAOK SEOK,[2,*] MATTHEW P. JACOBSON,[2] KEN A. DILL[2]
[1]*Department of Mathematics and Statistics, University of New Mexico,*
Albuquerque, New Mexico 87131
[2]*Department of Pharmaceutical Chemistry, University of California in San Francisco,*
San Francisco, California 94143-2240

Received 26 August 2003; Accepted 4 November 2003

Abstract: We consider the problem of loop closure, i.e., of finding the ensemble of possible backbone structures of a chain segment of a protein molecule that is geometrically consistent with preceding and following parts of the chain whose structures are given. We reduce this problem of determining the loop conformations of six torsions to finding the real roots of a 16th degree polynomial in one variable, based on the robotics literature on the kinematics of the equivalent rotator linkage in the most general case of oblique rotators. We provide a simple intuitive view and derivation of the polynomial for the case in which each of the three pair of torsional axes has a common point. Our method generalizes previous work on analytical loop closure in that the torsion angles need not be consecutive, and any rigid intervening segments are allowed between the free torsions. Our approach also allows for a small degree of flexibility in the bond angles and the peptide torsion angles; this substantially enlarges the space of solvable configurations as is demonstrated by an application of the method to the modeling of cyclic pentapeptides. We give further applications to two important problems. First, we show that this analytical loop closure algorithm can be efficiently combined with an existing loop-construction algorithm to sample loops longer than three residues. Second, we show that Monte Carlo minimization is made severalfold more efficient by employing the local moves generated by the loop closure algorithm, when applied to the global minimization of an eight-residue loop. Our loop closure algorithm is freely available at http://dillgroup.ucsf.edu/loop_closure/.

© 2004 Wiley Periodicals, Inc.   J Comput Chem 25: 510–528, 2004

Key words: loop closure; kinematic; loop modeling; Monte Carlo minimization

Introduction

We consider the problem of loop closure, i.e., finding structures of a segment in a chain molecule that are geometrically consistent with the rest of the chain structure. This problem has an important application in homology modeling,[1] when segments of insertions or deletions are to be modeled while the rest of the protein structure is relatively well known from structures of homologous proteins. Another useful application is in the area of Monte Carlo simulations, where alternative segment structures can be introduced as elementary localized moves.[2–11] These moves can lead to improved efficiency in conformational sampling. Unlike Cartesian moves, they avoid geometric distortions and the high energy penalty these entail. On the other hand, the deformation produced by these moves is limited to a segment, while uncoordinated torsion angle moves result in movement proportional to the distance of each atom from the perturbation axes, resulting in large uncontrolled moves. Other possible applications of the loop closure problem are discussed in ref. 12.

It is well known that the number of constraints is identical to the number of degrees of freedom (DOFs) in the case of loops with six free torsion angles, or three residue loops for proteins.[13] This means that, in general, such loops may be found as discrete solutions of the loop closure problem. This fact has been known for some time in the Kinematic theory of Mechanisms.[14] Kinematics is the branch of mechanics whose concern is the geometric analysis of motion, especially constrained displacements without regard to forces. The kinematic analysis of systems of rigid objects connected by flexible joints, such as multijointed robotic manipulators, exhibits many similarities with the geometric analysis of macromolecules, when the forces responsible for the motions are ignored and the main question of interest is the analysis of possible conformations consistent with the constraints associated with bond lengths and bond angles. In robotics, joints that allow one arm to

*These 2 authors contributed equally to this study.

Correspondence to: C. Seok; e-mail: chaok@maxwell.ucsf.edu

rotate about another at a fixed angle are called Rotator pairs or "**R**-pairs." The arm system analogous to a macromolecule with six rotatable bonds is a "6**R**" linkage. The kinematic analysis of these and other similar linkages leads to Fourier polynomials in the six rotation angles, τ_i, i.e., polynomials in the variables cos τ_i, sin τ_i. By introducing the half-angle transformation

$$u_i = \tan(\tau_i/2) \rightarrow \sin\tau_i = \frac{2u_i}{1+u_i^2}, \quad \cos\tau_i = \frac{1-u_i^2}{1+u_i^2}, \qquad i = 1, \ldots, 6,$$

a system results of polynomial equations in the u_i. A polynomial formulation offers several advantages, such as relative ease of solution, available theorems for the accurate enumeration of the number of solutions within a given region when there is only a discrete number, and in general, better understood numerical properties. For instance, the number of real roots of a univariate polynomial equation contained in an interval can be readily determined by Sturm's method.[15] No such method is available for more general, transcendental equations. Therefore, an advantage of our polynomial equation compared to the transcendental equation of Gō and Scheraga[13] is that the exact number of solutions can be found, which is important for satisfying microscopic reversibility in Monte Carlo simulations.[3] Methods from algebraic geometry[16,17] and homotopy theory[18] have been applied to such systems, and robust algorithms exist for the determination of their solutions, real or complex. A thorough discussion of robotic linkage systems can be found in the text by Duffy,[19] while informative expositions and reviews of the relevant literature can be found in the classic text by Hartenberg and Denavit,[14] and more recently, in the text by Hunt.[20] A relatively current survey is given in Manocha.[21]

The problem of closing 6**R** loops is central for the control of robotic manipulators, where in many common applications one end is fixed and the other (the "end effector") must be positioned at a specific location and with a given orientation. Adding a 7th rotator gives a system with one additional DOF, offering the possibility of continuous motion with two fixed ends. This problem, characterized as "The Mount Everest of robotic manipulators" by Freudenstein[22] was reduced to a single variable, 16th-degree polynomial equation by Lee and Liang.[23] In their solution, the 7th rotational DOF is used as a control parameter, and the real solutions obtained for the other angles once the 7th angle is fixed provide alternative closure configurations for the system. The method applies to systems with arbitrary axes of rotation, but the derivation is quite involved, and it is difficult to arrive at an intuitive understanding of its solutions and the implied chain displacements.

In this article we consider an important special case in which the 6**R** problem has an intuitively simple description: consider all the motions of a chain molecule that involve changes in only six backbone torsions. If these are arranged so that they form three coterminal pairs, then the segments between successive pairs will form effectively a coarser chain of three (closed case) or four (open case) rigid bodies, joined at the locations of the paired torsion axes. An illustration is given in Figure 1 for a tripeptide example, where the four rigid bodies are (N_1 $C_{\alpha1}$), ($C_{\alpha1}$ C_1 N_2 $C_{\alpha2}$), ($C_{\alpha2}$ C_2 N_3 $C_{\alpha3}$), and ($C_{\alpha3}$ C_3). If we now require the two end segments of the chain (N_1 $C_{\alpha1}$) and ($C_{\alpha3}$ C_3) to remain at a fixed position relative to each other, ($C_{\alpha3}$ C_3 N_1 $C_{\alpha1}$) forms a third segment. Now each of the three rigid units ($C_{\alpha1}$ C_1 N_2 $C_{\alpha2}$), ($C_{\alpha2}$ C_2 N_3 $C_{\alpha3}$), and ($C_{\alpha3}$ C_3 N_1 $C_{\alpha1}$) has two junctions on it, attaching to the other two units. Define the line connecting the two junctions on a unit as the virtual axis of the unit ($C_{\alpha1}$–$C_{\alpha2}$, $C_{\alpha2}$–$C_{\alpha3}$, and $C_{\alpha3}$–$C_{\alpha1}$). The motions of the middle two segments relative to the rest of the chain can only be composed of individual rotations of each about their respective virtual axes ($C_{\alpha1}$–$C_{\alpha2}$ and $C_{\alpha2}$–$C_{\alpha3}$) or joint rotations of the two as a unit about the third (fixed) axis ($C_{\alpha3}$–$C_{\alpha1}$). The three virtual axes form a triangle, with vertices at the three junctions ($C_{\alpha1}$, $C_{\alpha2}$, and $C_{\alpha3}$). If we rotate each of the units about its axis by some angle τ_i, $i = 1, 2, 3$, the rotatable bonds at either end of the unit maintain a fixed dihedral with the axis and each other (a dihedral formed by $C_{\alpha1}$–C_1, C_1–N_2, and N_2–$C_{\alpha2}$, for example). Any possible motion that a concerted change in the original six torsions is capable of can thus be described in terms of these three angles. If we now require that bond angles (θ_i) between the actual bonds at the junction of two segments remain at a given value, these motions become coupled. The feasible configurations where all constraints are satisfied form a discrete set, found as the solutions of a polynomial equation in the corresponding three variables u_i, $i = 1, 2, 3$. Having sets of rotation axes arranged in coterminal pairs is a natural property of polypeptide chain backbones where one encounters pairs of rotatable bonds at each C_α atom (with the exception of proline), and similar pairings are common in other molecules of interest, such as RNA where groupings of five pairwise coterminal rotatable bonds in the phosphate backbone are separated by relatively rigid sugar rings.

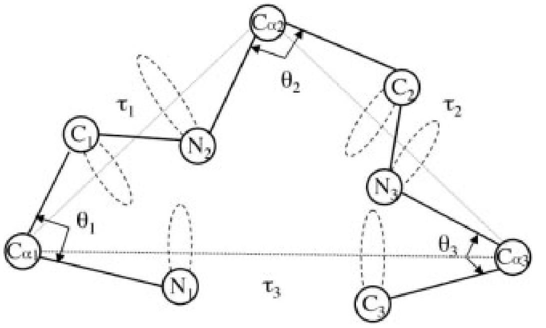

Figure 1. Definition of three variables τ_1, τ_2, and τ_3 and three constraints on θ_1, θ_2, and θ_3 in the canonical tripeptide loop closure problem.

In its simplest form our algorithm may utilize the torsion angles at three C_α atoms located consecutively along a peptide backbone. This is the "tripeptide loop closure" problem. The tripeptide loop closure problem was first considered by Gō and Scheraga,[13] who reduced the problem to solving a transcendental equation in a single variable in the case of planar peptide torsion angles. The method has found numerous applications and extensions. Bruccoleri and Karplus[24] allowed small variation in bond angles as a means of extending the method to cover normal variability of these parameters in proteins of known structures, and applied the method to loop modeling.[25] Dinner[7] produced a generalization to the nonplanar peptide case, still in terms of transcendental equations. More recently, Wedemeyer and Scheraga[12] derived a single-variable 16th-degree polynomial equation for the particular case of loop closure involving three consecutive residues with planar peptide torsions at canonical bond lengths and angles, i.e., when only three consecutive pairs of ϕ and ψ torsion angles are allowed to vary.

One of the generalizations possible with our algorithm is for the three pair of torsion angles with coterminal axes to be chosen along a molecular chain with arbitrary, fixed structure between successive pairs, including nonplanar peptide torsion angles. This generalization is useful for several reasons. Sampling with fixed bond angles and peptide torsion angles can significantly limit the coverage of conformational space,[24] and moreover, fluctuations of the order $\approx 10°$ for the bond angles and peptide torsion angles are not uncommon among proteins of known structure. Further, the method presented here allows for the torsion angles participating in the move to be chosen at arbitrary locations along the chain. This allows its application to diverse situations, such as to the modeling of longer loops and loops in polymers and nucleic acids. Although it is possible to derive a description in terms of a 16th-degree polynomial even if all the angles are chosen completely independently,[26] the choice of paired ϕ–ψ angles leads to a simple formulation in terms of three natural angle variables:

1. Choose three C_α carbons located successively (but not necessarily consecutively) along the chain, say $C_{\alpha i}$, $i = 1, 2, 3$.
2. Rotate the segment $C_{\alpha 1}, \ldots, C_{\alpha 2}$ by angle τ_1 about the axis $C_{\alpha 1}$–$C_{\alpha 2}$.
3. Rotate the segment $C_{\alpha 2}, \ldots, C_{\alpha 3}$ by angle τ_2 about the axis $C_{\alpha 2}$–$C_{\alpha 3}$.
4. Rotate the segment $C_{\alpha 1}, \ldots, C_{\alpha 2}, \ldots, C_{\alpha 3}$ by angle τ_3 about the axis $C_{\alpha 1}$–$C_{\alpha 3}$.
5. Choose the angles τ_i, $i = 1, 2, 3$ so that the bond angles N_i–$C_{\alpha i}$–C'_i assume (near) canonical values at each of the atoms $C_{\alpha i}$.

Satisfaction of the compatibility conditions in the last step is assured by the solution of the polynomial system mentioned above, and every real solution results in a distinct configuration. The analysis can easily be applied to chains of arbitrary structure (i.e., it is not limited to polypeptides), provided there exist pairs of coterminal rotatable bonds. In the robotics literature, **R**-joints with axes that have a common point are referred as "spherical pairs." We are thus studying the 6**R** system with three interconnected spherical pairs.[19] Problems of structure similar to the tripeptide loop closure problem are also common in another area of computational geometry: the motion planning for the assembly of four solid objects can be cast in identical mathematical form.[27]

Given that the general 6**R** problem can be described by a 16th-degree polynomial, it follows that there will always be an even number of real solutions, counting multiplicities, and, at most, 16 distinct real solutions are possible, leading in turn to at most 16 distinct loop configurations. Such an example has been found by Manseur and Doty[28] for a 6**R** robotic manipulator. Dodd et al.,[3] in their study of concerted rotations in polymer systems, report as many as 12 solutions in certain cases, but because they study the problem in its transcendental form they need to rely on expensive, exhaustive searches to arrive at a complete enumeration with confidence. For the canonical tripeptide loop closure, Wedemeyer and Scheraga[12] have found at most eight real solutions of the closure polynomial and, hence, at most eight distinct conformations. Our own studies with the more general peptide geometry have so far only discovered cases with at most 10 real solutions, and we believe that this might be a limitation due to the fact that the building blocks of the problem are of special form, perhaps not capable of covering the entire set of possible behaviors of the polynomial system unless a certain variability in the parameters is introduced. For example, the obtuseness of the bond angles at the C_α carbons should be contrasted with the fact that the angles between successive arms of the manipulator in ref. 28 are all $\pi/2$ except for one pair of parallel axes.

Even though it is clear that the analytical loop closure method, being exact, is much more efficient compared to numerical loop closure methods[10,29,30] for three residue loops, application of the analytical method to modeling longer loops has not yet been explored extensively. For loops of n torsion angles, $(n - 6)$ DOFs need to be sampled with some additional search method. Here we employ an existing loop construction method[31] to sample $(n - 6)$ torsion angles, and solve for the remaining six torsion angles using analytical loop closure. Other approaches such as a hierarchical method and a decimation method have been suggested by Wedemeyer and Scheraga[12] for sampling longer loops using an analytical loop closure method. We sample $(n - 6)$ torsion angles directly because it is possible to incorporate screenings for Ramachandran allowed regions and steric clashes. These screens enhance the efficiency of sampling because they can be applied at early stages, and nonpromising structures can be pruned out before the whole model loop is constructed.

We believe that an advantage of our work is the simplified, intuitive view of the tripeptide loop structure (or six free torsion angles in general), which enables us to develop insights for useful applications. General theory and methods are presented in the next section, and results of applications following that. Then we describe the simple view of the tripeptide loop closure, derive the loop closure equation, and present an efficient algorithm for solving the polynomial equation. Further generalizations to the case where the torsion angle pairs are chosen at arbitrary (noncontiguous) C_α atoms and to the case of an additional 7th dihedral are then discussed. A perturbation method for increasing the coverage of conformational space is also discussed. Applications to bond angle perturbations, longer loop modeling and Monte Carlo Minimization are presented, and then conclusions are given.

Theory and Methods

Loop Closure Formulation

We pose here the loop-closure problem in its simplest form as follows: given a molecular chain with inflexible bond lengths and bond angles, find all possible arrangements with the property that all bond vectors are fixed in space except for a contiguous set and such that the changes are made in at most six intervening dihedral angles. For convenience of presentation, we illustrate our deriva-

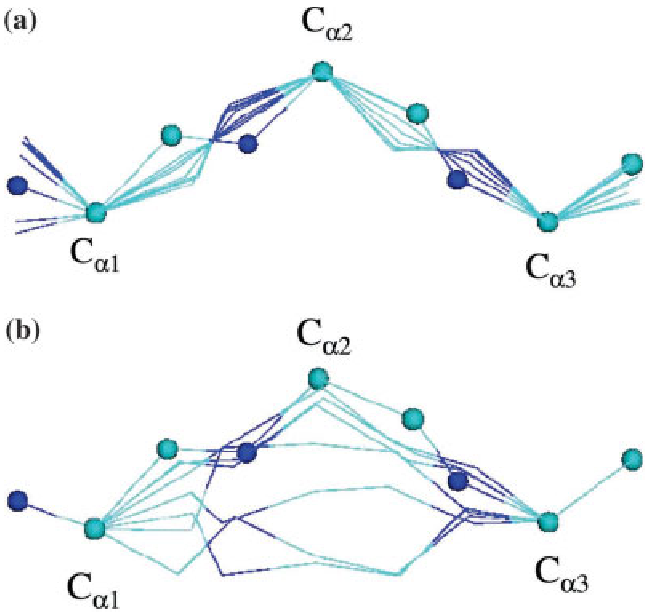

Figure 2. (a) Alternative configurations shown in the reference frame of the three fixed C_α atoms. (b) The same alternative loop closure configurations as in (a), but now in the original frame of the fixed atoms N_1, $C_{\alpha 1}$, $C_{\alpha 3}$, and C_3.

tion for the case of a tripeptide loop with occasional reference to more general cases.

Tripeptide Loop-Closure Equation

We view the six-torsion loop closure problem in a simplified representation as shown in Figure 1. A tripeptide loop example is shown in the figure, where four atoms N_1, $C_{\alpha 1}$, $C_{\alpha 3}$, and C_3 are fixed in space, and all other atom positions are to be determined. Atom types N, C_α, and C refer to nitrogen, alpha carbon, and carbonyl carbon, and the subscripts to the residue number (1, 2, or 3).

There are three variables and three constraints in this picture, which is equivalent to, but simpler than, the six-variable/six-constraint picture of Gō and Scheraga.[13] The three variables in the picture are the three rotation angles τ_i ($i = 1, 2, 3$) of the C_i and N_{i+1} atoms about the $C_{\alpha i}$–$C_{\alpha i+1}$ virtual bonds, where $i = 4$ is equivalent to $i = 1$. N_{i+1} is rotated with C_i because there is no free rotation involved between them. The τ_i rotations preserve all the bond lengths and angles except for the three bond angles θ_i ($:= \angle N_i C_{\alpha i} C_i$) shown in Figure 1. The condition that θ_i angles are equal to fixed values forms the three constraints in our problem. The τ_i angles are defined in the reference frame where all $C_{\alpha i}$ are fixed. $C_{\alpha 1}$ and $C_{\alpha 3}$ are fixed by definition, and so are the side lengths of the triangle formed by $C_{\alpha 1}$, $C_{\alpha 2}$, and $C_{\alpha 3}$. $C_{\alpha 2}$ therefore traces a circle about the $C_{\alpha 3}$–$C_{\alpha 1}$ axis. In the reference frame of Figures 1 and 2a, $C_{\alpha 2}$ is fixed and the rotation of $C_{\alpha 2}$ is replaced by an equivalent rotation of N_1 and C_3 about the same axis. Once the problem in this reference frame is solved, the N_1 and C_3 atoms (together with all other atoms in between) can be rotated back to the original frame by a reverse rotation about the same $C_{\alpha 3}$–$C_{\alpha 1}$ axis. This concept is illustrated with eight alternative loop conformations in the reference frame of the three C_α atoms in Figure 2a, and the corresponding picture in the original frame of fixed atoms N_1, $C_{\alpha 1}$, $C_{\alpha 3}$, and C_3 is shown in Figure 2b. Our formulation does not require planarity of the peptide torsion, and covers a more general case where arbitrary rigid structures intervene between the $C_{\alpha i}$–C_i and N_{i+1}–$C_{\alpha i+1}$ bonds.

In the derivation below, the bond vectors $\mathbf{C}_{\alpha i}\mathbf{C}_i$ and $\mathbf{C}_{\alpha i+1}\mathbf{N}_{i+1}$ (boldface symbol of a pair of atoms represents the bond vector of the pair) are first expressed in terms of τ_i angles and other fixed quantities, and then the θ_i angle constraints are written in terms of dot products of these vectors.

First consider the following unit vectors:

$$\hat{\mathbf{z}}_i = \mathbf{C}_{\alpha i}\mathbf{C}_{\alpha i+1}/|\mathbf{C}_{\alpha i}\mathbf{C}_{\alpha i+1}|, \quad \hat{\mathbf{r}}_i^\tau = \mathbf{C}_{\alpha i}\mathbf{C}_i/|\mathbf{C}_{\alpha i}\mathbf{C}_i|,$$
$$\hat{\mathbf{r}}_i^\sigma = \mathbf{C}_{\alpha i+1}\mathbf{N}_{i+1}/|\mathbf{C}_{\alpha i+1}\mathbf{N}_{i+1}|, \quad (1)$$

and define the following fixed angles in terms of these vectors:

$$\alpha_i = \cos^{-1}(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_{i-1}), \quad (2)$$

$$\eta_i = \cos^{-1}(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{r}}_i^\tau), \quad (3)$$

$$\xi_i = \cos^{-1}(-\hat{\mathbf{z}}_i \cdot \hat{\mathbf{r}}_i^\sigma), \quad (4)$$

where α_i, η_i, and ξ_i are all taken to be in the range $[0, \pi]$. These angles are shown in Figure 3 in the context of the C_α triangle.

We now define a right-handed local coordinate system by three unit vectors ($\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}_i$) for each τ_i rotation, where the reference axis $\hat{\mathbf{y}}$ is conveniently set to $\hat{\mathbf{y}} = (\hat{\mathbf{z}}_3 \times \hat{\mathbf{z}}_1)/|\hat{\mathbf{z}}_3 \times \hat{\mathbf{z}}_1|$ so that it is perpendicular to all $\hat{\mathbf{z}}_i$ defined in eq. (1), and to $\hat{\mathbf{x}}_i = \hat{\mathbf{y}} \times \hat{\mathbf{z}}_i$. As pictured in Figure 4a, the τ_i angle is now precisely defined to be the rotation angle of $\hat{\mathbf{r}}_i^\tau$ (or $\mathbf{C}_{\alpha i}\mathbf{C}_i$) about $\hat{\mathbf{z}}_i$ in this local coordinate system, and σ_i is defined similarly as the rotation angle of $\hat{\mathbf{r}}_i^\sigma$ (or $\mathbf{C}_{\alpha i+1}\mathbf{N}_{i+1}$) about $\hat{\mathbf{z}}_i$.

The angles τ_i and σ_i are related to each other because $\hat{\mathbf{r}}_i^\tau$ and $\hat{\mathbf{r}}_i^\sigma$ are rotated together as a rigid body. Figure 4a shows that τ_i and σ_i are related by the simple relation

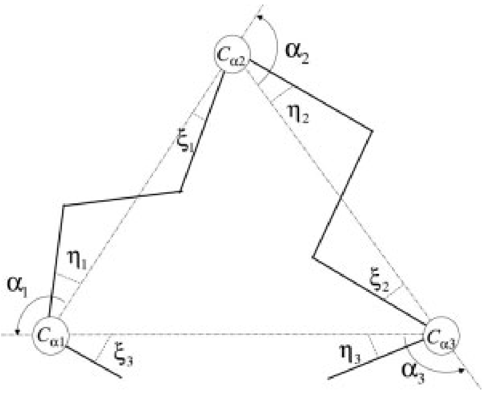

Figure 3. Definition of angle parameters α_i, η_i, and ξ_i.

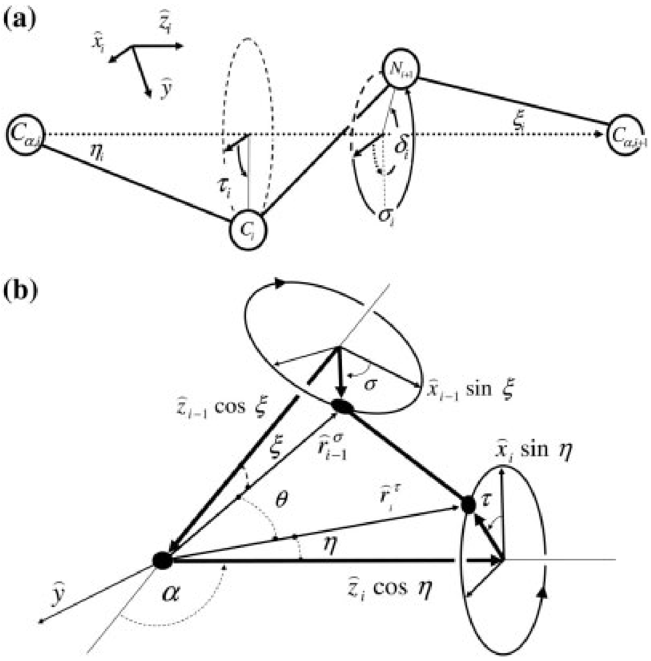

Figure 4. (a) A peptide unit along the $C_{\alpha i}$–$C_{\alpha i+1}$ virtual bond. In the local coordinate system, τ_i and σ_i are related by $\sigma_i = \tau_i + \delta_i$. (b) Geometric definitions at the $C_{\alpha i}$ junction. The black circle at the origin denotes the $C_{\alpha i}$ atom, while the vectors $\hat{\mathbf{r}}^{\sigma}_{i-1}$ and $\hat{\mathbf{r}}^{\tau}_i$ point to the N_i and C_i atoms, respectively.

$$\sigma_i = \tau_i + \delta_i, \tag{5}$$

where δ_i is the dihedral angle defined by the three vectors ($\mathbf{C}_i\mathbf{C}_{\alpha i}$, $\mathbf{C}_{\alpha i}\mathbf{C}_{\alpha i+1}$, $\mathbf{C}_{\alpha i+1}\mathbf{N}_{i+1}$), as illustrated in Figure 4a.

As can be seen in Figure 4b, the unit vectors $\hat{\mathbf{r}}^{\tau}_i$ and $\hat{\mathbf{r}}^{\sigma}_i$ are expressed in terms of the above defined unit vectors and angles as

$$\hat{\mathbf{r}}^{\tau}_i = \cos\eta_i\hat{\mathbf{z}}_i + \sin\eta_i(\cos\tau_i\hat{\mathbf{x}}_i + \sin\tau_i\hat{\mathbf{y}}),$$

$$\hat{\mathbf{r}}^{\sigma}_{i-1} = -\cos\xi_{i-1}\hat{\mathbf{z}}_{i-1} + \sin\xi_{i-1}(\cos\sigma_{i-1}\hat{\mathbf{x}}_{i-1} + \sin\sigma_{i-1}\hat{\mathbf{y}}). \tag{6}$$

The θ_i angle constraints can then be expressed in terms of $\hat{\mathbf{r}}^{\tau}_i$ and $\hat{\mathbf{r}}^{\sigma}_{i-1}$

$$\hat{\mathbf{r}}^{\tau}_i \cdot \hat{\mathbf{r}}^{\sigma}_{i-1} = \cos\theta_i. \tag{7}$$

Substitution of eq. (6) into eq. (7) gives the equations

$$\begin{aligned}\cos\theta_i + \cos\eta_i\cos\xi_{i-1}\cos\alpha_i = \sin\alpha_i(&\sin\xi_{i-1}\cos\eta_i\cos\sigma_{i-1}\\ &+ \cos\xi_{i-1}\sin\eta_i\cos\tau_i) + \sin\xi_{i-1}\sin\eta_i(\sin\tau_i\sin\sigma_{i-1}\\ &+ \cos\alpha_i\cos\tau_i\cos\sigma_{i-1}),\end{aligned} \tag{8}$$

with $i = 1, 2, 3$, where the dot products $(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_{i-1}) = \cos\alpha_i$, $(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{y}}) = 0$, $(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{x}}_{i-1}) = \sin\alpha_i$, $(\hat{\mathbf{x}}_i \cdot \hat{\mathbf{x}}_{i-1}) = \cos\alpha_i$, $(\hat{\mathbf{x}}_i \cdot \hat{\mathbf{y}}) = 0$, and $(\hat{\mathbf{z}}_{i-1} \cdot \hat{\mathbf{x}}_i) = -\sin\alpha_i$ have been used.

In a later section, σ_{i-1} is first eliminated from eq. (8) using eq. (5), and the three coupled equations for τ_i ($i = 1, 2, 3$) are reduced to a 16-degree polynomial for the single variable $u_3 = \tan(\tau_3/2)$. From the theory of polynomial systems[32] it follows that for every (real) solution there corresponds a unique (real) triplet (τ_1, τ_2, τ_3), so that, in general, there are at most 16 real solutions.

Equation (8), describing the rotation of the $C_{\alpha i}$–N_i and $C_{\alpha i}$–C_i bonds about the virtual bonds $C_{\alpha i-1}$–$C_{\alpha i}$ and $C_{\alpha i}$–$C_{\alpha i+1}$, respectively, is known in the theory of mechanisms as the equation for a *RR* joint with coterminal axes and with the two arms constrained to be at a fixed distance. It was derived in 1897 by Bricard in his study of flexible octahedra,[33] and considerable literature about it exists.[20,34] A geometrical analysis of the individual (uncoupled) θ_i constraint eqs. (7) and (8) is provided in Appendix A.

The Algorithm

Once the polynomial equation is obtained, all atomic coordinates in the loop can be determined. Before presenting a detailed derivation and solution of the polynomial equation, we give here a simple outline of the loop closure algorithm that finds the atom positions in the loop.

1. The polynomial coefficients are determined from the angles α_i, η_i, ξ_i, and δ_i: first, the angles η_3, ξ_1, δ_3 are determined from the coordinates of N_1, $C_{\alpha 1}$, $C_{\alpha 3}$, and C_3, and all other angles are computed from the given bond lengths and bond angles (canonical, or, more generally, for arbitrary, specified values of these). The coefficients of the 16th degree polynomial are then determined algorithmically following the steps described in the next section and Appendix B.
2. $u_3 = \tan(\tau_3/2)$ is obtained by solving the 16th-degree polynomial, as described later. $u_2 = \tan(\tau_2/2)$ and $u_1 = \tan(\tau_1/2)$ are determined from u_3 as described in Appendix C, and $\tau_i = 2\tan^{-1}u_i$ and $\sigma_i = \tau_i + \delta_i$ follow.
3. Positions of all the atoms are obtained from τ_i and σ_i: first, the reference frame is defined. The unit vector $\hat{\mathbf{z}}_3$ is determined from the coordinates of $C_{\alpha 1}$ and $C_{\alpha 3}$. $\hat{\mathbf{z}}_1$ is set arbitrarily except that the angle between $\hat{\mathbf{z}}_1$ and $\hat{\mathbf{z}}_3$ is α_1. $\hat{\mathbf{z}}_2 = -\hat{\mathbf{z}}_1 - \hat{\mathbf{z}}_3$ follows. $\hat{\mathbf{y}}$ and $\hat{\mathbf{x}}_i$ are computed from $\hat{\mathbf{z}}_i$. Next, $\hat{\mathbf{r}}^{\tau}_i$ and $\hat{\mathbf{r}}^{\sigma}_i$ ($i = 1, 2$) in the reference frame are obtained from τ_i and σ_i using eq. (6). All atom positions are then computed from these vectors in the reference frame. The unit vectors define $\tau^{(0)}_3$ that are determined from the fixed coordinates of N_1, $C_{\alpha 1}$, $C_{\alpha 3}$, and C_3. All atoms are then rotated about $\hat{\mathbf{z}}_3$ by $(\tau^{(0)}_3 - \tau_3)$ to bring them to the original frame.

Derivation of the Polynomial Equation

Equation (8) is converted to polynomial form in the variables w_i, u_i where

$$w_i := \tan(\sigma_i/2), u_i := \tan(\tau_i/2). \tag{9}$$

Using the half-angle formulas

$$\cos\tau = \frac{1-u^2}{1+u^2}, \quad \sin\tau = \frac{2u}{1+u^2}, \quad u = \tan\frac{\tau}{2}, \tag{10}$$

Equation (8) becomes

$$A_i w_{i-1}^2 u_i^2 + B_i w_{i-1}^2 + C_i w_{i-1} u_i + D_i u_i^2 + E_i = 0 \qquad (11)$$

where

$$\begin{aligned} A_i &= -\cos\theta_i - \cos(\alpha_i - \xi_{i-1} - \eta_i) \\ B_i &= -\cos\theta_i - \cos(\alpha_i + \xi_{i-1} - \eta_i) \\ C_i &= 4\sin\xi_{i-1}\sin\eta_i \\ D_i &= -\cos\theta_i - \cos(\alpha_i - \xi_{i-1} + \eta_i) \\ E_i &= -\cos\theta_i - \cos(\alpha_i + \xi_{i-1} + \eta_i). \end{aligned}$$

Equation (11) is called the tetrahedral equation,[33] because it describes the alternative shapes of the tetrahedral formed by the four fixed angles α, ξ, θ, and η. This equation is quadratic both in w and u, denoting that, in general, to each value of one of the dihedrals τ and σ there correspond two values of the other. After eliminating w_{i-1} from eq. (11) using eq. (5), because

$$w_i = \tan(\tau_i/2 + \delta_i/2) = \frac{\tan(\tau_i/2) + \tan(\delta_i/2)}{1 - \tan(\delta_i/2)\tan(\tau_i/2)} = \frac{u_i + \Delta_i}{1 - \Delta_i u_i} \qquad (12)$$

where we introduced $\Delta = \tan(\delta/2)$, we arrive at a system of three biquadratic (quadratic in two variables) equations in $u_i = \tan(\tau_i/2)$:

$$P_1(u_3, u_1) := \sum_{j,k=0}^{2} p_{jk}^{(1)} u_3^j u_1^k = 0, \qquad (13)$$

$$P_2(u_1, u_2) := \sum_{j,k=0}^{2} p_{jk}^{(2)} u_1^j u_2^k = 0, \qquad (14)$$

$$P_3(u_2, u_3) := \sum_{j,k=0}^{2} p_{jk}^{(3)} u_2^j u_3^k = 0, \qquad (15)$$

where the coefficients $p_{jk}^{(1)}$, $p_{jk}^{(2)}$, and $p_{jk}^{(3)}$ are defined in terms of the fixed angles α_i, θ_i, η_i, ξ_{i-1}, and δ_{i-1}. These coefficients and all other coefficients that follow below are derived in Appendix B.

Before proceeding with solving this system, we address the expected number of solutions. Although the classical Bezout theorem bounds the number of zeros of a system of polynomial equations by the product of their degrees (here $4^3 = 64$), a sharper result, referred as the "Bernshtein–Kusnirenko–Khovanskii (**BKK**) Theorem"[35] is known, which takes advantage of the fact that the above polynomials are not the most general 4th-degree polynomials in variables u_i, $i = 1, 2, 3$ (e.g., terms like u_i^4, $u_i^3 u^j$ or $u_i^2 u_j u_k$ are not present) and gives for the above system the upper bound as 16. Although we will not present the easy proof here, we must mention that this theorem is sharp, meaning that the number 16 is realizable for some sets of values of the coefficients. In the following discussion we carry out the elimination of variables in two steps in a similar manner as in ref. 12, taking advantage of the fact that each polynomial is bivariate so that variables can be eliminated one at a time. The final univariate polynomial is of order 16. Given the previous discussion, no redundancy can be present in this polynomial in general and all 16 solutions have potential physical significance. We have found at most 10 real solutions when we introduce variances in the peptide torsion and bond angles, in contrast to previous works[12,13] in which at most eight solutions were found in the rigid planar tripeptide case. However, given the rarity of such cases (3 in 1 million for the database we explored[36]) robustness issues may play a role. This does not mean that exceptional cases may still not be found where the number of distinct real solutions is 16. We are currently investigating this question.

The method of resultants (see Appendix C) is used to reduce the above equations to an equation for a single variable. In short, the variable u_1 is first eliminated from eqs. (13) and (15) to give

$$R_8(u_2, u_3) = \sum_{j,k=0}^{4} q_{jk} u_2^j u_3^k = 0, \qquad (16)$$

and u_2 is eliminated from eqs. (14) and (16) to give

$$R_{16}(u_3) = \sum_{j=0}^{16} r_{jk} u_3^j = 0. \qquad (17)$$

More specifically, eq. (16) is obtained by rewriting eqs. (13) and (15) as

$$P_1(u_3, u_1) = \sum_{k=0}^{2} L_k(u_3) u_1^k$$

and

$$P_2(u_1, u_2) = \sum_{j=0}^{2} M_j(u_2) u_1^j,$$

where $L_k := L_k(u_3)$ and $M_j := M_j(u_2)$ are themselves quadratics in u_3 and u_2, respectively (see Appendix B).

The resultant of the two biquadratics P_1 and P_2, which eliminates u_1, is given by the determinant

$$R_8(u_2, u_3) = \begin{vmatrix} L_2 & L_1 & L_0 & 0 \\ 0 & L_2 & L_1 & L_0 \\ M_2 & M_1 & M_0 & 0 \\ 0 & M_2 & M_1 & M_0 \end{vmatrix} = \sum_{j,k=0}^{4} q_{jk} u_2^j u_3^k = 0. \qquad (18)$$

We now write $R_8(u_3, u_3)$ as a quartic in u_3 introducing the functions $Q_j(u_3)$, quartics in u_3:

$$R_8(u_2, u_3) = \sum_{j=0}^{4} Q_j(u_3) u_2^j,$$

and eq. (15) as

$$P_3(u_2, u_3) = \sum_{j=0}^{2} N_j(u_3)u_2^j,$$

where the N_j are quadratics in u_3. The final resultant, which eliminates u_3 to give a degree 16 polynomial in u_1 is given by

$$R_{16}(u_3) = \begin{vmatrix} N_2 & N_1 & N_0 & 0 & 0 & 0 \\ 0 & N_2 & N_1 & N_0 & 0 & 0 \\ 0 & 0 & N_2 & N_1 & N_0 & 0 \\ 0 & 0 & 0 & N_2 & N_1 & N_0 \\ Q_4 & Q_3 & Q_2 & Q_1 & Q_0 & 0 \\ 0 & Q_4 & Q_3 & Q_2 & Q_1 & Q_0 \end{vmatrix} = \sum_{j=0}^{16} r_{jk}u_3^j = 0. \qquad (19)$$

One key advantage of the reduction to polynomial form carried out in the previous subsections is the availability of reliable software for the determination of polynomial zeros. The solution can be carried out by either directly solving the polynomial equation, or by reduction to the solution of a generalized eigenproblem.[21] For completeness we give a brief description of both schemes below. In our studies, the direct solution has proved to be more efficient.

Direct Solution and Sturm Chains

We use the polynomial solution package available from ACM.[37] This package uses Sturm's method[15] to determine the number of real zeros within an arbitrary interval. The intervals are bisected and refined until all the solutions are put in separate, tight intervals. The solutions are then refined using a secant method.

Generalized Eigenproblem Formulation

The above polynomial equation can be formulated as a generalized eigenproblem. Following Manocha,[21] we write $R_{16}(u_1)$ as a determinant of a matrix polynomial with matrix coefficients S_k:

$$det\left(\sum_{k=0}^{4} S_k u_1^k\right) = 0, \qquad (20)$$

which is equivalent to

$$det(\mathcal{B}u_1 - \mathcal{A}) = 0 \qquad (21)$$

with

$$\mathcal{B} := \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & S_4 \end{pmatrix}, \quad \mathcal{A} := \begin{pmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ -S_0 & -S_1 & -S_2 & -S_3 \end{pmatrix}, \qquad (22)$$

where all blocks are of size 6 × 6. The resulting generalized eigenproblem, $u_1\mathcal{B}\mathcal{X} = \mathcal{A}\mathcal{X}$, can be solved with the lapack routine **dggev.f,** for example. It is also possible to take advantage of the sparsity of the matrices $\mathcal{A}$, $\mathcal{B}$, if desired.

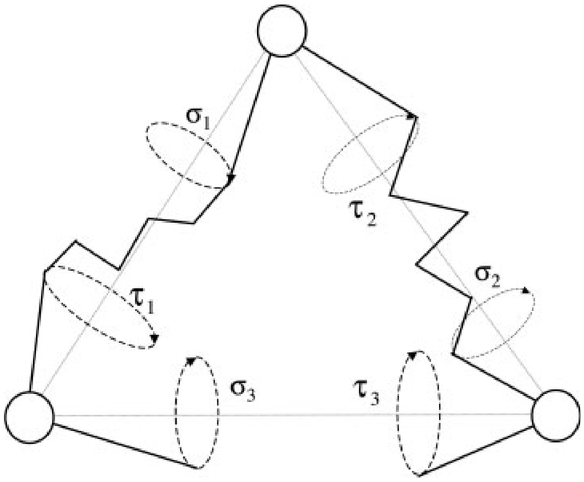

Figure 5. General chain loop closure.

Generalizations of the Method

Noncontiguous C_α Atoms

As is clear in Figure 1, the loop closure process involves three rotations about the axes $C_{\alpha i}$–$C_{\alpha i+1}$ (i = 1, 2, 3) and three constraints relating these rotations that ensure that the bond angles between the two rotatable bonds N_i–$C_{\alpha i}$ and $C_{\alpha i}$–C_i at the $C_{\alpha i}$ are set. The chain of atoms intervening between the $C_{\alpha i}$ is rotated rigidly. The problem is completely characterized by giving the angles α_i between the virtual bonds (which, together with one of the edges, say $C_{\alpha i}$–$C_{\alpha i+2}$, completely characterize the triangle $C_{\alpha i}$, $C_{\alpha i+1}$, $C_{\alpha i+2}$), the angles ξ_i, η_i formed by the rotatable bonds at each $C_{\alpha i}$ and the edges of that triangle, as well as the dihedrals δ_i. Nowhere in this construction is any assumption made about the intervening structure, nor are any such assumptions implicit in the derivation of the loop closure equations. Therefore, the algorithm can be applied without modification to moves involving arbitrary triads of C_α atoms (Fig. 5), i.e., the angle parameters α_i, ξ_i, η_i, and δ_i that completely determine the problem are defined in the same way as in Figures 3 and 4a from the three atoms at each vertex of the C_α triangle. This is a new feature relative to other algorithms. Of course, more general moves become possible now, where some of the intervening dihedrals are also changed, modifying the parameters of the basic triangle. To illustrate this additional flexibility we consider in the next subsection the simplest such move, namely the change of one additional dihedral. This introduces a continuous DOF to the problem, and it forms the basis of a Monte Carlo move.

Additional Dihedral Angle

We now consider a method of finding alternative local structures when an arbitrary dihedral angle is changed. Six angles need to be adjusted to compensate the change such that the chain structure is

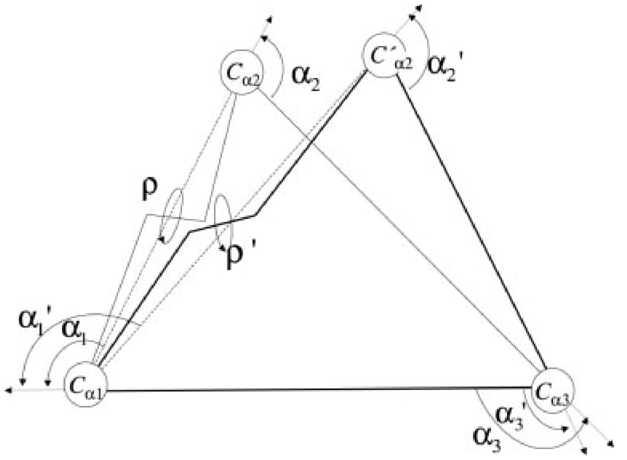

Figure 6. Deformation of the C_α triangle due to a dihedral (ρ) perturbation. Changes in α_i angles due to the triangle deformation are shown.

unchanged beyond the local region. Concerted angle perturbations of this kind can be used as elementary moves in Monte Carlo simulations to increase the sampling efficiency. A simple way is to adjust six consecutive angles adjacent to the driver angle.[7] The terminal atom position is changed (either $C_{\alpha 1}$ or $C_{\alpha 3}$ in Fig. 1) in this case, and the six angle loop closure problem can then be solved with the changed C$_\alpha$ triangle geometry. Here, we describe a more general and flexible method of compensating the angle change, in which six dihedrals to be adjusted are allowed to be separated in pairs arbitrarily in sequence, and the driver angle can be placed anywhere in between the adjusting dihedral pairs.

Figure 6 shows a case in which the driver angle ρ is placed on the left-hand side of the C$_\alpha$ triangle, as an example. As before, we consider three τ_i rotations separately, and then apply the θ_i constraints. This is possible because the net effect of the driver angle rotation in our simplified picture is to change some of the parameters for the C$_\alpha$ triangle geometry that are independent of τ_i rotations. The geometric parameters for the base of the triangle, $\overline{C_{\alpha 3}C_{\alpha 1}}$, η_3, ξ_1, are invariant because they are fixed by constraints, and those for the right side of the triangle, $\overline{C_{\alpha 2}C_{\alpha 3}}$, η_2, ξ_3, are also invariant because rotation due to the angles on the left side does not change the relative orientation of the atoms on the right. Those for the left side, $\overline{C_{\alpha 1}C_{\beta 2}}$, η_1, ξ_2, change because the driver angle rotates N_2 and $C_{\alpha 2}$, but not $C_{\alpha 1}$ and C_1. Due to the change in $\overline{C_{\alpha 1}C_{\alpha 2}}$, $\hat{\mathbf{z}}_i$ (i = 1, 2) and α_i (i = 1, 2, 3) change. Equation (8) can be then derived with the changed parameters.

These flexible concerted local moves are expected to improve efficiency of conformational search. A Monte Carlo with minimization method has been employed together with the concerted moves described above, and severalfold of improvement in efficiency has been observed compared to existing search methods (see later).

Bond Angle Perturbations

So far we have fixed bond lengths, bond angles, and peptide torsion angles at their canonical values in the loop closure algorithm, although there is no limitation on what specific values have to be used. However, real proteins exhibit a range of values depending on their chemical environment. When the rigid loop closure method is used to sample structures for real proteins, some structures cannot be sampled if the flexibility in bonds and angles is ignored. This fact was first noticed by Bruccoleri and Karplus (BK).[24] To test how much the rigid sampling can cover the real protein structure space, three-residue structures were deleted artificially from the Top500 database of high resolution, nonredundant protein structures,[36] and our exact loop closure algorithm was used to fill the gaps. About 27.5% of the gaps could not be filled with the rigid sampling. (The bond lengths and angles used are $\overline{NC_\alpha}$ = 1.45 Å, $\overline{C_\alpha C}$ = 1.52 Å, $\overline{CN}$ = 1.33 Å, $\angle NC_\alpha C$ = 111.6°, $\angle C_\alpha CN$ = 117.5°, and $\angle CNC_\alpha$ = 120.0°.) BK developed a search method to find minimal bond angle variations to close a given loop. Our method is used to vary peptide torsion angles as well as bond angles, because we now have a more general formula. We also present a much simpler, efficient method of perturbing bond angles, where no extensive search is involved.

We first present a method in which the minimum of the polynomial is moved by angle changes so that the minimum at least touches the axis to give roots, in a similar spirit to BK. The angles are perturbed by the minimum amount (so as to minimize the energy penalty). We then show a faster method that makes use of the knowledge of the direction of angle change that maximizes the probability of having loop-closure solutions. This method only determines the sign of angle change, but not the minimum magnitude.

Perturbation by Angle Search

The minimum of the polynomial and the derivatives of the minimum with respect to perturbed angles are computed, and a steepest descent search is performed to bring down the polynomial minimum to equal to or less than zero. A more sophisticated LBFGS minimization algorithm was tried, but the efficiency was similar. The steepest descent iteration is terminated when loop closure solutions are found, preset maximum angle perturbations are reached, or maximum number of iterations (set to 200) is reached.

The polynomial minimum is obtained by finding roots of $R'_{16}(u) = 0$ and comparing the polynomial values at the roots. The derivatives of the minimum with respect to perturbed angles are computed by a finite difference method. The minimum at the perturbed angles are computed by a Newton–Raphson minimization starting from the current minimum. Several parameters are introduced to accelerate the steepest descent iteration. The step size of the steepest descent minimization is increased or decreased (by a factor of f_i or f_d) depending on whether the previous iteration decreased or increased the minimum. The initial step size is chosen so that the largest component of angle change is equal to d_i. At each iteration, the largest angle component change is restricted to be at most d_m. $f_i = 9, f_d = 0.1, d_i = 0.1$, and $d_m = 0.0001$ were chosen by trial and error to maximize the number of loop closure solutions for the cyclopentapeptide example below. It is also found that angle perturbations prior to steepest descent help in finding more loop-closure solutions. For example, when the C_α triangle cannot be formed or is formed marginally because the base length $C_{\alpha 1}$–$C_{\alpha 3}$ is too short or too long to be reached with the canonical bond angles, the angles are perturbed by the maximum amount to allow for longer or shorter base lengths. In addition, when it is found that there exists no solution for the two-cone system for any vertex (see next section and Appendix A), angles are adjusted to maximize the overlap of the two cones as in the next subsection.

Simple Perturbation Method

We now present a simple bond angle perturbation method that does not require searching the bond angle space, thus solving the loop-closure problem only once. This can be accomplished by examining the components of the simple picture in Figure 1. Figure 9a in Appendix A shows σ_{i-1} rotation of $\hat{\mathbf{r}}^{\sigma}_{i-1}$ and τ_i rotation of $\hat{\mathbf{r}}^{\tau}_{i}$. Each vector traces a cone, so we call it a two-cone system. The two vectors have to satisfy the bond angle constraint eq. (7), and this limits the accessible ranges of σ_{i-1} and τ_i values. These ranges depend on the local geometry determined by ξ_{i-1}, η_i, α_i, and θ_i. Each vertex of the triangle in Figure 1 has a two-cone system, so there are three two cones in all. The loop closure solutions are determined by the intersection of the allowed σ_{i-1}/τ_i regions in the three two-cone systems. By construction, θ_i does not change the triangle geometry or any other parameters, but varies the accessible ranges of σ_{i-1}/τ_i. It is possible to determine whether to increase or decrease θ_i to maximize the accessible ranges at each two-cone, which in turn maximizes the overlaps of two-cone systems, and the possibility of closing the loop. This is done by classifying the two-cone types depending on how the extrema of σ_{i-1}/τ_i are arranged, and determining the effect of θ_i change on the extrema. The details are provided in Appendix A.

Results and Discussion

In this section we present an application of the angle perturbation method, then give two applications of the analytical loop closure, to longer loop modeling and Monte Carlo Minimization.

Test of the Angle Perturbation Methods

We apply the perturbation methods presented above to the three-residue gaps artificially deleted from the Top500 structures.[36] When fixed canonical angle parameters are assumed and the loop-closure algorithm is applied to fill the three-residue gaps, 22,981 (27.5%) out of total 83,327 gaps do not have loop-closure solutions. (Those loops including proline have been excluded in this test.) The number of missed gaps decreases dramatically with the simple perturbation method from earlier: 1249 (1.5%) and 469 (0.56%) with the maximum angle variation of 5 and 10 degrees, respectively. Note that only 3 $NC_{\alpha}C$ angles have been varied here. The full angle perturbation from above misses 209 (0.25%) and 23 (0.028%) for the maximum allowed perturbation 5 and 10 degrees, respectively, when nine angles are varied (three $NC_{\alpha}C$, two $C_{\alpha}CN$, two CNC_{α}, and two peptide torsion angles). In summary, the simple perturbation method is successful in covering most of the conformational space realized in the database, and the full angle perturbation can push the limit to almost perfection. Computation time increases only a few percent even with the full search method for this test because most of the loops have solutions, and only a few iterations are needed for angle search. Computation time increases more with perturbations when there are more cases in which loop-closure solutions are not found such as when applied to loop modeling or for exhaustive sampling as in the cyclopentapeptide example below.

Next, we consider the cyclopentapeptide Gly-Gly-Gly-Pro-Pro example for which Gō and Scheraga[38] and Bruccoleri and Karplus[24] sampled the conformational space. The two Pro ϕ angles are frozen, the two Pro ψ angles are varied with a grid of 5 deg, and the remaining six Gly ϕ/ψ torsions are solved for. We use the same bond lengths and angle parameters as BK, and 346 loops are closed when no perturbation is used. BK closed 1507 and 1565 loops with their fast and slow bond angle search method, respectively, varying nine bond angles with maximum variation of 5 degrees. Our full perturbation method closes 1517 loops when the same 9 angles (3 $NC_{\alpha}C$, three $C_{\alpha}CN$, three CNC_{α}, including two bond angles outside the C_{α} triangle) are varied with the same 5-degree maximum variation. Including perturbations in the additional three peptide torsional angles closes 1594 loops. Our full search method is thus comparable to that of BK in the number of closed loops, and our ability to add the peptide torsion DOFs increases the coverage of the conformational space slightly. The simple three-angle perturbation closes 819 closed loops, which is twice as many as those obtained without perturbation (346), but only half of those obtained with full perturbation. The total computation time increases steeply with increasing perturbation level: 0.26, 0.56, 17.6, and 24.0 s for no perturbation, simple 3-angle perturbation, 9-angle, and 12-angle perturbations, respectively, when scaled to an AMD 1800+ MP processor.

Application to Loop Modeling

Analytical loop closure finds a discrete set of loop conformations for a three-residue loop, but a longer loop has a continuous set of possible closed loop conformations. Sampling a longer loop therefore requires a strategy to sample the extra DOFs to be coupled with the analytical loop closure. The extra DOFs could be sampled either randomly or in an informed way. When the sampling is random, unfeasible conformations due to unfavorable ϕ/ψ angles or steric clashes would be screened out in later stages of loop prediction algorithms, but it would be more efficient if such structures are excluded during the loop sampling stage. We employ an existing loop construction algorithm,[31] which performs this by sampling in the allowed regions of the ϕ/ψ map in a discrete manner and screening possible side-chain clashes using a rotamer library. This algorithm (as implemented in the program PLOP[31]) is used to build the N-terminal and the C-terminal branches except for the three residue gap in the middle of a loop, and the analytical loop-closure algorithm is used to close the branches.

The performance of our coupled algorithm is compared to the recent work of Canutescu and Dunbrack called CCD (cyclic coordinate descent).[39] The CCD algorithm is a numerical loop-closure algorithm that is similar in spirit to the "random tweak" method,[29] solving first-order equations iteratively, but it is more robust and efficient. We take the same test set as in Table 2 of ref. 39, which consist of 10 loops each with lengths of 4, 8, and 12 residues (total of 30 loops) chosen from a set of nonredundant X-ray crystallographic structures. The comparison is summarized

Table 1. Minimum RMSD (in Å) of the Candidate Loops with Our Algorithm (CSJD) and the CCD Algorithm.

Four-residue loops			Eight-residue loops			12-residue loops		
Loop	CSJD	CCD	Loop	CSJD	CCD	Loop	CSJD	CCD
1dvjA_20	0.38 (4548)	0.61	1cruA_85	0.99 (2516)	1.75	1cruA_358	2.00 (4148)	2.54
1dysA_47	0.37 (2234)	0.68	1ctqA_144	0.96 (1754)	1.34	1ctqA_26	1.86 (3968)	2.49
1eguA_404	0.37 (170)	0.68	1d8wA_334	0.37 (1686)	1.51	1d4oA_88	1.60 (1802)	2.33
1ej0A_74	0.21 (1564)	0.34	1ds1A_20	1.30 (3506)	1.58	1d8wA_46	2.94 (3906)	4.83
1i0hA_123	0.26 (342)	0.62	1gk8A_122	1.29 (2362)	1.68	1ds1A_282	3.10 (1162)	3.04
1id0A_405	0.72 (528)	0.67	1i0hA_145	0.36 (1452)	1.35	1dysA_291	3.04 (2306)	2.48
1qnrA_195	0.39 (1064)	0.49	1ixh_106	2.36 (4448)	1.61	1eguA_508	2.82 (2106)	2.14
1qopA_44	0.61 (4284)	0.63	1lam_420	0.83 (2200)	1.60	1f74A_11	1.53 (3048)	2.72
1tca_95	0.28 (418)	0.39	1qopB_14	0.69 (3384)	1.85	1qlwA_31	2.32 (4780)	3.38
1thfD_121	0.36 (2958)	0.50	3chbD_51	0.96 (1838)	1.66	1qopA_178	2.18 (2014)	4.57
Average	0.40 (1181)	0.56	Average	1.01 (2525)	1.59	Average	2.34 (2924)	3.05

CCD results are taken from Table 2 of ref. 39; 5000 trials were performed per test loop with the CCD, so the minimum is among about 5000 closed loops. With CSJD, the number of candidate loops is taken to be always less than 5000 for each test loop, and shown in the parentheses.

in Table 1. The average of the best backbone RMSD obtained from CCD is 0.56, 1.59, and 3.05 Å for 4, 8, and 12 residue loops, respectively, with average computing time per closed loop of 31, 37, and 23 ms on an AMD 1800+ MP processor. Our coupled algorithm gives better average minimum RMSDs of 0.40, 1.01, and 2.34 Å, in almost two orders of magnitude less computing time (0.56, 0.68, and 0.72 ms per loop when scaled to the same processor). In addition, the minimum RMSD for individual test loops is better for 25 out of the 30 cases in Table 1. The conditions under which we perform the comparison actually disfavor our algorithm because we generate fewer loops. With CCD, the loops are obtained from 5000 trials (thus about 5000 loop candidates, given that the algorithm can close the loops 99.8% of the time). However, with our algorithm, the exact number of loop candidates is not the control parameter of the algorithm but rather the sampling resolution in the ϕ/ψ map. As the sampling resolution is increased, the number of loop conformations increases. For this comparison, we generate less than 5000 loop candidates for each test case, sometimes far less, which disfavors us in the comparison. The number of loop candidates for each test loop is also shown in Table 1.

The coupled algorithm is also compared with the algorithm as presented in ref. 31, which does not use the analytical loop closure and continues the discrete ϕ/ψ sampling instead to close the loop, which we call "numerical" closure here. When the same resolutions, thus the same sets of conformations for the residues outside of the three-residue closure segment, are used, the average best RMSD obtained is 0.29, 1.66, and 3.25 Å with computation time per loop of 0.73, 1.60, and 106 ms. Except for the short four-residue loops, which are easy both for the numerical and analytical closure due to the small number of DOFs, the analytical closure gives better RMSD in orders of magnitude shorter time per loop conformation, especially for the longest 12 residue loops. This is expected because the analytical closure can close branches more efficiently for the given sampling resolution for the branches. The number of conformations generated by the analytical closure method (which in essence has infinite sampling resolution) is much higher than the number generated by the algorithm without analytical closure. The average number of closed loops with the numerical closure is 459, 236, and 42 for the 4, 8, and 12 residue loops, respectively, compared to 1181, 2525, and 2924 with the analytical closure at the fixed branch sampling resolutions. In the tests performed here, in which the maximum number of loop candidates is held fixed at 5000, this is actually disadvantageous for the analytical closure, because more loops are generated using coarser sampling for the nonclosure residues. When a maximum of 5000 loops are generated, the numerical closure thus gives better RMSD (0.27, 1.04, and 1.89 Å) although with longer computing time per loop (8.5, 6.1, and 23 ms). This implies that the optimal number of closed loops to be sampled is different for the analytical and numerical closure, and more loop candidates must be sampled with the analytical closure. Clustering algorithms can be used to remove the redundancies in the candidates before more expensive refinement and rescoring steps.[31]

Finally, adding the bond angle perturbations is found not to affect the best RMSD compared to no perturbation, although more closed loops are found. Producing more high-quality loops by a biased perturbation that samples desired regions of ϕ/ψ would be more useful for loop modeling.

Application to Loop Optimization Using Monte Carlo Minimization

We employ the local moves described earlier as a perturbation strategy in the Monte Carlo Minimization (MCM) method of Li and Scheraga,[40] and apply the method to the global energy minimization of a protein loop. MCM is a global optimization method by which local energy barriers can be overcome with energy minimization of the perturbed structure before a Metropolis criterion[41] is applied.

We now take advantage of the fact that some steric barriers that are hard to overcome by random moves of individual atoms could be bypassed by coordinated moves of multiple atoms. It has been reported that adding such concerted moves in Metropolis Monte Carlo simulations improves the sampling efficiency.[2,3,5–7] Our local moves are more general than those previously applied to MC simulations, but it is straightforward to apply these moves to MC. The same Jacobian as in refs. 3 and 7 needs to be included to satisfy the microscopic reversibility. Here we apply the concerted moves to MCM for the first time, and show significant improvement in efficiency in finding the global minimum.

Several other strategies for local moves have been applied to the loop optimization problem (see references in ref. 42), and among these, Local Torsional Deformation (LTD)[42] has been one of the most efficient methods of perturbing cyclic or loop structures when combined with MCM. In LTD, torsion angles are perturbed only locally, i.e., no bonds are rotated beyond the perturbed region. In addition, only those perturbations that keep the bond between the last perturbed atom and the first unperturbed atom within a ring closure range (0.5–3.54 Å) are considered.[42] In our approach, one torsion angle (ϕ or ψ) in the loop is perturbed, and six other torsion angles (three pairs of ϕ and ψ) are adjusted to keep the perturbation local. The perturbed conformations thus do not have any large strains due to unrealistic bond angles or bond lengths. Such a move is compared with LTD below.

Results

It is not intuitively obvious whether using perturbations exactly satisfying geometrical constraints (Exact Loop Closure, ELC) would be significantly more efficient than approximate perturbations like LTD, because a physical energy function can correct for the inaccurate geometry in the process of energy minimization. Figures 7 and 8 show that the moves based on our ELC actually greatly enhance the performance of MCM compared to approximate LTD in finding low energy conformations. Figure 7 illustrates that ELC finds the (putative) global minimum about three times faster than LTD, and Figure 8 shows that ELC has a much higher probability of finding the correct global minimum than LTD when the optimization starts with a random initial structure. The details on the simulations are presented in the next section.

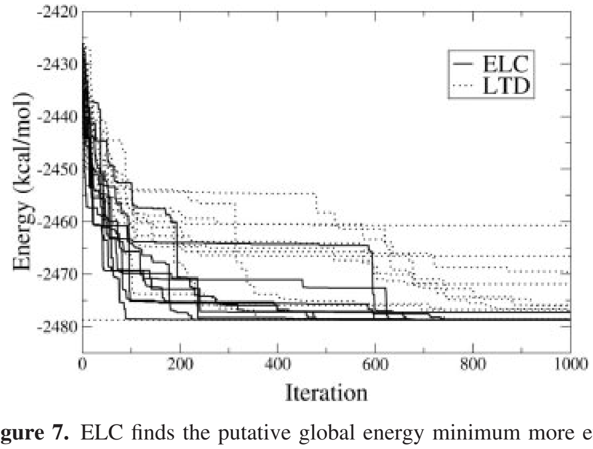

Figure 7. ELC finds the putative global energy minimum more efficiently than LTD. Ten MCM runs (1000 iterations, or minimizations, for each) are shown. Each run starts from the same initial conformations, which is 3.8 Å from the crystal structure. The ordinate represents the lowest energy found up to that iteration number. The (putative) global minimum is at −2478.7 kcal/mol, and is represented by the dotted line.

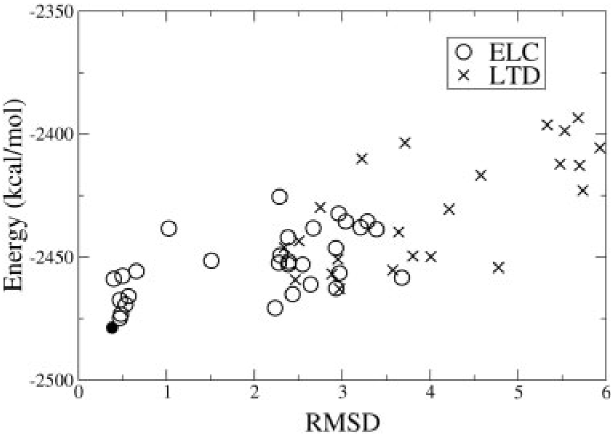

Figure 8. ELC finds lower energy and rmsd structures than LTD when MCM is started from diverse initial structures. The ordinate and abscissa are the lowest energy found and RMSD corresponding to the structure. The initial structures are generated by randomly choosing ϕ/ψ angles of the loop residues. One thousand iterations were performed for each of 30 random initial structures. Only 23 points are shown for LTD because seven other points have much higher energy and rmsd. The (putative) global minimum is represented by the back circle.

If keeping the backbone geometry within physically reasonable ranges improves efficiency, the same is expected for side chains. We thus draw the side chain torsion angles from a rotamer library. The rotamer probability distributions in the backbone dependent rotamer library of Dunbrack et al.[43] are used to perturb side-chain torsion angles. This rotamer method is compared with “random” side-chain perturbation method. Employing a rotamer library improves the performance: average lowest energies found after 10 runs of 1000 MCM iterations with rotamer and random methods are respectively, −2478.5 and −2477.2 kcal/mol for ELC, and −2473.2 and −2463.7 kcal/mol for LTD for the same initial loop structure as in Figure 7. Computations in Figures 7 and 8 were both performed with the side-chain rotamer library.

Methods

The energy function used is EEF1,[44] which is the CHARMM 19 polar hydrogen force field with a Gaussian implicit solvation model. An eight-residue loop (84–91) in Turkey egg lysozyme (pdb code 1351.pdb) is taken for this study because the (putative) global minimum of this loop is located very close to the crystal structure (about 0.3 Å). The loop RMSD is measured as the root-mean-square deviation in the main chain atoms of the loop when the three stem residues on both sides of the loop are optimally superimposed. The L-BFGS-b algorithm[45] by Zhu et al. is used for energy minimization with the gradient tolerance of 1 kcal/mol Å.

The details of modeling and parameters for both ELC and LTD are as follows. Hydrogen atoms are modeled on the crystal structure and the structure is energy minimized to remove bad contacts with harmonic constrains on heavy atoms with the force constant 5 kcal/mol. The resulting structure is 0.1 Å from the crystal structure. All other atoms are fixed except for the loop atoms in MCM. The temperature parameter kT for MCM is set to 1 kcal/mol for both ELC and LTD. In ELC, we choose three residues in the loop randomly whose ϕ/ψ angles compensate for the change of a driver torsion angle, which is also chosen randomly within the triangle formed by the three residues. The driver angle ρ is changed with uniform probability in the range $[\rho - f_1\pi, \rho + f_1\pi]$. $f_1 = 0.7$ was found to be optimal. Out of the multiple loop closure solutions, the closest solution to the current structure in RMSD is selected for the perturbation step in MCM. In LTD, four consecutive ϕ/ψ angles are perturbed with uniform probability in the range $[\rho - f_2\pi, \rho + f_2\pi]$, where $f_2 = 0.8$ was found to be optimal. We also tried 3–4 one or two consecutive angle movements following ref. 42, but they were less efficient. However, it has to be mentioned that comparisons of algorithms is not always straightforward, with a lot of parameters and technical details that can be varied. For both LTD and ELC, each side chain is perturbed independently with the probability 1/8, and the side-chain rotamer is selected from the backbone-dependent rotamer library with the backbone-dependent probability. The random side-chain perturbation method perturbs each side-chain torsion angle independently with the probability 1/8, and the angle value is drawn from a uniform range around the current value $[\chi - f\pi, \chi + f\pi]$, where $f = f_1$ for ELC and $f = f_2$ for LTD.

Conclusions

The bonded near-neighbor forces in a protein can be grouped into roughly three categories with respect to strength: hard forces associated with bond lengths, intermediate forces associated with bond angles, and soft forces associated with the ϕ/ψ dihedrals and side-chain angles. The forces associated with the ω dihedral of the peptide bond can be placed in the intermediate range. In considering a polypeptide chain it is tempting to concentrate on motions associated with the "soft" DOFs, i.e., those associated with ϕ/ψ. The other DOFs can be assumed to vary to a limited degree, although they can be fixed to arbitrary values as far as the geometric analysis is concerned.

The conformational space of a tripeptide unit (excluding N_1 and C'_3) can be seen as the Cartesian product of two circles, i.e., a torus. Exploring the conformation space of this simple system is straightforward in terms of the ϕ/ψ dihedrals. However, adding a constraint such as fixing the distance between $C_{\alpha 1}$ and $C_{\alpha 3}$ introduces a relationship between the two dihedrals, eq. (11), and this interdependence ("Rotation Transfer Function" or RTF) forms the basis of the analytical approach followed in this article. In general, fixed-distance constraints, whether resulting from NMR measurements for structure determination or as part of a strategy for exploring conformation space, imply sets of such transfer functions among angle DOFs and, when combined with the other "almost-rigid" constraints in a macromolecule, lead to a reduction in the dimensionality of the space of allowed motions. Allowing for small variability of additional DOFs provides a search algorithm with more room to maneuver, replacing barriers by narrow, passable corridors. Such constraint-compatible conformations form the natural low-energy terrain that needs to be explored thoroughly. Choosing coordinates that describe this terrain efficiently, is an essential part of this exploration, because the reduction in dimension comes at the price of complicated topology. Clearly, these ideas need not be limited to backbone motions, and extending them to side chains is essential if NMR derived distance constraints are to be included. In that regard we view the tripeptide loop closure and the variants presented here as one of many possible applications of the distance-angle relationship expressed by the basic RTF.

Several generalizations of the method presented here are possible. For example, the transfer functions are Fourier polynomials in the other angle variables as well. A reduction of these to polynomial form would lead to a new polynomial system, now in several additional variables. The zero sets become higher dimensional objects, and new methods can be brought to bear for finding closure solutions. We chose to treat these variables by simple search methods here, but a more complete search would be required if, for example, some energy criterion is included in the perturbation process.

The main advantage of a ϕ/ψ search method is that it avoids searching conformations that introduce distortions of the hard DOFs. The benefit of the method in reducing the size of the search space is not seriously affected if small variations in additional DOFs are allowed. Thin slivers of configurations replace the ϕ/ψ hypersurfaces, so that the volume of the allowed space is still dramatically reduced. To take full advantage of the intimate connection between the concerted moves idea and the true kinematic DOFs of the chain, a strategy of choosing moves should be informed about the effect of these moves vis. steric clashing with the rest of the chain as well as side chain placement. A possible extension could be the incorporation of obstacle avoidance and other similar ideas from robotic motion planning.

Acknowledgments

One of the authors (E.A.C.) would like to acknowledge the hospitality of the Dill group during several visits to UCSF. E.A.C. also acknowledges many useful conversations with John Chodera. We thank Dr. Bosco Ho for pointing out ref. 39, and Dr. Robert Bruccoleri for providing parameters for the cyclopentapeptide example.

Appendix A: Two-Cone Systems and the Rotation Transfer Function

In the body frame of the three fixed $C_{\alpha i}$ atoms, the $\mathbf{C}_{\alpha i}\mathbf{N}_i$ unit vector $\hat{\mathbf{r}}^{\sigma}_{i-1}$ and the $\mathbf{C}_{\alpha i}\mathbf{C}_i$ unit vector $\hat{\mathbf{r}}^{\tau}_i$ lie on cones about the $\mathbf{C}_{\alpha i-1}\mathbf{C}_{\alpha i}$ and $\mathbf{C}_{\alpha i}\mathbf{C}_{\alpha i+1}$ virtual bonds, respectively, assuming fixed bond lengths, angles, and dihedral ω [see Fig. 9a]. The σ_{i-1}

and τ_i rotations are not independent because the bond angle $N_i C_{\alpha i} C_i$ must be fixed (or in a limited range in general). If we can think of the cones of possible locations of the bonds about their corresponding virtual axes, then we must think of the angle constraint between the two bonds as a fixed distance condition between two generatrices of these cones as shown in Figure 9a. Clearly, to each position of one bond there can be at most two positions for the other. Because the four angles α, η, θ, ξ are constant, the alternative positions describe the possible conformations of the tetrahedral formed by these four angles in Figure 9b.

The ranges of these positions can change character and transit from both having one connected component (Fig. 10a) to where one of them splits to two disjoint arms (Fig. 10b).

At each two-cone, the σ_{i-1} and τ_i rotations are related by the θ_i bond angle constraint

$$\hat{\mathbf{r}}_i^{\tau} \cdot \hat{\mathbf{r}}_{i-1}^{\sigma} = \cos\theta, \tag{23}$$

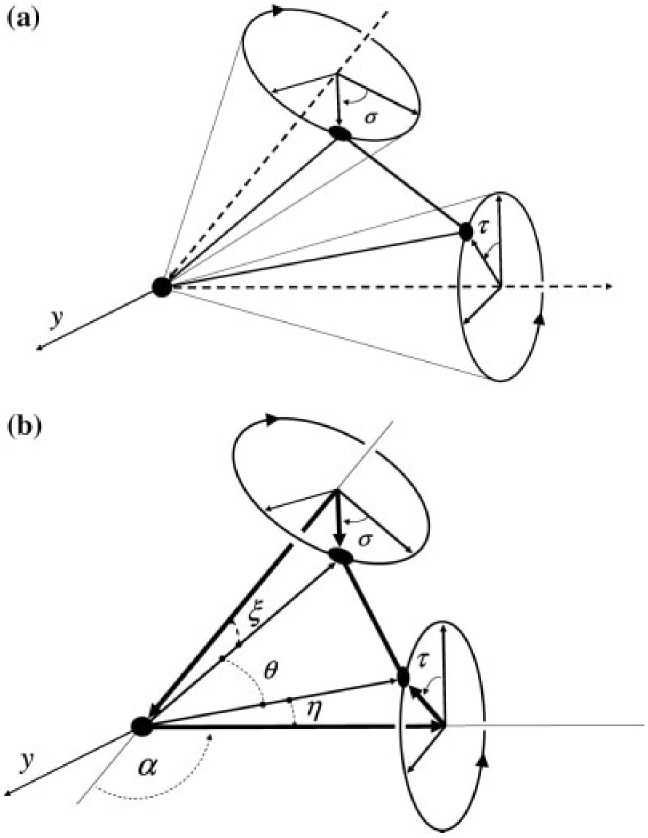

Figure 9. (a) The two cones at a double rotatable bond junction. The black circle denotes the C_α atom, the black oval on the τ-cone the end of the $\mathbf{C}_\alpha\mathbf{C}$ unit vector, and that on the σ-cone the end of $\mathbf{C}_\alpha\mathbf{N}$ unit vector. The dashed lines are the virtual bonds between the C_α atoms, and the line between the ovals shows the fixed distance constraint. (b) The angles at a double rotatable bond junction.

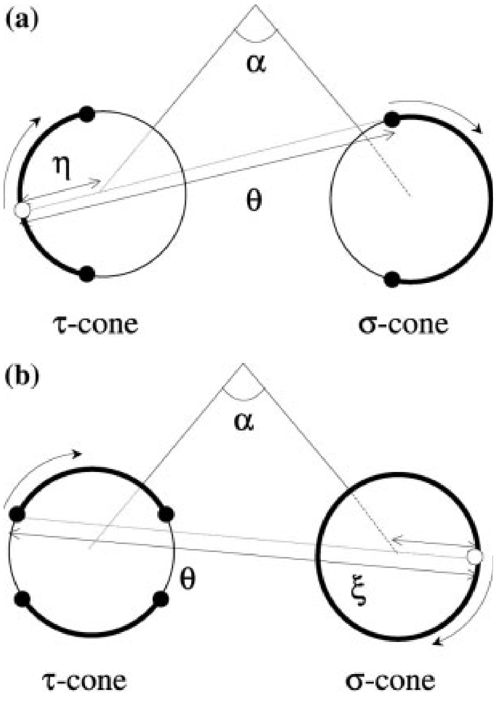

Figure 10. (a) A type of two cones (type IIb) in which both C (on the τ-cone) and N (on the σ-cone) trace connected segments. The black circles denote extreme positions of the N and C atoms. The white circle is the C position corresponding to the N at the black circle connected to it by a line. (b) A two-cone type (type IIIb) in which the C atom (on the τ-cone) traces two disjoint segments, and N (on the σ-cone) traces the whole circle.

which results in eq. (8) rewritten omitting subscripts for simplicity:

$$\begin{aligned}\cos\theta + \cos\eta\cos\xi\cos\alpha &= \sin\alpha(\sin\xi\cos\eta\cos\sigma + \cos\xi\sin\eta\cos\tau)\\ &\quad + \sin\xi\sin\eta(\sin\tau\sin\sigma + \cos\alpha\cos\tau\cos\sigma). \end{aligned}\tag{24}$$

To see the $\sigma - \tau$ relation more explicitly, $\tau := \tau_i$ is solved for given $\sigma := \sigma_{i-1}$, or σ given τ. Arranging eq. (24) as $a_t \cos\tau + b_t \sin\tau = c_t$ gives

$$\tau = \tau_o \pm \arccos(c_t/\sqrt{a_t^2 + b_t^2}), \tag{25}$$

where

$$\begin{gathered} a_t = \cos\alpha\sin\xi\sin\eta\cos\sigma + \sin\alpha\cos\xi\sin\eta, \\ b_t = \sin\xi\sin\eta\sin\sigma, \\ c_t = \cos\theta + \cos\alpha\cos\xi\cos\eta + \sin\alpha\sin\xi\cos\eta\cos\sigma, \\ \cos\tau_o = a_t/\sqrt{a_t^2 + b_t^2}, \quad \sin\tau_o = b_t/\sqrt{a_t^2 + b_t^2}. \end{gathered}\tag{26}$$

Given τ, σ is expressed as

$$\sigma = \sigma_o \pm \arccos(c_s/\sqrt{a_s^2 + b_s^2}), \tag{27}$$

where

$$a_s = \cos\alpha\sin\xi\sin\eta\cos\tau + \sin\alpha\sin\xi\cos\eta,$$

$$b_s = \sin\xi\sin\eta\sin\tau,$$

$$c_s = \cos\theta + \cos\alpha\cos\xi\cos\eta - \sin\alpha\cos\xi\sin\eta\cos\tau,$$

$$\cos\sigma_o = a_s/\sqrt{a_s^2 + b_s^2},\quad \sin\sigma_o = b_s/\sqrt{a_s^2 + b_s^2}. \quad (28)$$

Equation (25) has two solutions if $|c_t/\sqrt{a_t^2 + b_t^2}| < 1$, one if $= 1$, and none if > 1. The range of σ in which solutions exist is determined by σ_1 and σ_2 which are two roots of $c_t^2 = a_t^2 + b_t^2$, and likewise for τ. The roots can also be found by noting the following geometrical relations:

$$\hat{\mathbf{r}}^{\sigma}_{i-1}\cdot\hat{\mathbf{z}}_i = \cos(\theta \pm \eta),\quad \hat{\mathbf{r}}^{\tau}_i\cdot\hat{\mathbf{z}}_{i-1} = -\cos(\theta \pm \xi), \quad (29)$$

which give

$$\cos(\theta \pm \eta) = \sin\sigma\sin\xi\sin\alpha - \cos\xi\cos\alpha, \quad (30)$$

$$\cos(\theta \pm \xi) = \sin\tau\sin\eta\sin\alpha - \cos\eta\cos\alpha. \quad (31)$$

Equations (30) and (31) have roots if the following conditions are satisfied:

$$[\cos(\theta \pm \eta) + \cos(\alpha + \xi)][\cos(\theta \pm \eta) + \cos(\alpha - \xi)] \le 0, \quad (32)$$

$$[\cos(\theta \pm \xi) + \cos(\alpha + \eta)][\cos(\theta \pm \xi) + \cos(\alpha - \eta)] \le 0. \quad (33)$$

We call the roots of Equations (30) and (31) $\sigma_\pm$ and $\tau_\pm$. When there exist roots, there are six two-cone types, depending on the sign ($\pm$) of each root.

I. No solution
IIa. σ_+, τ_+
IIb. σ_-, τ_-
IIIa. σ_+, σ_-
IIIb. τ_+, τ_-
IVa. σ_+, τ_-
IVb. σ_-, τ_+

It can be seen that, in the case of IIa, increasing θ increases the allowable range of σ/τ, and decreasing θ increases the range for IIb (see Fig. 10a). This fact is used in the simple perturbation method described earlier. For other two-cone types, the θ values are increased in the perturbation algorithm.

The $\sigma_{i-1} - \tau_i$ relationship is more complicated than the σ_i–τ_i relationship ($\sigma_i = \tau_i + \delta_i$), but more detailed understanding of the σ–τ relationship at the junction of the two rotatable bonds is useful. First, it makes it possible to predict the effect of bond angle perturbations, as described earlier. Second, it also reveals the geometrical restriction on the side-chain location, especially C_β, due to the correlated movement of N_i, $C_{\alpha i}$, and C_i atoms as described by correlation of the σ_{i-1} and τ_i rotations. To demonstrate this, in Figure 10, we show the $\sigma - \tau$ relationship and the corresponding C_β positions obtained from the canonical tripeptide geometry, and compare with those extracted from the structure database Top500.[36] Figures 11a and b shows possible ranges for σ–τ and C_β when α is fixed at 90° (this value of α has the maximum density in the α distribution in the database). The database points are for $\alpha = 90 \pm 0.5°$. Clearly, the theoretical σ–τ curve computed from the canonical bond angles shows excellent agreement with the database points, the large majority of structures clustering about the Ramachandran allowable portion of the σ–τ curve (Fig. 11a). Reconstructing C_β using canonical angles also shows close agreement (Fig. 11b). This is the unimodal case (also illustrated in Fig. 10a), and it is the most commonly occurring configuration in the database. Figures 11c and d shows an example of a bimodal case (also illustrated in Fig. 10b), for $\alpha = 111°$, which is close to the second density maximum in the distribution of α. Here we see a small discrepancy, indicating that the structure is stressed, i.e., some of the parameters are off their typical values. The stress turns out to be even stronger if one compares the C_β distribution in the database to that reconstructed assuming typical angle values. We are studying these properties further, together with their possible application to side-chain optimization, especially when backbone flexibility is also taken into account.

Appendix B: Coefficients of the Polynomials

Equation (8) is written as a double Fourier series

$$0 = a_i + b_i\cos\sigma_{i-1} + c_i\cos\tau_i + d_i\cos\sigma_{i-1}\cos\tau_i + e_i\sin\sigma_{i-1}\sin\tau_i, \quad (34)$$

where the coefficients are

$$a_i = -\cos\theta_i - \cos\eta_i\cos\xi_{i-1}\cos\alpha_i$$

$$b_i = \sin\alpha_i\sin\xi_{i-1}\cos\eta_i$$

$$c_i = \sin\alpha_i\cos\xi_{i-1}\sin\eta_i$$

$$d_i = \cos\alpha_i\sin\xi_{i-1}\sin\eta_i$$

$$e_i = \sin\xi_{i-1}\sin\eta_i.$$

Now introduce the half-angle formulas Eqs. (9) and (10) into (34) to arrive at a system of three biquadratics in w_i, u_i, $i = 1, 2, 3$,

$$0 = a_i + b_i\frac{1 - w_{i-1}^2}{1 + w_{i-1}^2} + c_i\frac{1 - u_i^2}{1 + u_i^2} + d_i\frac{1 - w_{i-1}^2}{1 + w_{i-1}^2}\frac{1 - u_i^2}{1 + u_i^2} + e_i\frac{2w_{i-1}}{1 + w_{i-1}^2}\frac{2u_i}{1 + u_i^2},$$

or equivalently:

$$0 = a_i(1 + w_{i-1}^2)(1 + u_i^2) + b_i(1 - w_{i-1}^2)(1 + u_i^2) + c_i(1 + w_{i-1}^2)(1 - u_i^2) + d_i(1 - w_{i-1}^2)(1 - u_i^2) + e_i 4w_{i-1}u_i,$$

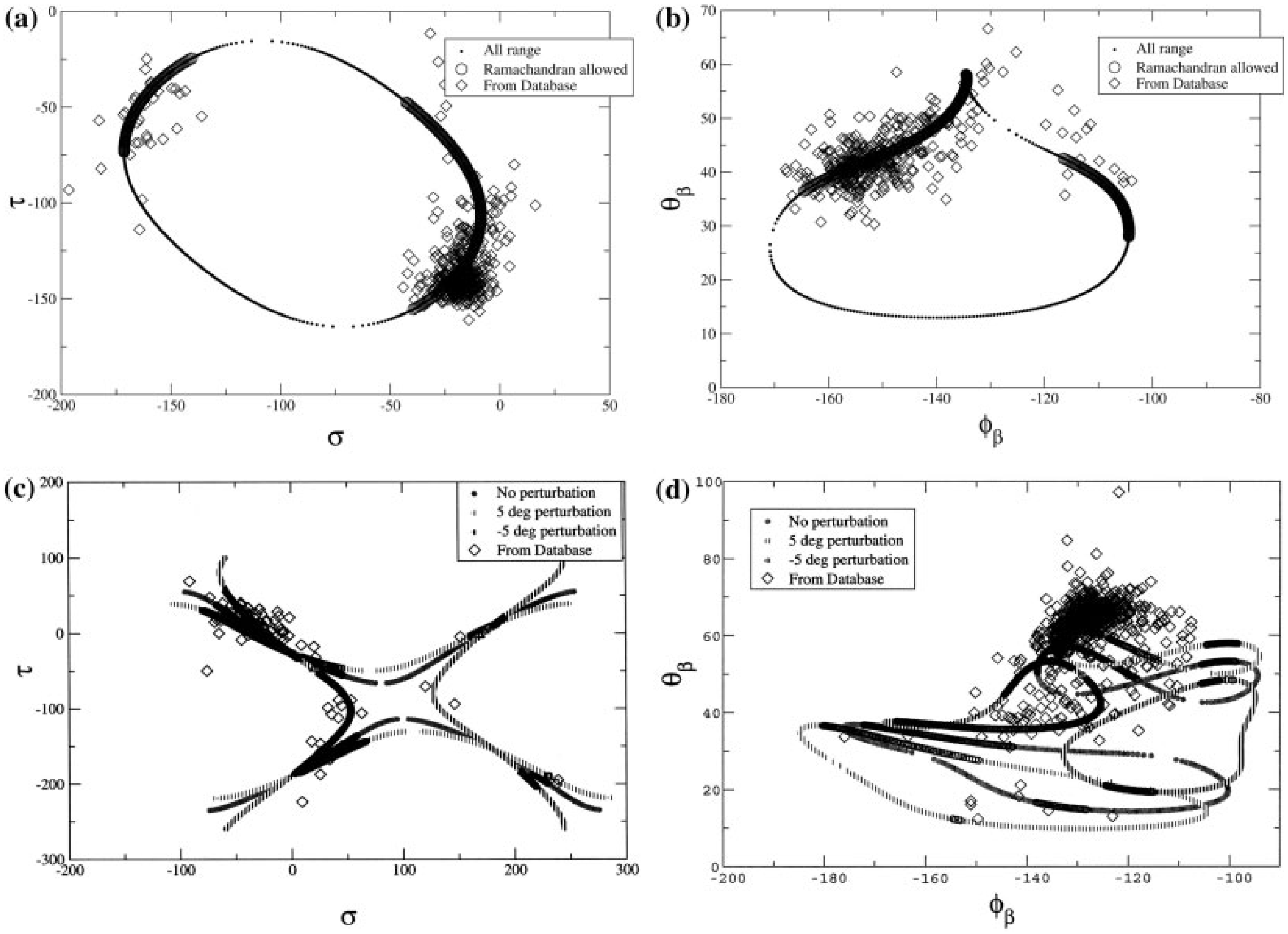

Figure 11. (a) The σ–τ relationship given by the Rotation Transfer Function for $\alpha = 90°$ and typical values of $\theta = 111.6°$, $\xi = 16.63°$, $\eta = 19.13°$, plotted together with σ–τ values in the database. (b) Plot of the location of C_β as computed from the σ–τ values in the theoretical curve of (a), against C_β positions from the database, both shown in spherical coordinates. (c) Same as (a), but for $\alpha = 111°$. Two more curves with η perturbation of 5° and −5° are shown together, which improve fit to the database points. The case is bimodal, although its character shifts as η is changed. (d) The C_β plots corresponding to the situation in (c).

Expanding and regrouping results in eq. (11):

$$A_i w_{i-1}^2 u_i^2 + B_i w_{i-1}^2 + C_i w_{i-1} u_i + D_i u_i^2 + E_i = 0 \qquad (35)$$

where

$$A_i = a_i - b_i - c_i + d_i = -\cos\theta_i - \cos(\alpha_i - \xi_{i-1} - \eta_i)$$

$$B_i = a_i - b_i + c_i - d_i = -\cos\theta_i - \cos(\alpha_i + \xi_{i-1} - \eta_i)$$

$$C_i = e_i = 4\sin\xi_{i-1}\sin\eta_i$$

$$D_i = a_i + b_i - c_i - d_i = -\cos\theta_i - \cos(\alpha_i - \xi_{i-1} + \eta_i)$$

$$E_i = a_i + b_i + c_i + d_i = -\cos\theta_i - \cos(\alpha_i + \xi_{i-1} + \eta_i).$$

We now eliminate the variables w_i: using the twist transformation, eq. (12),

$$w_i = \frac{u_i + \Delta_i}{1 - \Delta_i u_i}, \quad \Delta_i = \tan\delta_i/2,$$

in eq. (35) we find

$$A_i\left(\frac{u_{i-1} + \Delta_{i-1}}{1 - \Delta_{i-1}u_{i-1}}\right)^2 u_i^2 + B_i\left(\frac{u_{i-1} + \Delta_{i-1}}{1 - \Delta_{i-1}u_{i-1}}\right)^2 + C_i\frac{u_{i-1} + \Delta_{i-1}}{1 - \Delta_{i-1}u_{i-1}}u_i + D_i u_i^2 + E_i = 0 \quad (36)$$

Finally, the derivation of the coupled biquadratic polynomials eqs. (13), (14), and (15), is carried out by multiplying through by $(1 - \Delta_{i-1}u_{i-1})^2$ and regrouping. Because

$$\Delta = \frac{\sin\delta}{1 + \cos\delta}, \quad \Delta^2 = \frac{1 - \cos\delta}{1 + \cos\delta},$$

we multiply the resulting expressions through by $(1 + \cos \delta_{i-1})/2$ to arrive at the expression for the coefficients:

$$p^{(i)}_{22} = -\cos\theta_i - \cos\xi_{i-1}\cos(\alpha_i - \eta_i) - \cos\delta_{i-1}\sin\xi_{i-1}\sin(\alpha_i - \eta_i)$$

$$p^{(i)}_{21} = -2\sin\delta_{i-1}\sin\xi_{i-1}\sin\eta_i$$

$$p^{(i)}_{20} = -\cos\theta_i - \cos\xi_{i-1}\cos(\alpha_i + \eta_i) - \cos\delta_{i-1}\sin\xi_{i-1}\sin(\alpha_i + \eta_i)$$

$$p^{(i)}_{12} = -2\sin\delta_{i-1}\sin\xi_{i-1}\sin(\alpha_i - \eta_i)$$

$$p^{(i)}_{11} = 4\cos\delta_{i-1}\sin\xi_{i-1}\sin\eta_i$$

$$p^{(i)}_{10} = -2\sin\delta_{i-1}\sin\xi_{i-1}\sin(\alpha_i + \eta_i)$$

$$p^{(i)}_{02} = -\cos\theta_i - \cos\xi_{i-1}\cos(\alpha_i - \eta_i) + \cos\delta_{i-1}\sin\xi_{i-1}\sin(\alpha_i - \eta_i)$$

$$p^{(i)}_{01} = 2\sin\delta_{i-1}\sin\xi_{i-1}\sin\eta_i$$

$$p^{(i)}_{00} = -\cos\theta_i - \cos\xi_{i-1}\cos(\alpha_i + \eta_i) + \cos\delta_{i-1}\sin\xi_{i-1}\sin(\alpha_i + \eta_i).$$

Equations (13), (14), and (15) are now rewritten as

$$P_1(u_3, u_1) = \sum_{j=0}^{2}\left(\sum_{k=0}^{2} p^{(1)}_{jk} u_3^j\right) u_1^k = \sum_{j=0}^{2} L_j u_1^j,$$

$$P_2(u_1, u_2) = \sum_{k=0}^{2}\left(\sum_{j=0}^{2} p^{(2)}_{jk} u_2^k\right) u_1^j = \sum_{k=0}^{2} M_k u_1^k,$$

and

$$P_3(u_2, u_3) = \sum_{j=0}^{2}\left(\sum_{k=0}^{2} p^{(3)}_{jk} u_3^k\right) u_2^j = \sum_{j=0}^{2} N_j u_2^j,$$

where

$$L_j := L_j(u_3) := \sum_{k=0}^{2} p^{(1)}_{jk} u_3^j, \quad M_k := M_k(u_2) := \sum_{j=0}^{2} p^{(2)}_{jk} u_2^k,$$

and

$$N_j := N_j(u_3) := \sum_{k=0}^{2} p^{(3)}_{jk} u_3^k.$$

The resultant of P_1 and P_2, whose vanishing guarantees a common root in u_1, is given by the determinant

$$R_8(u_2, u_3) = \begin{vmatrix} L_2 & L_1 & L_0 & 0 \\ 0 & L_2 & L_1 & L_0 \\ M_2 & M_1 & M_0 & 0 \\ 0 & M_2 & M_1 & M_0 \end{vmatrix} = \begin{vmatrix} L_2 & L_0 \\ M_2 & M_0 \end{vmatrix}^2 - \begin{vmatrix} L_2 & L_1 \\ M_2 & M_1 \end{vmatrix} \begin{vmatrix} L_1 & L_0 \\ M_1 & M_0 \end{vmatrix}$$

Because all the nonvanishing elements are products of two quadratics in u_2 and two quadratics in u_3, the resultant is a biquartic in these variables, and has the form

$$R_8(u_2, u_3) = \sum_{j,k=0}^{4} q_{jk} u_2^j u_3^k.$$

Here, the $5 \times 5 = 25$ quantities q_{jk} are found in terms of products of the $a_{jk} := p^{(1)}_{jk}$ and $b_{jk} := p^{(2)}_{jk}$ by expressing as a sum of six tensor products:

We write R_8 as a quartic in u_2 introducing the functions Q_j, quartics in u_3:

$$R_8 = \sum_{j=0}^{4}\left(\sum_{k=0}^{4} q_{jk} u_3^k\right) u_2^j =: \sum_{j=0}^{4} Q_j u_2^j.$$

The final resultant, which eliminates u_2 to arrive at a degree 16 polynomial in u_3 is given by:

$$R_{16} = det(S)$$

where the matrix S is given as:

$$S(u_3) := \sum_{k=0}^{4} S_k u_3^k = \begin{pmatrix} N_2 & N_1 & N_0 & 0 & 0 & 0 \\ 0 & N_2 & N_1 & N_0 & 0 & 0 \\ 0 & 0 & N_2 & N_1 & N_0 & 0 \\ 0 & 0 & 0 & N_2 & N_1 & N_0 \\ Q_4 & Q_3 & Q_2 & Q_1 & Q_0 & 0 \\ 0 & Q_4 & Q_3 & Q_2 & Q_1 & Q_0 \end{pmatrix} \tag{37}$$

so that

$$S_k := \begin{pmatrix} c_{2k} & c_{1k} & c_{0k} & 0 & 0 & 0 \\ 0 & c_{2k} & c_{1k} & c_{0k} & 0 & 0 \\ 0 & 0 & c_{2k} & c_{1k} & c_{0k} & 0 \\ 0 & 0 & 0 & c_{2k} & c_{1k} & c_{0k} \\ q_{4k} & q_{3k} & q_{2k} & q_{1k} & q_{0k} & 0 \\ 0 & q_{4k} & q_{3k} & q_{2k} & q_{1k} & q_{0k} \end{pmatrix}$$

(where we defined $c_{ij} := p^{(3)}_{ij}$, with $c_{i3} = c_{i4} = 0$, $i = 0, 1, 2$). These matrices can be used directly in the matrix polynomial approach, which finds the solutions as eigenvalues of a "companion" matrix pencil. The computation of the polynomial coefficients for the direct approach requires some additional computations described below. We proceed by a Laplace expansion[46] of eq. (37) by complementary minors of order 3. First, we rearrange the rows of S:

$$det\ S = \begin{vmatrix} N_2 & N_1 & N_0 & 0 & 0 & 0 \\ 0 & N_2 & N_1 & N_0 & 0 & 0 \\ 0 & 0 & N_2 & N_1 & N_0 & 0 \\ 0 & 0 & 0 & N_2 & N_1 & N_0 \\ Q_4 & Q_3 & Q_2 & Q_1 & Q_0 & 0 \\ 0 & Q_4 & Q_3 & Q_2 & Q_1 & Q_0 \end{vmatrix}$$

$$= - \begin{vmatrix} N_2 & N_1 & N_0 & 0 & 0 & 0 \\ Q_4 & Q_3 & Q_2 & Q_1 & Q_0 & 0 \\ 0 & 0 & N_2 & N_1 & N_0 & 0 \\ 0 & N_2 & N_1 & N_0 & 0 & 0 \\ 0 & Q_4 & Q_3 & Q_2 & Q_1 & Q_0 \\ 0 & 0 & 0 & N_2 & N_1 & N_0 \end{vmatrix}$$

$$= - \sum_{1 \le i_1 < i_2 i_3 \le 6} (-1)^{i_1+i_2+i_3} det\ S(1, 2, 3; i_1, i_2, i_3)$$

$$\times\ det\ S(4, 5, 6; i_4, i_5, i_6)$$

where $S(1, 2, 3; i_1, i_2, i_3)$ is the 3×3 submatrix of S formed by elements in rows 1, 2, 3 and columns $i_1 < i_2 < i_3$. Also, $i_4 < i_5 < i_6$ and i_4, i_5, i_6 differ from i_1, i_2, i_3. We introduce the 3×5 submatrix

$$P := \begin{pmatrix} N_2 & N_1 & N_0 & 0 & 0 \\ Q_4 & Q_3 & Q_2 & Q_1 & Q_0 \\ 0 & 0 & N_2 & N_1 & N_0 \end{pmatrix}$$

and the 3×3 minors $T(i, j, k)$ formed by the columns i, j, and k of P. Then, the Laplace expansion of S in terms of the minors based on rows 1, 2, and 3, and their complements from rows 4, 5, and 6,[46] can be written compactly in the form:

$$det\ S = T(1, 2, 3)T(3, 4, 5) - T(1, 2, 4)T(2, 4, 5)$$
$$+ T(1, 2, 5)T(2, 3, 5) + T(1, 3, 4)T(1, 4, 5)$$
$$- T(1, 3, 5)T(1, 3, 5) + T(1, 4, 5)T(1, 2, 5)$$

The computation of the resultant proceeds with the nine quantities $T(i, j, k)$ above. Because they are sums of products of terms of the form $N_\alpha N_\beta Q_\gamma$ they are polynomials in u_1 of degree 8. We list the expressions for these below in terms of the N_i, Q_j involved. Once the Ts have been computed, we need to compute the products above, i.e., we need to compute 6 binary products of polynomials of degree 8. A certain amount of factoring can be utilized to further reduce the operational count of this procedure.

We give now the $T(i, j, k)$:

$$T(1, 2, 3) = \begin{vmatrix} N_2 & N_1 & N_0 \\ Q_4 & Q_3 & Q_2 \\ 0 & 0 & N_2 \end{vmatrix} = N_2 \begin{vmatrix} N_2 & N_1 \\ Q_4 & Q_3 \end{vmatrix}$$

$$T(1, 2, 4) = \begin{vmatrix} N_2 & N_1 & 0 \\ Q_4 & Q_3 & Q_1 \\ 0 & 0 & N_1 \end{vmatrix} = N_1 \begin{vmatrix} N_2 & N_1 \\ Q_4 & Q_3 \end{vmatrix}$$

$$T(1, 2, 5) = \begin{vmatrix} N_2 & N_1 & 0 \\ Q_4 & Q_3 & Q_0 \\ 0 & 0 & N_0 \end{vmatrix} = N_0 \begin{vmatrix} N_2 & N_1 \\ Q_4 & Q_3 \end{vmatrix}$$

$$T(1, 3, 4) = \begin{vmatrix} N_2 & N_0 & 0 \\ Q_4 & Q_2 & Q_1 \\ 0 & N_2 & N_1 \end{vmatrix} = -N_2 \begin{vmatrix} Q_2 & Q_1 \\ N_2 & N_1 \end{vmatrix} - Q_4 N_0 N_1$$

$$T(1, 3, 5) = \begin{vmatrix} N_2 & N_0 & 0 \\ Q_4 & Q_2 & Q_0 \\ 0 & N_2 & N_0 \end{vmatrix} = -N_2 \begin{vmatrix} N_2 & N_0 \\ Q_2 & Q_0 \end{vmatrix} - Q_4 N_0^2$$

$$T(1, 4, 5) = \begin{vmatrix} N_2 & 0 & 0 \\ Q_4 & Q_1 & Q_0 \\ 0 & N_1 & N_0 \end{vmatrix} = -N_2 \begin{vmatrix} N_1 & N_0 \\ Q_1 & Q_0 \end{vmatrix}$$

$$T(2, 3, 5) = \begin{vmatrix} N_1 & N_0 & 0 \\ Q_3 & Q_2 & Q_0 \\ 0 & N_2 & N_0 \end{vmatrix} = -N_1 \begin{vmatrix} N_2 & N_0 \\ Q_2 & Q_0 \end{vmatrix} - N_0^2 Q_3$$

$$T(2, 4, 5) = \begin{vmatrix} N_1 & 0 & 0 \\ Q_3 & Q_1 & Q_0 \\ 0 & N_1 & N_0 \end{vmatrix} = -N_1 \begin{vmatrix} N_1 & N_0 \\ Q_1 & Q_0 \end{vmatrix}$$

$$T(3, 4, 5) = \begin{vmatrix} N_0 & 0 & 0 \\ Q_2 & Q_1 & Q_0 \\ N_2 & N_1 & N_0 \end{vmatrix} = -N_0 \begin{vmatrix} N_1 & N_0 \\ Q_1 & Q_0 \end{vmatrix}$$

From these expressions, whose computation involves only 4 distinct 2×2 determinants, we can compute the final polynomial. This computation can be done analytically, by deriving the lengthy expressions for the coefficients of the final polynomial in terms of the coefficients of the original polynomials. These analytical expressions can be useful, especially if one wants to study the effect of varying parameters on the behavior of the solution of the tripeptide loop closure. For the calculations reported in this article, the computation of the coefficients was done numerically. In this case, it is optimal to compute the 8th degree polynomials associated with each of the $T(i, j, k)$ and then compute the six polynomial products (which can be easily reduced to five polynomial multiplications with appropriate factorizations).

Appendix C: Systems of Polynomials and Resultants

The resultant of a system of polynomials in several variables is a necessary and sufficient condition for the existence of a common solution. For two polynomials, $F_m(u)$ and $F_n(u)$ of degrees m and n, to have a common solution u they must have a factor in common, i.e., there must exist polynomials $g(u)$ and $h(u)$ of degrees $\le n - 1$ and $\le m - 1$, respectively, such that

$$gF_m + hF_n = 0.$$

This leads to a system of $m + n$ linear homogeneous equations for determining the coefficients of g and h, and the resultant is the determinant of the matrix associated with that system. We demonstrate how this works for two second order equations in a single variable. Let

$$f_1(u) = a_2u^2 + a_1u + a_0 = 0$$

$$f_2(u) = b_2u^2 + b_1u + b_0 = 0.$$

If these have a common root, say u^*, they must be of the form

$$f_1(u) = a_2(u - u^*)(u - u_1) = 0$$

$$f_2(u) = b_2(u - u^*)(u - u_2) = 0$$

so that there exist two polynomials of degree 1, $g(x) = b_2(u - u_2)$ and $h(x) = -a_2(u - u_1)$ such that

$$g(u)f_1(u) + h(u)f_2(u) = 0. \tag{38}$$

Because the roots are assumed unknown, we simply write

$$g(u) = g_1u + g_0, \quad h(u) = h_1u + h_0$$

and eq. (38) becomes

$$(g_1u + g_0)(a_2u^2 + a_1u + a_0) + (h_1u + h_0)(b_2u^2 + b_1u + b_0) = 0$$

or, grouping like powers of u together

$$(g_1a_2 + h_1b_2)u^3 + (g_1a_1 + g_0a_2 + h_1b_1 + h_0b_2)u^2 + (g_0a_1 + g_1a_0 + h_0b_1 + h_1b_0)u + (g_0a_0 + h_0b_0) = 0$$

which can be written in the equivalent form

$$(g_1 \quad g_0 \quad h_1 \quad h_0)\begin{pmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{pmatrix}\begin{pmatrix} u^3 \\ u^2 \\ u \\ 1 \end{pmatrix} = 0$$

so that the left and right null vectors give, respectively, the coefficients of the two factor polynomials and the (common) zero of the original pair. The rank deficiency of the coefficient matrix (and the vanishing of its determinant, i.e., the resultant) is the necessary and sufficient condition for the existence of these null vectors.

Once the vanishing of the determinant above has been established, finding u is straightforward; discarding the third equation implied above for the right null-vector (because it is dependent on the others), and moving the column associated with the component 1 to the right-hand side, we solve the resulting system for u using Cramer's rule:

$$u = \frac{\begin{vmatrix} a_2 & a_1 & 0 \\ 0 & a_2 & -a_0 \\ 0 & b_2 & -b_0 \end{vmatrix}}{\begin{vmatrix} a_2 & a_1 & a_0 \\ 0 & a_2 & a_1 \\ 0 & b_2 & b_1 \end{vmatrix}}$$

The above technique is applied to eqs. (18) and (19) to give u_2 and u_1, once u_3 is obtained:

$$u_2 = \frac{\begin{vmatrix} N_2 & N_1 & N_0 & 0 & 0 \\ 0 & N_2 & N_1 & N_0 & 0 \\ 0 & 0 & N_2 & N_1 & 0 \\ 0 & 0 & 0 & N_2 & -N_0 \\ 0 & Q_4 & Q_3 & Q_2 & -Q_0 \end{vmatrix}}{\begin{vmatrix} N_2 & N_1 & N_0 & 0 & 0 \\ 0 & N_2 & N_1 & N_0 & 0 \\ 0 & 0 & N_2 & N_1 & N_0 \\ 0 & 0 & 0 & N_2 & N_1 \\ 0 & Q_4 & Q_3 & Q_2 & Q_1 \end{vmatrix}},$$

where N_j and Q_j are functions of u_3 as described in Appendix B, and

$$u_1 = \frac{\begin{vmatrix} L_2 & L_1 & 0 \\ 0 & L_2 & -L_0 \\ 0 & M_2 & -M_0 \end{vmatrix}}{\begin{vmatrix} L_2 & L_1 & L_0 \\ 0 & L_2 & L_1 \\ 0 & M_2 & M_1 \end{vmatrix}},$$

where L_j and M_j are functions of u_3 and u_2, respectively, also given in Appendix B.

References

1. Tramontano, A.; Leplae, R.; Morea, V. Proteins 2001, 45, 22.
2. Gō, N.; Scheraga, H. A. Macromolecules 1978, 11, 552.
3. Dodd, L. R.; Boone, T. D.; Theodorou, D. N. Mol Phys 1993, 78, 961.
4. Wu, M. G.; Deem, M. W. Molecular Phys, 1999, 97, 559.
5. Wakana, H.; Wako, H.; Saitô, N. Int J Pept Protein Res 1984, 23, 315.
6. Knapp, E.-W.; Irgens-Defregger, A. J Comput Chem 1992, 14, 19.
7. Dinner, A. R. J Comput Chem 2000, 21, 1132.
8. Elofsson, A.; Le Grand, S. M.; Eisenberg, D. Proteins 1995, 13, 73.
9. Ulyanov, N. B.; Schmitz, U.; James, T. L. J Biomolecular NMR 1993, 3, 547.
10. Cahill, S.; Cahill, M.; Cahill, K. J Comp Chem 2003, 24, 1364.
11. Ulmschneider, J. P.; Jorgensen, W. L. J Chem Phys 2003, 118, 4261.
12. Wedemeyer, W. J.; Scheraga, H. A. J Comput Chem 1999, 20, 819.
13. Gō, N.; Scheraga, H. A. Macromolecules 1970, 3, 178.
14. Hartenberg, R. S.; Denavit, J. Kinematic Synthesis of Linkages; McGraw-Hill: New York, 1964.
15. Henrici, P. Applied and Computational Complex Analysis; Prentice-Hall: New York, 1974.
16. Sturmfels, B. In Applications of Computational Algebraic Geometry, Proceedings of Symposia in Applied Mathematics; Cox, D. A.; Sturmfels, B., Eds.; Am Math Soc 1997, p. 25.
17. Petitjean, S. J Math Imaging Vision 1999, 10, 1.
18. Wampler, C.; Morgan, A. Mech Mach Theory 1991, 26, 91.
19. Duffy, J. Analysis of Mechanisms and Robot Manipulators; Arnold: London, 1980.
20. Hunt, K. H. Kinematic Geometry of Mechanisms; Oxford: Oxford Univ Press, 1990.
21. Manocha, D. In Numerical Methods for Solving Polynomial Equations, Proceedings of Symposia in Applied Mathematics, Cox, D. A.; Sturmfels, B., Eds.; Am Math Soc 1997, p. 41.
22. Freudenstein, G. Mech Mach Theory 1973, 8, 151.
23. Lee, H.-Y.; Liang, C.-G. Mech Mach Theory 1988, 23, 219.
24. Bruccoleri, R. E.; Karplus, M. Macromolecules 1985, 18, 2767.

25. Bruccoleri, R. E.; Karplus, M. Biopolymers 1987, 26, 137.
26. Lee, H.-Y.; Liang, C.-G. Mech Mach Theory 1988, 23, 209.
27. Vermeer, P. Proceedings of the 13th ACM Symposium on Computational Geometry; Nice: France, June 1997.
28. Manseur, R.; Doty, K. L. Int J Robotics Res 1989, 8, 75.
29. Fine, R. M.; Wang, H.; Shenkin, P. S.; Yarmush, D. L.; Levinthal, C. Proteins 1986, 1, 342.
30. Zheng, Q.; Rosenfeld, R.; Vajda, S.; DeLisi, C. Protein Sci 1993, 2, 1242.
31. Jacobson, M. P.; Pincus, D. L.; Rapp, C. S.; Day, T. J. F.; Honig, B.; Shaw, D. E.; Friesner, R. A. Proteins, accepted.
32. Gelfand, I. M.; Kapranov, M. M.; Zelevinsky, A. V. Discriminants, Resultants and Multidimensional Determinants; Birkhäuser: Boston, 1994.
33. Bricard, R. J Math Pures Appl 1897, 3, 113.
34. Bennet, G. T. Proc Lond Math Soc 1912, 10, 309, 2nd Series.
35. Bernshtein, D. N. Functional Anal Appl 1975, 9, 183.
36. http://kinemage.biochem.duke.edu/databases/top500.php.
37. Hook, D. G.; McAree, P. R. Using Sturm Sequences to Bracket Real Roots of Polynomial Equations from "Graphics Gems;" Academic Press: New York, 1990. http://www.acm.org/pubs/tog/GraphicsGems/gems/Sturm/.
38. Gō, N.; Scheraga, H. A. Macromolecules 1970, 3, 188.
39. Canutescu, A. A.; Dunbrack, R. L. Protein Sci 2003, 12, 963.
40. Li, Z.; Scheraga, H. A. Proc Natl Acad Sci USA 1987, 84, 6611.
41. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. J Chem Phys 1953, 21, 1087.
42. Baysal, C.; Meirovitch, H. J Phys Chem A 1997, 101, 2185.
43. http://www.fccc.edu/research/labs/dunbrack/bbdep.html.
44. Lazaridis, T.; Karplus, M. Proteins 1999, 35, 133.
45. Zhu, C.; Byrd, R. H.; Lu, P.; Nocedal, J. L-BFGS-B; Northwestern Univ: Evanston, IL, 1996.
46. Iyanaga, S.; Kawada, Y., Eds. Encyclopedia Dictionary of Mathematics; MIT Press: Cambridge, MA, 1980, p. 349.