Conformational Sampling by a General Linearized Embedding Algorithm

Gordon M. Crippen,* Andrew S. Smellie,† and Wendy W. Richardson

College of Pharmacy, University of Michigan, Ann Arbor, Michigan 48109

Received 16 March 1992; accepted 25 June 1992

Linearized embedding is a variant on the usual distance geometry methods for finding atomic Cartesian coordinates given constraints on interatomic distances. Instead of dealing primarily with the matrix of interatomic distances, linearized embedding concentrates on properties of the metric matrix, the matrix of inner products between pairs of vectors defining local coordinate systems within the molecule. We developed a pair of general computer programs that first convert a given arbitrary conformation of any covalent molecule from atomic Cartesian coordinates representation to internal local coordinate systems enforcing rigid valence geometry and then generate a random sampling of conformers in terms of atomic Cartesian coordinates that satisfy the rigid local geometry and a given list of interatomic distance constraints. We studied the sampling properties of this linearized embedding algorithm vs. a standard metric matrix embedding program, DGEOM, on cyclohexane, cycloheptane, and a cyclic pentapeptide. Linearized embedding always produces exactly correct bond lengths, bond angles, planarities, and chiralities; it runs at least two times faster per structure generated, and is successful as much as four times as often at refining these structures to full agreement with the constraints. It samples the full range of allowed conformations broadly, although not perfectly uniformly. Because local geometry is rigid, linearized embedding's sampling in terms of torsion angles is more restricted than that of DGEOM, but it finds in some instances conformations missed by DGEOM. © 1992 by John Wiley & Sons, Inc.

INTRODUCTION

A commonly occurring problem in chemistry is to calculate a molecule's conformation or conformations, if any, that satisfy a given set of geometric constraints. This most often arises in the determination of the conformation of small proteins in solution by nuclear magnetic resonance (NMR), where the constraints are restrictions on some dihedral angles from coupling constant measurements, upper bounds on the distances between a few specified pairs of atoms from nuclear Overhauser effect (NOE) measurements, plus all the *a priori* bond lengths, bond angles, van der Waals radii of atoms, and the planarity or chirality of various groups. The obvious way to solve this is by local minimization of a **penalty function,** which consists of a sum of terms, one for each constraint, such that a term is zero if the constraint is satisfied or monotonically increasingly positive as the violation of the constraint increases. The difficulty is that the optimization starts from an arbitrarily chosen conformation and proceeds nearly always to a local minimum where the penalty function is greater than zero. That is, no small perturbation of the conformation can improve the remaining violation of constraints. Historically, the first practical approach to the problem was the distance geometry EMBED algorithm,[1] which uses the constraints to produce better starting conformations, in the sense that the subsequent minimization of the penalty function more often succeeds in reaching zero, i.e., complete agreement with the constraints. Another way to avoid the attrition problem is to vary the penalty function during the course of minimization.[2] The other popular approach is to begin with a conformation typically calculated by EMBED and seek a low-energy conformation still satisfying the constraints by using simulated annealing and molecular dynamics with a potential function that is the sum of an empirical intramolecular energy function and the penalty function.[3] In any case, these procedures produce, more or less efficiently, a collection of conformers, each obeying the original constraints more or less accurately, or else some indication that the constraints are mutually contradictory. The set of output structures is supposed to be a more or less thorough and more or less representative random sample of the theoretically allowed conformation space. The sampling properties of these methods has been particularly hotly debated.

*To whom all correspondence should be addressed.

†Current address: BioCAD Corp., 1390 Shorebird Way, Mountain View, CA 94043.

CCC 0192-8651/92/101262-13

To explain our new linearized embedding procedure, we must first briefly summarize the EMBED algorithm.[1] First, almost all the **local constraints** are expressed as upper and lower bounds on the distances between specified pairs of atoms: bond lengths, bond angles, rigidity of some ring systems, and torsion angle constraints that do not involve the sign of the dihedral angle. Other local constraints are included only later as terms in the penalty function: signed torsion angle constraints, chirality of asymmetric centers, and planarity of some ring systems. **Explicit constraints,** such as the results of NMR experiments, are similarly expressed as upper and/or lower bounds on certain interatomic distances. Van der Waals contact distances otherwise provide lower bounds for the remaining interatomic distances, but these do little to restrict the overall conformation. At this point, most interatomic distances have only a weak lower bound and essentially no upper bound at all, in the sense that these are far from being the greatest lower bounds and the least upper bounds consistent with the given constraints. However, the effects of the few strong constraints can be propagated in various ways to all distances, a process termed **bound smoothing.** All this preliminary work, especially the bound smoothing, can be rather time consuming but is done only once for a given molecule and its explicit constraints. Then, for each random structure to be generated, the algorithm simply chooses an $n_a \times n_a$ matrix of random interatomic **trial distances,** each within its corresponding lower and upper bounds, where there are n_a atoms in the molecule. This matrix is in turn converted to its corresponding $n_a \times n_a$ **trial metric matrix.** The trial metric matrix corresponds in general to some conformation in $\mathbb{R}^n$ (or in no space at all, due to an incorrect choice of trial distances), but the nearest rank 3 metric matrix, built up out of the three largest positive eigenvalues and corresponding eigenvectors of the trial metric matrix, can be directly converted to a set of atomic **trial coordinates** in $\mathbb{R}^3$. The trial coordinates have the correct dimensionality, but they no longer obey the original geometric constraints, in general. **Refined coordinates** are found by local numerical minimization of the constraint violations as a function of atomic coordinates starting from the trial coordinates. Many variations on EMBED have been programmed,[4,5] but the version used here for comparison purposes is the DGEOM code, as furnished by QCPE.[6] This was based upon early programs due to Crippen and coworkers but greatly modified primarily by Jeffrey Blaney and also Andrew Dearing and J. Scott Dixon.[7–9]

In this study, we further develop the linearized embedding approach.[10] Instead of concentrating on atomic coordinates and interatomic distances, atoms are positioned indirectly once a set of local coordinate systems are determined, one system for each group of atoms having fixed positions with respect to each other under the assumption of rigid valence geometry. Thus, all the local constraints are built into the initial **linearized representation** of the molecule, as described below. Each conformation to be generated is represented primarily by an $n_u \times n_u$ metric matrix, giving the relative orientations of the n_u local coordinate system axis unit vectors. Then, as in EMBED, the trial metric matrix is converted to unit vector trial coordinates in $\mathbb{R}^3$ and a penalty function is minimized, but with respect to dihedral angles rather than atom coordinates. Sampling the allowed conformation space is still done in terms of choosing many different random trial metric matrices and producing refined atomic Cartesian coordinates from each. In what follows, we describe major changes to the linearized embedding algorithm and compare its sampling with results from DGEOM.

METHODS

Linearized Representation

Depending upon the approximations one wants to make, a molecule can be idealized in many different ways, such as a continuously varying electron density function, a list of atomic coordinates, or an abstract graph with atoms for nodes and bonds for edges. Here, we assume the atoms are points in $\mathbb{R}^3$, all the atoms of a molecule are connected by covalent bonds, bond lengths and vicinal bond angles are fixed, and the only variation in conformation comes from rotating about single bonds. (Technically, we also assume that all rings are rigid, so that the calculation of the pseudorotation of cyclohexane is done by putting extra ring closure constraints on the acyclic hexane molecule.) Let a set of atoms with fixed relative positions be called a **rigid group.** Then, one can think of the molecule as a set of n_r rigid groups linked together by $n_r - 1$ rotatable bonds. Because all cycles are assumed rigid, the rigid groups cannot be connected together in any sort of loop, so they form a tree graph. Choose one relatively central rigid group as the root of the tree, called here the first rigid group, and number the rest 2, ..., n_r in breadth-first order. The atoms on either end of a rotatable bond can be considered members of either rigid group, and we choose to put both in the rigid group higher in the tree. Figure 1 shows the example of biphenylmethane, where the methyl is the first rigid group at the root of the molecular tree, and it includes two hydrogens, the central carbon A, and the carbon atoms B and C. The other two rigid groups are the remainders of the two phenyl rings. Here, $n_r = 3$ and the two rotatable bonds are AB and AC. We will refer to B as the **root atom** for the second rigid group, A as its **parent atom,** and AB as its

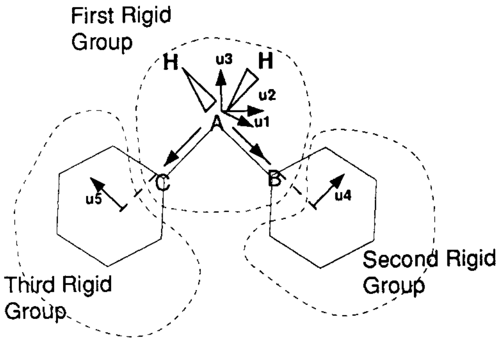

Figure 1. Molecule showing assignment of rigid groups. The two bonds, from A to B and from A to C, are rotatable torsions. The atom labeled B is the root atom for the second rigid group and the vector from A to B is the bond vector for the second rigid group. Similarly, the atom labeled C is the root for the third rigid group and the vector from A to C is its bond vector.

defining bond. For the third rigid group, these are C, A, and AC, respectively. In general, the defining bond is the rotatable bond linking a rigid group to its parent rigid group, the root atom is on the end of the defining bond nearer the rigid group, and the parent atom is on the other end.

The central feature of the linearized representation of a molecule is that each rigid group has a local coordinate system that moves with it as the molecule changes conformation. In Figure 1, the nonplanar first rigid group requires a full three-dimensional coordinate system with coordinate axes, called **unit vectors,** labeled $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$. Subsequent rigid groups use the corresponding defining bond vector as one axis, and then one or two additional unit vectors, depending upon whether the group is planar or nonplanar, respectively. For every atom, there is a linear combination of unit vectors and α coefficients that describes its position.

$$\mathbf{p}_i^1 = \mathbf{w} + \sum_{j=1}^{n_u^1} \alpha_{ij}^1 \mathbf{u}_j^1$$
$$\mathbf{p}_i^r = \mathbf{p}_{\text{root}}^r + \alpha_{iv}^r \mathbf{v}^r + \sum_{j=1}^{n_u^r} \alpha_{ij}^r \mathbf{u}_j^r \qquad (1)$$

where $\mathbf{p}_i^r$ is the position vector of the ith atom in the rth rigid group, $\mathbf{w}$ is the translation vector from the fixed external frame of reference to the centroid of the first rigid group, $\mathbf{u}_j^r$ is the jth unit vector in the rth rigid group, n_u^r is the number of unit vectors that define the rth rigid group ($n_u^1 = 1, 2,$ or 3; subsequent $n_u^r = 0, 1,$ or 2), $\mathbf{v}^r$ is the normalized defining bond vector for the rth rigid group, and α_{ij}^r is the coefficient associated with the ith atom and jth unit vector (or the bond vector $\mathbf{v}^r$) in the rth rigid group. In terms of atom positions already defined by unit vectors and αs of previous rigid groups,

$$\mathbf{v}^r = \frac{\mathbf{p}_{\text{root}}^r - \mathbf{p}_{\text{parent}}^r}{\|\mathbf{p}_{\text{root}}^r - \mathbf{p}_{\text{parent}}^r\|} \qquad (2)$$

The converse is also true. Every unit vector can be described by a linear combination of atom positions.

$$\mathbf{u}_j^1 = \sum_{i=1}^{n_a^1} \beta_{ij}^1(\mathbf{p}_i^1 - \mathbf{w})$$
$$\mathbf{u}_j^r = \sum_{i=1}^{n_a^r} \beta_{ij}^r(\mathbf{p}_i^r - \mathbf{p}_{\text{root}}^r - \alpha_{iv}^r \mathbf{v}^r) \qquad (3)$$

where β_{ij}^r is the coefficient associated with the ith atom and jth unit vector in the rth rigid group, and n_a^r is the number of atoms in the rth rigid group. As the molecule changes conformation, the unit vectors change direction, but the αs and βs remain constant.

The LINIZE program finds a linearized representation for a molecule. Its input is the atomic coordinates of the molecule, which element each atom is, and the simple, unlabeled connectivity between pairs of atoms. Its output is the organization of the molecule into rigid groups, the αs and βs for all possible conformations, as well as the particular coordinates for the unit vectors corresponding to the given conformation.

LINIZE begins with the given connectivity, atom coordinates, and a unique label for each atom that starts with its atomic symbol. A series of depth-first searches are made to determine the most central atom in the molecular graph and identify those atoms involved in cycles. Bond types are determined for each bond in the molecule using the connectivity data, each atom's chemical identity, and simple rules of chemistry.

Next, the program must break the molecule up into a collection of rigid groups, as defined above. A single bond that is not part of a ring and does not link a terminal atom is viewed as a rotatable bond and must connect two rigid groups. To minimize the propagation of errors associated with the linearization process, the first rigid group is the one containing the central atom. Subsequent rigid groups are numbered in order of a breadth-first traversal of the tree of rigid groups. Each subsequent rigid group has not only a collection of atoms belonging directly to it but also references to its root atom, parent atom, and defining bond, all of which technically are part of the parent rigid group in the tree.

Once the rigid group organization is complete, we calculate unit vector coordinates and αs for each rigid group from the atomic coordinates. The centroid for the first rigid group, $\mathbf{w}$, amounts to the overall translation vector of the molecule from the origin of the atomic coordinates external reference frame. Subtracting this from the coordinates of each atom gives **reduced coordinates,** $\mathbf{c}_i^r$, that are local to the rigid group.

$$\mathbf{w} = \frac{1}{n_a^1} \sum_{i=1}^{n_a^1} \mathbf{p}_i^1$$
$$\mathbf{c}_i^1 = \mathbf{p}_i^1 - \mathbf{w} \qquad (4)$$

For the subsequent rigid groups, reduced coordinates are calculated by subtracting the root atom's coordinates and the component of the (normalized) bond vector:

$$\mathbf{c}_i^r = \mathbf{p}_i^r - \mathbf{p}_{\text{root}}^r - \alpha_{iv}^r \mathbf{v}^r \tag{5}$$

where

$$\alpha_{iv}^r = (\mathbf{p}_i^r - \mathbf{p}_{\text{root}}^r) \cdot \mathbf{v}^r \tag{6}$$

The reduced coordinates are used to calculate the **inertial tensor** for each rigid group:

$$T^r = \sum_{i=1}^{n_a^r} \mathbf{c}_i^r (\mathbf{c}_i^r)^T \tag{7}$$

where $(.)^{\text{T}}$ denotes vector transpose. The Jacobi method[11] is used to determine all the eigenvalues and (normalized) eigenvectors for T^r. There may be from one to three nonzero eigenvalues for the first rigid group. Subsequent rigid groups may have zero to two nonzero eigenvalues. The number of nonzero eigenvalues defines the dimension of space spanned by the reduced coordinates of the atoms of the rigid group, and the corresponding eigenvectors form a basis set for the space, denoted as the unit vectors $\mathbf{u}_j^r$ in eq. (1). By construction, the eigenvectors are always orthogonal to the bond vector, and adding the latter to the basis gives an expanded basis for the original coordinates of the atoms in the rigid group. Then, the αs are simply the atom coordinates in the rigid group's local basis, and are calculated by

$$\begin{aligned} \alpha_{ij}^1 &= \mathbf{c}_i^1 \cdot \mathbf{u}_j^1 \\ \alpha_{ij}^r &= (\mathbf{p}_i^r - \mathbf{p}_{\text{root}}^r) \cdot \mathbf{u}_j^r \end{aligned} \tag{8}$$

and by eq. (6).

These unit vectors and αs are accurate for rigid groups that fully span three dimensions, but for those rigid groups that do not, these trial values may lead to some large errors. The worst case occurs when almost planar or almost linear groups are treated. Suppose LINIZE begins with the X-ray crystal structure of a molecule involving a chain of aromatic ring systems linked by single bonds. Because of small experimental errors and numerical imprecision, a few atoms of the first rigid group may end up just barely out of the plane. LINIZE would give a basis set of two vectors, neglecting the third eigenvector because of its very small eigenvalue. Use of this basis set results in small errors in the positions of some atoms, which may be in turn the parent and root atoms for a subsequent nearly planar rigid group, etc. If these errors are allowed to propagate down a long chain, the errors at the end could be substantial.

To avoid this type of problem, we refine the unit vector coordinates and αs for each rigid group, independently of the others, starting with the first rigid group and proceeding in the usual breadth-first ordering. In each case, we minimize a penalty function that is just the total squared deviation between the calculated atom positions according to eq. (1) and the original atom positions.

$$E^r = \sum_{i=1}^{n_a^r} (\mathbf{p}_{i,\text{calc}}^r - \mathbf{p}_{i,\text{original}}^r)^2 \tag{9}$$

For the first rigid group, no adjustment is required if it is clearly nonplanar, but if it is planar we perform a least-squares fit to find the plane that best describes the positions of all atoms in the rigid group. Two orthonormal vectors that describe the plane are chosen as the fixed unit vectors, and E^1 is minimized with respect to the α_{ij}^1s. For subsequent rigid groups, an orthonormal basis set is defined consisting of always the normalized bond vector, which is now fixed in the refined parent rigid group, and two basis vectors calculated by orthogonalization of the approximate unit vectors and the bond vector. For a nonplanar rigid group, such as a methylene, the two refined unit vectors are calculated by

$$\begin{aligned} \mathbf{u}_1^r &= \mathbf{b}_2^r \sin\theta + \mathbf{b}_3^r \cos\theta \\ \mathbf{u}_2^r &= \mathbf{b}_2^r \cos\theta + \mathbf{b}_3^r \sin\theta \end{aligned} \tag{10}$$

in terms of the two basis vectors $\mathbf{b}_2^r$ and $\mathbf{b}_3^r$ as a function of θ, the angle of rotation about the bond vector. (For a planar group, we use only the first of these two equations.) This transformation guarantees that the optimal unit vectors will always be orthogonal to the bond vector and to each other. In this situation, the variables involved in the minimization of E^r are the αs associated with both the unit vectors and the bond vector and θ. For minimizations, we used the conjugate gradient algorithm by Shanno and Phua.[12] Although this protocol treats rigid groups individually, we found it performed more reliably than minimizing a more complicated penalty function including orthonormality terms with respect to all αs and all unit vector coordinates of all rigid groups at once.

The full linearized description of the molecule requires not only the αs to go from unit vector coordinates to atom coordinates according to eq. (1) but also the βs for the inverse calculation via eq. (3). Although linearized embedding does not require βs, other conformational calculations do, so we describe their determination here for completeness. For rigid group r, let $\vec{\beta}^r = (\beta_{1,j}^r, \ldots, \beta_{n_a^r,j}^r, \beta_{\text{root},j}^r, \beta_{\text{parent},j}^r)$ be the vector of desired βs for the jth unit vector, let $\tilde{\mathbf{u}}_j^r = (u_{j,x}^r, u_{j,y}^r, u_{j,z}^r, 0)$ be the known coordinates of the jth unit vector with a zero appended, and let

$$D^r = \begin{pmatrix} p_{1,x}^r & \cdots & p_{n_a^r,x}^r & p_{\text{root},x}^r & p_{\text{parent},x}^r \\ p_{1,y}^r & \cdots & p_{n_a^r,y}^r & p_{\text{root},y}^r & p_{\text{parent},y}^r \\ p_{1,y}^r & \cdots & p_{n_a^r,z}^r & p_{\text{root},z}^r & p_{\text{parent},z}^r \\ 1 & \cdots & 1 & 1 & 1 \end{pmatrix} \tag{11}$$

Clearly, the root and parent entries are not included in $\vec{\beta}^1$ and D^1. Then, solving

$$D^r\vec{\beta}^r = \hat{\mathbf{u}}_j^r \qquad (12)$$

determines the βs for this unit vector in this rigid group, along with the side condition that they sum to zero to make them unique. Because this is always an underdetermined system, we solve it by computing the Moore–Penrose generalized inverse[13] for D^r.

LINIZE uses eq. (1) to calculate atom coordinates from unit vector coordinates because it requires the fewest αs. However, it does require the full organization of the molecule in terms of rigid groups with their root and parent atoms and implies that *all* atom coordinates need to be calculated in order of their rigid groups. A more convenient form, especially for calculating the coordinates of an arbitrary atom i, is

$$\mathbf{p}_i = \mathbf{w} + \sum_j \alpha_{ij}\mathbf{u}_j \qquad (13)$$

where the index j runs over all unit vectors in the entire molecule. Similarly, eq. (3) becomes

$$\mathbf{u}_j = \sum_i \beta_{ij}(\mathbf{p}_i - \mathbf{w}) \qquad (14)$$

where i runs over all atoms in the whole molecule. Clearly, one can convert from eq. (1) to eq. (13) by noting that the parent and root atoms for one rigid group can be written as a linear combination of αs and unit vectors from previous rigid groups. Linearized embedding uses eq. (13), as we shall see.

Linearized Embedding

In overview, the linearized embedding algorithm begins with the linearized form of the molecule and a set of explicit interatomic distance constraints. The current implementation of the algorithm, the program LE, deals only with intramolecular constraints acting on a single, covalently connected molecule, although there is no fundamental reason why it could not be generalized to handle multiple molecules with intermolecular constraints as well. The linearized representation itself embodies a number of constraints on certain linear combinations of unit vector dot products in order that individual rigid groups of atoms are not distorted and so as to maintain the rigid valence geometry relative orientations between adjacent rigid groups. The explicit distance constraints can be put in a similar form and added to the list of implicit constraints. Like the bound smoothing step in EMBED, these linear constraints imply some generally valid contraction of the ranges allowed to some of the metric matrix elements (the matrix of all unit vector dot products). This helps in choosing a random trial metric matrix, but much more important is to adjust the random choices so they precisely obey all the linear constraints. Then trial unit vector coordinates in R^3 are found, but now by a new method not involving the eigenvalues of the metric matrix. The advantage is that these trial coordinates exactly obey all the implicit constraints from the rigid valence geometry assumption. These trial coordinates are refined by optimizing a penalty function of explicit distance and van der Waals constraint violations with respect to the dihedral angles of all the rotatable bonds. The outcome is a set of Cartesian coordinates for the unit vectors, which can be readily converted to atomic coordinates. As usual, many different random refined structures can be produced by choosing different random trial metric matrices. The following paragraphs expand on this outline in an attempt to highlight the key equations without drowning in programming detail, a matter of some 5000 lines of C source code.

We start with the linearized representation of the molecule in terms of n_a atoms, n_u unit vectors, and n_r rigid groups, necessarily joined together by $n_r - 1$ rotatable bonds. Suppose there is an explicit distance constraint between atoms a and b of the form $d_{a,b} < v_{a,b}$. (Distance lower bounds and equalities can be treated in exactly the same way.) Now, because the coordinates of these atoms are expressible in terms of the unit vector coordinates and the fixed coefficients

$$\mathbf{p}_a = \mathbf{w} + \sum_{i=1}^{n_u} \alpha_{a,i}\mathbf{u}_i \quad \text{and} \quad \mathbf{p}_b = \mathbf{w} + \sum_{i=1}^{n_u} \alpha_{b,i}\mathbf{u}_i \qquad (15)$$

then

$$\begin{aligned} d_{a,b}^2 &= \|\mathbf{p}_a - \mathbf{p}_b\|^2 \\ &= \sum_{i=1}^{n_u} (\alpha_{i,i} - \alpha_{b,i})^2 + 2\sum_{i<j}^{n_u} (\alpha_{a,i} - \alpha_{b,i}) \\ &\quad \times (\alpha_{a,j} - \alpha_{b,j})\mathbf{u}_i \cdot \mathbf{u}_j < v_{a,b}^2 \end{aligned} \qquad (16)$$

because $\mathbf{u}_i^2 = 1$. Other linear constraints on the $\mathbf{u}_i \cdot \mathbf{u}_j$s arise from the mutual orthogonality of unit vectors belonging to the same rigid group

$$\mathbf{u}_i \cdot \mathbf{u}_j = 0 \qquad \forall i, j \in \text{same rigid group} \qquad (17)$$

and from the orthogonality of the unit vectors $\mathbf{u}_j$ in any but the first rigid group to their defining bond vector

$$\sum_{i=1}^{n_u} (\alpha_{a,i} - \alpha_{b,i})\mathbf{u}_i \cdot \mathbf{u}_j = 0 \qquad (18)$$

where atoms a and b are the parent and root atoms defining the bond. A last type of implicit constraint comes from the orthogonality of a unit vector $\mathbf{u}_j$ to its normalized defining bond vector

$$\frac{\mathbf{p}_a - \mathbf{p}_b}{\|\mathbf{p_a} - \mathbf{p_b}\|} = \sum_{k=1}^{n_u} \alpha_{ab,k}\mathbf{u}_k \qquad (19)$$

and the fixed but not necessarily 90° angle between the bond vector and a unit vector $\mathbf{u}_i$ in the parent rigid group, i.e., the group on the other end of the a—b bond. Then

$$|\mathbf{u}_i \cdot \mathbf{u}_j| \leq \sqrt{1 - \alpha_{ab,i}^2} \qquad (20)$$

which is equivalent to saying that $\mathbf{u}_i \cdot \mathbf{u}_j$ achieves maximal and minimal values at the *cis* and *trans* conformations given by the sine of the angle between $\mathbf{u}_i$ and the bond vector.

In EMBED, bound smoothing raises the lower bounds and lowers the upper bounds on many of the interatomic distances. The equivalent operation here is to start with the default limits

$$-1 = \lambda_{ij} \leq \mathbf{u}_i \cdot \mathbf{u}_j \leq v_{ij} = 1 \qquad (21)$$

combined with all the linear inequalities enumerated above, and successively maximize and minimize each $\mathbf{u}_i \cdot \mathbf{u}_j$ by linear programming. This is the best bounds contraction that can be derived from given constraints, but the CPU time required is large compared to the rest of the steps in the algorithm. As illustrated in Figure 2, the region of random $\mathbf{u}_i \cdot \mathbf{u}_j$ sampling later in the algorithm is contracted to some hyperrectangle in $\mathbb{R}^{n_u}$, while the actual feasible region may be much smaller due to the frequently occurring equality and near-equality constraints. Consequently, we use a much faster approximation to the linear programming process that still substantially contracts the bounds of a minority of the metric matrix elements while keeping in mind that we will have to refer to the original constraints later. The smoothing procedure involves choosing one constraint (other than a van der Waals constraint) and solving it for a particular $\mathbf{u}_i \cdot \mathbf{u}_j$, resulting in an upper bound, for example.

$$\mathbf{u}_i \cdot \mathbf{u}_j \leq \sum_{k,l} c_{k,l} \mathbf{u}_k \cdot \mathbf{u}_l \qquad (22)$$

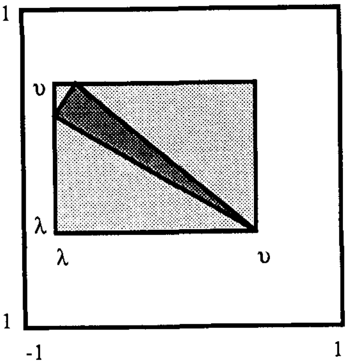

Figure 2. Two variable example of a typical bound smoothing situation. The initial bounds of eq. (21) correspond to the large white box, but the detailed constraints allow only the feasible region shown as a dark triangle. Bound smoothing can contract only to the lightly shaded rectangle, thus still permitting many infeasible choices of $\mathbf{u}_i \cdot \mathbf{u}_j$s.

Then, substituting the current limits on the other variables

$$\mathbf{u}_i \cdot \mathbf{u}_j \leq \left[\sum_{k,l} c_{k,l} \begin{cases} v_{kl} & \text{if } c_{k,l} > 0 \\ \lambda_{kl} & \text{if } c_{k,l} < 0 \end{cases} \right] < v_{ij} \qquad (23)$$

may produce a tighter upper bound for $\mathbf{u}_i \cdot \mathbf{u}_j$. Similarly, lower bounds can be raised. Repeat for all constraints and all $\mathbf{u}_i \cdot \mathbf{u}_j$s until no further improvements can be made. In general, the range for most $\mathbf{u}_i \cdot \mathbf{u}_j$s remains 2, as in eq. (21), but for some the range contracts to 1 or less, typically when $\mathbf{u}_i$ and $\mathbf{u}_j$ lie in adjacent rigid groups. On average, we see a contraction of only a few percent. This smoothing step completes all the preliminary calculations.

For each conformer to be generated, we cycle through this and the following steps. First, choose values for all elements of the (symmetric) metric matrix by taking independent random numbers uniformly distributed between the corresponding bounds. This metric matrix does not in general satisfy the original list of linear constraints for reasons explained above. Suppose there are altogether n_c linear constraints, of which often nearly half, or n_e, are equalities. There are $n_v = n_u(n_u - 1)/2$ variable entries in the random metric matrix, and generally $n_v \gg n_e$. Use the equalities to eliminate n_e of the variables from the remaining $n_c - n_e$ linear inequality constraints, and solve these by subgradient optimization. Subgradient optimization applied to solving a set of inequalities amounts to simply finding the worst violated inequality given the current approximation to the solution and taking a step along its gradient to reduce the violation, according to a step size choice given by Sandi.[14] For the kinds of problems we are trying to solve here, the procedure tends to oscillate between two different inequalities having nearly opposite gradient vectors and thus makes little progress per iteration. In the spirit of modifications outlined by Fletcher,[15] we average the two gradients when this behavior arises. Typically, the random metric matrix converges to a trial metric matrix accurately satisfying all the linear constraints within 10 or 20 iterations, but two features are absolutely essential for this rapid convergence: the gradient averaging and using the equalities to eliminate variables, rather than including them as explicit constraints during subgradient optimization.

The trial metric matrix satisfies all the linear constraints, but it does not in general have 3 positive and $n_u - 3$ zero eigenvalues, the condition for embeddability in $\mathbb{R}^3$. If we generate unit vector coordinates in the usual way[1] from the three largest positive eigenvalues and their eigenvectors, the local constraints, such as unit vector orthonormality within rigid groups, are grossly violated. Instead, we generate unit vector coordinates in $\mathbb{R}^3$ exactly satisfying all the local constraints, while attempting to otherwise agree with the trial metric matrix in a least-squares sense. The procedure is to first place the orthonormal unit vectors of the first rigid group in an arbitrary orientation, say parallel to the coordinate axes. Then, for every subsequent rigid group let $\mathbf{v}$ be the first unit vector of the group and let $U = (u_{i,j})$ be the $n \times 3$ array of the n determined unit vectors from previous rigid groups such that the dot product of $\mathbf{v}$ and each of these fixed unit vectors was involved in at least one of the linear constraints. (Using *all* previously determined unit vectors tends to dilute the quality of the least-squares fitting of $\mathbf{v}$ by striving to fit many elements of the trial metric matrix that are very random.) The problem is to

$$\text{minimize} \sum_{i=1}^{n} \left(t_i - \sum_{j=1}^{3} u_{i,j} v_j \right)^2 \qquad (24)$$
$$\text{subject to } \mathbf{v} \cdot \mathbf{b} = 0$$

where $\mathbf{b}$ is the current rigid group's defining bond vector and $\mathbf{t}$ is the vector of desired dot products between $\mathbf{v}$ and the already determined unit vectors of U, as specified by the trial metric matrix. The solution to this constrained optimization problem can be found by solving

$$\begin{pmatrix} U^T U & \mathbf{b} \\ \mathbf{b}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{t}^T U \\ 0 \end{pmatrix} \qquad (25)$$

where λ is the Lagrange multiplier for the orthogonality constraint. The resulting $\mathbf{v}$ must be subsequently normalized. Technically, this is not the same as adding a normalization constraint to eq. (24), but that would make the problem nonlinear. Given that all this is just a heuristic to find promising initial coordinates for the unit vectors, knowing full well they will not satisfy the original linear constraints, dealing with a nonlinear constrained optimization hardly seems worth the trouble. There is one special case not covered by our calculation of $\mathbf{v}$, namely, when it is the first unit vector in the second rigid group and the first rigid group was planar. Then, $n = 2$, $\mathbf{u}_1 = (1, 0, 0)$, $\mathbf{u}_2 = (0, 1, 0)$, the normalized bond vector $\mathbf{b} = (b_x, b_y, 0)$, and $\mathbf{t} = (m_{1,3}, m_{2,3})$, where $M = (m_{i,j})$ is the trial metric matrix. Then, if $-1 \le \cos\theta = b_y m_{1,3} - b_x m_{2,3} \le 1$ the optimal

$$\mathbf{v} = (b_y \cos\theta, -b_x \cos\theta, \sin\theta) \qquad (26)$$

Otherwise; the solution lies in the plane

$$\mathbf{v} = \begin{cases} (b_y, -b_x, 0) & \text{whichever is better} \\ (-b_y, b_x, 0) & \end{cases} \qquad (27)$$

If the current rigid group is planar, we are done, but if not, the second unit vector is just $\pm\mathbf{v} \times \mathbf{b}$ with the sign chosen to give the correct chirality of the rigid group.

The trial unit vector coordinates just calculated are certainly in $\mathbb{R}^3$ and obey all the local constraints of orthonormality within a rigid group, correct local chiral centers, and orthogonality to corresponding bond vectors. However, they do not necessarily obey the linear constraints that the trial metric matrix did, not do the implied atom coordinates obey van der Waals minimal interatomic distances. The last step is to find refined unit vector coordinates and hence atom coordinates by minimizing a penalty function, starting from the trial coordinates. To maintain the precise local geometry, this unconstrained local minimization is carried out with respect to dihedral angles about the $n_r - 1$ rotatable bonds. We used Shanno's conjugate gradient minimizer with Beale restarts[12] applied to the error function

$$f = \sum_{i,j} \begin{cases} \left(\dfrac{d_{ij}^2}{v_{ij}^2} - 1 \right)^2 & \text{if } d_{ij} > v_{ij} \\ 0 & \text{if } \lambda_{ij} \le d_{ij} \le v_{ij} \\ \left(\dfrac{\lambda_{ij}^2}{d_{ij}^2} - 1 \right)^2 & \text{if } d_{ij} < \lambda_{ij} \end{cases} \qquad (28)$$

where the sum runs over van der Waals lower bounds on 1–4 interatomic distances and beyond, and over the explicit interatomic distance constraints. Van der Waals and explicit constraints may be weighted differently. For example, we find it improves convergence to first minimize f with a low weight on van der Waals compared to explicit constraints, and then follow with a second round of minimization weighting them equally. The first pass locates the correct overall conformation without massive van der Waals violations, and the second pass makes small changes to the conformation to relieve interatomic overlaps.

RESULTS

Cyclohexane

The main use of the EMBED algorithm has been to determine conformations of small proteins from the results of NMR studies. There has been considerable concern about the breadth and evenness of sampling the allowed conformation space. With such large molecules and complicated sets of constraints, it has been hard enough to decide whether the constraints are mutually consistent, i.e., whether there is any allowed set of conformations to sample at all, much less whether all major types of conformations have been seen and that the sampling was in some sense representative. As a test on a relevant, known system, Havel[16] demonstrated agreement between the sampling of polyalanine chains generated by his DG-II program,[17] an advanced implementation of EMBED, with the distributions expected by polymer theory. Here, we initially tested LE with the much simpler case of cyclohexane, where the full conformation space is known analytically, for example, by Dress's interatomic distances derivation,[1] or by an analysis of the linearized representation.[18] Disregarding the hydrogens and insisting on uniform C—C bond lengths and uniform tetrahedral

C—C—C bond angles, the chair conformation is an isolated point in conformation space, the boat and skew-boat conformations form a continuous, locally one-dimensional closed loop traversed by a single degree of pseudorotation freedom, and there are no other conformations.

As we have seen above, calculating the linearized representation for even hexane is a nontrivial operation, so we used the LINIZE program, but this has so much chemical knowledge built into it that one must start with a chemically correct C_6H_{14} molecule rather than just the carbon skeleton. Consequently, we began with the default *n*-hexane structure generated by the commercial Quanta program, which happens to give C—C bond lengths of 1.529 Å and C—C—C bond angles of 112.7° (greater than the perfect tetrahedral angle of 109.47°), corresponding to a 1–3 distance of 2.546 Å. To compare as closely as possible with the analytic results, all van der Waals radii were set to zero, leaving only the three explicit distance constraints: $d(\mathrm{C1, C6}) = 1.529$ Å, $d(\mathrm{C1, C5}) = d(\mathrm{C2, C6}) = 2.546$ Å. A total of 100 sets of refined coordinates were produced from 100 sets of trial coordinates (what we will call a 100/100 success ratio) in 0.251 s of CPU time per trial structure on a Sun SparcStation II. The 25 s for the run includes both the initial set-up and analysis of the general constraints, as well as the 100 repetitions of the steps associated with each structure, but the majority of the time was in the latter. The maximum violation of an explicit distance constraint by any trial structure was 3.2 Å, although in general it was on the order of 0.5 Å, indicating that usually even the trial structures nearly satisfied the ring closure constraint. Figure 3 shows a scatter plot of the C1—C2 vs. C3—C4 dihedral angles. Seven of the structures cluster around the chair conformation, 1 is at the mirror image chair, and the remaining 92 are distributed broadly but not entirely uniformly around the boat pseudorotation loop. The maximum allowed constraint violation in all these studies was 0.1 Å, so the clustering of the conformations was a little fuzzy. To ensure that this maximal constraint violation would always be obeyed, the final minimization of the error function with respect to dihedral angles had to be carried out to a squared gradient magnitude of 0.005 or less. A perfect cyclohexane chair with perfect tetrahedral bond angles would have dihedral angles [±60°, ∓60°, ±60°, ∓60°, ±60°, ∓60°], but due to our greater C—C—C angle and limited precision, chair dihedral angles ranged in magnitude from 44–56°.

Like LINIZE-LE, the traditional embedding program, DGEOM, deduces its local constraints by starting with an arbitrary, unconstrained conformation of the molecule in question. To make as close a comparison as possible between LE and DGEOM, we started the latter with the same *n*-hexane conformation including all hydrogens, the same required distance accuracy and gradient norm (otherwise the default options), the same zero van der Waals constraints, and the same three explicit ring closure constraints. This produced 100/108 structures in 0.51 s per trial structure. The eight failures were really due to local minima in the error function during the refinement step. It is clear in Figure 4 that DGEOM locates only the chair conformation and its mirror image, apparently due to its well-known tendency to favor conformations having long interatomic distances.

However, in defense of DGEOM, if one starts with only six carbon atoms but specifies explicitly all 15 distance constraints among them (6 fixed bond lengths, 6 fixed bond angles, and otherwise zero

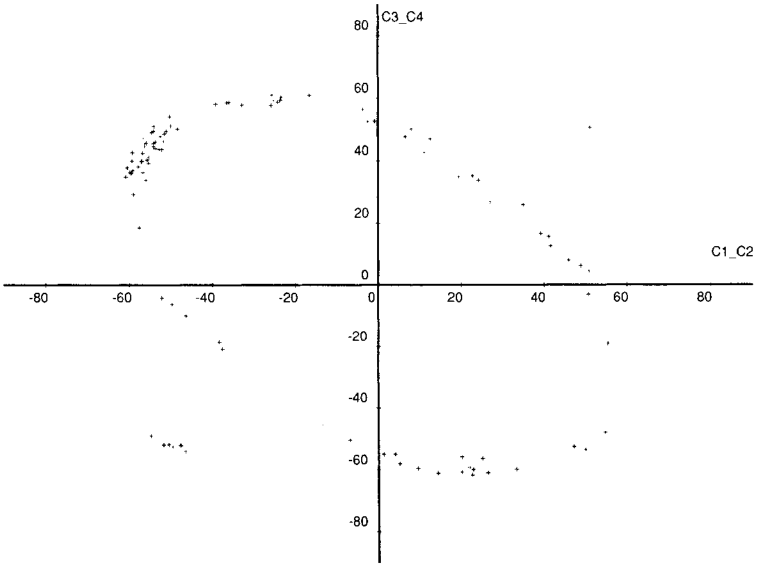

Figure 3. Scatter plot of two dihedral angles for 100 conformers of cyclohexane as generated by LE.

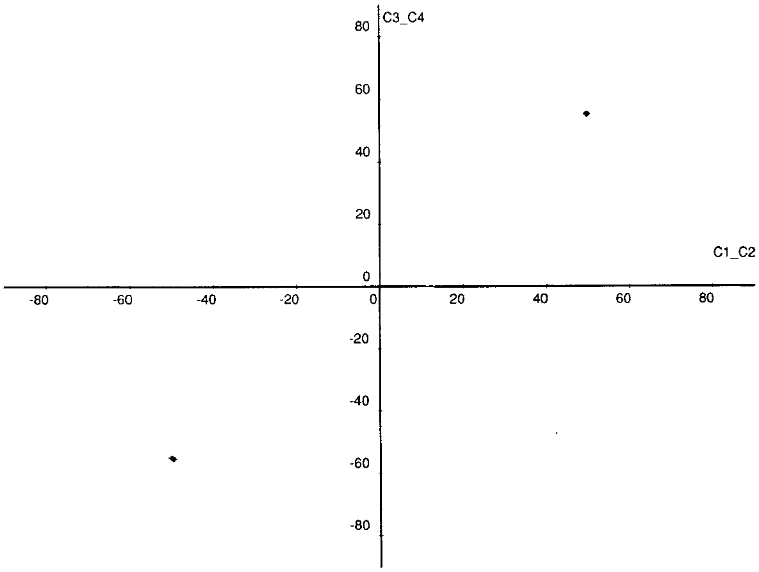

Figure 4. Scatter plot of two dihedral angles for 100 conformers of cyclohexane as generated by DGEOM (cf. Fig. 3).

lower bounds on other intercarbon distances), then the sampling is dramatically better. All 100/100 structures converge, averaging 0.03 s per structure. Five are chair conformations and the other 95 uniformly sample the boat pseudorotation, resulting in a picture essentially like Figure 3. Why the presence of hydrogens, all with zero van der Waals radii, should damage the sampling is not yet clear.

Cycloheptane

The set of conformations for cycloheptane allowed by rigid valence geometry has long been known to consist of a continuous one-dimensional boat pseudorotation loop and a similar chair pseudorotation loop. These paths have been mapped out in some detail, and no other conformations are known.[18] Proceeding as before from the standard heptane structure produced by Quanta (without energy minimization), and using zero van der Waals radii and the same ring closure explicit constraints, LE generates 500/500 structures in 0.494 s per trial structure. It is a little harder to see the two nonintersecting pseudorotation loops without going to a seven-dimensional display, but Figure 5 is one of the best choices of a pair of dihedral angles. Note that somewhat fuzzy curved lines are clearly visible, somewhat unevenly sampled (although the projection from 7 to 2 dimensions also has the effect of making the sampling appear heavy in some places), but obviously a broad range of conformations has been examined. The picture is clearer if we exploit the sevenfold

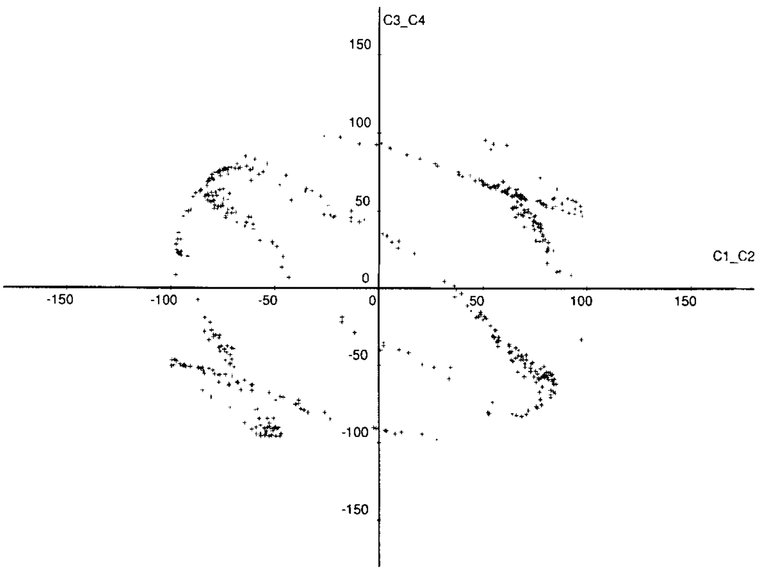

Figure 5. Scatter plot of two dihedral angles for 500 conformers of cycloheptane as generated by LE.

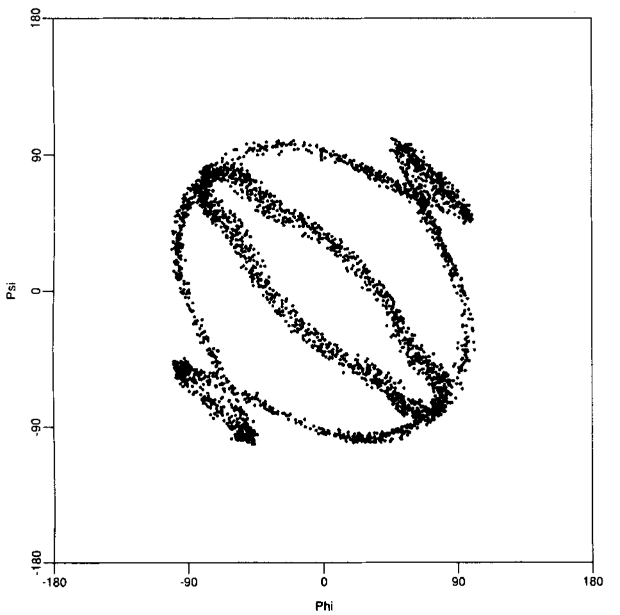

Figure 6. An enhanced scatter plot with 3500 values of two dihedral angles of cycloheptane obtained by cyclic permutation of the atom labels for the 500 conformers in Figure 5.

symmetry of the search by superimposing the scatterplot for the C1—C2 vs. C3—C4 dihedral angles onto that for the C2—C3 vs. C4—C5 angles, etc., as shown in Figure 6. The outer "double figure eight" pattern is the chair pseudorotation one-dimensional continuum, while the inner simple elliptical loop is the boat pseudorotation. In the full seven dimensions, the chair path does not actually cross itself and does not really touch the boat path.

Once again, we made a matching run with DGEOM, resulting in 500/1275 successful conformers, requiring 0.791 s of CPU per trial structure. The scatterplot equivalent to Figure 5 is Figure 7, which shows only 10 tight clusters, 2 on the boat pseudorotation loop and 8 on the chair loop. Although there is some diversity in the structures it found, they are so tightly clustered that the majority of the allowed set of conformations is untouched, giving the erroneous impression that cycloheptane has only a few rigid conformations. As in the case of cyclohexane, the sampling is much better if only the seven carbons are used and all the intercarbon constraints are specified explicitly. Then, 500/500 successful conformers are produced in 0.682 s per structure, and the sampling, shown in Figure 8, thoroughly covers the chair pseudorotation while missing the boat altogether.

DPLPE

DPLPE is a conformationally restricted opioid receptor selective enkephalin analog, [D-Pen^2, L-Pen^5]enkephalin, Tyr-D – $\overline{\text{Pen-Gly-Phe-L-Pen}}$, where Pen, penicillamine, is β,β-dimethylcysteine. Quite aside from some interesting NMR studies on the solution conformation of such molecules,[19] just the constraints of rigid valence geometry, van der Waals contacts, and closing the disulfide bridge make this a challenging test case. As with the small cyclic alkanes, we required a maximum distance constraint violation of 0.1 Å instead of the more customary 0.5 Å for molecules of DPLPE's size. Van der Waals radii were set to 90% of the default values used in DGEOM because there appear to be no cyclic conformations compatible with full radii. The only explicit distance constraints were $d(2{:}S^{\gamma}, 5{:}S^{\gamma}) = 2.025$ Å, $d(2{:}C^{\beta}, 5{:}S^{\gamma}) = 3.08$ Å, and $d(5{:}C^{\beta}, 2{:}S^{\gamma}) = 3.23$ Å, where the number before the colon is the residue

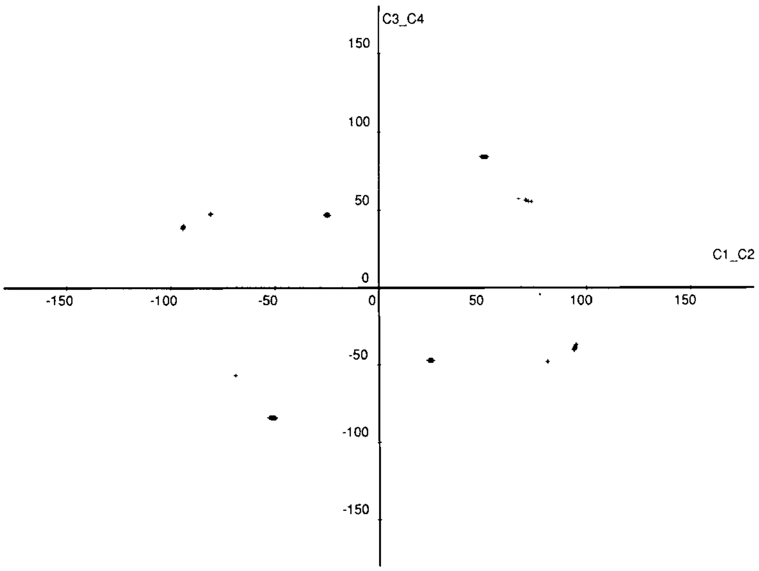

Figure 7. Scatter plot of two dihedral angles for 500 conformers of cycloheptane as generated by DGEOM starting from the full C_7H_{16} molecule (cf. Fig. 5).

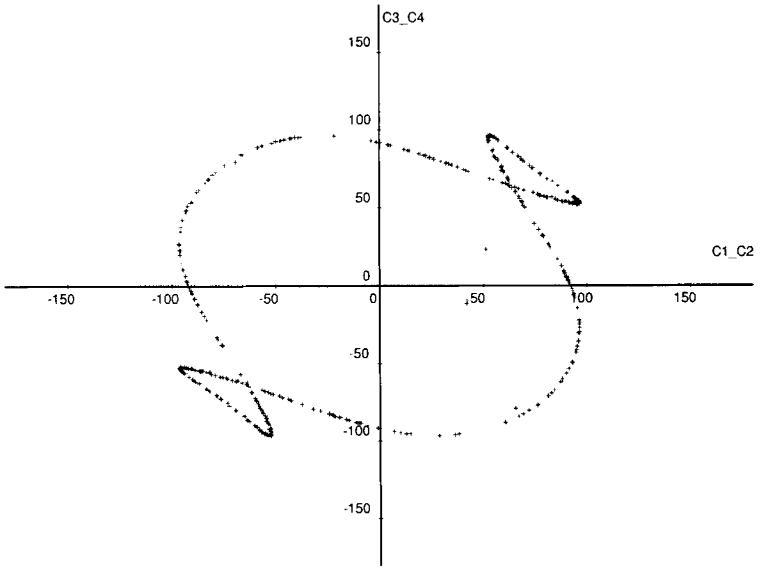

Figure 8. Scatter plot of two dihedral angles for 500 conformers of cycloheptane as generated by DGEOM starting from the C_7 skeleton.

number and the symbol after is the fairly standard amino acid atom label. LE produced 60/100 conformers in 27.4 seconds per trial structure. This is the first time the constraint set was challenging enough to reduce the success rate down from 100 to 60%. The equivalent run with DGEOM found 100/607 structures in 50.0 s per trial structure, a success ratio of only 16.5%.

It is harder to comment on the breadth and uniformity of sampling because we have no *a priori* knowledge about what conformations are really allowed, there are 24 rotatable bonds (peptide bonds were held *planar* and *trans*), and there were only 60 structures from LE. To be perfectly equitable, only the first 60 of the 100 DGEOM structures were examined. LE shows that the 2:C$^\alpha$—2:C$^\beta$ dihedral is −60° or 180°, the 5:C$^\alpha$—5:C$^\beta$ dihedral is ±60°, the disulfide 2:S$^\gamma$—5:S$^\gamma$ dihedral falls in the range −60° to +60°, and the remaining dihedrals seem to scatter broadly throughout their full 360° range. Figure 9 shows, for example, the four clusters of values for 2:C$^\alpha$—2:C$^\beta$ vs. 5:C$^\alpha$—5:C$^\beta$. In the equivalent plot from DGEOM, Figure 10, the tight clusters are spread out yet they miss the 2:C$^\alpha$—2:C$^\beta$ value of −60° entirely. Part of the difference may be due to the way DGEOM treats the molecule as 91 independently moveable atoms subject only to a maximum interatomic distance violation of 0.1 Å, while LE views it as 40 unit vectors maintaining local orthogonality constraints to very high precision. The missing conformations

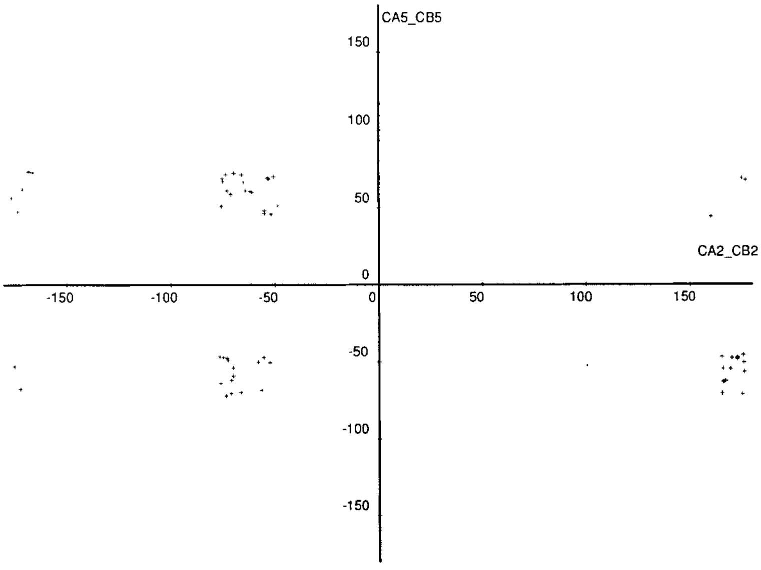

Figure 9. Scatter plot of two side-chain dihedral angles, χ_1 of residue 2 and χ_1 of residue 5, for 60 conformers of DPLPE as generated by LE.

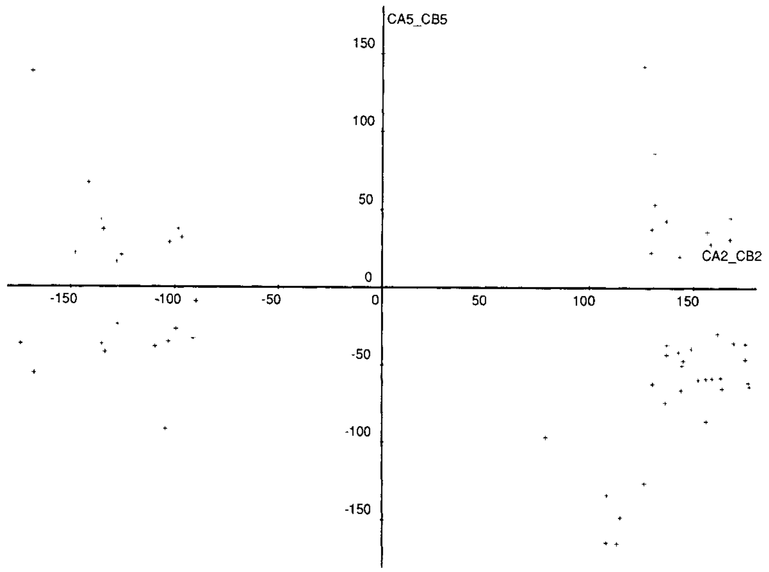

Figure 10. Scatter plot of two side-chain dihedral angles, χ_1 of residue 2 and χ_1 of residue 5, for 60 conformers of DPLPE as generated by DGEOM.

may once again be due to DGEOM's preference for the more extended *trans* conformation of 2:C$^\alpha$—2:C$^\beta$.

CONCLUSIONS

Although we have certainly not carried out an exhaustive complexity study for LE, these three examples show that it is consistently twice as fast per trial structure than DGEOM is, and then for challenging sets of constraints the trial structures can be as much as four times more likely to refine successfully. Moreover, LE shows a wider and more even sampling of the available conformations that can even be called roughly satisfactory in the cyclohexane and cycloheptane tests, where we know what the correct sampling should be. This is not to say that DGEOM is a bad program. On the contrary, it is a mature and robust piece of software that has consistently located unanticipated conformations in a broad spectrum of situations, including multiple molecule problems that our LE program cannot yet treat.

It is interesting to compare these results with some work by Havel,[17] where he experimented with a variation on linearized embedding he calls "angular embedding." He notes that the angles between unit vectors can be treated just like the distances between points for the purpose of bound smoothing at the triangle inequality level, whereas the dot products between unit vectors cannot be used this way. When the results of this operation are converted back to bounds on the dot products, we observe only a tiny contraction in the bounds, another reason why our algorithm solves the linear constraints on the dot products explicitly after choosing a random metric matrix. Thus, we are able to extract more information directly out of the constraints at still a reasonable cost than the angular embedding method could by concentrating on upper and lower bounds. Unfortunately, he concludes that "... angular embedding as well as, by inference, Crippen's linearized embedding, are unlikely to be useful in the determination of the structures of long-chain polymers from NOE distance information." True, DPLPE is only a pentapeptide, but the outlook for bigger problems is considerably brighter than that. Despite the inherent numerical stability problems associated with internal coordinate representations of large molecules, such as dihedral angles or linearization, preliminary work now in progress shows that LINIZE and LE work with 20 residue polypeptides.

This work was supported by grants from the National Institutes of Health (GM37123 and DA06746). W.W.R. was supported as an E. Mead Johnson–AFPE Memorial Fellow. The authors thank Joel Dinverno for assistance in setting up the comparison studies between LE and DGEOM.

References

1. G.M. Crippen and T.F. Havel, in D. Bawden, Ed., *Chemometrics Research Studies Series*, Research Studies Press (Wiley), New York, 1988.
2. W. Braun and N. Go, *J. Mol. Biol.*, **186,** 611 (1985).
3. M. Nilges, G.M. Clore, and A.M. Gronenborn, *FEBS Lett.*, **229,** 317 (1988).
4. G.M. Crippen, *J. Math. Chem.*, **6,** 307 (1991).
5. A.S. Smellie, CONSTRICTOR, Oxford Molecular Ltd., Terrapin House, University Science Area, South Parks Road, Oxford, UK, 1989.
6. J.M. Blaney, DGEOM, Quantum Chemistry Program Exchange, Department of Chemistry, Indiana University, Bloomington, IN 47405, 1989.

7. C.E. Peishoff, J.S. Dixon, and K.D. Kopple, *Biopolymers*, **30,** 45 (1990).
8. R.P. Sheridan, R. Nilakantan, J.S. Dixon, and R. Venkataraghavan, *J. Med. Chem.*, **29,** 899 (1986).
9. J. Lautz, H. Kessler, J.M. Blaney, R.M. Scheek, and W.F. van Gunsteren, *Int. J. Peptide Protein Res.*, **33,** 281 (1989).
10. G.M. Crippen, *J. Comp. Chem.*, **10,** 896 (1989).
11. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1986.
12. D.F. Shanno and K.H. Phua, *ACM Trans. Math. Softw.*, **6,** 618 (1980).
13. N.N. Gupta, *IEEE Trans. Systems, Man Cybernet.*, **SMC-1,** 89 (1971).
14. C. Sandi, in N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, Eds., *Combinatorial Optimization*, Wiley, New York, 1979, pp. 73–91.
15. R. Fletcher, *Practical Methods of Optimization*, John Wiley, New York, 1987.
16. T.F. Havel, *Biopolymers*, **29,** 1565 (1990).
17. T.F. Havel, *Prog. Biophys. Molec. Biol.*, **56,** 43 (1991).
18. G.M. Crippen, *J. Comp. Chem.*, **13,** 351 (1992).
19. H.I. Mosberg, K. Sobczyk-Kojiro, P. Subramanian, G.M. Crippen, K. Ramalingam, and R.W. Woodard, *J. Am. Chem. Soc.*, **112,** 822 (1990).