Molecular Dynamics with Constraints

M. K. MEMON,* R. W. HOCKNEY, AND S. K. MITRA

Department of Computer Science, University of Reading,
Whiteknights Park, Reading RG6 2AX, England

Received September 21, 1980; revised April 20, 1981

In this paper the molecular dynamics method has been extended to include constraints in order to permit the computer simulation of molecular systems with strong intramolecular bonding. In this constraint dynamics method the units of each molecule are spatially adjusted to keep a preassigned geometry. The method has been tested for reliability, efficiency and accuracy on a simulation of H_2O microsystem. The important conservation laws (e.g., energy, momentum and angular momentum) are satisfied.

1. INTRODUCTION

The method of molecular dynamics (MD) [1] has been successfully applied to the study of simple non-polar liquids [2, 3], liquid metals [4], molecular liquids [5], ionic liquids [6], water [7] and melting phenomena [8, 9]. The speed of an MD calculation can be increased by reducing the number of degrees of freedom. In a polyatomic system, the fast internal vibrations are usually decoupled from the rotational and translational motions. They can be frozen by applying constraint dynamics, so that the bond lengths and angles are kept fixed at a constant value during the MD run. For example, a nitrogen molecule becomes a rod, a water molecule a rigid triangle and butane (C_4H_{10}) a non-rigid solid with one internal rotation [10, 7, 11]. This method has been proposed by Ryckaert *et al.* [12] for integrating the cartesian equations of motion of a system of particles subject to holonomic constraints. van Gunsteren and Berendsen [13] used this method for macromolecules.

In principle the properties of a collection of N classical particles can be obtained by solving the set of N equations of motion

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} - \mathbf{F}_i(\mathbf{r}_i \cdots \mathbf{r}_N) = 0, \tag{1}$$

where $i = 1, 2, \ldots, N$, m is the mass and $\mathbf{F}$ and $\mathbf{r}$ are the force and position vectors, respectively. This is the basis of the MD calculation. In constraint dynamics these equations of motion are solved together with holonomic constraints on the coor-

* Present address: Institute of Physics and Technology, University of Sind, Jamshoro (Sind), Pakistan.

0021-9991/81/100345–12$02.00/0

Copyright © 1981 by Academic Press, Inc.
All rights of reproduction in any form reserved.

dinates of the particle. The condition of constraint can be expressed as equations connecting of the particles (at a given time) by:

$$c_k(\mathbf{r}_1, \mathbf{r}_2, ..., t) = 0. \tag{2}$$

These constraints maintain the preassigned relationship within a given set of particles during the time evolution of the system. A simple form of such constraints can be expressed in the following way [14]:

$$d_{ij}^2 - (\mathbf{r}_i(t) - \mathbf{r}_j(t))^2 = 0, \tag{3}$$

where d_{ij} is the length of the rigid bond between the particles i and j.

In constraint dynamics the equations of motion become

$$m_i \frac{\partial^2 \mathbf{r}_i}{\partial t^2} - \mathbf{F}_i - \mathbf{G}_i = 0, \tag{4}$$

where $\mathbf{G}_i$ is the force due to l constraints c_k involving the ith particle and can be written in terms of Lagrangian multipliers λ_k;

$$\mathbf{G}_i = -\sum_{k=1}^{l} \lambda_k(t)\, \nabla_i c_k. \tag{5}$$

In Section 2 we shall develop a method of solving Eq. (3) by using the leapfrog algorithm. In Section 3 we have proposed a new constraint equation:

$$d_{ij} - \sqrt{(\mathbf{r}_i(t) - \mathbf{r}_j(t))^2} = 0. \tag{6}$$

The relative efficiency of method I, consisting of the leapfrog scheme for Eqs. (3) to (5), and method II, consisting of the leapfrog scheme for Eqs. (4) to (6), are analysed in detail in Section 4 for a model water-like microsystem in three-dimensional space.

2. Leapfrog Algorithm for Constraint Dynamics

The leapfrog algorithm is usually more efficient than Verlet or Beeman algorithms [15]. It requires fewer operations and less storage and the energy conservation is better satisfied. A molecular dynamics program based on the leapfrog method has been written by Eastwood *et al.* [16]. They have also described the procedures to calculate energy, pressure, etc., from the positions and velocities of the particles in the system. Here we shall discuss the modifications necessary to implement the leapfrog algorithm in the constraint dynamics equations (4) and (5).

In the leapfrog method the velocity, $\dot{\mathbf{r}}_i$ of the ith particle at discrete timestep $n + \frac{1}{2}$ is calculated using the velocity of the particle at previous timestep $n - \frac{1}{2}$ and the contribution from the forces, i.e.,

$$\dot{\mathbf{r}}_i(n + \tfrac{1}{2}) = \dot{\mathbf{r}}_i(n - \tfrac{1}{2}) + (\mathbf{F}_i(n) + \mathbf{G}_i(n))\,\Delta t/m_i, \tag{7}$$

where Δt is the fixed timestep and m_i is the mass of the particle i. The position of the particle is calculated at timestep $n+1$ from its position at previous timestep n and its velocity at time level $n+\frac{1}{2}$, i.e.,

$$\mathbf{r}_i(n+1)=\mathbf{r}_i(n)+\dot{\mathbf{r}}_i(n+\tfrac{1}{2})\,\Delta t. \tag{8}$$

If constraint forces $\mathbf{G}_i$ given by Eq. (5) are used in Eq. (7) to calculate the trajectory we see that the constraints (Eq. (2)) are satisfied only to the first order in Δt. This error in constraints will grow with time. Ryckaert *et al.* [12] has developed a very elegant method for the Verlet algorithm, where the Lagrange multipliers at time n are determined by requiring that the constraints remain exactly fulfilled at time $n+1$. This gives an error in the trajectory calculation of the same order as the error implicit in the algorithm, yet the constraints are perfectly fulfilled at each time level. To incorporate this in Eqs. (7) and (8) we rewrite these equations in the following form:

$$\dot{\mathbf{r}}_i\left(n+\frac{1}{2}\right)=\dot{\mathbf{r}}_i\left(n-\frac{1}{2}\right)+\mathbf{F}_i(n)\frac{\Delta t}{m_i}+\delta\mathbf{r}_i(n+1)/\Delta t, \tag{9}$$

$$\mathbf{r}_i(n+1)=\mathbf{r}_i(n)+\dot{\mathbf{r}}_i(n+\tfrac{1}{2})\,\Delta t, \tag{10}$$

where

$$\delta\mathbf{r}_i(n+1)=-\frac{(\Delta t)^2}{m_i}\sum_{k=1}^{l}\gamma_k[\nabla_i c_k]_n, \tag{11}$$

where γ_k are obtained from constraint equation

$$c_k(\{\mathbf{r}_i(n+1)\})=0. \tag{12}$$

For ease of calculation we express Eqs. (9) in terms of unconstrained (primed) and constrained (unprimed) quantities. So

$$\mathbf{r}_i(n+1)=\mathbf{r}'_i(n+1)+\delta\mathbf{r}_i(n+1), \tag{13}$$

$$\dot{\mathbf{r}}'_i(n+\tfrac{1}{2})=\dot{\mathbf{r}}_i(n-\tfrac{1}{2})+\mathbf{F}_i(n)\,\Delta t/m_i, \tag{14}$$

$$\mathbf{r}'_i(n+1)=\mathbf{r}_i(n)+\dot{\mathbf{r}}'_i(n+\tfrac{1}{2})\,\Delta t. \tag{15}$$

Finally γ_k can be calculated by using Eqs. (11), (13) and the constraint Eq. (12), i.e.,

$$c_k(\{\mathbf{r}'_i(n+1)+\delta\mathbf{r}_i(n+1)\})=0. \tag{16}$$

Ryckaert *et al.* [12] have already obtained the following explicit equations for the rigid constraints of the type given by Eq. (3).

$$c_k\equiv[\mathbf{r}'_i(n+1)+\delta\mathbf{r}_i(n+1)+\mathbf{r}'_j(n+1)-\delta\mathbf{r}_j(n+1)]^2-d_{ij}^2=0 \tag{17}$$

and

$$\delta \mathbf{r}_i = -\frac{2(\Delta t)^2}{m_i} \sum_{k=1}^{l} \gamma_k(\mathbf{r}_i(n) - \mathbf{r}_i(n)). \tag{18}$$

In Section 4 we shall apply these equations ((13 to (15) and (17)) to simulate a system of water molecules constrained to form rigid triangles.

3. A New Rigid Constraint

In the last section we saw that for the rigid constraint equations (17) used in Ryckaert *et al.* [12] one has to.solve the simultaneous quadratic equations (17) and (18). These equations can be viewed as a particular point of a four-dimensional hyper surface which is analogous to a parabola in two-dimensional space. The iterative procedure generally used to solve such equations (the Newton iteration) obtains successive approximation to its roots from the gradients (a hyper plane) at the approximate root points on the hyper surface, for example, if we have to solve

$$\mathbf{f}(\mathbf{x}) = 0, \tag{19}$$

where x is an N-dimensional column vector of the roots to Eq. (19) and $\mathbf{f}(\mathbf{x})$ is a column vector of the N simultaneous non-linear equations that are to be satisfied. In the Newton iteration for the roots of Eq. (19), the $\nu + 1$th iteration is given by

$$\mathbf{x}^{\nu+1} = \mathbf{x}^{\nu} - [\mathbf{f}'(\mathbf{x}^{\nu})]^{-1}\mathbf{f}(x^{\nu}), \tag{20}$$

where $\mathbf{f}'(\mathbf{x}^{\nu})$ is the first derivative of $\mathbf{f}$ with respect to $\mathbf{x}$ and is an $N \times N$ matrix with element (i, j) equal to $\partial f_i/\partial x_j$ and $\mathbf{f}(\mathbf{x}^{\nu})$ is a column vector containing the values f for given approximate roots $\mathbf{x}^{\nu}$. The convergence of Eq. (20) will depend on the initial guess $\mathbf{x}^0$ and on the gradients $\mathbf{f}'$. If the initial approximation to $\mathbf{x}$ is very near a root and if the gradients in this region are continuous and smoothly varying, the convergence to the root will be very rapid. In Eq. (20) one has to calculate the inverse of the matrix $\mathbf{f}'(\mathbf{x}^{\nu})$ at each iteration, which may be very expensive, especially if the size of the matrix is large. However, for the present system (Eq. (17)), the gradient $\mathbf{f}'(x^{\nu})$ is a slowly varying function of $\mathbf{x}$ near a root and one can use the initial gradient throughout the iteration, thus saving the time'for calculating more than the first matrix inverse of $\mathbf{f}'(\mathbf{x}^{\nu})$. This will, of course, slow down the convergence but the overall efficiency will be better. An iterative method based on this idea has been used by Ryckaert *et al.* [12] to solve Eqs. (17) and (18) (henceforth this method will be called method I).

If instead of using the constaint Eq. (3) we use Eq. (6), then

$$c_k = d_{ij} - |\mathbf{r}_i - \mathbf{r}_j|,$$

we get

$$c_k = |\mathbf{r}_i'(n+1) + \delta\mathbf{r}_i(n+1) - \mathbf{r}_j'(n+1) - \delta\mathbf{r}_j(n+1)| - d_{ij} = 0, \tag{21}$$

which can also be solved by using the above iterative procedure (henceforth this method will be called method II). Although method II requires square root calculation (Eq. (21)) it is more efficient than method I because for a given accuracy method II converges faster. The right-hand side of Eq. (21) constitutes a hyper surface whose gradients vary less rapidly with $\delta\mathbf{r}_i$ than the gradients of the hyper surface constructed by method I from Eq. (17). Therefore in method II one can use the constant initial gradient approach more effectively. A quantitative comparison of these two methods is carried out by simulating a system of water molecules.

4. Comparison of the Methods

A water molecule can be regarded as two hydrogen atoms and one oxygen atom rigidly bonded together and the length of bonds O–H and H–H constant. Here we shall apply the constraint dynamics method based on method I and method II to simulate a system of 128 water molecules. In addition to constraint forces the oxygen and hydrogen atoms are regarded as doubly and singly charged soft spherical ions, respectively. The interionic force law used in the calculation has the following form [16]:

$$F(r_{ij}) = \frac{q_i q_j}{r_{ij}^2}\left[1 + \operatorname{sign}(q_i q_j)\frac{(\sigma_i + \sigma_j)^p}{r_{ij}}\right], \tag{22}$$

TABLE I

The Parameters Used in the Simulation of Water Molecules to Test the Numerical Accuracy of the Model

Parameter	Value
O–H bond length, B_{OH}	0.9584 Å
H–H bond length, B_{HH}	1.5151 Å
H$\hat{\mathrm{O}}$H bond angle, θ	104.45°
charge of hydrogen ion, q_1	0.3298e
charge of oxygen ion, q_2	−0.65966e
mass of hydrogen ion, m_1	1.0
mass of oxygen ion, m_2	16.0
hardness parameter, p	8
redius ratio $\sigma_{\mathrm{O}}/\sigma_{\mathrm{H}}$	0.4545
equilibrium separation, $\sigma_0 + \sigma_0$	0.96 Å
equilibrium separation, $\sigma_{\mathrm{O}} + \sigma_{\mathrm{H}}$	1.536 Å
equilibrium separation, $\sigma_{\mathrm{H}} + \sigma_{\mathrm{H}}$	2.112 Å
timestep, DT	3×10^{-16} sec

Note. e is the absolute value of the electronic charge ~ 1.602×10^{-19} coulombs and masses are given in units of atomic weight.

where q_i and σ_i are the charge and hard sphere radius of the ion i and p measures the strenth of repulsion. All the parameters used in the present calculation are given in Table I.

A flowchart for the constraint dynamics computer program is shown in Fig. 1. We start with a random configuration of water molecules, where each molecule exactly satisfies the bond constraints. All starting velocities are initialized to zero. Then the following steps are followed for each timestep cycle (using numbering of Fig. 1):

2. The force on each ion due to all other ions is calculated by using the force law equation (22) and unconstrained velocities are calculated (Eq. (14)).

3. A molecule is selected and unconstrained position of each atom is calculated (Eq. (15)). Procedure "ADJUST," which calculated γ_k from the current unconstrained and previous constrained positions, is called. (Eq. (11)). Details of "ADJUST" for method I and method II are different and will be discussed later in this section.

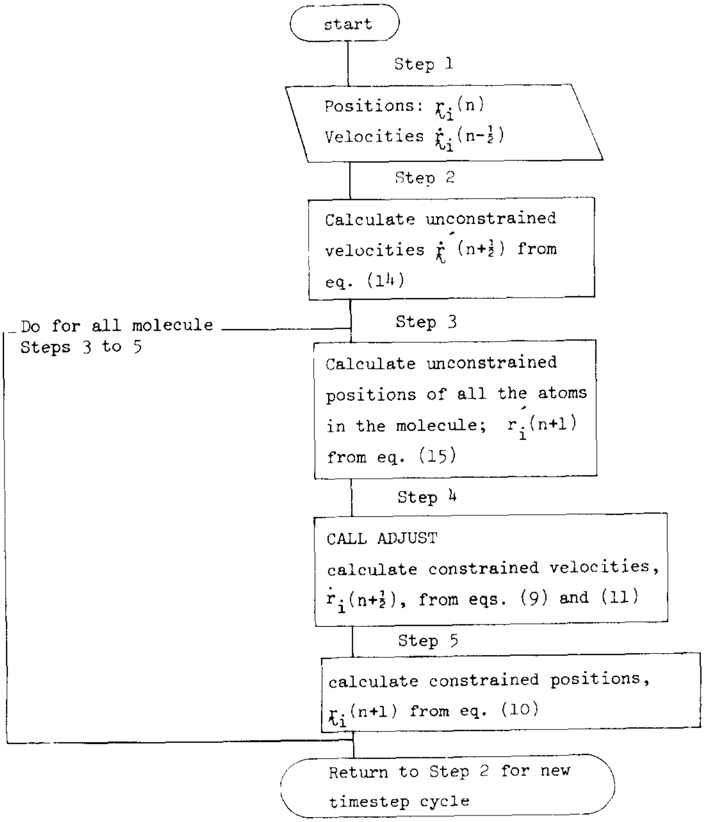
start
Step 1
Positions: $\underset{\sim}{r}_i(n)$
Velocities $\dot{\underset{\sim}{r}}_i(n-\frac{1}{2})$
Step 2
Calculate unconstrained velocities $\dot{\underset{\sim}{r}}'(n+\frac{1}{2})$ from eq. (14)
Do for all molecule Steps 3 to 5
Step 3
Calculate unconstrained positions of all the atoms in the molecule; $r_i'(n+1)$ from eq. (15)
Step 4
CALL ADJUST calculate constrained velocities, $\dot{r}_i(n+\frac{1}{2})$, from eqs. (9) and (11)
Step 5
calculate constrained positions, $\underset{\sim}{r}_i(n+1)$ from eq. (10)
Return to Step 2 for new timestep cycle

FIG. 1. Flowchart of constraint dynamics program.

4. Constraint forces constructed from γ_k are used to calculate the constrained velocities (Eq. (13)).

5. Constrained positions are calculated from these velocities using Eq. (15).

Steps 3 to 5 are repeated for all the molecules. For the next timestep the current velocities and positions are used in step 2.

4.1. *Procedure "ADJUST"*

For a water molecule to have constant O–H bond length and constant H–O–H bond angle we need three bond constraints. If the positions of the hydrogen and oxygen atoms are $\mathbf{r}_1^0$, $\mathbf{r}_2^0$ and $\mathbf{r}_3^0$ at any given timestep they must conform to:

$$|\mathbf{r}_1^0-\mathbf{r}_2^0|=B_{\mathrm{HH}}\ \text{(bond length H–H)}$$

$$|\mathbf{r}_1^0-\mathbf{r}_3^0|=B_{\mathrm{OH}}\ \text{(bond length O–H)},$$

and

$$|\mathbf{r}_2^0-\mathbf{r}_3^0|=B_{\mathrm{OH}}.$$

Application of method I (Ryckaert *et al.* [12]) to a water molecule (Eq. (17) and (18)) gives the following set of equations:

$$(\mathbf{r}_1'+\delta\mathbf{r}_1-\mathbf{r}_2'-\delta\mathbf{r}_2)^2-B_{\mathrm{HH}}^2=0,$$

$$(\mathbf{r}_1'+\delta\mathbf{r}_1-\mathbf{r}_3'-\delta\mathbf{r}_3)^2-B_{\mathrm{OH}}^2=0,$$

$$(\mathbf{r}_2'+\delta\mathbf{r}_2-\mathbf{r}_3'-\delta\mathbf{r}_3)^2-B_{\mathrm{OH}}^2=0,$$

and

$$\delta\mathbf{r}_i=1/m_i\sum_j g_{ij}(\mathbf{r}_i^0-\mathbf{r}_j^0), \tag{23}$$

where $\mathbf{r}_i^0\equiv\mathbf{r}_i(n)$, constrained positions at timestep n.

Following Ryckaert *et al.* [12] we have introduced $g_{ij}=-2(\Delta t)^2\gamma_k$, which is the contribution of the kth constraint (between particle i and j) to the displacement $\delta\mathbf{r}_i$. We note that $g_{ij}=g_{ji}$ and $g_{ii}=0$; altogether we shall have three unknowns, g_{12}, g_{13} and g_{23}. For ease of the iterative calculation we expand Eqs. (23) and write them in a matrix form:

$$\mathbf{S}+2\mathbf{Ag}+\mathbf{W}(g^2)=0, \tag{24}$$

where

$$\mathbf{S}\equiv\begin{bmatrix}-B_{\mathrm{HH}}^2+(\mathbf{r}_{12}')^2\\ -B_{\mathrm{OH}}^2+(\mathbf{r}_{13}')^2\\ -B_{\mathrm{OH}}^2+(\mathbf{r}_{23}')^2\end{bmatrix}, \tag{25}$$

$$\mathbf{g}\equiv\begin{bmatrix}g_{12}\\ g_{13}\\ g_{23}\end{bmatrix}, \tag{26}$$

and

$$\mathbf{A} \equiv \begin{bmatrix} \mu_{12}\mathbf{r}_{12}^0 \cdot \mathbf{r}'_{12} & \frac{1}{m_1}\mathbf{r}_{13}^0 \cdot \mathbf{r}'_{12} & -\frac{1}{m_2}\mathbf{r}_{23}^0 \cdot \mathbf{r}'_{12} \\ \frac{1}{m_1}\mathbf{r}_{12}^0 \cdot \mathbf{r}'_{13} & \mu_{13}\mathbf{r}_{13}^0 \cdot \mathbf{r}'_{13} & \frac{1}{m_3}\mathbf{r}_{23}^0 \cdot \mathbf{r}'_{13} \\ -\frac{1}{m_2}\mathbf{r}_{12}^0 \cdot \mathbf{r}'_{23} & \frac{1}{m_3}\mathbf{r}_{13}^0 \cdot \mathbf{r}'_{23} & \mu_{23}\mathbf{r}_{23}^0 \cdot \mathbf{r}'_{23} \end{bmatrix}, \tag{27}$$

where

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j,$$

$$\mu_{ij} = \frac{1}{m_i} + \frac{1}{m_j},$$

$m_1 = m_2 \equiv$ mass of the hydrogen atom,

$m_3 =$ mass of oxygen of atom and $\mathbf{W}$ is a vector whose elements are quadratic function of g's.

To solve Eq. (24) we have used the iterative procedure suggested by Ryckaert *et al.* [12], where the νth iterated value $\{g^{\nu}\}$ is obtained by solving linearized equation derived by substituting $\{g^{\nu-1}\}$ values in the quadratic term. This may be shown to be the same as using the Newton iteration with a constant gradient (the initial one) in all iterations.

Similarly using method II (Eq. (21)) we get a set of non-linear equations:

$$\sqrt{(\mathbf{r}'_{12} + \delta\mathbf{r}_1 - \delta\mathbf{r}_2) \cdot (\mathbf{r}'_{12} + \delta r_1 - \delta\mathbf{r}_2)} - B_{\mathrm{HH}} = 0,$$

$$\sqrt{(\mathbf{r}'_{13} + \delta\mathbf{r}_1 - \delta\mathbf{r}_3) \cdot (\mathbf{r}'_{13} + \delta\mathbf{r}_1 - \delta\mathbf{r}_3)} - B_{\mathrm{OH}} = 0,$$

$$\sqrt{(\mathbf{r}'_{23} + \delta\mathbf{r}_2 - \delta\mathbf{r}_3) \cdot (\mathbf{r}'_{23} + \delta\mathbf{r}_2 - \delta\mathbf{r}_3)} - B_{\mathrm{OH}} = 0,$$

and

$$\delta\mathbf{r}_i = \frac{1}{m_i}\sum_j g_{ij}\hat{\mathbf{r}}_{ij}^0, \tag{28}$$

where $\hat{\mathbf{r}}_{ij} = (\mathbf{r}_i^0 - \mathbf{r}_j^0)/|\mathbf{r}_i^0 - \mathbf{r}_j^0|$ and "hat" on the position vectors denote unit vectors. These equations, when Taylor expanded and written in matrix form, become:

$$\mathbf{S} + \mathbf{Ag} + \mathbf{O}(g^2) = 0, \tag{29}$$

where

$$\mathbf{S} \equiv \begin{bmatrix} -B_{\mathrm{HH}} + |\mathbf{r}'_1 - \mathbf{r}'_2| \\ -B_{\mathrm{OH}} + |\mathbf{r}'_1 - \mathbf{r}'_3| \\ -B_{\mathrm{OH}} + |\mathbf{r}'_2 - \mathbf{r}'_3| \end{bmatrix}, \tag{30}$$

$$\mathbf{A} \equiv \begin{bmatrix} \mu_{12}\hat{\mathbf{r}}^0_{12} \cdot \hat{\mathbf{r}}'_{12} & \dfrac{1}{m_1}\hat{\mathbf{r}}^0_{13} \cdot \mathbf{r}'_{12} & -\dfrac{1}{m_2}\mathbf{r}^0_{23} \cdot \mathbf{r}'_{12} \\ \dfrac{1}{m_1}\hat{\mathbf{r}}^0_{12} \cdot \hat{\mathbf{r}}'_{13} & \mu_{13}\hat{\mathbf{r}}^0_{13} \cdot \hat{\mathbf{r}}'_{13} & \dfrac{1}{m_3}\hat{\mathbf{r}}^0_{23} \cdot \hat{\mathbf{r}}'_{13} \\ -\dfrac{1}{m_2}\hat{\mathbf{r}}^0_{12} \cdot \hat{\mathbf{r}}'_{23} & \dfrac{1}{m_3}\hat{\mathbf{r}}^0_{13} \cdot \hat{\mathbf{r}}'_{23} & \mu_{23}\hat{\mathbf{r}}^0_{23} \cdot \hat{\mathbf{r}}'_{23} \end{bmatrix}. \qquad (31)$$

As compared to Eq. (27), here in Eq. (31) all the position vectors are replaced by corresponding unit vectors. We have used two iterative schemes to solve Eq. (29). In the first method (IIa, $\{g^{\nu+1}\}$ at $\nu+1$th iteration is calculated from the linearized equation (29) using $\{g^{\nu}\}$ values in the non-linear part of the equation and A^{-1} is calculated at each iteration. The iterations continue until all the components of **S** lie

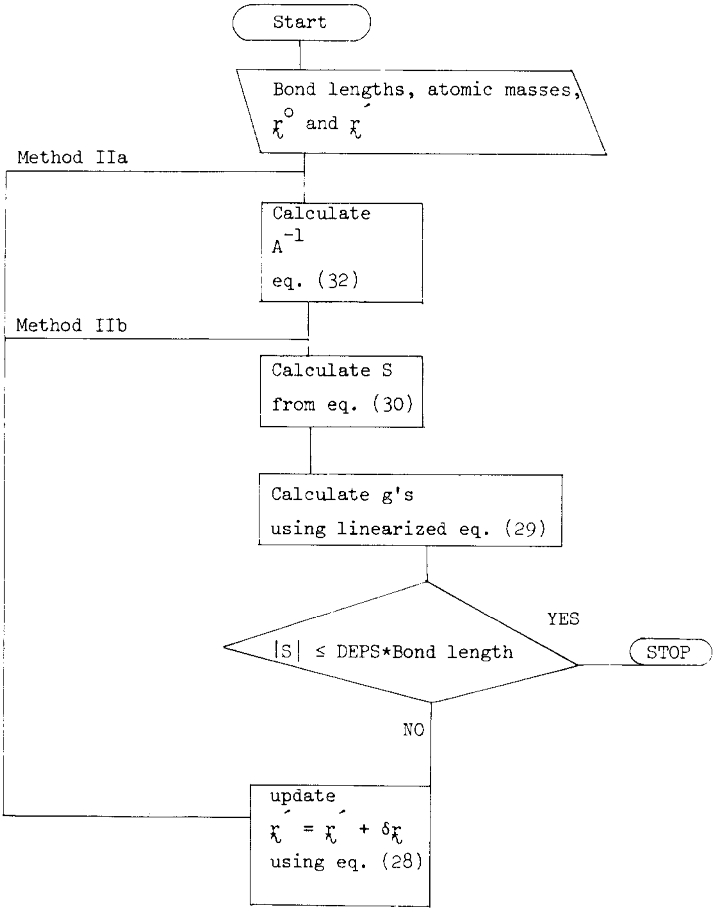

FIG. 2. Flowchart of Procedure "ADJUST," methods IIa and IIb.

within the desired accuracy, DEPS. In the second method (IIb) the inverse matrix A^{-1} is calculated only during the first iteration and the rest of the process is carried out as before. Thus if Eq. (28) is formally written as

$$\mathbf{f}(g) = 0,$$

the iterative method IIa can be described as

$$\mathbf{g}^{N+1} = \mathbf{g}^{N} = \{\mathbf{f}'(g^{N})\}^{-1}\mathbf{f}(g^{N}),$$

where $\mathbf{f}(g^{N}) \equiv \mathbf{S}$ (Eq. (30)) and $\mathbf{f}'(g^{N}) \equiv \mathbf{A}$ (Eq. (31)). Similarly method IIb can be described as

$$\mathbf{g}^{N+1} = \mathbf{g}^{N} - \{f'(g^{0})\}^{-1}\mathbf{f}(g^{N}).$$

It is important to note here that the value of $\mathbf{r}'$ in these calculation are updated at each iteration by $\mathbf{r}' + \delta\mathbf{r}(g^{N})$. A flowchart for these two methods is given in Fig. 2.

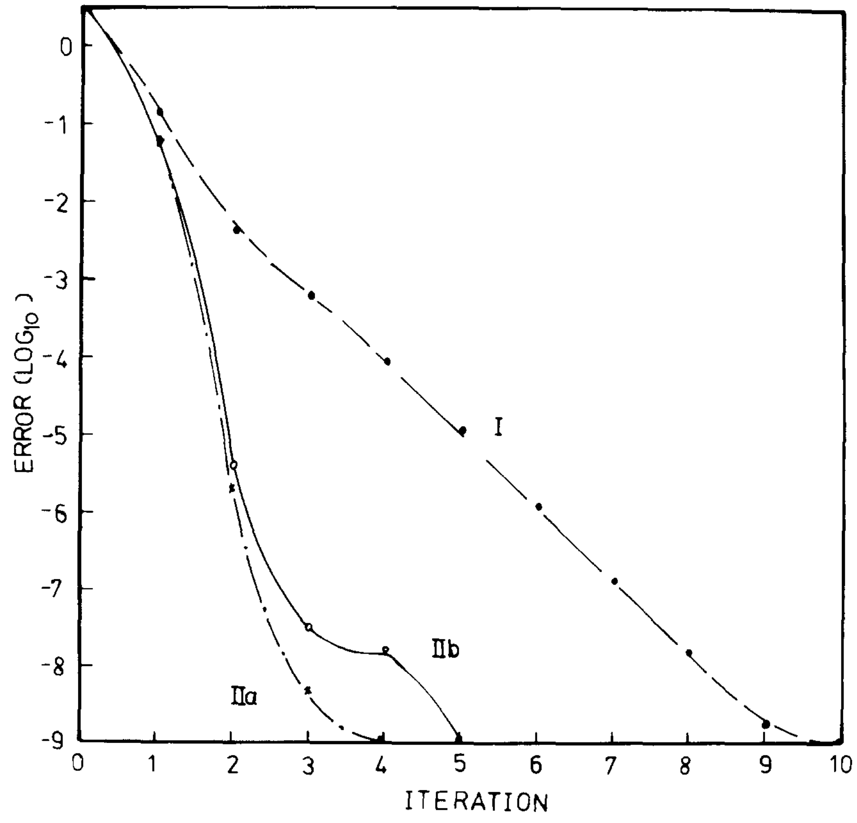

FIG. 3. Convergence of "ADJUST" methods percentage error at each iteration.

TABLE II

Relative Efficiency of "ADJUST"

	Time for matrix inversion (10^{-4} sec)		Total time (10^{-4} sec)
Method I	6	10	28
Method IIa	7	4	30
Method IIb	7	5	20

4.2. *Comparison of method* I *and method* II

We have simulated three water systems using the same initial conditions but different "ADJUST" procedures (as defined above) on the CDC 7600. The constraints in each calculations are fulfilled to an accuracy, DEPS $= 10^{-9} *$ bond length. The positions and velocities in each simulation were found to be in complete agreement up to eleven decimal places. By measuring the time taken by each "ADJUST" procedure to calculate the constraint forces we found method IIb to be the most economic. For the sake of comparison, in Fig. 3 we have plotted the error in bond length against the number of iterations for all three procedures. Here we see the Ryckaert *et al.* [12] method (I) takes ten iterations to achieve an accuracy of the order 10^{-9} of the bond length whereas the new methods IIa and IIb take four and five iterations, respectively, to achieve the same accuracy. The same unconstrained and constrained positions were used in all these calculations. Although method IIa takes fewer iterations it is about 50% slower than method IIb because each matric inversion takes a comparatively large amount of time. In Table II we have given the time statistics for all three methods. The time taken to perform each matrix inversion including the calculation of its elements is 6 μsec for method I, compared with 7 μsec for method II; in the latter, calculation of each element requires an additional division operation to calculate the unit vectors (Eq. (31)). The total time for performing the calculation for a single molecule by method I is about 0.8 msec more than the time taken by method IIb.

5. Conclusions

The results of the above section clearly show that the most efficient and economic "ADJUST" procedure is method IIb. This procedure is based on the constraint equation

$$c_{ij} = |\mathbf{r}_i - \mathbf{r}_j| - d_{ij}$$

and uses a single matrix inversion method for the iterative solution. All of the methods discussed here are found to be equivalent with regard to the calculations of positions, velocities and thermodynamic quantities. The conservation laws, e.g., total energy, momentum and angular momentum, etc., are also satisfied.

ACKNOWLEDGMENTS

We would like to thank Dr. M. Amini and Professor H. J. C. Berendsen for fruitful discussions and helpful comments. One of us (M.K.M.) is also grateful to the Sind University and Government of Pakistan, for financial support. We also acknowledge the helpful comments by one of the referees of this paper.

REFERENCES

1. B. J. ALDER AND T. E. WAINWRIGHT, *J. Chem. Phys.* **31** (1959), 459.
2. A. RAHMAN, *Phys. Rev. A* **405** (1964), 136.
3. L. VERLET, *Phys. Rev.* **159** (1967), 98.
4. D. SCHIFF, *Phys. Rev.* **186** (1969), 151.
5. J. B. BERNE AND G. D. HARP, *Adv. Chem. Phys.* **1** (1970), 515.
6. L. V. WOODCOCK, *Chem. Phys. Lett.* **10** (1971), 257.
7. A. RAHMAN AND F. H. STILLINGER, *J. Chem. Phys.* **55** (1971), 3336.
8. R. M. J. COTTERILL, W. D. KRISTENSEN, AND E. J. JENSEN, *Phil Mag.* **30** (1974), 245.
9. M. AMINI AND R. W. HOCKNEY, *J. Non-Cryst. Solids* **31** (1979), 447.
10. J. BAROJAS, D. LEVESQUE, AND B. QUENTREC, *Phys. Rev. A* 7 (1973), 1092.
11. J. P. RYCKAERT AND A. BELLEMANS, *Chem. Phys. Lett.* **30** (1975), 123.
12. J. P. RYCKAERT, G. CICCOTTI, AND H. J. C. BERENDSEN, *J. Comput. Phys.* **23** (1977), 327.
13. W. P. VAN GUNSTEREN AND H. J. C. BERENDSEN, *Mol. Phys.* **34** (1977), 1311.
14. H. GOLDSTEIN, "Classical Mechanics," Chap. 1, Addison–Wesley, Reading, Mass., 1977.
15. M. AMINI, J. W. EASTWOOD, AND R. W. HOCKNEY, "Time integration in Particle Models," Reading University Science Report RCS 127, Reading University, U. K. (Jan. 1980).
16. J. W. EASTWOOD, D. LAWRENCE, AND R. W. HOCKNEY, *Comput. Phys. Commun.* **19** (1980), 215.
17. L. PAULING, "The Nature of Chemical Bond," p. 224, Cornell Univ. Press, New York, 1960.
18. H. L. LEMBERG AND F. H. STILLINGER, *J. Chem. Phys.* **62** (1975), 1677.
19. H. POPKIE, H. KISTENMACHER, AND E. CLEMENTI, *J. Chem. Phys.* **59** (1973), 1325.