Journal of Magnetic Resonance **152,** 288–302 (2001)
doi:10.1006/jmre.2001.2413, available online at http://www.idealibrary.com on IDEAL®

Internal Coordinates for Molecular Dynamics and Minimization in Structure Determination and Refinement

Charles D. Schwieters* and G. Marius Clore†

*Computational Bioscience and Engineering Laboratory, Center for Information Technology, National Institutes of Health, Building 12A, Bethesda, Maryland 20892-5624; and †Laboratory of Chemical Physics, National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, Building 5, Bethesda, Maryland 20892-0510

E-mail: Charles.Schwieters@nih.gov, clore@speck.niddk.nih.gov

Received April 23, 2001; revised July 10, 2001; published online September 6, 2001

We present a software module which allows one to efficiently perform molecular dynamics and local minimization calculations in internal coordinates when incorporated into a molecular dynamics package. We have implemented a reference interface to the NIH version of the X-PLOR structure refinement package and we show that the module provides superior torsion-angle dynamics functionality relative to the native X-PLOR implementation. The module has been designed in a portable fashion so that interfacing it with other packages should be relatively easy. Other features of the module include the ability to define rather general internal coordinates, an accurate integration algorithm which can automatically adjust the integration step size, and a modular design, which facilitates extending and enhancing the module.

1. INTRODUCTION

In the past ten years efficient algorithms have made computationally tractable the use of internal coordinates in molecular dynamics simulations of systems of biological interest (having more than say 100 atoms). Internal coordinates are an attractive alternative to the Cartesian coordinates of each atom when particular degrees of freedom are not of interest. For example, in the process of NMR structure determination and refinement in which one seeks molecular structures consistent with experimental NMR data, the bond lengths and bond angles are generally taken as fixed—and no information about these features is generally available from the NMR experiments. If these known coordinates are removed from the local optimizations and molecular dynamics simulations, the conformational search space becomes smaller and can be more rapidly sampled. For example, typical proteins have approximately $N_a/3$ torsion angles compared with $3N_a$ coordinates in atomic Cartesian space, where N_a is the number of atoms. Hence, the conformational space is about an order of magnitude smaller if torsion angles are used. Furthermore, in torsional angle molecular dynamics (TAMD), it is typical that the timestep required to maintain a given level of energy conservation is about 10 times larger than that required in atomic Cartesian space because the high frequency bond bending and stretching motions have been removed. Other aspects of the simulation might also be made more efficient because bond and bond-angle forces no longer need be calculated and because there are fewer coordinates to update in the integrator. However, these final two aspects have not been found to make a significant contribution to dynamics run times in practice.

An efficient recursive algorithm for dynamics in internal coordinates was originally introduced in the robotics literature (*1–4*). This algorithm was then implemented for TAMD in X-ray and NMR refinement packages (*5–7*) and in a more general purpose molecular dynamics package (*8, 9*). In this paper we report the implementation of a general internal variable dynamics module (IVM) for efficient molecular dynamics. It allows general hinge definitions including those used in TAMD, but it also allows more general coordinates which are appropriate when some degrees of freedom are of interest and others are not; for instance, in the refinement problem of a two protein complex in which the backbone coordinates of the isolated protein structures are already known.

The IVM also includes local minimization routines (Powell method conjugate gradient and steepest descent) so that these techniques can be conveniently employed in the same coordinate system. Our package employs an efficient sixth-order predictor-corrector integrator, which requires one force evaluation per timestep and allows for automatic timestep adjustment. We have implemented loop constraints to maintain bond lengths in ring topologies, although as yet we have found the feature to be of limited use. Finally, the code has been developed in a highly modular fashion to make the addition of new hinge definitions, integrators, and minimizers a relatively simple task. The IVM is not a stand-alone program as it does not have code to evaluate forces and lacks support for file formats, etc. It is currently interfaced to the NIH version of X-PLOR (*10*), and we intend to integrate the IVM into other packages.

In the next section, we derive the equations of motion in internal coordinates and outline the recursive solution. In

1090-7807/01

Section 3 we document the details of the current implementation. In Section 4 we give two examples of using the IVM in the refinement of NMR structures. Finally, Section 5 contains some concluding remarks. The software can be obtained as part of the NIH version of X-PLOR (*10*) or by contacting the authors.

2. FORMULATION

In Cartesian coordinates, Newton's equations are well known,

$$-\nabla_{q_i} \mathsf{V} = m_i \frac{d^2 q_i}{dt^2}, \qquad [1]$$

where q_i is the position of the ith atom, m_i is its mass, and V is the total potential energy. Equation [1] reflects the fact that the atomic coordinates are coupled only in the potential energy term. However, in internal coordinates, Newton's equation reads

$$-\nabla_{\theta} \mathsf{V} = \mathcal{M} \frac{d^2 \theta}{dt^2} + \mathcal{C}, \qquad [2]$$

where θ describes a vector of internal coordinates; $\mathcal{C}$ is a vector of Coriolis forces present because these coordinates are noninertial; and $\mathcal{M}$ is a mass matrix which is, in general, not diagonal and the elements of which vary as the coordinates change in time. An equation for the mass matrix is given in Section 2.4. Naively, in solving Eq. [2], one would expect to expend computational effort proportional to the cube of the number of internal coordinates, making its solution more expensive than the evaluation of the forces, and resulting in unacceptably poor performance for most molecules of biological interest. However, Jain *et al.* (*1, 2*) and Bae and Haug (*3, 4*) have come up with recursive algorithms to solve Eq. [2] with effort directly proportional to the number of internal coordinates if the molecule is decomposed into a hierarchical tree structure as described below. We outline this recursive algorithm in Section 2.5.

Following (*1*) we decompose a molecule of interest into collections of one or more atoms which we group together in rigid bodies referred to as clusters. Within a given cluster, the relative positions of the atoms are specified. An arbitrary cluster is then chosen as the base and covalently bonded clusters are assigned as children. This process is repeated until all the clusters have been placed in the tree. A cluster tree decomposition appropriate for torsion-angle-only dynamics is depicted in Fig. 1.

The clusters are connected by "hinges" which allow motion of one cluster relative to its parent. These hinges permit those degrees of freedom appropriate to those internal coordinates which one wishes to allow. Hence, freely rotating and translating clusters with three or more atoms would have six degrees of freedom, while torsion-angle-only motion would be represented by a hinge with a single rotational degree of freedom. A concrete example of hinge coordinates is given in Section 2.1.

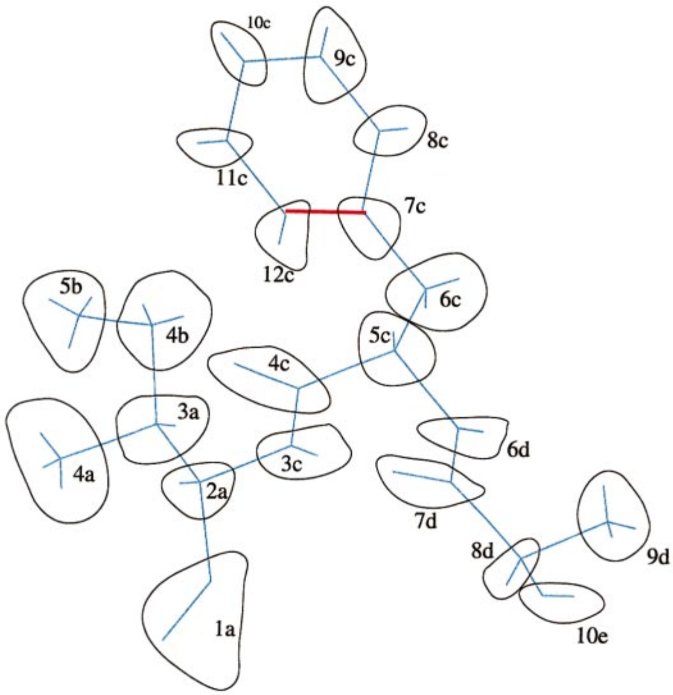

FIG. 1. The tree structure of a three-residue protein fragment decomposed into clusters appropriate for torsion-angle dynamics. The numbers represent the cluster level (distance from the base), while the letters denote the branch. The red bond in the ring of the phenyl group shows where a bond must be broken in order to impose the tree structure. However, in refinement calculations the relative position of the atoms in the phenyl group are known and thus are usually grouped into a single cluster.

In order to form the tree topology, one must disregard covalent bonds closing rings and loops such as those arising from disulfide bonds. These bonding relationships can be reasserted using one of several methods: an appropriate bonding potential energy term can be employed, the bond can be explicitly constrained in the dynamics, or, for small rings, the ring can be treated as a flexible (nonrigid) cluster (*11*), with only desired ring degrees of freedom active. The IVM allows for the first solution and also implements the second ring-closing technique, as described in Section 2.6.

In the tree structure, each cluster is identified by a pair of indices. The first identifies the cluster level and varies between 1 and N_t. We term the level 1 cluster the base, and those clusters at the ends of the branches the tips. The second index labels the particular branch at a given level. This label is not needed for unbranched structures and is usually omitted in this paper for clarity. Further notes on notation:

- Quantities with superscripts (o), (i), or (c) denote the appropriate initial quantity, value in internal coordinates, or value in Cartesian coordinates, respectively.
- R denotes a rotation matrix.

- $\mathbb{1}$ represents a unit matrix of appropriate dimension.
- Superscript T denotes the transpose operation.
- A tilde over a vector denotes the tensor associated with the vector's cross product, i.e., in matrix representation

$$\tilde{a} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}, \qquad [3]$$

such that $\tilde{a}b = a \times b$, for vectors a and b.

2.1. Example Hinge Coordinates

As a concrete example of the coordinates used we consider the branchless tree segment depicted in Fig. 2 with the positions of atom n in cluster k represented as

$$q_{k,n} = q_{k-1,0} + R_{k-1}\big(q^{(0)}_{k,0} - q^{(0)}_{k-1,0} + q^{(i)}_k\big) + R_k\big(q^{(0)}_{k,n} - q^{(0)}_{k,0}\big), \qquad [4]$$

with the rotation matrix $R_k = R_{k-1}R^{(i)}_k$. (Note that here the second subscript n denotes the atom within this cluster and not the branch.) Hinge k corresponds to the bond between atoms at positions $q_{k,0}$ and $q_{k-1,b}$. Displacements from the parent cluster, such as those due to a bond stretch, are represented by the relative internal position $q^{(i)}_k$, while internal rotations relative to the parent cluster are represented by the rotation matrix $R^{(i)}_k$.

Consider the example case in which the only allowed degrees of freedom between clusters k and $k-1$ are s_k, the displacement from equilibrium $s^{(0)}_k$ of the bond between atoms at positions $q_{k,0}$ and $q_{k-1,b}$, such that

$$q^{(i)}_k = \big(s_k + s^{(0)}_k\big)e^{(0)}_k, \qquad [5]$$

where e_k is the unit vector in the direction $q_{k,0} - q_{k-1,b}$ and τ_k, the torsion angle about this bond resulting in the rotation matrix about this bond, $R^{(i)}_k$, whose time derivative is given by

$$\dot{R}^{(i)}_k = \dot{\tau}\tilde{e}^{(0)}_k R^{(i)}_k. \qquad [6]$$

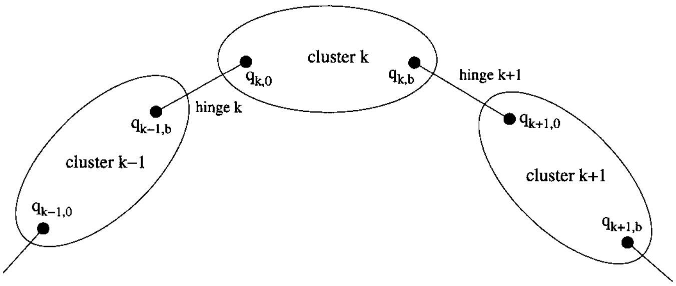

FIG. 2. Hierarchical hinge/cluster decomposition. As there are no branches in this example, only a single cluster index is used here.

Then the velocity of atom n in cluster k is

$$\begin{aligned}\dot{q}_{k,n} &= \dot{q}_{k-1,0} + \dot{R}_{k-1}\big(q^{(0)}_{k,0} - q^{(0)}_{k-1,0} + q^{(i)}_k\big) \\ &\quad + R_{k-1}\dot{q}^{(i)}_k + \dot{R}_k\big(q^{(0)}_{k,n} - q^{(0)}_{k,0}\big) \\ &= \dot{q}_{k-1,0} + \tilde{\omega}_{k-1}\big(q_{k,n} - q_{k-1,0}\big) + e_k\dot{s}_k \\ &\quad + R_{k-1}\tilde{e}^{(0)}_k R^{(i)}_k\big(q^{(0)}_{k,n} - q^{(0)}_{k,0}\big)\dot{\tau}_k, \end{aligned} \qquad [7]$$

where we have used the relationship $dR/dt = \tilde{\omega}R$, where R is rotation matrix describing a rotation relative to a fixed reference frame and ω is the associated angular velocity. Likewise, the time dependence of the rotation matrix obeys the following recursive relation

$$\dot{R}_k = \tilde{\omega}_k R_k = \tilde{\omega}_{k-1}R_k + R_{k-1}\tilde{e}^{(0)}R^{(i)}_k\dot{\tau}_k. \qquad [8]$$

If we define $v_k \equiv \dot{q}_{k,0,}$, then, from Eq. [7] we have

$$v_k = v_{k-1} + \tilde{\omega}_{k-1}(q_{k,0} - q_{k-1,0}) + e_k\dot{s}_k, \qquad [9]$$

and from Eq. [8] we can also write

$$\begin{aligned}\tilde{\omega}_k - \tilde{\omega}_{k-1} &= R_{k-1}\tilde{e}^{(0)}_k R^{(i)}_k R^T_k\dot{\tau}_k \\ &= R_{k-1}\tilde{e}^{(0)}_k R^T_{k-1}\dot{\tau}_k \\ &= \tilde{e}_k\dot{\tau}_k, \end{aligned} \qquad [10]$$

rewritten as

$$\tilde{\omega}_k = \tilde{\omega}_{k-1} + \tilde{e}_k\dot{\tau}_k. \qquad [11]$$

Equations [9] and [11] allow one to determine the linear and angular velocities of cluster k recursively from the internal velocities $\dot{s}$ and $\dot{\tau}$ and from its parent's linear and angular velocities. The general recursive equations for angular and linear velocities are given in the next section.

2.2. General Recursive Expression for Spatial Velocity

The spatial velocity of cluster k is defined (*12*) as the block vector

$$V_k = \begin{pmatrix} \omega_k \\ v_k \end{pmatrix}. \qquad [12]$$

Equations [9] and [11] can be generalized in the following recursive expression for the spatial velocity of the kth cluster

$$V_k = \begin{pmatrix} \omega_k \\ v_k \end{pmatrix} = \phi^T_{k,k-1}V_{k-1} + H^T_k\dot{\theta}_k. \qquad [13]$$

Here

$$\phi_{k,k-1} = \begin{pmatrix} \mathbb{1} & \tilde{q}_k - \tilde{q}_{k-1} \\ 0 & \mathbb{1} \end{pmatrix}, \qquad H_k = \begin{pmatrix} h_k^{(a)} & 0 \\ 0 & h_k^{(t)} \end{pmatrix}, \tag{14}$$

where $h_k^{(a)}$ and $h_k^{(t)}$ are matrices, the rows of which respectively denote the angular and translational degrees of internal freedom of hinge k. These directions should be normalized, such that $H_k H_k^T = \mathbb{1}$. Typically, the base cluster ($k = 1$) will be allowed all degrees of rotational and translational freedom ($H = \mathbb{1}$). The vector θ_k contains all of the allowed internal degrees of freedom of hinge k.

In practice, the coordinates used for integration can be different from those in which the angular velocity is expressed. For instance, it is convenient to use the four Euler parameters (quaternion representation) to describe the orientation of a rigid body, while only three angular velocities are expressed in the equations of motion. For simplicity, we ignore this subtlety in our current notation.

Time differentiation of Eq. [13] leads to the recursive equation for the spatial acceleration

$$\begin{aligned} \alpha_k \equiv \dot{V}_k &= \dot{\phi}_{k,k-1}^T V_{k-1} + \phi_{k,k-1}^T \alpha_{k-1} + \dot{H}_k^T \dot{\theta}_k + H_k^T \ddot{\theta}_k \\ &= \phi_{k,k-1}^T \alpha_{k-1} + H_k^T \ddot{\theta}_k + a_k, \end{aligned} \tag{15}$$

where a_k is the kth cluster's Coriolis acceleration

$$a_k = \begin{pmatrix} 0 \\ \tilde{\omega}_k (v_k - v_{k-1}) \end{pmatrix} + \begin{pmatrix} \tilde{\omega}_k & 0 \\ 0 & \tilde{\omega}_k \end{pmatrix} H_k^T \dot{\theta}_k. \tag{16}$$

2.3. Forces and the Equations of Motion

The force on atom n in cluster k is

$$f_{k,n}^{(c)} = -\nabla_{q_{k,n}} \mathsf{V}, \tag{17}$$

where V is the total potential energy. Recall that, within a cluster, atomic velocities are given by

$$\dot{q}_{k,n} = v_k + \tilde{\omega}_k (q_{k,n} - q_{k,0}). \tag{18}$$

Time differentiation gives the rigid-body atomic accelerations which are substituted into Newton's equation of motion for cluster k to give the equations of motion

$$\begin{aligned} f_k^{(c)} \equiv \sum_n f_{k,n}^{(c)} &= \sum_n m_{k,n} \ddot{q}_{k,n} \\ &= m_k \dot{v}_k + m_k \tilde{\omega}_k \tilde{\omega}_k (q_{k,c} - q_{k,0}), \end{aligned} \tag{19}$$

where $q_{k,c}$ is the position of the cluster's center of mass and m_k is the cluster mass. The moment about point $q_{k,0}$ is denoted $N_k^{(c)}$,

$$N_k^{(c)} \equiv \sum_n (\tilde{q}_{k,n} - \tilde{q}_{k,0})\, f_{k,n}^{(c)}. \tag{20}$$

Again replacing the forces using $f_{k,n}^{(c)} = m_{k,n} \ddot{q}_{k,n}$, and using accelerations appropriate for a rigid body yields the equation for moment in terms of rigid body velocities and accelerations

$$N_k^{(c)} = m_k (\tilde{q}_{k,c} - \tilde{q}_{k,0}) \dot{v} + I_k \dot{\omega}_k + \tilde{\omega}_k I_k \omega_k, \tag{21}$$

where I_k is the inertia tensor about $q_{k,0}$:

$$I_k = \sum_n m_{k,n} [|q_{k,n} - q_{k,0}|^2 \mathbb{1} - (q_{k,n} - q_{k,0})(q_{k,n} - q_{k,0})^T]. \tag{22}$$

Thus, the spatial equations of motion in the absence of hinges can then be written as

$$F_k^{(c)} = M_k \alpha_k + b_k, \tag{23}$$

where

$$F_k^{(c)} = \begin{pmatrix} N_k^{(c)} \\ f_k^{(c)} \end{pmatrix} \tag{24}$$

$$b_k = \begin{pmatrix} \tilde{\omega}_k I_k \omega_k \\ m_k \tilde{\omega}_k \tilde{\omega}_k (q_{k,c} - q_{k,0}) \end{pmatrix} \tag{25}$$

$$M_k = \begin{pmatrix} I_k & m_k (\tilde{q}_{k,c} - \tilde{q}_{k,0}) \\ -m_k (\tilde{q}_{k,c} - \tilde{q}_{k,0}) & m_k \mathbb{1} \end{pmatrix}. \tag{26}$$

Equation [23] is a form of the Newton–Euler equation (12).

Now, a hinge couples adjacent clusters by means of equal and opposite force on the two clusters. At the position of $q_{k,0}$ we write the spatial force which parent cluster $k - 1$ exerts on cluster k as F_k, and the spatial force from daughter cluster $k + 1$ as $-\phi_{k+1,k} F_{k+1}$. If a cluster has more than one daughter, the force is the sum of such terms.

We can then write the total force on cluster k as the sum of hinge forces and external forces, and thus the spatial equation of motion for cluster k within a tree is written as

$$F_k - \phi_{k+1,k} F_{k+1} + F_k^{(c)} = M_k \alpha_k + b_k. \tag{27}$$

Now, the atom-based forces can be expressed in terms of internal coordinates as

$$\begin{aligned} T_k^{(c)} &= -\nabla_{\theta_k} \mathsf{V}^{(c)} \\ &= H_k \sum_{k'} \phi_{k,k'} F_{k'}^{(c)}, \end{aligned} \tag{28}$$

where

$$\phi_{k,k'} = \begin{cases} \phi_{k'+1,k'} \dots \phi_{k,k-1} & k > k' \\ \mathbb{1} & k = k'. \\ 0 & k < k' \end{cases} \quad [29]$$

Equation [28] is obtained from the definitions of the internal coordinates and the definition of $F_k^{(c)}$. But we can write the potential energy as a sum of terms expressed in internal and atomic Cartesian coordinates, $\mathsf{V}^{(i)}$ and $\mathsf{V}^{(c)}$, respectively:

$$\mathsf{V} = \mathsf{V}^{(i)}(\theta_1, \dots, \theta_{N_t}) + \mathsf{V}^{(c)}. \quad [30]$$

F_k can be decomposed into a component acting in the constrained degrees of freedom to enforce constraints, and a component in the allowed degrees of freedom arising from energy terms which depend explicitly on the appropriate internal coordinate. This later force component is projected out as

$$T_k^{(i)} \equiv H_k F_k. \quad [31]$$

It is seen that the projection of the hinge force in the allowed degrees of freedom is precisely the force in those degrees of freedom due to potential terms which explicitly depend on these coordinates:

$$T_k^{(i)} = -\nabla_{\theta_k} \mathsf{V}^{(i)}. \quad [32]$$

Thus, we have the option of expressing individual potential (and force) function terms in either internal or atomic coordinates, whichever are most convenient.

Finally, Eq. [27] can be regarded as a recursion relation that is complimentary to that for acceleration

$$F_k = \phi_{k+1,k} F_{k+1} - F_k^{(c)} + M_k \alpha_k + b_k, \quad [33]$$

where one determines the hinge forces starting from the tip and working toward the base.

2.4. Spatial Operator Notation and the Internal Coordinate, Mass Matrix

If cluster $k = 0$ is chosen to be at rest, Eq. [13] yields

$$\begin{aligned} V_0 &= 0 \\ V_1 &= H_1^T \dot\theta_1 \\ V_2 &= \phi_{21}^T V_1 + H_2^T \dot\theta_2 = \phi_{21}^T H_1^T \dot\theta_1 + H_2^T \dot\theta_2 \\ V_3 &= \phi_{32}^T V_2 + H_3^T \dot\theta_3 = \phi_{31}^T H_1^T \dot\theta_1 + \phi_{32}^T H_2^T \dot\theta_2 + H_3^T \dot\theta_3, \end{aligned} \quad [34]$$

and so on. These equations are rewritten in spatial operator notation (*1*, *2*) as

$$V \equiv \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ \vdots \end{pmatrix} = \phi^T H^T \dot\theta, \quad [35]$$

where

$$\phi = \begin{pmatrix} \mathbb{1} & \phi_{21} & \phi_{31} & \cdots \\ 0 & \mathbb{1} & \phi_{32} & \cdots \\ 0 & 0 & \mathbb{1} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad [36]$$

$$H = \begin{pmatrix} H_1 & 0 & 0 & \cdots \\ 0 & H_2 & 0 & \cdots \\ 0 & 0 & H_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad [37]$$

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \end{pmatrix}. \quad [38]$$

Likewise,

$$\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \end{pmatrix} = \phi^T H^T \ddot\theta + a, \quad [39]$$

with

$$a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \end{pmatrix}. \quad [40]$$

The representation of the Cartesian force in internal coordinates becomes

$$T^{(c)} = H\phi F^{(c)}. \quad [41]$$

The recursion relation Eq. [33] for the hinge force is rewritten as

$$F = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \end{pmatrix} = \phi\big[M\alpha + b - F^{(c)}\big], \qquad [42]$$

with the block-diagonal spatial operator mass matrix defined as

$$M = \begin{pmatrix} M_1 & 0 & 0 & \cdots \\ 0 & M_2 & 0 & \cdots \\ 0 & 0 & M_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \qquad [43]$$

Finally, we use the operator counterpart of Eq. [31] combined with Eqs. [39] and [42] to obtain the equations of motion in internal coordinates as

$$HF = T^{(i)} = \mathcal{M}\ddot{\theta} + \mathcal{C} \qquad [44]$$

with the internal variable mass matrix and Coriolis terms defined respectively as

$$\mathcal{M} = H\phi M\phi^T H^T, \qquad [45]$$

and

$$\mathcal{C} = H\phi\big[Ma + b - F^{(c)}\big]. \qquad [46]$$

Note that $\mathcal{M}$ is a nondiagonal matrix whose elements change with time (ϕ and H depend on cluster position). From this point, one then can use the recursive algorithm described below to solve for $\ddot{\theta}$ with computational effort proportional to the number of clusters.

For a branched molecule, the spatial equations take exactly the same form as Eqs. [35]–[46], where now all clusters with the same level have the same index k. The components of the spatial operator quantities themselves become block vectors and matrices. For example, if there are two branches at the kth level, the internal velocity vector and the $\phi_{k,k-1}$ matrix respectively become

$$V_k \rightarrow \begin{pmatrix} V_{ka} \\ V_{kb} \end{pmatrix} \qquad [47]$$

and

$$\phi_{k,k-1} \rightarrow \begin{cases} \begin{pmatrix} \mathbb{1} & \tilde{l}_{ka} & 0 & 0 \\ 0 & \mathbb{1} & 0 & 0 \\ 0 & 0 & \mathbb{1} & \tilde{l}_{kb} \\ 0 & 0 & 0 & \mathbb{1} \end{pmatrix} & \text{if clusters } ka \text{ and } kb \text{ have distinct parents} \\[2ex] \begin{pmatrix} \mathbb{1} & \tilde{l}_{ka} \\ 0 & \mathbb{1} \\ \mathbb{1} & \tilde{l}_{kb} \\ 0 & \mathbb{1} \end{pmatrix} & \text{if clusters } ka \text{ and } kb \text{ have a common parent,} \end{cases} \qquad [48]$$

where $l_{ka} = q_{ka,0} - q_{k-1,0}$. Thus, the generalization of the formulae from unbranched to branched structures is straightforward.

2.5. Recursive Solution of the Equations of Motion

Jain *et al.* (*1*, *2*) have formulated a recursive methodology for solving Eq. [2] for the internal coordinate accelerations. For completeness, we include the algorithm here. For proof of its correctness, see (*1*, *2*).

Introduce these ancillary quantities for each cluster k

$$P_k = \phi_{k+1,k} P_k^+ \phi_{k+1,k}^T + M_k \qquad [49a]$$

$$D_k = H_k P_k H_k^T \qquad [49b]$$

$$G_k = P_k H_k^T D_k^{-1} \qquad [49c]$$

$$P_k^+ = (\mathbb{1} - G_k H_k) P_k \qquad [49d]$$

$$z_k = \phi_{k+1,k} z_{k+1}^+ + P_k \alpha_k + b_k - F_k^{(c)} \qquad [49e]$$

$$\epsilon_k = T_k - H_k z_k \qquad [49f]$$

$$\nu_k = D_k^{-1} \epsilon_k \qquad [49g]$$

$$z_k^+ = z_k + G_k \epsilon_k, \qquad [49h]$$

where the recursion sweeps from the tip to the base with the boundary conditions at the tip

$$P_{N_t}^+ = 0, \quad z_{N_t}^+ = 0. \qquad [50]$$

Then, the updated accelerations are calculated by sweeping from base to tip with the recursion relations

$$\alpha_k^+ = \phi_{k,k-1}^T \alpha_{k-1} \qquad [51a]$$

$$\ddot{\theta}_k = \nu_k - G_k^T \alpha_k^+ \qquad [51b]$$

$$\alpha_k = \alpha_k^+ + H_k^T \ddot{\theta}_k + a_k, \qquad [51c]$$

with $\alpha_0^+ = 0$.

Calculation of the equations of motion entails three sweeps over each molecule. In the first sweep from base to tip Eq. [13]

is used to compute spatial velocities. Second is a sweep from tip to base in which the quantities in Eq. [49] are computed. Finally, accelerations are computed using Eqs. [51]. The most computationally expensive step is the second sweep due to the computation of matrix quantities. Typically, Eq. [49a] is the cycle-limiting step resulting in compute time scaling linearly with the number of atom clusters.

2.6. Loop Constraints

Loop topologies such as those caused by disulfide bonds in proteins and by sugar rings in nucleic acids are broken in the tree decomposition of the molecules into rigid clusters. One approach to reimpose the broken bonds is to apply correction forces such that the cluster accelerations are consistent with the bond constraints.

To address the problem of applying bond-length constraints, we consider the situation of two independent trees with the constraint of a single bond between them. This artificial system contains no loop topologies, but allows a simple formulation of the bond-constraint problem and is sufficiently general for the current discussion.

Following (*13*), we write the constraint the bond constraint C as

$$C : |q_{1t} - q_{2t}| - c = 0, \qquad [52]$$

where q_{1t} and q_{2t} are the positions of atoms in two tip clusters in trees 1 and 2, respectively, and c is the associated nominal bond length. Differentiating constraint C twice with respect to time results in the following relation between atomic positions, velocities, and accelerations:

$$(\ddot{q}_{1t} - \ddot{q}_{2t}) \cdot (q_{1t} - q_{2t}) + |\dot{q}_{1t} - \dot{q}_{2t}|^2 = 0. \qquad [53]$$

We enforce the constraint by means of an equal and opposite force between the two atoms

$$f_{1t} = -f_{2t} = \lambda(q_{1t} - q_{2t}), \qquad [54]$$

where λ remains to be determined. For simplicity, we consider the case of two independent trees, of which one contains the tip atom at position q_{1t} and the other contains an atom at position q_{2t}. The internal coordinates of the two trees are described by θ_1 and θ_2, respectively. In analogy to Eq. [41], the force on the tip atoms can be represented in internal coordinates as

$$-J_1^T f_{1t}, \quad J_2^T f_{2t}, \qquad [55]$$

respectively, where

$$J_1^T = H_1 \phi_1 B_1, \qquad [56]$$

and B_1 projects out the Cartesian velocity of the constraint atom on tip 1; i.e.,

$$v_{1t} = B_1^T V_1 = J_1 \dot{\theta}_1. \qquad [57]$$

Then the equations of motion for the chains become

$$\mathcal{M}_1 \ddot{\theta}_1 + \mathcal{C}_1 = T_1^{(i)} - J_1^T \lambda(q_{1t} - q_{2t}) \qquad [58a]$$

$$\mathcal{M}_2 \ddot{\theta}_2 + \mathcal{C}_2 = T_2^{(i)} - J_2^T \lambda(q_{1t} - q_{2t}). \qquad [58b]$$

The internal coordinate acceleration of each chain is broken into two parts

$$\ddot{\theta}_1 = \ddot{\theta}_{1f} + \Delta\ddot{\theta}_1, \qquad [59]$$

where $\ddot{\theta}_{1f}$ is the acceleration in the absence of the constraint,

$$\ddot{\theta}_{1f} = \mathcal{M}_1^{-1}\big(T_1^{(i)} - \mathcal{C}_1\big), \qquad [60]$$

and $\Delta\ddot{\theta}_1$ is the correction so that C is obeyed:

$$\Delta\ddot{\theta}_1 = -\mathcal{M}_1^{-1} J_1^T \lambda(q_{1t} - q_{2t}). \qquad [61]$$

Likewise, the atomic accelerations become

$$\ddot{q}_{1t} = \ddot{q}_{1tf} + \Delta\ddot{q}_{1t}, \qquad [62]$$

with

$$\Delta\ddot{q}_{1t} = J_1 \Delta\ddot{\theta}_1. \qquad [63]$$

Therefore,

$$\ddot{q}_{1t} = \ddot{q}_{1tf} + J_1 \Delta\ddot{\theta}_1 \qquad [64a]$$

$$= \ddot{q}_{1tf} - J_1 \mathcal{M}_1^{-1} J_1^T \lambda(q_{1t} - q_{2t}), \qquad [64b]$$

and

$$\ddot{q}_{2t} = \ddot{q}_{2tf} + J_2 \mathcal{M}_2^{-1} J_2^T \lambda(q_{1t} - q_{2t}). \qquad [65]$$

Combining these expressions for tip atom accelerations with Eq. [53] results in an equation in which λ is the only unknown. This equation can be solved for λ:

$$\lambda = \big[(q_{1t} - q_{2t})^T \big(J_1 \mathcal{M}_1^{-1} J_1^T + J_2 \mathcal{M}_2^{-1} J_2^T\big)(q_{1t} - q_{2t})\big]^{-1} \times \big[(\ddot{q}_{1tf} - \ddot{q}_{2tf})^T (q_{1t} - q_{2t}) + |\dot{q}_{1t} - \dot{q}_{2t}|^2\big]. \qquad [66]$$

In this equation the free accelerations have been calculated using the recursive algorithm, while the quantities $J_1 \mathcal{M}_1^{-1} J_1^T$ are also calculable by a recursive algorithm (*2, 13*).

During integration with finite timestep size, there will be drift in the positions and velocities such that they violate the constraint condition C. Thus, it is desirable to periodically reenforce C with an approach which minimizes the violation of conservation of energy. Such a procedure has been implemented in an efficient fashion.

Unfortunately, this procedure becomes expensive when there are multiple constrained loop closures, as in nucleic acids: the λ_l's for each loop are coupled together and computation becomes dominated by the computation of the cross terms $J_l \mathcal{M}^{-1} J_{l'}^T$. This problem might be alleviated to some extent by the Lagrange multiplier approach of (*14*). Yet another approach would be to choose the coefficients λ_l such that the associated constraint equations [52] are exactly obeyed at each timestep, as in SHAKE (*15, 16*).

More serious, however, is that in TAMD, the above approach employs the incorrect coordinates for the sugar rings. The appropriate coordinates for the rings are ring pucker coordinates (*17*). However, pucker motion involves a combination of torsion angles and bond angles, the latter not being available in TAMD. One possible solution to this problem within the IVM is to free the bond angles to allow them to change in concert with the torsion angles in dynamics calculations.

3. IMPLEMENTATION

3.1. Integrator

Traditionally, molecular dynamics codes would use the velocity Verlet algorithm (*18*) for reasons of computational simplicity, low memory storage requirements, and its efficiency and stability. In internal coordinates, the acceleration depends on the velocity, requiring that the velocity Verlet algorithm be modified. This fact and the extra computational effort required to solve the internal variable equations of motion lead one to rethink the choice of integration algorithm. The original TAMD implementation in X-PLOR (*5, 6*) utilized the Runge–Kutta algorithm, but this has the distinct disadvantage of requiring three force evaluations at each timestep. The program Dyana (*7*) employed a modified Verlet algorithm to account for the velocity-dependent forces. Our code is built in a modular fashion, so that it has been straightforward to implement a fourth-order Runge–Kutta algorithm (*19*) and a modified Verlet algorithm in addition to a sixth-order predictor-corrector integrator (PC6) (*20*), which also requires only one force evaluation per timestep.

In the sixth-order predictor corrector used here, the vector of internal coordinates and its first five scaled time derivatives at time t are denoted

$$\theta^{(n)}(t) = \frac{\Delta t^n}{n!} \frac{d^n \theta(t)}{dt^n}, \quad n = 0 \ldots 5. \qquad [67]$$

If $q^T = (\theta^{(0)^T}, \ldots, \theta^{(5)^T})$, then the prediction step becomes

$$q^{(p)}(t + \Delta t) = Wq(t), \qquad [68]$$

where

$$W = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 3 & 6 & 10 \\ 0 & 0 & 0 & 1 & 4 & 10 \\ 0 & 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \qquad [69]$$

Using the updated positions and velocities, the scaled acceleration $a = \Delta t^2 \ddot{\theta}/2$ is then calculated using the recursive algorithm, and the state vector is then corrected

$$q(t + \Delta t) = q^{(p)}(t + \Delta t) + U \Delta a\ , \qquad [70]$$

where $\Delta a = a - a^{(p)}$ and

$$U = \begin{pmatrix} 3/16\mathbb{1} \\ 251/360\mathbb{1} \\ \mathbb{1} \\ 11/18\mathbb{1} \\ 1/6\mathbb{1} \\ 1/60\mathbb{1} \end{pmatrix}. \qquad [71]$$

In our experience the PC6 integrator was found to be the most efficient: it allowed larger timesteps than those in the Verlet algorithm and required but one force evaluation per timestep. Figure 3 displays the energy error per timestep for three

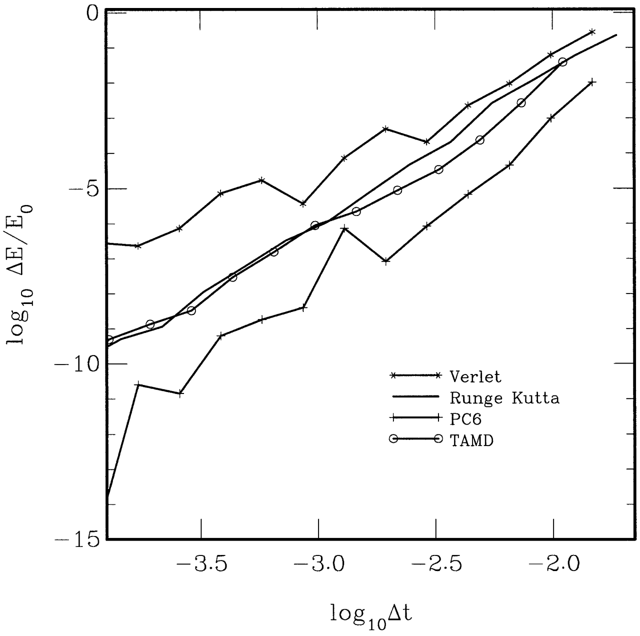

FIG. 3. The error in energy as a function of step size for three IVM integrators and the TAMD routine native to X-PLOR. Note that the IVM Runge–Kutta and native TAMD step sizes have been divided by 4 and 3, respectively, to reflect the number of force evaluations at each timestep.

integration schemes for the torsion-angle dynamics of the 57 residue B1 domain of protein G (*21*). Also shown in the figure is the result for the TAMD routine native to X-PLOR version 3.851, which employs a Runge–Kutta integrator. The error is defined as

$$\delta E = \langle (E_i - \langle E \rangle)^2 \rangle / \langle E \rangle, \qquad [72]$$

where E_i is the energy at timestep i and the angle brackets denote average over timesteps. In the figure, the timestep size of the Runge–Kutta results was divided by 4 and the IVM and native TAMD results were divided by 3 to reflect the fact that they take that many force evaluations each timestep. Using the ΔE metric, one sees that the two Runge–Kutta algorithms perform approximately equivalently and slightly better than the Verlet algorithm. Figure 3 clearly shows the performance advantage for the PC6 approach. The PC6 protocol requires more storage and computation than the Verlet algorithm, but this is mitigated by the overhead of the recursive algorithm. Example timings on our platforms suggest that the CPU overhead of the PC6 versus the Verlet is a few percent of run time. This efficiency penalty is readily offset by a factor of 3 or so increase in step size allowed by the PC6 algorithm. However, for a given set of internal coordinates (i.e., not torsion angles), a different integrator may be more appropriate. The Verlet and Runge–Kutta routines are available in the IVM package, and new algorithms can be implemented in a straightforward fashion.

3.1.1. Self-Adjusting Timestep

The native integrators in the X-PLOR program require that one specify the size of the timestep. As an alternative, one can specify an error tolerance in the total energy and the timestep is then appropriately adjusted to meet this tolerance. We employed the approach used in Dyana (*7*) in which the timestep is scaled by the ad hoc factor

$$\sqrt{1 + \frac{\Delta E_0 - \Delta E}{\tau \Delta E}}, \qquad [73]$$

where ΔE_0 and ΔE are the target and observed energy errors, respectively, and τ is the response time in units of the molecular dynamics timestep size. In addition, a step is thrown out and the step size halved if the energy error is greater than a threshold (typically 10% of the total energy).

We implemented both fixed and implicit timestep approaches and found that the latter is more flexible and convenient, particularly when using the same simulated annealing protocol with different hinge definitions; for instance, the same protocol may be used for torsion angle and Cartesian space dynamics. The error tolerance is made a set fraction of the bath temperature and then the annealing protocols can be identical, with a larger timestep automatically chosen for the torsion-angle-only dynamics.

3.1.2. Constant Temperature Dynamics

Constant temperature dynamics is desirable for the simulated annealing optimization used in structure refinement. We have implemented coupling of the simulated system to a temperature bath using two rudimentary approaches appropriate for structure determination dynamics: using velocity rescaling and using a velocity-dependent force (*22*). Here we describe the velocity-scaling approach as it is the more convenient when using automatic timestep selection. At each step, all of the internal velocities are scaled by

$$\sqrt{1 + \frac{T_0 - T}{\tau T}}, \qquad [74]$$

where T and T_0 are the system and bath temperatures, respectively, and τ is the response time in units of the molecular dynamics timestep size.

Some care must be taken in determining the total energy in the presence of a bath, since the timestep depends upon it. We used the approach employed in Dyana (*7*) in which temperature coupling was achieved by velocity rescaling *after* the step was taken to evaluate the error in the system energy.

More appropriate for molecular dynamics simulations attempting to reproduce a canonical ensemble would be a thermostat using Nosé–Hoover chains (*23, 24*). Use of the Nosé–Hoover approach would yield more accurate dynamics and would still allow the current automatic timestep adjustment algorithm due to the fact that there is a conserved total energy. However, this thermostat has not yet been implemented in the current framework.

3.1.3. Startup

The issue of dynamics initialization becomes important particularly when many short dynamics runs are required, as is usually the case in simulated annealing protocols for NMR structure determination and refinement. There are a number of tasks that must be accomplished upon startup. These tasks are itemized along with the solutions we employed to achieve acceptable performance:

- The internal coordinates must be defined. For torsion-angle dynamics, we have automated the generation of the appropriate hinge definitions.
- The internal coordinates must be determined such that they are consistent with a given set of Cartesian coordinates. Given a defined tree topology, this mapping is unique and scales linearly with the number of atoms.
- Internal velocities must be generated that are as consistent as possible with the given atomic velocities. Because the atomic velocities are free to display nonallowed motion within a fixed cluster and other motion not allowed by the hinge definitions, this mapping is not one-to-one. We pose the process as an

optimization problem with the objective functional

$$\frac{1}{2}\left(V_0^{(c)} - V^{(c)}\right)^T M^{(c)}\left(V_0^{(c)} - V^{(c)}\right), \qquad [75]$$

where $V_0^{(c)}$ is a vector of the given atomic velocities and $V^{(c)}$ are the atom velocities determined from a given set of internal coordinate velocities θ. We have chosen to scale the objective function by the atomic masses, represented in the diagonal atomic mass matrix $M^{(c)}$.

The atomic velocities are linearly related to the internal coordinate velocities by

$$V^{(c)} = J\dot{\theta}, \qquad [76]$$

with

$$J^T = H\phi B, \qquad [77]$$

and with B^T the rectangular matrix which converts cluster spatial velocities to atomic velocities. The internal velocities that minimize Eq. [75] are then

$$\dot{\theta} = \left(J^T M^{(c)} J\right)^{-1} J^T M^{(c)} V_0^{(c)}. \qquad [78]$$

Now, the effort for solving this equation scales as the cube of the number of internal coordinates if we use a brute-force approach. Clearly, we would like to avoid this cost if possible. Fortunately, this equation has the same form as that for the cluster equations of motion, Eq. [2], with the matrix

$$J^T M^{(c)} J = H\phi B M^{(c)} B^T \phi^T H^T \qquad [79]$$

having the same form as the mass matrix in internal coordinates Eq. [45], so that we can reuse the algorithm from Section 2.5 if these substitutions are made

$$M \rightarrow BM^{(c)}B^T \qquad [80a]$$

$$\ddot{\theta} \rightarrow \dot{\theta} \qquad [80b]$$

$$T \rightarrow J^T M^{(c)} V_0^{(c)}, \qquad [80c]$$

and we set $a = b = f^{(c)} = 0$. This allows the velocities of the internal coordinates to be determined with effort proportional to the number of atom clusters.

- The integrator must be initialized. The PC6 integrator requires no extra steps at initialization if one is willing to settle for decreased accuracy during the first few steps: the initial values of $d^n\theta/dt^n$ are set to zero for $n = 3 \ldots 5$. This approximation was found to be adequate in the current applications of the IVM, but a different approach may be desirable in other circumstances.

3.2. Local Minimization

In local minimization, the mass matrix is not involved, so that the algorithm of Section 2.5 is not necessary. However, it was deemed convenient to provide a local minimization facility to augment the integrator. For one, this approach allows for local minimization after a dynamics run using the same coordinates. Also, this addition removes the need for any separate rigid-body minimization routine.

Again, the minimizer was implemented using a modular design so that multiple algorithms can be implemented with ease. Currently, the most efficient algorithm implemented is a Powell method derived from the same IMSL code which was used in X-PLOR (*25*).

To obtain the gradient in internal coordinates, $T = T^{(c)} + T^{(i)}$, Eq. [41] must be solved. This can be accomplished using the recursive formula

$$z_k = F_k^{(c)} + \phi_{k+1,k} z_{k+1} \qquad [81a]$$

$$T_k^{(c)} = H_k z_k, \qquad [81b]$$

solved from tip to base, with $z_k = F_k^{(c)}$ for the tip clusters.

3.3. Hinge Types Implemented Thus Far

As of this writing, hinges have been implemented which allow the following categories of motion:

- Rotations appropriate for rigid bodies consisting of one, two, or three-plus atoms. Euler parameters (*26*) are used for acceleration calculations, while XYZ Euler angles (*27*) are used to calculate gradients with respect to the internal coordinates.
- Rotations plus translations appropriate for rigid bodies of one or more atoms.
- Torsion-angle-only motion.
- Translation-only motion. Rigid body motion without rotations.

It is also simple to fix atoms in space by placing them in a fixed inertial cluster. We note that adding new hinge definitions is a straightforward process.

3.4. Programming Considerations

This IVM was written in the C++ programming language, an appropriate compromise between performance and ease of development and maintenance. Since we wish this module to be included in multiple software packages, careful attention was paid to both modularity and to performance. For example, so that new hinge definitions would be easy to implement, runtime polymorphism was used in the description of the cluster-plus-parent hinge object. The Hinge Node virtual base class encapsulates the behavior of a generic hinge-plus-cluster object, while it is specialized for each specific type of hinge motion, be it torsion angle, full translation, translation-plus-rotation, etc. Attention was paid so that virtual class methods were not used on too fine a granularity because of performance considerations (the calls are indirect and cannot be inlined). Another example of the balance between performance and convenience can be found in the two-vector template classes employed in the IVM. One class was used for arrays, whose size can vary from one invocation to the next, such as those that contain the internal coordinate positions and velocities, and another was used for vectors of a size (generally small) fixed at compile time. This latter class was deemed desirable to obviate the need for heap storage and size information for many small vectors and it facilitates added compile-time optimization opportunity because of the fixed loop sizes.

Despite the attention paid to coding detail here, the code certainly suffers a runtime performance deficit relative to the equivalent code written in Fortran because of the relative compiler maturity and language-level features such as Fortran's no-alias guarantee. However, we are pleased with our choice of C++ as it allows rapid development, ease of code readability, and ease of upkeep/modification. Finally, we find the current performance quite adequate.

4. EXAMPLES OF USE FOR NMR STRUCTURE DETERMINATION AND REFINEMENT

We are actively using the IVM presented in this paper for refining NMR structures within the NIH X-PLOR implementation. We present two examples of use below.

4.1. Protein G: Torsion-Angle Dynamics

Here we present an example of NMR structure determination of the B1 domain of protein G (*21*) to compare the IVM with the TAMD functionality native to X-PLOR version 3.851 (*5, 6*) and to demonstrate the power of the variable timestep functionality in the IVM. In the calculation, experimental nuclear Overhauser effect (NOE) derived interproton distance restraint terms and dihedral restraint terms were employed in addition to potential terms used to enforce reasonable covalent geometry and atom–atom separation. All atomic masses were set to 100 AMU.

The starting coordinates consisted of an extended structure, and the annealing protocol employed is summarized as follows:

- Initial all-atom Cartesian coordinate Powell minimization using only bond, angle, and improper energy terms.
- High temperature torsional angle dynamics at 2000 K for 1000 steps with all potential terms included except atom–atom repulsion.
- A dynamics loop in which atom–atom repulsion is added in 10 increments of 100 steps each.
- A cooling loop in which the temperature is reduced from 2000 to 100 K in increments of 25 K—with 39 dynamics steps taken at each temperature.
- Final all-atom Powell minimization.

All dynamics calculations were performed with a timestep of 10 fs in the internal variable coordinate space consisting of 321 torsion angles, with all aromatic rings grouped into rigid clusters. Coupling to a temperature bath was achieved by means of the velocity-dependent force method. Numerical results were obtained from an executable generated by SGI Irix compilers at full optimization.

As mentioned previously, the native TAMD procedure employs a Runge–Kutta integration scheme which requires three force evaluations per timestep, while the PC6 algorithm requires but one force evaluation per timestep. The resulting run time of the IVM-based protocol was approximately half that of the native TAMD routine, which is somewhat larger than that expected purely on the number of force evaluations. We believe that the extra overhead is due to the greater generality of the IVM, and its use of C++ as opposed to the Fortran in which the native TAMD routine was written.

We then made the following change to the protocol: we replaced the fixed length timesteps in the dynamics calculations with timesteps that varied in the fashion described in Section 3.1.1. The scale factor of Eq. [73] for the timestep was chosen using the target energy error ΔE_0 set to 0.001 kcal/K times the bath temperature, and the system-bath coupling was changed to the velocity scaling method. The number of timesteps was allowed to vary in order to complete fixed-length-in-time dynamics runs: 3 ps for the high temperature equilibration, 0.3 ps for each of the 10 dynamics simulations in the atom–atom repulsion loop, and 0.78 ps for each temperature value in the cooling loop. These values for the total integration times in the variable-time protocol were chosen so that the first two dynamics components had the same values as for the fixed-time protocols, while the integration time of each dynamics run within the cooling loop was twice that of the fixed-time protocols.

In Fig. 4 the refinement energy and root-mean-square deviation (RMSD) from the lowest energy structure are shown for 10 structures calculated by each of the three refinement protocols. The fixed timestep, PC6 method resulted in one structure with high energy and an RMSD value larger than 4 Å. Hence, this structure is not visible in the figure. One can see that the fixed timestep IVM protocol produces fewer good structures than does the native TAMD routine, with the former producing only two structures in the cluster of low energy and low RMSD structures. This is as we would expect due to the fact that for equal step size, the Runge–Kutta algorithm has better accuracy than the predictor-corrector algorithm.

On the other hand Fig. 4 shows that the variable timestep protocol obtains results of quality equal to the TAMD protocol. Moreover, the run time for the variable timestep protocol is about half that of the fixed timestep IVM protocol, and it is four times faster than the TAMD protocol. This reduction in run time is manifest even though the total time of integration was significantly larger (about 80 ps for the variable timestep protocol versus approximately 50 ps for the fixed

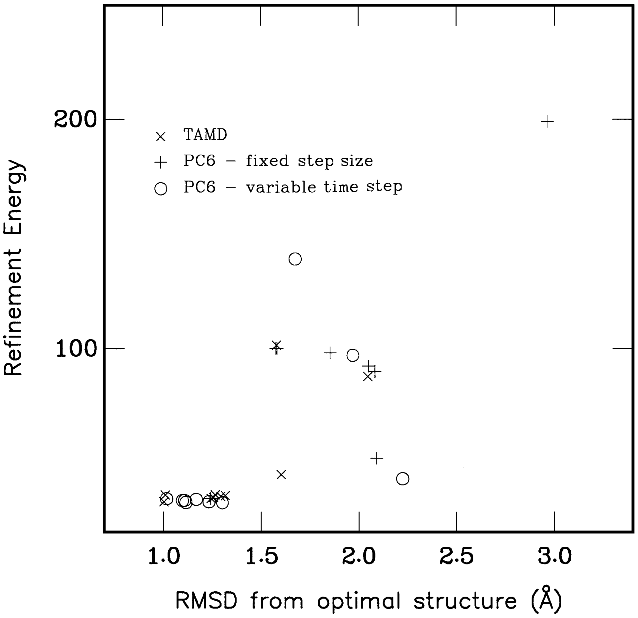

FIG. 4. Refinement energy vs RMS deviation from the minimum energy structure for refinement using three integration methods: the native TAMD integrator, PC6 integrator with fixed step size, and PC6 integrator with the variable timestep feature enabled. The fixed timestep PC6 methods resulted in one structure with high energy and RMSD values larger than 4 Å.

timestep protocols). This integration time was adjusted to give results equivalent to those of the TAMD protocol. Significantly better results were obtained when a longer integration time was used. For instance, if the total integration time is appropriately doubled (resulting in a run time approximately that of the fixed time IVM protocol), we found that all 10 structures converged.

The integration step size Δt and system temperature for one of the PC6 structures are shown as a function of step number in Fig. 5. Downward spikes in Δt occur at the beginnings of dynamics simulations when potential parameters have been changed discontinuously and also when there are atom–atom collisions. At these steps the integrator detects a large error in energy conservation and halves the step size. Note that despite this fact, Δt stays relatively constant as the temperature decreases due to the fact that ΔE_0, the energy-conservation tolerance, decreases as the temperature is lowered.

Note that when using a fixed step size it is appropriate to tune the step size for each protocol. It is thus clear that the auto-adjusting timestep feature provides much in the way of convenience. In fact, the exact same (variable step size) protocol could be used in all-degrees-of-freedom dynamics; the correct step size would be chosen by the IVM.

4.2. Two Protein Complex

Here we present an example of using the IVM module to determine the structure of the complex of enzyme IIA$^{\mathrm{Glc}}$ with the histidine phosphocarrier protein HPr, the crystal structures of which are reported in (*28, 29*), respectively. For this example, we reexamine the protocol used in (*30, 31*) for determining the structure of the complex from NMR data given the X-ray structures of the two proteins in isolation. Experimental NMR NOE-derived interproton distance, dipolar coupling, and *J*-coupling restraints were used, as in (*30*). In addition to terms corresponding to these data, the refinement energy included bond, angle, improper, dihedral, atom–atom repulsion, radius of gyration (*32*), and knowledge-based dihedral (*33*) potential terms. The X-ray structures at various separations and orientations were used as starting structures.

In the original protocol, two copies each of HPr and IIA$^{\mathrm{Glc}}$ were utilized in the dynamics calculations, with the extra copies kept fixed in space. In the primary set of coordinates, all atoms not belonging to a sidechain at the interface between the two proteins are bound to their corresponding copies using the X-PLOR noncrystallographic symmetry (NCS) potential. We refer to this original protocol as NCS.

We implemented a new protocol in which all dynamics and minimizations were carried out within the IVM. In the IVM protocol the interfacial sidechain atoms were allowed their torsional degrees of freedom during the dynamics calculations. The remaining atoms in IIA were fixed in space, while the

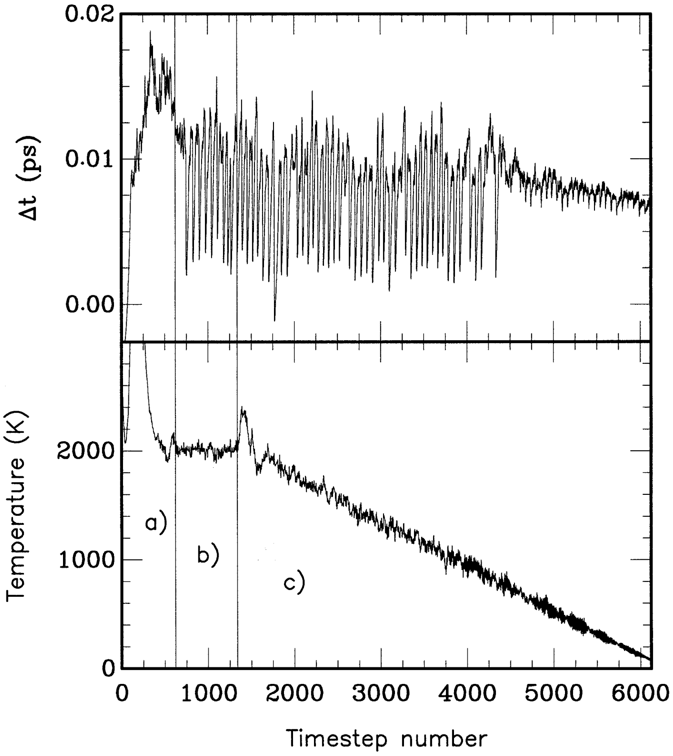

FIG. 5. Timestep size and system temperature during a structure determination run of protein G. The three regions of the dynamics protocol are identified as (a) high temperature equilibration, (b) addition of the atom–atom repulsion potential, and (c) the cooling loop.

noninterfacial atoms in HPR were allowed to rotate and translate as a rigid body.

In both cases, the protocol is summarized as follows:

- Rigid-body minimization, with atom–atom repulsion slowly added.
- High temperature, equilibration dynamics at 3000 K (3000 steps with the NCS protocol and 10 ps, approximately 600 steps with the IVM).
- A molecular dynamics simulated annealing cooling loop in which the force constants and atom–atom repulsion radii are increased. The NCS protocol utilized 103 steps of 2 fs at each temperature, while at each step, the IVM used a time duration of 350 fs resulting in approximately 20 integration steps.
- All atom minimization for the NCS protocol, and rigid body plus flexible sidechain torsion minimization using the IVM.
- Rigid body minimizations first with the atom–atom repulsion and radius of gyration terms turned off and then with those terms turned back on again.

All atomic masses were set to 100 AMU. Four additional pseudo-atoms were used to define the dipolar-coupling frame of reference (*34*). Both protocols treated these as a rigid body restricted to rotation motion only. Before and after depictions of IIA$^{\mathrm{Glc}}$ and HPr are shown in Fig. 6.

With the above two protocols on the SGI Irix platform, the IVM performed the calculations about five times faster than the NCS protocol. For reference, the IVM protocol was modified to allow all internal degrees of freedom for the interfacial sidechain atoms during the dynamics portion of refinement and the resulting time was approximately the same as that using the NCS protocol.

All of the resulting structures from both protocols agreed to within 0.5 Å RMS deviation of all nonhydrogen atoms. The NCS structures had somewhat lower refinement energies: this was apparently due to the fact that there was some distortion of the covalent geometry in the noninterfacial portion of the proteins; this distortion was not possible in the IVM protocol, as those portions were rigid. In particular, the dipolar coupling energy term is rather sensitive to tiny changes in bond orientation. Further, there is no experimental justification for distorting the covalent geometry. The bottom line is that the differences between structures calculated with different initial velocities using the same protocol were the same size as differences between structures calculated using the other protocol.

This example indicates two strengths of the IVM. It allows us to simply consider just those degrees of freedom which are probed in the experiment. As a result of this feature we were able to obtain results equivalent to those obtained by a much more complicated and costly procedure. Secondly, the IVM provides convenience in that all minimizations and dynamics simulations can be carried out within the same framework.

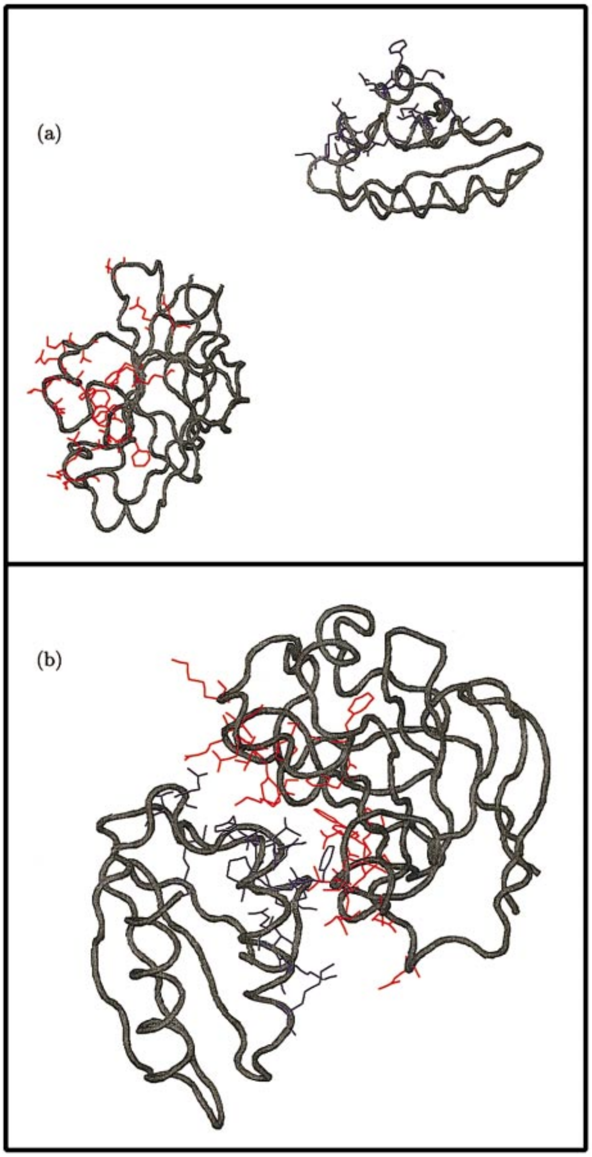

FIG. 6. Hierarchical refinement of the HPr/IIA$^{\mathrm{Glc}}$ complex. Flexible interfacial side chains and backbone atoms are displayed. Panel (a) depicts a starting point configuration while panel (b) shows the final structure. In the figure, the backbone atoms are represented by gray tubes. The flexible regions of HPr and IIA$^{\mathrm{Glc}}$ are represented by blue and red lines, respectively. This figure was created with the VMD–XPLOR visualization program (*35, 36*).

5. CONCLUSION

In this paper we have presented the internal variable module for molecular dynamics and minimization calculations in internal coordinates. The IVM has been shown to be an efficient full-featured implementation of the recursive tree decomposition, acceleration evaluation algorithm which allows loop topologies to be treated correctly. Furthermore, the extensible, modular design allows new features to be easily incorporated. The IVM is currently actively used by our group in the determination of protein and nucleic acid structures by NMR.

It should be noted that constraining arbitrary internal coordinates can result in unphysical dynamical behavior. For example, it has been shown that constraining, bond angle motion raises the effective barrier for torsion motion (*37*). In simulated annealing molecular dynamics calculations used in structure determination, such barriers are mitigated by strong experiment-based, potential energy terms and by employing a suitably high initial temperature in the annealing protocol. Moreover, the detailed dynamical trajectory is of no interest. However, when performing molecular dynamics in nonstructure determination contexts, the implications of freezing degrees of freedom must be given careful consideration.

Possible future enhancements include the addition of one or more Monte Carlo minimization algorithms, adding a more modern local minimization methodology, such as TNPACK (*38, 39*), and better treatment of loop constraints. Finally, most of the algorithms which the IVM employs are amenable to efficient parallelization. For NMR structure determination and refinement, parallelization efforts are usually better focused on processing multiple structures simultaneously. However, other applications have different parallelization specifications, which could benefit from a parallelized version of this IVM.

ACKNOWLEDGMENT

We thank Attila Szabo for helpful comments.

REFERENCES

1. A. Jain, N. Vaidehi, and G. Rodriguez, A fast recursive algorithm for molecular dynamics simulation, *J. Comput. Phys.* **106,** 258 (1993).

2. G. Rodriguez, A. Jain, and K. Kreutz-Delgado, Spatial operator algebra for multibody system dynamics, *J. Austron. Sci.* **40,** 27 (1992).

3. Dae-Sung Bae and E. J. Haug, A recursive formulation for constrained mechanical system dynamics: Part I. Open loop systems, *Mech. Struct. Mach.* **15,** 359 (1987).

4. Dae-Sung Bae and E. J. Haug, A recursive formulation for constrained mechanical system dynamics: Part II. Closed loop systems. *Mech. Struct. Mach.* **15,** 481 (1987).

5. L. M. Rice and A. T. Brünger, Torsion angle dynamics: Reduced variable conformational sampling enhances crystallographic structure refinement, *Proteins* **19,** 277 (1994).

6. E. G. Stein, L. M. Rice, and A. T. Brünger, Torsion-angle molecular dynamics as a new efficient tool for NMR structure calculation. *J. Magn. Reson.* **124,** 154–164 (1997), doi: 10.1006/jmre.1996.1027.

7. P. Güntert, C. Mumenthaler, and K. Wüthrich, Torsion angle dynamics for NMR structure calculation with the new program Dyana, *J. Mol. Biol.* **273,** 283 (1997).

8. A. M. Mathiowetz, A. Jain, N. Karasawa, and W. A. Goddard, III, Protein simulation using techniques suitable for very large systems: The cell multipole method for nonbond interactions and the Newton–Euler inverse mass operator method for internal coordinate dynamics, *Proteins* **20,** 227 (1994).

9. N. Vaidehi, A. Jain, and W. A. Goddard, III, Constant temperature constrained molecular dynamics: The Newton–Euler inverse mass operator method, *J. Phys. Chem.* **100,** 10,508 (1996).

10. G. M. Clore and A. M. Gronenborn, New methods of structure refinement for macromolecular structure determination by NMR, *Proc. Nat. Acad. Sci. U.S.A.* **95,** 5891–5898 (1998); *available at* http://nmr.cit.nih.gov/xplor.nih/. [Review]

11. A. Jain and G. Rodriguez, Recursive flexible multibody system dynamics using spatial operators, *J. Guid. Control Dynam.* **15,** 1453 (1992).

12. H. Goldstein, "Classical Mechanics," Chap. 5, Addison–Wesley, Reading, MA (1980).

13. G. Rodriguez, A. Jain, and K. Kreutz-Delgado, A spatial operator algebra for manipulator modeling and control, *Int. J. Rob. Res.* **10,** 371 (1991).

14. A. K. Mazur, Symplectic integration of closed chain rigid body dynamics with internal coordinate equations of motion, *J. Chem. Phys.* **111,** 1407 (1999).

15. J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of *n*-alkanes, *J. Comput. Phys.* **23,** 327 (1977).

16. W. F. van Gunsteren and H. J. C. Berendsen, Algorithms for macromolecular dynamics and constraint dynamics, *Mol. Phys.* **34,** 1311 (1977).

17. D. Cremer and J. A. Pople, A general definition of ring puckering coordinates, *J. Am. Chem. Soc.* **76,** 1354 (1975).

18. L. Verlet, Computer "experiments" on classical fluids. I. Thermodynamical Properties of Lennard–Jones molecules, *Phys. Rev.* **159,** 98 (1967).

19. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Numerical Recipes in C," p. 569, Cambridge Univ. Press, Cambridge, UK (1988).

20. M. P. Allen and D. J. Tildesley, "Computer Simulation of Liquid," p. 340, Clarendon Press, Oxford (1987).

21. A. M. Gronenborn, D. R. Filpula, N. Z. Essig, A. Achari, M. Whitlow, P. T. Wingfield, and G. M. Clore, A novel, highly stable fold of the immunoglobulin binding domain of streptococcal protein-G, *Science* **253,** 657 (1991).

22. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, Molecular dynamics with coupling to an external bath, *J. Chem. Phys.* **81,** 3684 (1984).

23. S. Nosé, A unified formulation of the constant temperature molecular dynamics methods, *J. Chem. Phys.* **81,** 511 (1984).

24. G. J. Martyna, M. L. Klein, and M. Tuckerman, Nosé–Hoover chains: The canonical ensemble via continuous dynamics, *J. Chem. Phys.* **97,** 2635 (1992).

25. M. J. D. Powell, Restart procedures for the conjugate gradient method, *Math. Program.* **12,** 241–254 (1977).

26. D. J. Evans and S. Murad, Singularity free algorithm for molecular dynamics simulation of rigid polyatomics, *Mol. Phys.* **34,** 327 (1977).

27. H. Goldstein, "Classical Mechanics," p. 608, Addison–Wesley, Reading, MA (1980).

28. M. D. Feese, L. Comolli, N. D. Meadow, S. Roseman, and S. J. Remington, Structural studies of the Escherichia coli signal transducing protein IIA(Glc): Implications for target recognition, *Biochem.* **36,** 16,087 (1997).

29. Z. C. Jia, J. W. Quail, E. B. Waygood, and L. T. J. Delbaere, The 2.0-angstrom resolution structure of escherichia-coli histidine-containing phosphocarrier protein HPr—A redetermination, *J. Biol. Chem.* **268,** 22,490 (1993).

30. G. S. Wang, J. M. Louis, M. Sondej, Y. J. Seok, A. Peterkofsky, and G. M. Clore, Solution structure of the phosphoryl transfer complex between the signal transducing proteins HPr and IIA(Glucose) of the Escherichia coli phosphoenolpyruvate: Sugar phosphotransferase system, *EMBO J.* **19,** 5635 (2000).

31. G. M. Clore, Accurate and rapid docking of protein–protein complexes on the basis of intermolecular nuclear Overhauser enhancement data and dipolar couplings by rigid body minimization, *P. Nat. Acad. Sci. USA* **97,** 9021 (2000).

32. J. Kuszewski, A. M. Gronenborn, and G. M. Clore, Improving the packing and accuracy of NMR structures with a pseudopotential for the radius of gyration, *J. Am. Chem. Soc.* **121,** 2337 (1999).

33. J. Kuszewski and G. M. Clore, Source of and solutions to problems in the refinement of protein NMR structures against torsion angle potentials of mean force, *J. Magn. Reson.* **146,** 249–254 (2000).

34. G. M. Clore, A. M. Gronenborn, and N. Tjandra, Direct refinement against residual dipolar couplings in the presence of rhombicity of unknown magnitude, *J. Magn. Reson.* **131,** 159–162 (1998).

35. C. D. Schwieters and G. M. Clore, The VMD–XPLOR visualization package for NMR structure refinement, *J. Magn. Reson.* **149,** 239 (2001); *available at* http://vmd-xplor.cit.nih.gov/.

36. W. Humphrey, A. Dalke, and K. Schulten, VMD—Visual molecular dynamics, *J. Mol. Graph.* **14,** 33–38 (1996): *available at* http://www.ks.uiuc.edu/Research/vmd/. VMD–XPLOR is based on the VMD program.

37. W. F. van Gunsteren and M. Karplus, Effect of constraints on the dynamics of macromolecules, *Macromolecules* **15,** 1528 (1982).

38. P. Derreumaux, G. Zhang, T. Schlick, and B. Brooks, A truncated Newton minimizer adapted for CHARMM and biomolecular applications, *J. Comput. Chem.* **15,** 532 (1994).

39. D. X. Xie and T. Schlick, Efficient implementation of the truncated-Newton algorithm for large-scale chemistry applications. *SIAM J. Optimiz.* **10,** 132 (1999).