FORME: An interactive package for protein backbone deformation

P. Tuffery,† M.C. Vaney,* J.P. Mornon* and S. Hazout†

†Unité de Recherches Biomathématiques et Biostatistiques, INSERM U263, Université Paris 7, Paris, France
**Laboratoire de Minéralogie-Cristallographie, CNRS URA09, Universités Paris 6 et 7, Paris, France*

The FORME package presented herein is designed for modeling purposes: It allows interactive deformation of the protein backbone. General formalism on transformations is introduced and the operators of stretching inside an "acceptance area" and stretching with end-block invariance (i.e., governed by a translational moving) are described. A discussion is presented on the choice of strategy to achieve an interactive deformation tool. Perspectives about complex transformations are presented.

Keywords: proteins, graphic display, backbone deformation

INTRODUCTION

In the last ten years, molecular graphics have become a general tool for molecular biologists, as they provide a particularly accurate support for three-dimensional (3D) visualization. During this period, due to the evolution of our understanding of molecules, molecular visualization packages have evolved from pure imaging toward a sophisticated interactive tool.[1–4] Not only does the display provide a good infographic media, the user may modify the imaged data so that it becomes the most adequate to his understanding. Animating data relies on a dual concept: graphic animation of successive data sets, and real-time transformation of the data. Graphic animation ability is essentially used to visualize successive steps of a molecular dynamics simulation. On the other hand, most modeling procedures rely on transforming the coordinate set. These procedures lead to a bottleneck in the modeling work because homology is less and less restricted to some local modifications,[5–7] and in some cases leads to modifications of the whole backbone.[8] Though real-time transformations using a force field can be resolved for small molecules, this task seems impossible when there exist many degrees of freedom (as for large molecules) because the numerical calculation of the transformation cannot proceed interactively in a reasonable amount of time. Usually, packages allow such manipulation in two ways: direct manipulation through a series of single-axis rotations. This process is often tedious and long because each move breaks the global geometry of the molecule and requires many corrective operations. The second approach consists in linking structural fragments taken from a motif library.[9] However, due to the small number of known structures, this situation is far from allowing any desired conformation to be generated. The FORME package is designed to perform protein backbone deformations by focusing on a purely geometric approach. It is based on finding a combination of single rotations to perform a given transformation. Most of its procedures have been derived from analytic work on CPK models and are governed by the following rules:

(1) Only geometry guides the motion. As yet, no energetical or sterical considerations are taken into account in the procedures.
(2) The motion is supported by the main chain. Side chains are kept in their initial conformations relative to the local peptide chain (N–Cα–C′). Only rotations around N–Cα and Cα–C′ bonds are allowed (except for Proline, for which only rotations around the Cα–C′ bond are allowed).
(3) The transformation operator quickly organizes the motion at a global level, avoiding a loss of the molecule's geometry. It performs a given modification by selecting from a local subset of the backbone the bonds that will act and by calculating the associated values of the rotation angles.

Although this approach leads to some immediate restrictions concerning bond length or torsion angle flexibility, it is satisfactory to reach quickly an approximation of a state that is close to a given conformation and that may be further refined. The operator's experience may be sufficient to avoid sterical conflicts or impossible conformations. It is expected that if transformations are of good enough quality, many hypotheses can be tested with much less computational work than is required by molecular dynamics. This approach supposes that a user-friendly interface exists and that particular care has been taken to permit interactive control. FORME is designed to allow quasi-continuous local geometric modifications. For this purpose deformation operators, such as stretching or twisting of the backbone, have been implemented.

First, we present some general considerations about protein backbone deformation by analyzing the degrees of free-

Address reprint requests to Dr. Hazout at the Unité de Recherches Biomathématiques et Biostatistiques, INSERM U263, Université Paris 7, Tour 53, 2 place Jussieu, 75251 Paris cedex 05, France.
Received 23 October 1990; accepted 13 November 1990

dom necessary to perform a given transformation, such as stretching or twisting. Then two types of stretching operators are developed and described: stretching inside an "acceptance area" and stretching with end-block invariance. Finally, we discuss perspectives for the design of complex operators.

GENERAL CONSIDERATIONS

Deformation operators

Analysis of manipulations of the CPK model shows that only a restricted set of operators are needed to perform most of the desired deformations. Stretching and twisting can be considered "primitive" operators, with more complex motions combining these two (Figure 1). Some subcases can be defined from characteristics of the different areas of the molecule. For example, stretching the end of a strand introduces different conditions than stretching the middle of the strand. The act of stretching the middle of a strand must preserve the relative orientation of the two extremities; i.e., the two end-blocks perform a translational motion relative to each other (which will be called "end-block invariance").

These primitive operators can then be combined to create more complex operators.

Deformation regions

Before performing a protein distortion, the backbone is divided into different parts: active regions (denoted R_1 in Figure 2) where the rotation axes are located, passive regions that undergo the move and that are structurally invariant (denoted R_2). The end-block (R_0) above these regions is considered fixed. The other end-block (R_3) may be mobile

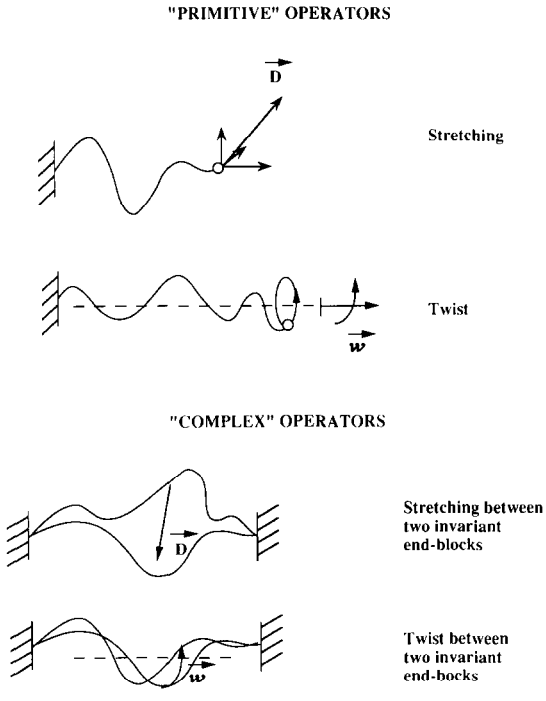

Figure 1. Modeling primitive operators. Complex operators can be derived by combining primitive operators

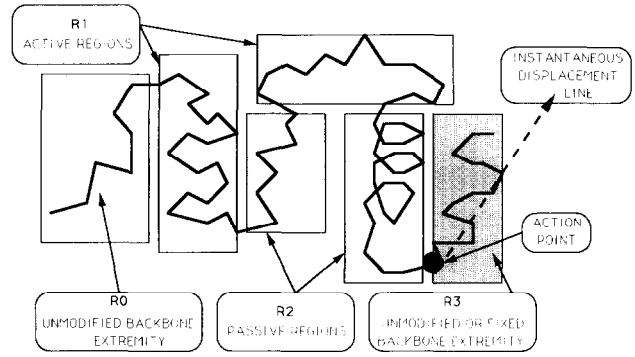

Figure 2. Segmentation of protein backbone for displacement

with or without constraint. (For example, it may be constrained to translations along an axis.) The active region may be composed of backbone fragments located in different parts of the molecule; for example, two loops at the extremities of the helix may constitute an active region to displace this structure without distortion.

Strategy for deformation

Two ways of tackling the problem exist that may lead to different efficiencies.

(1) Given a starting molecule conformation and the final location of a subset of points, the objective is to find a path (i.e., a series of transformations) between the two structures that satisfy the geometric constraints. This approach was not tried because of the following points: First, the solution requires the use of optimization methods and leads to a loss of interactivity because in general, the number of parameters is important. For example, for a protein of 200 residues one needs to control 400 parameters (ϕ and ψ angles) and 400 equality constraints for lengths and angles. Moreover, multiple solutions may satisfy the constraints. In that case, one may arrive at a solution that is not possible by continuity with the pre-existing coordinate set. This may create a problem, for example, in the case of motion within a loop if the side chains change their orientations with respect to the extremities. More generally, the problem is how to preserve the global topology of the molecule.

(2) Given a starting set of coordinates, the objective is to vary the set continuously to reach a given point in space. This approach seems more convenient for an interactive approach. First, the validity of the obtained solution is preserved as the transformations take into account the global structure of the protein. Second, a judicious choice of approximations allows the design of simple real-time algorithms. An exact resolution of a physical system of forces applied to an atom is not "realistic" because the multiple nonlinear equations involved will require optimization methods. The solution may be reached approximatively because the active regions provide, in general, sufficient degrees of freedom to perform the transformation and to facilitate the optimal choice of rotating axes. A simple

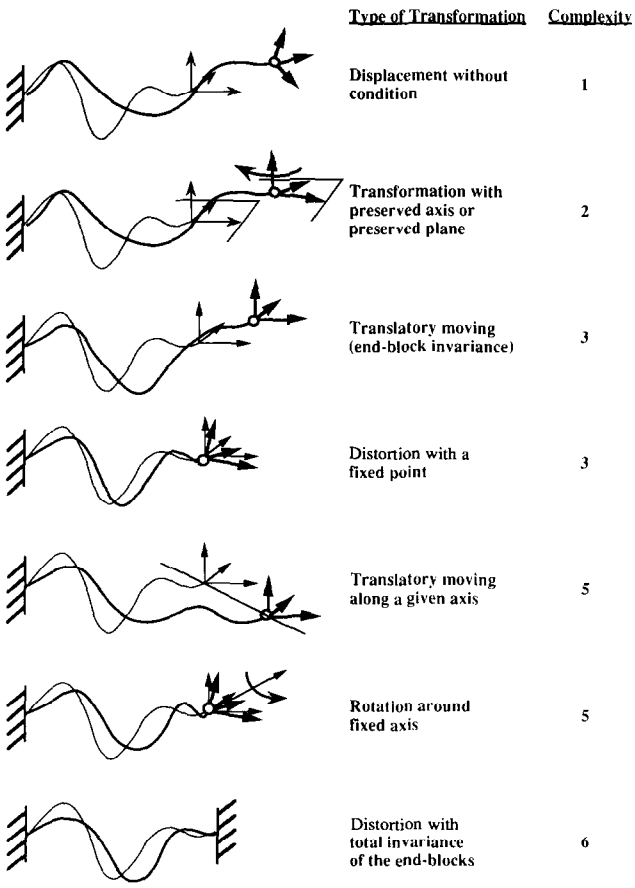

Figure 3. Analysis of degrees of freedom necessary per transformation. Complexity denotes the number of degrees of freedom necessary per motion

count of the theoretical number of degrees of freedom necessary for the different types of transformations is reported in Figure 3. As shown, the number of required degrees of freedom varies from one to six; it depends mainly on constraints that will be applied, particularly to the passive regions (R_2 and R_3). For example, simple stretching requires only one degree of freedom but stretching while maintaining one point along a trajectory requires at least two degrees of freedom, and moving along a trajectory while keeping a reference point invariant requires at least five degrees of freedom. This theoretical count can be reduced if some approximations are introduced. For example, a constraint on the trajectory may be solved by a one-degree motion if we accept the idea that only proximity of the trajectory is required. The move can then be constructed as a sum of simple moves that keep one point in a volume, for example, a cylinder surrounding the theoretical trajectory. In the same way, motion involving end-block invariance may be solved with only three degrees of freedom if an accurate choice of the rotation axes is made, so that the displacement remains close to the trajectory. In such a strategy, a crucial point is the choice of the rotating bonds so that final point is close enough to the trajectory. Consequently, we proceed by determining the rotation axes corresponding to the number of degrees of freedom and calculating the associated angles.

METHODOLOGY

Simplified representation of an operator

An operator consists of a combination of several rotations that obtain exactly or approximately the desired modification. We have defined a simplified representation to study the transformations easily. Figure 4a shows in 3D space how an arbitrary vector **a** is modified by two independent rotations of axis $\mathbf{u_0}$ and $\mathbf{u_1}$. (Note that these vectors are of modulus 1.) In 3D space, the extremity of **a** describes part of a circle from **a** to **i** in a plane P_0 orthogonal to $\mathbf{u_0}$, and then part of another circle from **i** to **b** in a plane P_1 orthogonal to $\mathbf{u_1}$. The simplified representation (Figure 4b) corresponds to drawing only the two circles of planes P_0 and P_1. It may be generalized to several rotations. For building complex operators from combination of rotations, one denotes the rotation vector by u_i (i = 1, 2, . . .) and the associated angle by θ_i. (For general information about 3D transformations, see references 10 and 11.)

Elementary operators

Coordinates (x, y, z) are expressed in a local reference defined by the data of two independent vectors $\mathbf{u_0}$ and $\mathbf{u_1}$ and are constructed as follows:

$$p_{01} = [(\mathbf{u_0} \times \mathbf{u_1}) \times \mathbf{u_0}, \mathbf{u_0} \times \mathbf{u_1}, \mathbf{u_0}]$$

($\times$ denotes a vector product). Using this convention, a reference defined by $\mathbf{u_0}$ and $\mathbf{u_1}$ is denoted p_{01}.

In this reference, elementary rotation is always about the z-axis.

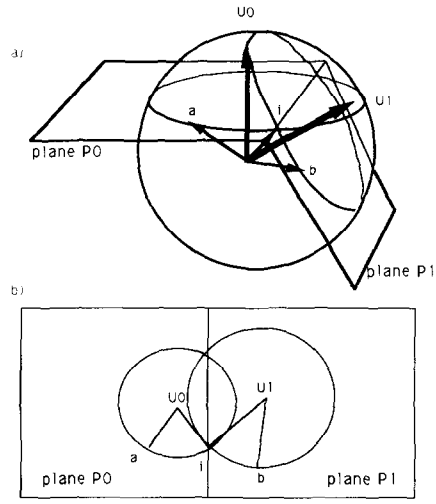

Figure 4. Simplified representation of a transformation defined by two rotations around u_0 and u_1: (a) the 3D trajectory of the extremity of direction a to give i and b by successive rotations around u_0 and u_1; and (b) simplified representation of this move

Elementary operators are used as follows:

(1) Rotation matrix for an angle θ about the z-axis ($\mathbf{u_0}$):

$$\mathbf{R}(\mathbf{u_0}, \theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(2) Matrix of change of reference p_{01} to reference p_{10} ("seesaw" motion):

$$\mathbf{B}(\mathbf{u_0}, \mathbf{u_1}) = \begin{pmatrix} -\cos\alpha_{01} & 0 & \sin\alpha_{01} \\ 0 & -1 & 0 \\ \sin\alpha_{01} & 0 & \cos\alpha_{01} \end{pmatrix}$$

where α_{01} is the angle between the vectors $\mathbf{u_0}$ and $\mathbf{u_1}$ oriented by the vector $(\mathbf{u_0} \times \mathbf{u_1})$. We can remark that $\mathbf{B}(\mathbf{u_0}, \mathbf{u_1}) = \mathbf{B}(\mathbf{u_1}, \mathbf{u_0})$.

(3) Matrix of change of reference p_{01} to reference p_{02}:

$$\mathbf{P}(\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_0}) = \begin{pmatrix} \cos\alpha_{102} & \sin\alpha_{102} & 0 \\ -\sin\alpha_{102} & \cos\alpha_{102} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This is equivalent to a rotation of $(-\alpha_{102})$ around $\mathbf{u_0}$, where α_{102} is the angle between $(\mathbf{u_0} \times \mathbf{u_1})$ and $(\mathbf{u_0} \times \mathbf{u_2})$ oriented by the vector $\mathbf{u_0}$.

Combinations of these operators allow for general changes of references.

Design of primitive operators: stretching

Stretching consists in pulling the protein's backbone out from a point (called the "action point") in a given instantaneous direction. Different types of displacement conditions can be introduced:

(1) pulling out in a direction without any displacement or structure control
(2) displacement of the control point along a curve (a straight line in a simple case) without structure control
(3) displacement of the control point along a curve, maintaining end-block invariance

The following argument is made by considering spatial directions and does not take atomic locations into account.

The first case may be solved by a single rotation about some axis. This case is not of much interest because of the absence of controls. However, this transformation, using one degree of freedom, may be used to solve the second case. A solution may be obtained to correctly follow a curve by defining an "acceptance area," i.e., a spatial neighborhood of the curve in which the point is constrained to move. In general, single rotations do not yield a good fit to the curve, but this fit may be approximated through successive rotations.

The third case requires additional degrees of freedom. The theoretical solution may involve two degrees of freedom for motion on a trajectory, and three degrees of freedom to restore invariance. We have found that this may be approximated by a system of three rotations. One degree of freedom is responsible for motion, two degrees to preserve two directions. (A reference consists of three orthogonal vectors, but knowing two of them allows one to construct the third.)

Acceptance areas

The acceptance area is identified as a cylinder of given radius whose axis is along the instantaneous direction **D** defined by the user. A nonlinear trajectory can be approximated by a series of straight segments. The objective is to keep the action point inside the acceptance area. One axis is chosen in active regions such that it satisfies the following condition: The best axis corresponds to a maximal value of the determinant of the matrix composed of the three following vectors: $\mathbf{u_0}$, the unitary vector representing the rotation axis; **OA**, a vector linking the origin point O of $\mathbf{u_0}$ with the action point A; and **D**, the displacement direction. (See Figure 5.) This condition allows one to reintroduce the action point A in the acceptance area and to progress in the direction **D**. To reestablish the trajectory at each rotation, one changes the displacement vector **D** by a linear combination as follows:

$$\mathbf{D_{new}} = \alpha\mathbf{D} + (1 - \alpha)\,\mathbf{h} \qquad (1)$$

where **h** is the vector orthogonal to the absolute direction and defined by the two points A (the action) and H (the projection of the action point on the cylinder axis), and α is a function of the modulus of **h** (i.e., $|\mathbf{h}|$), which varies between 0 and 1. The chosen function is:

$$\alpha = exp(-|\mathbf{h}| \cdot k) \qquad (2)$$

with k an arbitrary constant. The larger $|\mathbf{h}|$ is, the more important it is to reestablish the trajectory instead of advancing the action point in the direction **D**.

End-block invariance

Approximation by three rotations We assume here that the displaced protein backbone corresponding to an unmodified region is defined by a reference of three arbitrary orthogonal vectors ($\mathbf{u_0}$, $\mathbf{v_0}$, $\mathbf{w_0}$). (Only the first two vectors are necessary.) To stretch the backbone into the direction **D**, we must choose three axes of rotation ($\mathbf{u_0}$, $\mathbf{u_1}$, $\mathbf{u_2}$). The first axis is identical to one of the reference vectors. Also, after rotations through angles θ_0, θ_1 and θ_2, the reference

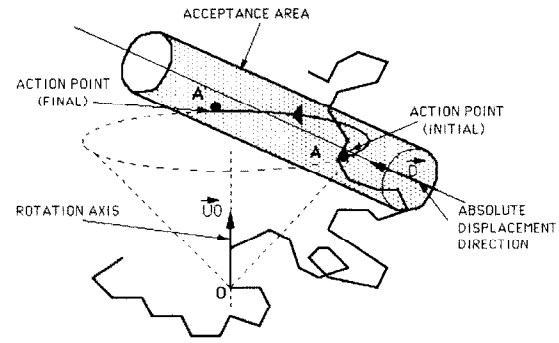

Figure 5. Trajectory approximation of a displacement. Direction D represents the absolute displacement direction approximated by the motion. The acceptance area is shown as a cylinder surrounding a line supporting **D**. *Rotation around u_0 of the action point A leads to A' inside the acceptance area*

axes again become identical. Figures 6a and 6b show respectively the intermediate states of the vectors $\mathbf{u_0}$ and $\mathbf{v_0}$ after the three rotations (two vectors defining the reference, the third deduced from them). One defines these vectors in the different reference directions by using the elementary operators. Then, using the invariance of $\mathbf{u_0}$ and $\mathbf{v_0}$, one obtains the following equations:

$$\mathbf{u_0} = \mathbf{B}(\mathbf{u_1}; \mathbf{u_0}) \cdot \mathbf{R}(\mathbf{u_1}, \theta_1/2) \cdot \mathbf{B}(\mathbf{u_2}, \mathbf{u_1}) \cdot \mathbf{R}(\mathbf{u_2}, \theta_2) \cdot \mathbf{B}(\mathbf{u_1}, \mathbf{u_2}) \cdot \mathbf{R}(\mathbf{u_1}, \theta_1/2) \cdot \mathbf{B}(\mathbf{u_0}, \mathbf{u_1}) \cdot \mathbf{u_0} \quad (3)$$

$$\mathbf{v_0} = \mathbf{B}(\mathbf{u_1}; \mathbf{u_0}) \cdot \mathbf{R}(\mathbf{u_1}, \theta_1/2) \cdot \mathbf{B}(\mathbf{u_2}, \mathbf{u_1}) \cdot \mathbf{R}(\mathbf{u_2}, \theta_2) \cdot \mathbf{B}(\mathbf{u_1}, \mathbf{u_2}) \cdot \mathbf{R}(\mathbf{u_1}, \theta_1/2) \cdot \mathbf{B}(\mathbf{u_0}, \mathbf{u_1}) \cdot \mathbf{R}(\mathbf{u_0}, \theta_0) \cdot \mathbf{v_0} \quad (4)$$

By developing these expressions, one obtains two equations between θ_0, θ_1 and θ_2:

$$tan(\theta_0/2) = sin(\alpha_{12})\, sin(\theta_1/2)/(sin(\alpha_{01})cos(\alpha_{12}) - sin(\alpha_{12})cos(\alpha_{01})cos(\theta_1/2)) \quad (5)$$

$$tan(\theta_2/2) = sin(\alpha_{01})\, sin(\theta_1/2)/(sin(\alpha_{12})cos(\alpha_{01}) - sin(\alpha_{01})cos(\alpha_{12})cos(\theta_1/2)) \quad (6)$$

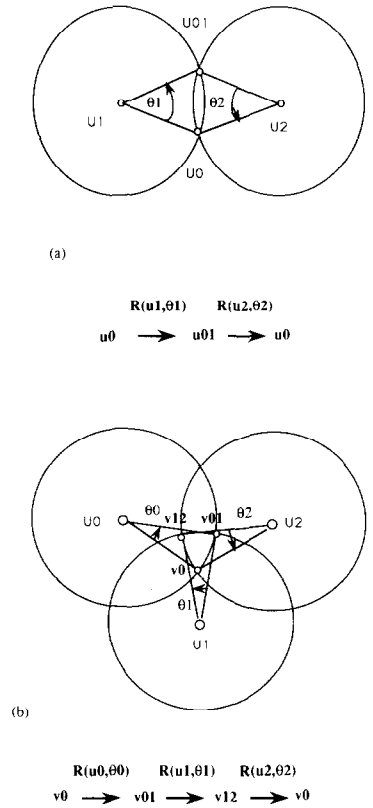

Figure 6. Simplified representations of a system to solve the translational motion of a reference by three rotations around u_0, u_1 and u_2. The first and second axes of reference are u_0 and v_0, respectively: (a) simplified representation of a two-rotation system preserving direction u_0 by successive rotations around u_1 and u_2; and (b) simplified representation of a three-rotation system preserving direction v_0 by successive rotations around u_0, u_1 and u_2

where α_{01}, α_{02} and α_{12} correspond to the angles between the two vectors $(\mathbf{u_2} \times \mathbf{u_0})$ and $(\mathbf{u_2} \times \mathbf{u_1})$, oriented by $\mathbf{u_2}$; between $(\mathbf{u_1} \times \mathbf{u_0})$ and $(\mathbf{u_1} \times \mathbf{u_2})$, oriented by $\mathbf{u_1}$; and between $(\mathbf{u_0} \times \mathbf{u_1})$ and $(\mathbf{u_0} \times \mathbf{u_2})$, oriented by $\mathbf{u_0}$, respectively. One notices the symmetry between Equations (3) and (4): θ_0, α_{01} and α_{12} are changed by θ_2, α_{12} and α_{01}, respectively. Independent of this demonstration, one can deduce from the construction of the representation of Figure 6a that $\theta_1/2$ is equal to the angle between $(\mathbf{u_1} \times \mathbf{u_0})$ and $(\mathbf{u_1} \times \mathbf{u_2})$ and that $\theta_2/2$ is equal to the angle between $(\mathbf{u2} \times \mathbf{u_1})$ and $(\mathbf{u_2} \times \mathbf{u_0})$. Consequently the data of the three vectors $(\mathbf{u_0}, \mathbf{u_1}, \mathbf{u_2})$ leads to a unique displacement of the backbone. The strategy of this stretching relies on the unique choice of rotation axes among the different bonds of the active regions. (Further details are given in the next section.)

Approximation by four rotations In some cases a degree of freedom for the motion must exist. Considering the preceding strategy, the operator can be designed simply as following: a three-rotation motion around $\mathbf{u_1}$, $\mathbf{u_2}$ and $\mathbf{u_3}$ maintains the invariant vector $\mathbf{u_0}$. (A similar procedure was used above to preserve the invariance of $\mathbf{u_0}$ in the three-rotation case.) This is equivalent to a rotation about the vector $\mathbf{u_0}$. Following rotation around $\mathbf{u_0}$ maintains the invariance of the reference.

Choice of rotating bonds

The choice of rotating axes must satisfy two conditions:

(1) Efficiency of all the rotations relative to the motion. For a one-rotation motion, the rotating bond is simply chosen as the maximum momentum axis. For a three- or four-rotation motion, concordance between the directions of rotation required by the geometry of the system and efficiency must be tested. (For example, all rotations in Figure 6b have the same orientation.)

(2) Verification of the constraints on the motion. Because each step of the motion has to be small, rotation values have to be close to zero. This is most strongly required for axes that have great momentum relative to the displacement direction. It can be seen graphically (Figure 6a) that small rotation values for directions $\mathbf{u_1}$ and $\mathbf{u_2}$ are obtained for quasi coplanarity of the directions $\mathbf{u_0}$, $\mathbf{u_1}$ and $\mathbf{u_2}$ and large values for the angles α_{01} and α_{12}. Rotation values for $\mathbf{u_0}$ can hardly be close to zero with such a system, but this problem can be overcome by choosing $\mathbf{u_0}$ as the axis closest to the displacement direction. By performing approximations of trigonometric functions for small values of angles (θ_0, θ_1 and θ_2) in Equations (4) and (5), one obtains:

$$\theta_0/2 = (\theta_1/2)\, sin(\alpha_{12})/sin(\alpha_{01} - \alpha_{12}) \quad (7)$$

$$\theta_2/2 = (\theta_1/2)\, sin(\alpha_{01})/sin(\alpha_{12} - \alpha_{01}) \quad (8)$$

A small value for θ_2 is obtained when $(\alpha_{12} - \alpha_{01})$ is as close to $\pi/2$ as possible. Previous conditions on angles α_{12} and α_{01} show that one possible system is for α_{01} and α_{02} to be close to $\pi/4$, with all axes quasi-coplanar.

Extension to a four-rotation motion leads to tetrahedral construction for $\mathbf{u_1}$, $\mathbf{u_2}$ and $\mathbf{u_3}$, similar to Figure 6b, where $\mathbf{u_0}$ replaces $\mathbf{v_0}$.

Control on the move

The strategy of deformation allows two types of control:

(1) Control of the trajectory can be achieved by

- choosing an appropriate value of k in Equation (2) to control the radius of the cylinder surrounding the displacement direction
- limiting the maximum rotation values for the degrees of freedom, to keep the action point close to the trajectory
- possibly redefining the instantaneous displacement direction at each step.

(2) Control on the speed is achieved by limiting the maximum rotation values.

Implementation

User interface

FORME is organized as shown in the flowchart (Figure 7). Access to the deformation algorithm is performed via a menu system. Fundamental and independent modules for the deformations are:

(1) Organizing the deformation

- preselecting the type of motion. For simplicity only stretching, twisting and complex types of motions have been implemented.
- defining the active regions. This is achieved by selecting the atoms at the beginnings and the ends of active regions. For convenience it is possible to add or delete regions, as well as to save and label them for different work sessions.
- defining the action point, where a force is supposed to be applied. In the case of twisting, nonatomic locations can be defined, so that the twist axis may be independent of the coordinates of the atoms defining the active regions.
- defining the force direction via the mouse or the dial-box, at any time.

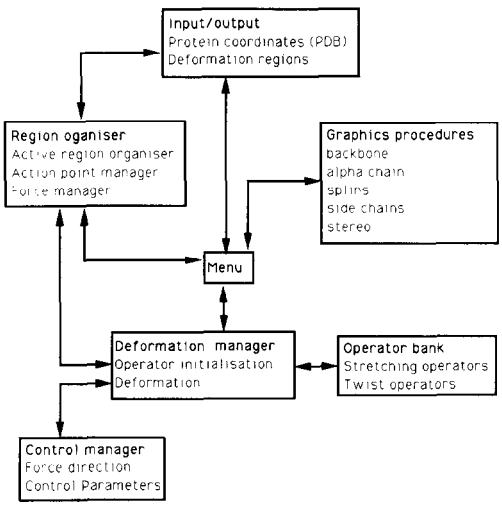

Figure 7. Flowchart of FORME

(2) Performing the deformation

- The correct initialization of active regions and action point allows FORME to automatically select the appropriate operator that corresponds to the type of operation desired by the user. For complex operations (like stretching with complete end-block invariance), primitive operators are automatically assigned their sub-deforming zones. Interaction with the deformation is then handed via the dials. At any time during the motion, dials allow one to control the stretching direction or twist axis, by defining their x-, y- and z-values. This allows the operator to interactively define the trajectory. A special dial controls the "speed" of the deformation. It is assigned to either the maximum value of the main rotation, or to the radius of the cylinder approximating the trajectory. (A small radius involves small motions.) At each step, the rotating axes are highlighted.
- An "undo" system has been implemented to return to an intermediate protein location.

Hardware and programming considerations

FORME has been written in the C programming language on a Stardent 1500 workstation, and visualization is done using the DORE graphics library. A stereo system based on separation of two images rotated through $+3°$ and $-3°$ has been implemented. Input and output of Protein Data Bank files is possible.

RESULTS AND DISCUSSION

Design of the operators

The choice of a design based mainly on the reestablishment of directions rather than on operators that compute the exact positions of the final state has proven efficient. In fact the combination of low speed for the transformation (through simple reduction of the rotations allowed at each step) and the direct control of the direction of the motion seems sufficient to reach a location very close to what was desired. Some cases where exact atomic positions are required can be solved by an iterative process until satisfactory proximity is achieved.

At the moment, our work has mainly been oriented toward designing efficient primitive operators that allow FORME to perform the following types of motion:

- simple stretching
- stretching with end-block invariance (See Color Plate 2.)
- simple twisting (See Color Plate 3.)

Our current work is to improve the efficiency of:

- stretching restricted to internal motion (for example, a local deformation of a coil in which no atoms are moved outside). This operator is designed as a combination of two stretchings with end-block invariance whose effects are added so that the coordinates of the extremity of the active region remain constant.
- twisting restricted to internal motion.

Further developments of FORME will include the analysis of larger scale motions, such as sheet twisting or helix rotation, as combinations of the primitive operators.

Efficiency

Our main goal was to make a user-friendly and efficient tool. Tests on proteins of various sizes show no major loss of user interaction for larger sizes. Pertinence may be measured through a real modeling work. We report here study cases of α_1-antitrypsin. (See Color Plates.) A more elaborate model has been described.[8] The goal was to connect the two chains that are split in the crystal form. Our option was to add a strand A7 to the *A*-sheet. (See Engh et al.[8] for conventions.) This required that we correctly position the two extremities of strand C1 and strand A4. As this simple operation leads to contact between the two strands of the small chain, it was necessary to displace the C-terminal strand, which made it necessary to move the *A*-helix. In constructing the A7 strand, care was taken to move the side chains into sterically possible positions, and the manual alignment of the new strand was done as was best possible.

Guides to the motion

At the moment, we have chosen not to implement (steric or energetic) controls to analyze the possibility of a motion. This was done because it seems possible that such automatic controls may decrease the number of possible motions by forbidding "unacceptable" motions, such as crossing strands, that allow one to reach easily a satisfactory conformation. This would require many local deformations in response to local steric contacts that are not necessarily wanted in the final conformation; in the end, local steric contacts are removed by further refinement of the final state. However, operators make it possible to proceed as follows for large steric contacts: A warning along the chain indicates the importance of the steric contacts. On request, contacts more than a given threshold are automatically removed, operators being automatically assigned.

CONCLUSION

FORME concepts provide an efficient approach to the problem of chain deformations. Simple operators constructed from combinations of rotations allow the user to perform primitive motions, such as stretching or twisting in real time on standard workstations. We intend to analyze more complex motions composed of these primitive operators. Because the operators work on any system consisting of a set of sticks and balls, generalization to other macromolecular systems may be tackled.

REFERENCES

1 Jones, T.A. FRODO: a graphics model building and refinement system for macromolecules. *J. Appl. Crystallogr.* 1978, **11**, 268–272

2 Cherfils, J., Vaney, M.C., Morize, I., Surcouf, E., Colloc'h, N. and Mornon, J.P. MANOSK: A graphics program for analysing and modeling molecular structures and functions. *J. Mol. Graphics* 1988, **6**, 155–160

3 Shomburg, D. and Reichett, J., BRAGI: A comprehensive protein modeling program system. *J. Mol. Graphics* 1988, **6**, 161–165

4 Ferrin, T.E., Huang, C.C., Jarvis, L.E. and Langridge, R. The MIDAS display system. *J. Mol. Graphics* 1988, **6**, 13–27

5 Rath, V.J., Newgard, C.B., Sprang, S.R., Goldsmith, E.J. and Fletterick, R.J. Modeling the biochemical differences between rabbit muscle and human liver phosphorilase. *Proteins* 1987, **2**, 225–235

6 Giranda, V.L., Chapman, M.S. and Rossman, M.G. Modeling of the human intercellular adhesion molecule-1, the human rhinovirus major group receptor. *Proteins* 1990, **7**, 227–233

7 Madison, E.L., Goldsmith, E.J., Gerard, R.D., Gething, M.-J.H. and Sambrook, J.F. Serpin-resistant mutants of human tissue-type plasminogen activator. *Nature* 1989, **339**, 721–724

8 Engh, R.A., Wright, H.T. and Huber, R. Modeling the intact form of the α1-proteinase inhibitor. *Prot. Eng.* 1990, **6**, 469–477

9 Shomburg, D. and Reichelt, J. BRAGI: A comprehensive protein modeling program system. *J. Mol. Graphics* 1988, **6**, 161–165

10 Foley, J.D. and Van Dam, A. *Fundamentals of Interactive Computer Graphics*. Addison–Wesley, 1982

11 Newman, W.M. and Sproul, R.F. *Principles of Interactive Computer Graphics* McGraw–Hill, 1981