Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of *n*-Alkanes

JEAN-PAUL RYCKAERT*, GIOVANNI CICCOTTI†, AND
HERMAN J. C. BERENDSEN‡

Centre Européen de Calcul Atomique et Moléculaire (CECAM),
Bâtiment 506, Université de Paris XI, 91405 Orsay, France

Received July 19, 1976

A numerical algorithm integrating the $3N$ Cartesian equations of motion of a system of N points subject to holonomic constraints is formulated. The relations of constraint remain perfectly fulfilled at each step of the trajectory despite the approximate character of numerical integration. The method is applied to a molecular dynamics simulation of a liquid of 64 *n*-butane molecules and compared to a simulation using generalized coordinates. The method should be useful for molecular dynamics calculations on large molecules with internal degrees of freedom.

1. INTRODUCTION

The method of molecular dynamics (MD), which has been widely used in the past for studying simple liquids and solids, has more recently been applied to molecular systems with internal degrees of freedom such as N_2 [1], H_2O [2] and even C_4H_{10} [3]. In applying the MD method three problems arise: (a) the choice of a suitable mechanical model, (b) the derivation of the equations of motion of the system and (c) the choice of an efficient algorithm for the numerical integration of these equations.

In polyatomic molecules, the fast internal vibrations are usually decoupled from rotational and translational motions and can therefore be frozen by introducing a certain number of rigid bonds and angles in the skeleton of the molecule. For example, N_2 becomes a rod, H_2O a rigid triangle and C_4H_{10} a nonrigid solid with one internal rotation [1, 2, 3]. The classical way to treat such systems is in terms of generalized coordinates (Lagrange–Hamilton formalism), but as the number of internal degrees of freedom increases it rapidly becomes harder to write down explicitly the appropriate equations of motion.

[*] Faculté des Sciences, Université Libre de Bruxelles, Brussels, Belgium.

[†] Gruppo Nazionale di Struttura della Materia, Consiglio Nazionale delle Ricerche, and Istituto di Fisica "G. Marconi," Università di Roma, Roma, Italy.

[‡] Laboratory of Physical Chemistry, The University of Groningen, Zernikelaan, NL-8002 Groningen, the Netherlands.

ISSN 0021-9991

Some time ago, Orban and Ryckaert [4] suggested the use of cartesian equations of motion in order to describe the dynamical behavior of n-alkane chains, built up of n CH_2 or CH_3 point units connected by $n - 1$ rigid bonds and $n - 2$ rigid angles between adjacent bonds. These equations are in fact the Lagrange equations of motion of the first kind [5], in which the forces of constraint appear explicitly; the dependence of these forces on the positions and velocities of the centers of force is obtained from the relations of constraint (see Section 2 or [4]). On integrating these equations numerically, Orban and Ryckaert obtained quite promising preliminary results. However, one difficulty remained in their approach. The constraints are satisfied exactly at some initial time, but not at later times, because in the numerical integration of a large set of coupled differential equations the computed trajectory deviates more and more from the true one as time proceeds. This is a consequence of the approximate character of the algorithm which is used, and in the present case means that the numerical values of the constrained bond lengths and angles gradually depart from their original values. It follows that after a sufficiently long time the character of the system is strongly modified. In principle, the time step can be reduced sufficiently to obtain an acceptable discrepancy in the constraints after the total time of integration, but this would be very inefficient. In order to avoid these difficulties in the numerical computations we have developed a method which is still based on cartesian coordinates, but leads now to a trajectory in which all constraints are fulfilled exactly at each step of the integration. This is obtained without any loss of precision.

Consider a system of N interacting points subject to l holonomic constraints

$$\sigma_k(\{\mathbf{r}\}) = 0 \qquad (k = 1, ..., l). \tag{1}$$

The force acting on each point can be divided into two contributions: the force $\mathbf{F}_i$ due to the potential energy and the force of constraint $\mathbf{G}_i$ due to all constraints σ_k involving the ith particle. $\mathbf{G}_i$ can be written [5]

$$\mathbf{G}_i = - \sum_{k=1}^{l} \lambda_k(t) \, \nabla_i \sigma_k , \tag{2}$$

where the $\{\lambda_k(t)\}$ are a set of l Lagrangian multipliers depending only on time.

Let us suppose that the integration algorithm used takes into account the time derivatives of forces up to order $n - 2$ (n is often equal to 2), and that it is equivalent to a Taylor expansion of the coordinates $\mathbf{r}(t + \Delta t)$ up to order $(\Delta t)^m$, where Δt is the time step. In order to be consistent with the integration algorithm, the derivatives of $\mathbf{G}$, and hence of $\{\lambda\}$ also have to be known up to order $n - 2$. To avoid the errors implied in the conservation of the constraints, which are accumulative, we employ a method by which the l relations of constraint are exactly fulfilled at time $t + \Delta t$.

Thus instead of solving for $\{\lambda_k^{(n-2)}(t)\}$, a set of parameters $\{\gamma_k\}$ are obtained. In Section 2 we show that the substitution of $\{\gamma_k\}$ for $\{\lambda_k^{(n-2)}(t)\}$ means that the calculation of coordinates at successive steps is still exact up to $(\Delta t)^m$, and thus can be applied in any molecular dynamics algorithm. In addition, the constraints are now automatically fulfilled.

In Section 3, we apply the procedure to the well-known algorithm used by Verlet [6]. This algorithm is particularly suitable for use with our method because the forces of constraint have not to be evaluated explicitly at time t. Their effect is obtained instead from the calculation of the set of l parameters $\{\gamma_k\}$. The l equations giving them are quadratic in $\{\gamma\}$; they can be solved by an iterative process.

In Section 4 the method is applied to a liquid system of 64 n-butane molecules considered as a system of 256 CH_3 or CH_2 "particles," but subject additionally to five constraints per molecule (three rigid bonds and two rigid angles). A numerical comparison between the trajectory obtained by integration of the equations of the system in generalized coordinates, and the path obtained with the method described in Section 3, shows that for this relatively simple system the two methods are equally economic in terms of computing time.

In Section 5 an alternative iterative procedure is described that allows the constraints to be satisfied at each step without solving the equations for $\{\gamma_k\}$ explicitly. The method is based on successively satisfying each of the constraints and is suitable for straightforward application to large molecules with a complicated set of constraints.

The main advantage of the method introduced here is that it allows the use of equations of motion in cartesian form in the simulation of the molecular dynamics of complex molecules, with elimination of the motion of irrelevant internal degrees of freedom. This opens the way to study complex (macro) molecular systems, where, on the one hand, the number of degrees of freedom must be reduced as much as possible and on the other hand, the use of generalized coordinates is prohibitively complicated. The application to the study of liquid n-alkanes (both pure liquid and mixtures) with arbitrary n is straightforward. In the Appendix we give explicitly the equations necessary to deal with such systems.

2. Integration of the Cartesian Equations of Motion of an N-Point System Subject to Holonomic Constraints: Method of Lagrangian Multipliers

Consider a system of N interacting particles. Let $\mathbf{r}_i$ and $\dot{\mathbf{r}}_i$ be the position and the velocity of the ith particle ($i = 1,..., N$) and let $V(\{\mathbf{r}\})$ be the potential energy of the system. We suppose that the system is subject to l holonomic constraints:

$$\sigma_k(\{\mathbf{r}(t)\}) \equiv (\mathbf{r}_i(t) - \mathbf{r}_j(t))^2 - d_{ij}^2 = 0 \qquad (k = 1,..., l), \tag{2.1}$$

where k is a label for the rigid bond (ij) of length d_{ij}. The $3N$ Lagrangian equations of motion of the first kind, in which the forces of constraint appear explicitly, are given [5] by

$$m_i\ddot{\mathbf{r}}_i = \mathbf{F}_i + \mathbf{G}_i = -\nabla_i V - \sum_{k=1}^{l} \lambda_k \nabla_i\sigma_k \qquad (i = 1,..., N), \tag{2.2}$$

where $\mathbf{F}_i$ is the force on i resulting from the potential energy V, $\mathbf{G}_i$ is the force on i due to all constraints (2.1) involving i, and λ_k are the Lagrangian multipliers associated with σ_k. The $3N$ equations (2.2) and the l relations of constraint (2.1) together constitute $3N + l$ equations for $3N + l$ unknowns $(\{\mathbf{r}(t)\}, \{\lambda(t)\})$. If the initial configuration $(\{\mathbf{r}_i(0)\}, \{\dot{\mathbf{r}}_i(0)\})$ is specified and all constraints are satisfied, the trajectory of the system is uniquely determined by (2.1) and (2.2).

Let us look for the analytical solution of the equations of motion (2.2) by writing

$$\lambda_k(t) = \sum_0^\infty{}_n\, \lambda_k^{(n)}(t_0)\, \frac{(t - t_0)^n}{n!}. \tag{2.3}$$

We can obviously calculate $\mathbf{r}_i(t,\{\lambda(t)\})$ as an explicit function of the set $\{\lambda^{(n)}(t_0)\}$. The latter, as we shall see, can easily be obtained from (2.1). With the use of (2.3) we find a Taylor expansion for the solution of (2.2) as

$$\begin{aligned}\mathbf{r}_i(t, \{\lambda(t)\}) &\equiv \mathbf{r}_i(t, \{\lambda^{(n)}(t_0)\}) \\ &= \mathbf{r}_i(t_0) + \dot{\mathbf{r}}_i(t_0)(t - t_0) + \frac{1}{m_i} \sum_2^\infty{}_n \frac{(t - t_0)^n}{n!} \\ &\quad \cdot \left\{ \mathbf{F}_i^{(n-2)}(t_0) - \sum_1^l{}_k \sum_0^{n-2}{}_p \binom{n - 2}{p} \lambda_k^{(p)}(t_0) [(\nabla_i\sigma_k)^{(n-2-p)}]_{t=t_0} \right\},\end{aligned} \tag{2.4}$$

where

$$\mathbf{F}_i^{(s)}(t_0) \equiv \left[\frac{d^s}{dt^s}\, \mathbf{F}_i(t)\right]_{t=t_0} = \sum_1^N{}_j \frac{\partial \mathbf{F}_i^{(s-1)}(t_0)}{\partial \mathbf{r}_j} \cdot \dot{\mathbf{r}}_j(t_0)$$

and

$$[(\nabla_i\sigma_k)^{(s)}]_{t=t_0} \equiv \left[\frac{d^s}{dt^s}\, (\nabla_i\sigma_k)\right]_{t=t_0} = \sum_1^N{}_j \nabla_j[(\nabla_i\sigma_k)^{(s-1)}]_{t=t_0} \cdot \dot{\mathbf{r}}_j(t_0)$$

are the time derivative- of order s of the functions $\mathbf{F}_i$ and $\mathbf{G}_i$ computed at time $t = t_0$.

In order to compute the set $\{\lambda_k^{(s)}(t_0)\}$, $s = 0, 1, \ldots$, we have merely to exploit the information contained in Eqs. (2.1). As these hold at any time, all time derivatives of (2.1) must vanish. We write the system of equations

$$\sigma_k^{(s+2)}(\{\mathbf{r}(t_0)\}) \equiv \left[\frac{d^{s+2}}{dt^{s+2}}\, \sigma_k(\{\mathbf{r}(t)\})\right]_{t=t_0} = 0 \qquad (k = 1, \ldots, l).$$

An explicit evaluation of the time derivatives gives

$$\begin{aligned}\sigma_k^{(s+2)}(\{\mathbf{r}(t_0)\}) &= \sum_1^N{}_j \sum_0^{s+1}{}_\alpha \binom{s + 1}{\alpha} \mathbf{r}_j^{(s+2-\alpha)}(t_0) \cdot [(\nabla_j\sigma_k)^{(\alpha)}]_{t=t_0} \\ &= \sum_1^N{}_j \sum_0^{s+1}{}_\alpha \binom{s + 1}{\alpha} \frac{1}{m_j} \left\{ \mathbf{F}_j^{(s-\alpha)}(t_0) - \sum_1^l{}_p \sum_0^{s-\alpha}{}_\beta \binom{s - \alpha}{\beta} \lambda_p^{(\beta)}(t_0) \right. \\ &\qquad \left. [(\nabla_j\sigma_p)^{(s-\alpha-\beta)}]_{t_0} \right\} [(\nabla_j\sigma_k)^{(\alpha)}]_{t_0} = 0.\end{aligned} \tag{2.5a}$$

The last line is obtained by substitution of $\mathbf{r}_j^{(s+2-\alpha)}$ by the corresponding term in Eq. (2.4). Isolating the terms of the highest order (s) in λ, i.e., the terms with $\alpha = 0$ and $\beta = s$, we can rewrite Eq. (2.5a) as

$$\sum_{1}^{N}{}_{j} \frac{1}{m_j} \left\{ \mathbf{F}_j^{(s)}(t_0) - \sum_{1}^{l}{}_{p} \lambda_p^{(s)}(t_0)[\nabla_j\sigma_p]_{t_0} \right\} \cdot [\nabla_j\sigma_k]_{t_0}$$

$$+ \mathscr{F}(\{\lambda_p^{(s-1)}(t_0),\dots, \lambda_p^{(0)}(t_0)\}, \{\mathbf{r}(t_0), \dot{\mathbf{r}}(t_0)\}) = 0. \tag{2.5b}$$

Here, $\mathscr{F}$ is a known function of the time derivatives of λ's of order less than s and of the coordinates and momenta at time t_0. We see that all $\{\lambda_k^{(s)}\}$ ($s = 0, 1,\dots$) can be obtained by successively inverting the system of linear equations (2.5b) for $s = 0, 1,\dots$. In this way Eqs. (2.2) are reduced to ordinary second order differential equations which can be integrated numerically [4].

The difficulty now arises that the constraints are fulfilled only to the order in the time step which is implicit in the use of a specific algorithm, and the discrepancy will grow in time much more rapidly than linearly. To overcome this problem, we have developed a method that automatically incorporates the exact constraints (2.2), but does not introduce any additional errors in the computation of the trajectory.

Consider the case when the algorithm used involves an error in the coordinates of order $(\Delta t)^{m+1}$. In the simplest case of Taylor's expansion, this is the order of the first omitted term. If the algorithm uses the time derivatives of the forces up to order $n - 2$, corresponding to the nth derivatives of the space coordinates, we have to compute $\{\lambda^{(0)}(t_0),\dots, \lambda^{(n-2)}(t_0)\}$ and we will have at worst

$$\sigma(\{\mathbf{r}^A(t)\}) = O[(\Delta t)^{(m+1)}]$$

where $\mathbf{r}_i{}^A(t)$ denotes the values obtained from the algorithm for $\mathbf{r}_i(t)$. Instead of this, let us compute from Eq. (2.5) only the first $(n - 3)$ derivatives of the λ_k's at t_0 and replace $\{\lambda_k^{(n-2)}(t_0)\}$ by a set of parameters $\{\gamma_k\}$ such that the relations

$$\sigma_k(\{\mathbf{r}(t, \gamma_{k'})\}) = 0 \tag{2.7}$$

are satisfied. Because the $\lambda_k^{(n-2)}$ appear in the expansion Eq. (2.4) always multiplied by $(\Delta t)^n$, it follows that

$$\lambda_k^{(n-2)}(t_0) - \gamma_k = O[(\Delta t)^{m+1-n}]. \tag{2.8}$$

Thus, comparing again with Eq. (2.4), we see that the difference between the trajectory computed with $\gamma_k(t_0)$ and that computed with $\lambda_k^{(n-2)}(t_0)$ is of $O[(\Delta t)^{m+1}]$. This is of the same order as the error implicit in the algorithm, but the constraints are now perfectly fulfilled.

To make the discussion more specific, we write the value of $\mathbf{r}_i(t_0 + \Delta t)$, given by the algorithm, as the sum of two contributions:

$$\mathbf{r}_i(t_0 + \Delta t, (\lambda_k^{(0)}(t_0),\ldots, \lambda_k^{(n-3)}(t_0), \gamma_k\})$$

$$= \mathbf{r}_i'(t_0 + \Delta t, \{\lambda_k^{(0)}(t_0),\ldots, \lambda_k^{(n-3)}(t_0)\}) + \delta\mathbf{r}_i(t_0 + \Delta t, \{\gamma_k\}) \tag{2.9}$$

such that $\mathbf{r}_i'$ is independent of γ_k and $\delta\mathbf{r}_i$ is linear in γ_k and depends further only on $\{\mathbf{r}_i(t_0)\}$. Substituting Eq. (2.9) into (2.1), we find for the kth constraint involving the pair (i, j):

$$2(\mathbf{r}_j' - \mathbf{r}_i') \cdot (\delta\mathbf{r}_j - \delta\mathbf{r}_i) + (\delta\mathbf{r}_j - \delta\mathbf{r}_i)^2 = d_{ij}^2 - (\mathbf{r}_j' - \mathbf{r}_i')^2. \tag{2.10}$$

The $\delta\mathbf{r}_i$ are given in terms of Taylor's expansion by (Eq. (2.4)):

$$\delta\mathbf{r}_i(t_0 + \Delta t, \{\gamma_k\}) = \frac{1}{m_i}\frac{(\Delta t)^n}{n!}\sum_{1}^{l}{}_k\, \gamma_k(\nabla_i\sigma_k)_{t_0}\,. \tag{2.11}$$

The resulting set of l equations (2.10) (one for each constraint) constitutes a matrix equation for the vector $(\gamma_k\}$.

Two remarks can be added. First, Eqs. (2.10) are nonlinear, but can be solved in a quite efficient way by iteration. This is justified for $\Delta t \to 0$ because then the terms nonlinear in γ_k are small. Thus the iteration procedure can be initiated by substituting $\{\gamma_k = 0\}$ in all nonlinear terms, and the iteration always converges rapidly to the physical solution of the second-degree equations. Second, the method is particularly suitable and efficient for use with the simple algorithm of Verlet [6]. In this case n in Eq. (2.9) is equal to 2 and no λ's have to be computed at all. In the next section we shall describe the method for this particular case in detail.

3. Numerical Integration Method Using the Method of Undetermined Parameters

As we have already mentioned in the Introduction, our aim is to perform molecular dynamics calculations with the cartesian equations of motion for polyatomic molecules visualized as systems of interacting point-atoms subject to rigid bonds. The method of undetermined parameters explained in the previous section has now to be combined with a specific integration scheme. In practice we have used the well-known algorithm [6] given by (h is the time step)

$$u(h) = -u(-h) + 2u(0) + h^2\ddot{u}(0) + O(h^4), \tag{3.1}$$

$$\dot{u}(0) = (u(h) - u(-h))/2h + O(h^2). \tag{3.2}$$

Using the method of Section 2 Eqs. (2.9), (2.10), and (2.11), the algorithm (3.1) becomes for each point of the system:

$$\mathbf{r}_i(h) = \mathbf{r}_i(h, \{\gamma_k\}) = \mathbf{r}_i'(h) + \delta\mathbf{r}_i(h)$$

with

$$\begin{aligned} \mathbf{r}_i'(h) &= -\mathbf{r}_i(-h) + 2\mathbf{r}_i(0) + (h^2/m_i)\,\mathbf{F}_i(0), \\ \delta\mathbf{r}_i(h) &= (h^2/m_i) \sum_{k=1}^{l} \gamma_k [\nabla_i \sigma_k]_{t_0}, \end{aligned} \tag{3.3}$$

the $\{\gamma_k\}$ being obtained from

$$\sigma_k(\{\mathbf{r}_i(h, \{\gamma_k\})\}) = 0 \qquad (k = 1,..., l). \tag{3.4}$$

The argument of Section 2, Eq. (2.8) shows that

$$\gamma_k(0) = \lambda_k(0) + O(h^2) \qquad (k = 1,..., l). \tag{3.5}$$

Thus, the algorithm (3.3), (3.4) leads to the required trajectory exact up to the third order in h (as usual), but for which the constraints are now perfectly obeyed. Moreover, the $\{\lambda_k(0)\}$ need no longer be evaluated: the computation of the forces of constraint at time $t = 0$ is now converted in the evaluation of l parameters $(\gamma_k\}$.

We can illustrate this last step in the procedure in the case of rigid constraints (2.1), i.e.,

$$(\mathbf{r}_j(t) - \mathbf{r}_i(t))^2 - d_{ij}^2 = 0, \tag{3.6}$$

where d_{ij} is the constant length of the (ij) pair. Following the methods already discussed we can obtain from (2.11),

$$\begin{aligned} &2(\mathbf{r}_j'(h) - \mathbf{r}_i'(h)) \Big(-h^2 \sum_{k=1}^{l} \gamma_k \left[\left(\frac{\nabla_j}{m_j} - \frac{\nabla_i}{m_i}\right)\sigma_k\right]_{t=0} \\ &\quad + h^4 \sum_{k=1}^{l}\sum_{k'=1}^{l} \gamma_k\gamma_{k'} \left[\left(\frac{\nabla_j}{m_j} - \frac{\nabla_i}{m_i}\right)\sigma_k\right]_{t=0} \left[\left(\frac{\nabla_j}{m_j} - \frac{\nabla_i}{m_i}\right)\sigma_{k'}\right]_{t=0} \\ &= d_{ij}^2 - (\mathbf{r}_j'(h) - \mathbf{r}_i'(h))^2. \end{aligned} \tag{3.7}$$

If there are l rigid bonds ij, we have to solve a system of l quadratic equations in γ_k ; this can be achieved by means of the following iterative procedure. The nth iterated value $\{\gamma_k^{[n]}\}$ is obtained by solving the linearized equations derived by substituting the $\{\gamma_k^{[n-1]}\}$ in the quadratic terms of (3.7). The process is initiated with $\{\gamma_k^{[0]} = 0\}$ and converges in a few steps because the quadratic terms in (3.7) are proportional to h^4 and give only small contributions for the usual time steps used in molecular dynamics. In a numerical test of this method on liquid n-butane (Section 4), three or four iterations are sufficient to satisfy the constraints to a relative discrepancy in the displacements of order 10^{-10}.

In the Appendix, we give the explicit form of (3.7) for an n-point chain simulating an n-alkane; we consider the semirigid model of a chain for which all bonds between adjacent atoms of the chain and all angles between adjacent bonds are rigid, i.e., a total of $(2n - 3)$ rigid constraints.

Algorithm (3.3) is not self-starting; we must therefore evaluate two successive configurations of the system $\{\mathbf{r}_i(0)\}$, $\{\mathbf{r}_i(h)\}$ in order to initiate the integration. If $(\{\mathbf{r}_i(0), \dot{\mathbf{r}}_i(0)\})$ is the initial state of the system such that all constraints are satisfied, we can obtain $\{\mathbf{r}_i(h)\}$ to a reasonable approximation (error in $O(h^3)$) by applying the general method given in Section 2 to the Taylor expansion up to the second order. The method for computing the $\{\gamma_k\}$ is identical to the procedure developed for the algorithm (3.3), (3.4).

4. The Case of n-Butane: Comparison with Integration in Generalized Coordinates

In order to compare our integration method in cartesian coordinates with the classical method involving generalized coordinates, we have applied both methods to a system of 64 n-butane molecules. The model of the n-butane molecule was previously described in Ref. [3]. For present purposes we recall that the molecule is represented as a four-point system with three rigid bonds between adjacent points (C–C bonds) and two rigid angles of 109°28′ between adjacent bonds (C–C–C angles). The whole system is enclosed in a cubical box with periodic boundary conditions; the density $\rho = 0.675$ g/cm^3 matches that of liquid butane and the kinetic energy corresponds to a temperature of $\sim$200°K.

1. *Integration in Generalized Coordinates*

Lagrange equations of the second kind are written in the following generalized coordinates [3]: three coordinates for the center of mass of each molecule; three Eulerian angles giving the orientation of a reference frame attached to the molecule; and one angle of internal rotation around the C–C central bond for each molecule. Those equations are integrated with the Gear algorithm used previously by Rahman and Stillinger [2].

2. *Integration in Cartesian Coordinates*

Lagrange equations of the first kind are integrated by the method of undetermined parameters given in Section 3. These equations of motion and the equations giving the $\{\gamma_k\}$ of n-butane are given in the Appendix for n-alkanes of any length.

Let $\{\mathbf{r}^{(1)}(t)\}$ and $\{\mathbf{r}^{(2)}(t)\}$ be the trajectory of the system (256 CH_3 or CH_2 particles) obtained, respectively, by methods 1 and 2, starting with the same initial configuration at $t = t_0$. ($\{\mathbf{r}^{(2)}(t)\}$ is calculated from the corresponding generalized coordinates at time t.) We define

$$\langle|\,\delta\mathbf{r}(t)|\rangle = \left[\sum_{1}^{256}{}_i\, (\mathbf{r}_i^{(1)}(t) - \mathbf{r}_i^{(2)}(t))^2/256\right]^{1/2}, \tag{4.1}$$

$$\langle r(t)\rangle = \left[\sum_{1}^{256}{}_i\, (\mathbf{r}_i^{(1)}(t) - \mathbf{r}_i^{(1)}(0))^2/256\right]^{1/2}. \tag{4.2}$$

The calculations in each case were carried out with the same value of the time step, i.e., $h = 1.95\ 10^{-15}$ sec, over a period $T = 1.56\ 10^{-13}$ sec sufficiently long to observe an appreciable modification of the system (the normalized center of the mass velocity–autocorrelation function decreases from 1 to 0.3 during this time interval). Equations (4.1) and (4.2) being zero at time t_0, we obtained for $t = t_0 + T$

$$\langle|\,\delta\mathbf{r}(t_0 + T)|\rangle = 1.1\ 10^{-4}\ \text{Å},$$

$$\langle\mathbf{r}(t_0 + T)\rangle = 0.6\ \text{Å}.$$

Hence, after a time T, both numerical paths have diverged with a relative average discrepancy of 2.10^{-4}. On a larger scale of time, no drift of the total energy was observed in either case. Instead, it oscillates around a stable value with an amplitude of $\sim 10^{-3}$ of the kinetic energy. The computer times required for a single integration step were, respectively, 3.25 sec in method 1 and 1.30 sec in method 2 on an IBM 370/168.

Both methods seem then comparable in efficiency for an n-butane liquid. The advantage of the cartesian method, as we mentioned earlier, is that it can be readily extended to larger n-alkanes.

5. An Alternative Procedure for Coordinate Resetting

The method described in Section 3 for "reseting" the coordinates by solving Eqs. (2.10) requires a matrix inversion at each step in order to compute the γ_k. An alternative method based on a physical picture of the process by which the constraints are corrected is described below. The procedure is valid for algorithms requiring no derivatives of the forces, such as the Verlet algorithm.

Let the kth constraint be given by

$$\sigma_k = (\mathbf{r}_i - \mathbf{r}_j)^2 - d_{ij}^2 = 0. \tag{5.1}$$

In the method described in Section 3, the constraints are satisfied by adding displacement vectors $\delta\mathbf{r}_i$ to the vectors $\mathbf{r}_i'$, which resulted from a nonconstrained time step, such that $\sigma_k = 0$ for $\{\mathbf{r}_i = \mathbf{r}_i' + \delta\mathbf{r}_i\}$. According to Eq. (3.3) the displacements are given for the Verlet algorithm by

$$\begin{aligned}\delta\mathbf{r}_i &= -\frac{1}{m_i}(\Delta t)^2 \overset{l}{\underset{1}{\sum}}{}_k\, \gamma_k[\nabla_i\sigma_k]_{t_0} \\ &= -\frac{2(\Delta t)^2}{m_i} \overset{l}{\underset{1}{\sum}}{}_k\, \gamma_k\mathbf{r}_{ij}(t_0),\end{aligned} \tag{5.2}$$

where $\mathbf{r}_{ij} \equiv \mathbf{r}_i - \mathbf{r}_j$ is the vector associated to the kth rigid bond.

If we define $g_{ij} = -2(\Delta t)^2\gamma_k$, then $g_{ij}\mathbf{r}_{ij}(t_0)/m_i$ is the contribution of the kth constraint to the displacement $\delta\mathbf{r}_i$. Likewise $g_{ji}\mathbf{r}_{ji}(t_0)/m_j$ is the contribution of the same constraint to the displacement $\delta\mathbf{r}_j$. Obviously $g_{ji} = g_{ij}$. In a physical picture this means that constraint forces of equal magnitudes and opposite orientations are applied to the particles i and j, in the direction of the bond between i and j for the configuration of the system at time t_0. Since the γ_k are uniquely determined, the g_{ij} are also uniquely determined. It thus follows that any convergent procedure that results in satisfying all constraints by displacements of the form

$$\delta\mathbf{r}_i = \sum_j g_{ij}\mathbf{r}_{ij}(t_0)/m_i, \tag{5.3}$$

while $g_{ji} = g_{ij}$ for all constrained pairs (i, j) and $g_{ij} = 0$ for all other pairs will give results equivalent to those of the method described in the previous sections.

Our alternative method, referred to as the procedure "SHAKE," is an iterative method that considers all constraints in succession.[1]

Given the particle i the procedure corrects its position (and that of j) for the action of the kth constraint according to

$$\delta^k\mathbf{r}_i = g_{ij}\mathbf{r}_{ij}(t_0)/m_i, \tag{5.4a}$$

$$\delta^k\mathbf{r}_j = -g_{ij}\mathbf{r}_{ij}(t_0)/m_j. \tag{5.4b}$$

Consequently the position $\mathbf{r}_i'$, initially produced by the nonconstrained time step (cf., Eqs. (2.9) and (3.3)), is corrected with $\sum_k \delta^k\mathbf{r}_i$ and the next particle is considered.

In this way, for each constraint a quadratic equation in g_{ij} is obtained. Defining

$$\mathbf{r}' = \mathbf{r}_i' + \sum_{k'<k} \delta^{k'}\mathbf{r}_i - \Big(\mathbf{r}_j' + \sum_{k'<k} \delta^{k'}\mathbf{r}_j\Big)$$

and

$$\delta\mathbf{r} = \delta^k\mathbf{r}_i - \delta^k\mathbf{r}_j,$$

[1] Copies of the Fortran subroutine "SHAKE" are available on request by writing to the third author.

$\mathbf{r} = \mathbf{r}_i(t_0) - \mathbf{r}_j(t_0)$, $g = g_{ij}$ and $d = d_{ij}$, we can write the constraint relations as

$$(\mathbf{r}' + \delta\mathbf{r})^2 - d^2 = 0 \tag{5.5}$$

with

$$\delta\mathbf{r} = \left(\frac{1}{m_1} + \frac{1}{m_2}\right) g\mathbf{r},$$

which yields the equation

$$2\left(\frac{1}{m_1} + \frac{1}{m_2}\right) \cdot g \cdot (\mathbf{r} \cdot \mathbf{r}') + \left(\frac{1}{m_1} + \frac{1}{m_2}\right)^2 g^2\mathbf{r}^2 = d^2 - \mathbf{r}'^2. \tag{5.6}$$

The fulfillment of the kth constraint partially destroys that of the previous constraints $k' < k$. Therefore the procedure is subsequently iterated until every σ_k is less than a specified tolerance. As a result of the completed procedure a total correction $\{\delta\mathbf{r}_i\}$ is obtained according to Eq. (5.3), where now g_{ij} is equal to the sum of all g_{ij}'s obtained in the successive iterations.

For computational efficiency Eq. (5.6) is solved only to first order for each constraint, i.e., the term in g^2 is neglected. The iterative nature of the procedure assures that each quadratic equation is finally also solved within the specified tolerance.

The procedure was compared with the matrix method for a molecular dynamics step on a single decane molecule with a time step of 4×10^{-15} sec. Both the initial configuration $\{\mathbf{r}(t_0)\}$ and the configuration $\{\mathbf{r}'\}$ after an unconstrained step were taken from the dynamics run. The relative performance of the two methods is very dependent on the required accuracy of the resetting procedure: the number of iterations required and the computer time for SHAKE resetting increase roughly proportional to the negative logarithm of the tolerance, while for the matrix method practically all the time is spent on matrix inversion and a high accuracy can be attained with little additional effort. Both methods give identical numerical results within the specified tolerance.

For a relative tolerance of 10^{-7} in the constraints both methods used the same central processor time (90 msec on a CDC Cyber 74-16); for better accuracies the matrix method is faster, while for lower accuracies SHAKE is to be preferred. For a tolerance of 10^{-10}, SHAKE was roughly twice as slow as for a tolerance of 10^{-7}.

SHAKE has the advantage of being generally applicabile also to very large molecules containing hundreds of atoms. For small molecules in accurate molecular dynamics runs, however, the matrix resetting method is to be preferred.

VI. Conclusions

In this paper, we have described an algorithm for the integration of the cartesian equations of motion of an N point system subject to holonomic constraints; the relations of constraint remain satisfied exactly at each step in the trajectory.

For liquid *n*-butane we have compared the method with the corresponding integration scheme in generalized coordinates. The two methods are of similar efficiency; this is expected to remain so for larger molecules. In fact, the suitable time step for the method of undetermined parameters is of the same order of magnitude as those generally chosen in other molecular dynamics studies [1, 2, 3].

For the *n*-butane case, the computer time per step is 2.5 times smaller in the cartesian method although the calculation of the forces are performed once per step in both methods. The difference in the computer time is mainly due to the greater simplicity of the cartesian method, in which all calculations are performed in a unique reference frame with a unique set of variables. For the same reason, there is no significant difference in the required storage of data used in both methods. We can therefore claim that if we add holonomic constraints to a system of N interacting free points, the numerical integration of the new equations of motion does not introduce new technical problems, the time step remains approximatively the same, and the computer time does not increase much, because the handling of the forces of constraint does not involve complicated manipulations. Apart from the calculation of these forces, the program remains exactly equivalent to that for simple liquids.

The new method would be adequate for the study of the physical properties of systems involving molecules with a large number of internal degrees of freedom, in which case the equations of motion in generalized coordinates become rather impractical.

Molecular dynamics studies of pure *n*-decane and *n*-decane–solvent mixtures are now in progress.

APPENDIX: Cartesian Forces of Constraints for an *n*-Point Semirigid Chain (*n*-Alkane)

The model of an *n*-alkane molecule is an extension of that used for *n*-butane [3]. It is a semirigid (linear) chain of *n*-CH_2 or CH_3 groups considered as interacting points: the rigidity of the chain results of $(n - 1)$ fixed bonds between adjacent groups (C–C bonds of length $a = 1.53$ Å) and $(n - 2)$ fixed angles (C–C–C of angle $\theta = 109°.28'$). The internal motion of the chain results in $(n - 3)$ internal rotations around the C–C bonds. The physical forces in this system are due to all pair interactions (as Lennard-Jones interactions, for example) between groups belonging to different molecules or between groups belonging to the same molecule (from the third neighboring ones).

If we number $i = 1, 2,..., n - 1, n$ the groups from one terminal group to the other, let $\{\mathbf{r}_i\}$, $\{\dot{\mathbf{r}}_i\}$, $\{\mathbf{F}_i\}$, and $\{\mathbf{G}_i\}$ be, respectively, their positions, velocities, physical forces, and forces of constraint. All relations of constraint can be written as

$$(\mathbf{r}_{i+1} - \mathbf{r}_i)^2 - a^2 = 0 \qquad (i = 1,..., n - 1;\ \text{bond constraint}), \tag{A.1}$$

$$(\mathbf{r}_{i+2} - \mathbf{r}_i)^2 - b^2 = 0 \quad (i = 1,..., n - 2;\ \text{angle constraint}),\ b = 2a \sin \theta/2. \tag{A.2}$$

In order to get a single expression of $\mathbf{G}_i$ in terms of the $\{\lambda\}$, (2.2), we label the relations of constraint (A.1) and (A.2) and their corresponding Lagrangian multipliers in the following way:

$c = 1, 3, 5,..., (2n - 3)$ corresponds to the successive bond constraints (A.1) between pairs of adjacent groups 21, 32,..., $n(n - 1)$.

$c = 2, 4,..., (2n - 4)$ corresponds to the successive angle constraints (A.2) between second neighbor groups 31, 42,..., $n(n - 2)$. For each group of the chain, the equations of motion (2.2) become

$$m_i\ddot{\mathbf{r}}_i = \mathbf{F}_i + \mathbf{G}_i \tag{A.3}$$

where

$$\begin{aligned}\mathbf{G}_i = {} & -2\lambda_{2i-4}(\mathbf{r}_i - \mathbf{r}_{i-2}) - 2\lambda_{2i-3}(\mathbf{r}_i - \mathbf{r}_{i-1}) \\ & + 2\lambda_{2i-1}(\mathbf{r}_{i+1} - \mathbf{r}_i) + 2\lambda_{2i}(\mathbf{r}_{i+2} - \mathbf{r}_i).\end{aligned} \tag{A.4}$$

(For $i = 1, 2,..., (n - 1)$ and n, some terms involving nonexisting groups must be dropped in (A.4).)

Equation (A.3) can be integrated with the algorithm given in Section 3. Applying our procedure (3.3), (3.4), we obtain the following expressions

$$\mathbf{r}_i(h) = \mathbf{r}_i'(h) + \delta\mathbf{r}_i(h) \tag{A.5}$$

where

$$\mathbf{r}_i(h) = -\mathbf{r}_i(-h) + 2\mathbf{r}_i(0) + (h^2/m_i)\,\mathbf{F}_i(0),$$

$$\delta\mathbf{r}_i(h) = (h^2/m_i)\,\mathbf{G}_i\,,$$

with

$$\begin{aligned}\mathbf{G}_i = {} & 2[-\gamma_{2i-4}(\mathbf{r}_i - \mathbf{r}_{i-2}) - \gamma_{2i-3}(\mathbf{r}_i - \mathbf{r}_{i-1}) \\ & + \gamma_{2i-1}(\mathbf{r}_{i+1} - \mathbf{r}_i) + \gamma_{2i}(\mathbf{r}_{i+2} - \mathbf{r}_i)].\end{aligned} \tag{A.6}$$

The $\{\gamma_k\}$ are a set of $(2n - 3)$ parameters corresponding to the $\{\lambda_k(0)\}$ through relation (3.5) if the $\{\gamma_k\}$ are solutions of the $(2n - 3)$ Eqs. (3.8) obtained by substituting (A.5) in (A.1) and (A.2).

In order to obtain the solution of this set of $(2n - 3)$ quadratic equations in the $(2n - 3)$ $\{\gamma_k\}$ by the iterative procedure of Section 3, it is convenient to write them in the following matricial form where the kth line corresponds to the equation involving the kth constraint

$$A(\{\mathbf{r}(0)\}, \{\mathbf{r}'(h)\})\boldsymbol{\gamma} = \mathbf{B}(\{\mathbf{r}(0)\}, \{\mathbf{r}'(h)\}, \{\boldsymbol{\gamma}\}), \tag{A.7}$$

where A is a $(2n - 3, 2n - 3)$ matrix whose elements are zero except for those given below, $\boldsymbol{\gamma}$ is the column vector $(\gamma_1, \gamma_2, ..., \gamma_{2n-3})$, and $\mathbf{B}$ is the column vector whose elements collect all terms nonlinear in γ in the kth equation.

If k is odd (constraint between groups labeled, $(k+1)/2$, $((k+1)/2)+1$)

$$B_k = a^2 - (\mathbf{r}'_{(k+3)/2} - \mathbf{r}'_{(k+1)/2})^2 - \frac{h^4}{m^2}(\mathbf{G}_{(k+3)/2} - \mathbf{G}_{(k+1)/2})^2,$$

$$A_{k,k-3} = \frac{4h^2}{m}(\mathbf{r}'_{(k+3)/2} - \mathbf{r}'_{(k+1)/2}) \cdot (\mathbf{r}_{(k+1)/2} - \mathbf{r}_{(k-3)/2}),$$

$$A_{k,k-2} = \frac{4h^2}{m}(\mathbf{r}'_{(k+3)/2} - \mathbf{r}'_{(k+1)/2}) \cdot (\mathbf{r}_{(k+1)/2} - \mathbf{r}_{(k-1)/2}),$$

$$A_{k,k-1} = -\frac{4h^2}{m}(\mathbf{r}'_{(k+3)/2} - \mathbf{r}'_{(k+1)/2}) \cdot (\mathbf{r}_{(k+3)/2} - \mathbf{r}_{(k-1)/2}),$$

$$A_{k,k} = -\frac{8h^2}{m}(\mathbf{r}'_{(k+3)/2} - \mathbf{r}'_{(k+1)/2}) \cdot (\mathbf{r}_{(k+3)/2} - \mathbf{r}_{(k+1)/2}),$$

$$A_{k,k+1} = -\frac{4h^2}{m}(\mathbf{r}'_{(k+3)/2} - \mathbf{r}'_{(k+1)/2}) \cdot (\mathbf{r}_{(k+5)/2} - \mathbf{r}_{(k+1)/2}),$$

$$A_{k,k+2} = \frac{4h^2}{m}(\mathbf{r}'_{(k+3)/2} - \mathbf{r}'_{(k+1)/2}) \cdot (\mathbf{r}_{(k+5)/2} - \mathbf{r}_{(k+3)/2}),$$

$$A_{k,k+3} = \frac{4h^2}{m}(\mathbf{r}'_{(k+3)/2} - \mathbf{r}'_{(k+1)/2}) \cdot (\mathbf{r}_{(k+7)/2} - \mathbf{r}_{(k+3)/2}).$$

If k is even (constraints between groups labeled $k/2$ and $(k/2)+2$)

$$B_k = b^2 - (\mathbf{r}'_{(k+4)/2} - \mathbf{r}'_{k/2})^2 - \frac{h^4}{m^2}(\mathbf{G}_{(k+4)/2} - \mathbf{G}_{k/2})^2,$$

$$A_{k,k-4} = \frac{4h^2}{m}(\mathbf{r}'_{(k+4)/2} - \mathbf{r}'_{k/2}) \cdot (\mathbf{r}_{k/2} - \mathbf{r}_{(k-4)/2}),$$

$$A_{k,k-3} = \frac{4h^2}{m}(\mathbf{r}'_{(k+4)/2} - \mathbf{r}'_{k/2}) \cdot (\mathbf{r}_{k/2} - \mathbf{r}_{(k-2)/2}),$$

$$A_{k,k-1} = -\frac{4h^2}{m}(\mathbf{r}'_{(k+4)/2} - \mathbf{r}'_{k/2}) \cdot (\mathbf{r}_{(k+2)/2} - \mathbf{r}_{k/2}),$$

$$A_{k,k} = -\frac{8h^2}{m}(\mathbf{r}'_{(k+4)/2} - \mathbf{r}'_{k/2}) \cdot (\mathbf{r}_{(k+4)/2} - \mathbf{r}_{k/2}),$$

$$A_{k,k+1} = -\frac{4h^2}{m}(\mathbf{r}'_{(k+4)/2} - \mathbf{r}'_{k/2}) \cdot (\mathbf{r}_{(k+4)/2} - \mathbf{r}_{(k+2)/2}),$$

$$A_{k,k+3} = \frac{4h^2}{m}(\mathbf{r}'_{(k+4)/2} - \mathbf{r}'_{k/2}) \cdot (\mathbf{r}_{(k+6)/2} - \mathbf{r}_{(k+4)/2}),$$

$$A_{k,k+4} = \frac{4h^2}{m}(\mathbf{r}'_{(k+4)/2} - \mathbf{r}'_{k/2}) \cdot (\mathbf{r}_{(k+8)/2} - \mathbf{r}_{(k+4)/2}).$$

Acknowledgments

The authors wish to thank Professor A. Bellemans, Dr. G. Jacucci, and Dr. J. Orban for many stimulating discussions and suggestions. They are much indebted to Dr. I. R. MacDonald for the critical lecture of the manuscript. One of us (J.P.R.) is also grateful to the "Institut pour l'Encouragement de la Recherche Scientifique dans l'Industrie et l'Agriculture" for financial support.

References

1. J. Barojas, D. Levesque, and B. Quentrec, *Phys. Rev. A* 7 (1973), 1092.
2. A. Rahman and F. H. Stillinger, *J. Chem. Phys.* **55** (1971), 3336; *J. Chem. Phys.* **60** (1974), 1545.
3. J. P. Ryckaert and A. Bellemans, *Chem. Phys. Lett.* **30** (1975), 123.
4. J. Orban and J. P. Ryckaert, Report on CECAM Workshop, "Methods in Molecular Dynamics," 1974, unpublished.
5. T. C. Bradbury, "Theoretical Mechanics," Chap. XI, Wiley International Editions, New York, 1968.
6. L. Verlet, *Phys. Rev.* **159** (1967), 98.

Printed in Belgium