JOURNAL OF COMPUTATIONAL PHYSICS 109, 318–328 (1993)

Dangers of Multiple Time Step Methods*

JEFFREY J. BIESIADECKI†

Jet Propulsion Laboratory, 4800 Oak Grove Drive, M/S 301–345, Pasadena, California 91109

AND

ROBERT D. SKEEL

Department of Computer Science and Beckman Institute, University of Illinois at Urbana-Champaign, 1304 West Springfield Avenue, Urbana, Illinois 61801-2987

Received January 4, 1993

In this work we demonstrate two potential dangers of multiple time step techniques for numerical integration of differential equations. This idea of using different time steps for different interactions was proposed in 1978 for molecular dynamics but undoubtedly has other useful applications. The use of multiple time stepping with the popular Verlet method was proposed in a 1991 paper. However, the method advocated in this paper does not retain the symplectic (or canonical) property of the Verlet method, which is an abstract property satisfied by the flow of any Hamiltonian system, of which molecular dynamics is an example. Recent work reported in the literature suggests that this property is important for the long-time integration of Hamiltonian dynamical systems. We perform experiments on linear and nonlinear problems comparing symplectic and nonsymplectic multiple time stepping extensions of the Verlet method. We observe that in the nonsymplectic case either instability or dissipation becomes evident after a long integration. However, our experiments also indicate that it is quite possible to obtain an artificial "resonance" for the symplectic method that is much worse than that for the nonsymplectic methods.

1. INTRODUCTION

In this work we demonstrate two potential dangers of multiple time step techniques for numerical integration of differential equations. The use of different time steps for different interactions was proposed [10] in 1978 for molecular dynamics (MD), but the idea undoubtedly has other useful applications. In molecular dynamics only a small percentage of interactions require a very small step size in order to be resolved accurately; for most interactions much larger time steps are adequate. This idea is developed further in a recent paper [6], which uses multiple time steps in a way that generalizes the popular Encke/Störmer/leapfrog/Verlet method. The theme of this paper is to evaluate different electrostatic interactions with different time increments depending on the distance. Pairs of particles are grouped into distance classes whose memberships change as the particles move. Method coefficients are chosen so that if all particle pairs were to belong to one distance class, the method would reduce to a single time step Verlet method. A speedup by a factor of 5.4 was observed with no significant loss of accuracy. (A popular alternative is to ignore completely long-range interactions beyond a certain "cut off" distance, but this can significantly change the behavior of the molecule [1, p. 155].)

Recent work reported in the literature suggests that for the long-time integration of Hamiltonian dynamical systems, such as occurs in molecular dynamics, one should use methods that preserve the symplectic (or canonical) property of the *flow*. The multiple time stepping method advocated in [6] does not retain the symplectic property of the Verlet method and it requires additional storage of past positions. We have performed experiments comparing this method to a symplectic method and to another nonsymplectic method, neither of which require the additional storage. Experiments on linear and nonlinear problems (very small systems that model bondlength and Lennard–Jones interactions, respectively) show striking superiority of the symplectic methods on long enough time intervals. We observe that for nonsymplectic multiple time stepping either instability or dissipation becomes evident after a long integration.

On the other hand, it is suggested [6] that resonance may be a problem for the symplectic method, and we confirm this for linear and nonlinear problems. We show that it is quite possible to obtain an artificial "resonance" for the symplectic method that is much worse than that for the nonsymplectic methods. This situation arises because the large

* This work was supported by National Science Foundation Grant DMS 90 15533 and National Institutes of Health Grant PHS 2 P41 RR05969-03.

† Work performed while at the University of Illinois.

0021-9991/93 $5.00

time step discretization (which resolves the slowly varying interactions) introduces periodic impulses of a low enough frequency to excite natural high frequencies arising from the fast interactions, which are being integrated with a smaller time step. The resonance waxes and wanes on a short timescale. However, it might be the case that the irregularities of a large nonlinear system make such resonance very unlikely.

The simple experiments reported in this paper are for situations in which the step size used for each interaction remains fixed. This is insufficient for nonbonded interactions, and we are currently investigating a strategy for permitting the step size to increase or decrease as the distance between interacting particles changes without sacrificing symplecticness.

Section 2 states the equations used for molecular dynamics, and Section 3 defines the property of being symplectic. Section 4 describes the multiple time step algorithms, and these are compared in Sections 5 and 6 with three numerical experiments.

2. MOLECULAR DYNAMICS SIMULATIONS

A molecular dynamics simulation of $\mathcal{N}$ particles based on principles of Newtonian mechanics requires the solution of the following system of second-order ordinary differential equations:

$$m_k \frac{d^2}{dt^2}\mathbf{r}_k(t) = \mathbf{F}_k(\mathbf{r}), \qquad k = 1, 2, ..., \mathcal{N}, \tag{1}$$

where m_k, $\mathbf{r}_k$, and $\mathbf{F}_k$ are the mass of, position of, and force acting upon particle k, respectively. The force $\mathbf{F}_k = -\nabla_k V(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{\mathcal{N}})$, where the potential energy V is a sum of contributions from nonbonded two-body forces and bonded 2-, 3-, and 4-body forces. The nonbonded forces consist of Coulomb forces due to charges (or assigned fractional charges) of particles and the Van der Waals forces, and the number of such interactions is $\frac{1}{2}\mathcal{N}(\mathcal{N}-1)$, which is much greater than the number of bonded interactions.

The Verlet algorithm [12] is a standard way of solving system (1). It obtains the positions of the particles at discrete time steps, ih, where i is the step number and h is the size of an individual time step. Defining the vector $\mathbf{x} = (\mathbf{r}_1^T, \mathbf{r}_2^T, ..., \mathbf{r}_{\mathcal{N}}^T)^T$ and the vector $\mathbf{f} = (\mathbf{F}_1^T/m_1, \mathbf{F}_2^T/m_2, ..., \mathbf{F}_{\mathcal{N}}^T/m_{\mathcal{N}})^T$, the algorithm approximately satisfies $\ddot{\mathbf{x}}(ih) = \mathbf{f}(\mathbf{x}(ih))$ for each step i of the simulation. The formula used by the Verlet algorithm to obtain approximations $\mathbf{x}_i \approx \mathbf{x}(ih)$ to new particle positions is

$$\mathbf{x}_{i+1} = 2\mathbf{x}_i - \mathbf{x}_{i-1} + h^2\mathbf{f}_i, \tag{2}$$

which is second-order accurate. If velocities are wanted, the Verlet algorithm specifies that they be calculated with

$$\dot{\mathbf{x}}_i = \frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2h}. \tag{3}$$

Typically one would start a simulation knowing $\mathbf{x}_0$ and $\dot{\mathbf{x}}_0$, so Eq. (3) can be used to eliminate the reference to $\mathbf{x}_{-1}$ from (2) to obtain for $\mathbf{x}_1$ Eq. (7) of Section 4.1.

3. SYMPLECTIC INTEGRATION METHODS

Hamiltonian systems of differential equations have special properties that are retained by symplectic (also called canonical) integration methods. A Hamiltonian is a function $H(\mathbf{q}, \mathbf{p})$, where $\mathbf{q}, \mathbf{p} \in \mathbb{R}^d$. The system of differential equations associated with H is defined to be [9]

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \qquad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, \qquad 1 \leqslant i \leqslant d. \tag{4}$$

(Here subscript i refers to the ith element of a vector.) Equivalently, (4) can be written as

$$\dot{\mathbf{z}} = J \cdot \nabla H, \qquad \text{where} \quad \mathbf{z} = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix}, \qquad J = \begin{pmatrix} 0 & I_d \\ -I_d & 0 \end{pmatrix}.$$

In molecular dynamics simulations, the Hamiltonian of interest is the formula for total energy expressed as a function of positions and momenta. There are $d = 3\mathcal{N}$ degrees of freedom, $\mathbf{q}$ contains the position of each particle (referred to as $\mathbf{x}$ earlier), and $\mathbf{p}$ contains the momentum of each particle. Define M to be the diagonal matrix of particle masses, with

$$M_{3k-2,3k-2} = M_{3k-1,3k-1} = M_{3k,3k} = m_k, \qquad k = 1, 2, ..., \mathcal{N}.$$

Then, the Hamiltonian is given by

$$H(\mathbf{q}, \mathbf{p}) = \tfrac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + V(\mathbf{q}), \tag{5}$$

where the first term is the kinetic energy and the second is the potential energy. The force is $-\nabla V(\mathbf{q})$.

The *t-flow* of a Hamiltonian system [9] is the mapping ϕ_t from $\mathbf{z}(0)$ to $\mathbf{z}(t)$ which is the solution of (4) at time t with initial values $\mathbf{z}(0)$. A quick example, from [9], is the Hamiltonian with $d = 1$ and $H(q, p) = \frac{1}{2}q^2 + \frac{1}{2}p^2$, for which the flow is

$$\phi_t(q, p) = \begin{pmatrix} q\cos t + p\sin t \\ -q\sin t + p\cos t \end{pmatrix}.$$

Flow of a Hamiltonian system has an area-preserving property [2]. Let Σ be an oriented two-dimensional surface in $\mathbb{R}^{2d}$. In the case of a Hamiltonian with one degree of freedom, the signed area of Σ is equal to the signed area of $\phi_t(\Sigma)$. With multiple degrees of freedom, the sum of the signed areas of the projection of Σ onto the (p_i, q_i)-plane, $i = 1, 2, ..., d$, is preserved. Any mapping that possesses this property for arbitrary Σ is called symplectic or canonical.

One way to check if a mapping χ is symplectic is to see if

$$(\nabla\chi)^T J(\nabla\chi) = J, \tag{6}$$

where $\nabla\chi$ is the Jacobian matrix of the mapping. This equation and other characterizations are discussed in [9]. It is easy to verify that (6) holds for the example earlier.

It will be useful to know that the composition of two symplectic mappings $\phi = \phi^{[2]} \circ \phi^{[1]}$ is also symplectic and that the inverse of a symplectic mapping is symplectic.

Numerical approximations ψ_h to ϕ_h are sought to calculate positions and momenta through iteration,

$$\mathbf{z}_{n+1} = \psi_h(\mathbf{z}_n) = \underbrace{\psi_h(\psi_h(\cdots\psi_h}_{n+1}(\mathbf{z}_0)\cdots)) \approx \phi_{(n+1)h}(\mathbf{z}_0).$$

It is desirable for ψ_h to be symplectic itself, at least disregarding round-off error; this mimics certain properties of the physical system. For example, because of the area-preservation of phase space, the system will not "damp out" under repeated iteration. In particular, there is considerable experimental evidence [3, 7, 8] suggesting symplectic methods perform better than nonsymplectic ones.

4. MULTIPLE TIME STEP ALGORITHMS

Three multiple time step methods are compared: Verlet-I and Verlet-II, which are introduced by Grubmüller *et al.* [6], and Verlet-X, which is new. Simplified versions of these algorithms will be presented where it is assumed that there are only two distance classes, "near" and "far." Forces $\mathbf{f}^{\text{near}}$ in the "near" class are those that are changing rapidly, so they will be evaluated every time step. The rest of the forces $\mathbf{f}^{\text{far}}$ are in the "far" class and will be evaluated only every *macro*-step, because they change more slowly. The length of a macro-step is N times the length of a normal time step, for some integer N. The more general case would be to have an arbitrary number of distance classes. In [6], distance classes are numbered zero through n, with forces in class j (referred to as $\mathbf{f}^{(j)}$) evaluated every 2^j time steps and $N = 2^n$.

Because particles move during the course of a simulation, forces between them may change distance classes. But in this paper membership will be considered fixed. Problems that arise even with this easier special case are illustrated in Sections 5 and 6.

A constraint imposed on multiple time step algorithms by [6] is that they satisfy "Verlet equivalence." This means that if all forces are in a class with step size Nh, then the method should give the same results as the standard Verlet algorithms with step size Nh.

4.1. *Verlet Algorithm*

The standard Verlet algorithm, given earlier with (2) and (3), can be rewritten in such a way that velocities are made use of in the calculation of new positions. The description of distance-class methods will use the formulas derived below, but differ in how the $\mathbf{f}$ term (which specifies the forces to evaluate) is defined.

First, to obtain an equation for $\mathbf{x}_{i+1}$, use (3) to eliminate $\mathbf{x}_{i-1}$ from (2). Rearrangement yields

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h\dot{\mathbf{x}}_i + \frac{h^2}{2}\mathbf{f}_i. \tag{7}$$

Then, to obtain an equation for $\dot{\mathbf{x}}_{i+1}$ use (3) and later plug in (2) to see

$$\begin{aligned}\dot{\mathbf{x}}_{i+1} - \dot{\mathbf{x}}_i &= \frac{\mathbf{x}_{i+2} - \mathbf{x}_i}{2h} - \frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2h} \\ &= \frac{\mathbf{x}_{i+2} - 2\mathbf{x}_{i+1} + \mathbf{x}_i}{2h} + \frac{\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}}{2h} \\ &= \frac{1}{2h}(h^2\mathbf{f}_{i+1} + h^2\mathbf{f}_i).\end{aligned}$$

Rearranged, this is

$$\dot{\mathbf{x}}_{i+1} = \dot{\mathbf{x}}_i + \frac{h}{2}\mathbf{f}_i + \frac{h}{2}\mathbf{f}_{i+1}. \tag{8}$$

Equations (7) and (8) minimize round-off error and define a method proposed in [11].

Finally, notation is used that will be convenient for distance-class methods, and an intermediate velocity term is introduced to obtain

$$\dot{\mathbf{x}}_{Ni+k+1/2} = \dot{\mathbf{x}}_{Ni+k} + \frac{h}{2}\mathbf{f}_{Ni+k} \tag{9}$$

$$\mathbf{x}_{Ni+k+1} = \mathbf{x}_{Ni+k} + h\dot{\mathbf{x}}_{Ni+k+1/2} \tag{10}$$

$$\dot{\mathbf{x}}_{Ni+k+1} = \dot{\mathbf{x}}_{Ni+k+1/2} + \frac{h}{2}\mathbf{f}_{Ni+k+1}, \tag{11}$$

where i is the current macro-step, k ranges from zero to $N-1$, and $Ni+k$ is the current time step. For the standard method (without multiple time steps) $\mathbf{f}^{\text{far}}$ is evaluated every step, so the force term is just

$$\mathbf{f}_{Ni+j} = \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j}) + \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni+j}).$$

Equation (9) is the transformation

$$\psi^{[1]}(\mathbf{q}, \mathbf{p}) = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} + \dfrac{h}{2}M\mathbf{f}(\mathbf{q}) \end{pmatrix}$$

which maps

$$\begin{pmatrix} \mathbf{x}_{Ni+k} \\ M\dot{\mathbf{x}}_{Ni+k} \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{x}_{Ni+k} \\ M\dot{\mathbf{x}}_{Ni+k+1/2} \end{pmatrix}.$$

By using (6) and the fact that the Jacobian matrix of $M\mathbf{f}$ is symmetric (because it is the Hessian of the function $-V$), $\psi^{[1]}$ is symplectic. Mapping $\psi^{[2]}$ for Eq. (10) is similarly seen to be symplectic; $\psi^{[1]}$ is used again for (11). Therefore, the Verlet method is symplectic, since it is just the composition of these mappings,

$$\psi = \psi^{[1]} \circ \psi^{[2]} \circ \psi^{[1]}.$$

4.2. *Verlet-I Algorithm*

This method scales forces in each class by the ratio of its step size to the smallest step size. In [6],

$$\mathbf{f}_{Ni+j} = \sum_{l=0}^{n} 2^l \delta_{j,\, 2^l \lfloor j/2^l \rfloor} \mathbf{f}^{(l)}(\mathbf{x}_{Ni+2^l \lfloor j/2^l \rfloor}),$$

where $\delta_{i,j} = 1$ if $i = j$, and 0 otherwise.

In general the time steps of each class need not be h times some power of two. Using the assumption of having only two different time steps, one obtains

$$\mathbf{f}_{Ni+j} = \begin{cases} \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j}) + N\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}), & j = 0, \\ \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j}), & j = 1, 2, \ldots, N-1. \end{cases}$$

The mapping corresponding to one macro-step of Verlet-I is the composition of simple symplectic mappings and thus is also symplectic.

Note that "far" forces are felt only at the beginning and end of each macro-step. A physical interpretation would be two equally large impulses at time $t = Nih-$ and $t = Nih+$. A possible problem arising from this scheme will be discussed and demonstrated in Section 6.

4.3. *Verlet-II Algorithm*

This algorithm includes forces from the previous macro-step to increase accuracy, and to spread out the effects of "far" forces over the entire macro-step (unlike Verlet-I). Forces are to be of the form

$$\mathbf{f}_{Ni+j} = \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j}) + a_j \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + b_j \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N})$$

for some choice of coefficients a_j and b_j. The idea is to approximate linear extrapolation of the "far" force, giving

$$\begin{aligned} &a_j \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + b_j \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N}) \\ &\quad \approx \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + \frac{j}{N}(\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) - \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N})). \end{aligned} \quad (12)$$

However, the choice of a_j and b_j that satisfies extrapolation exactly gives a method that does not satisfy Verlet equivalence [6]. The Verlet-II algorithm is derived by finding coefficients that are as close as possible to those in linear extrapolation with the constraint of Verlet equivalence.

The result of a least-squares derivation obtained in [6] for a full set of distance classes is

$$\begin{aligned} \mathbf{f}_{Ni+j} = \sum_{l=0}^{n} &(a^{(l)}_{j-2^l\lfloor j/2^l \rfloor} \mathbf{f}^{(l)}(\mathbf{x}_{Ni+2^l\lfloor j/2^l \rfloor}) \\ &+ b^{(l)}_{j-2^l\lfloor j/2^l \rfloor} \mathbf{f}^{(l)}(\mathbf{x}_{Ni+2^l\lfloor j/2^l \rfloor - 2^l})) \end{aligned}$$

with coefficients

$$a^{(l)}_j = 3 \cdot 2^{l+1} \frac{(2^{2l} - 2^l + 1) - j(2^l - 2)}{(2^l + 1)(2^{2l+1} + 1)},$$

$$b^{(l)}_j = \frac{2(-2^{2l+1} + 3 \cdot 2^l - 1) + 6j(2^l - 1)}{2^{2l+1} + 1}.$$

With our assumption that forces are in the innermost ($\mathbf{f}^{(0)} \equiv \mathbf{f}^{\text{near}}$) or outermost ($\mathbf{f}^{(n)} \equiv \mathbf{f}^{\text{far}}$) distance classes only, this leads to

$$\begin{aligned} \mathbf{f}_{Ni+j} = &a^{(0)}_0 \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j}) + b^{(0)}_0 \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j-1}) \\ &+ a^{(n)}_j \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + b^{(n)}_j \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N}). \end{aligned}$$

Note that $a^{(0)}_0 = 1$, $b^{(0)}_0 = 0$, and $N = 2^n$, so the superscripts are dropped and the simplified version of Verlet-II is

$$\mathbf{f}_{Ni+j} = \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j}) + a_j \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + b_j \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N}),$$

$$a_j = 6N \frac{(N^2 - N + 1) - j(N-2)}{(N+1)(2N^2+1)},$$

$$b_j = \frac{2(-2N^2 + 3N - 1) + 6j(N-1)}{2N^2 + 1}.$$

This method is not symplectic. One attractive feature it has, however, is that the coefficients a_j and b_j are bounded, unlike coefficients used in Verlet-I and Verlet-X, which makes Verlet-II less sensitive to variations in the forces.

4.4. *Verlet-X Algorithm*

This algorithm also approximates linear extrapolation. Rearrange the left side of (12) to obtain

$$\begin{aligned} &(a_j + b_j)\, \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) - b_j(\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) - \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N})) \\ &\quad \approx \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + \frac{j}{N}(\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) - \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N})). \end{aligned} \quad (13)$$

Presumably $\|\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) - \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N})\| \ll \|\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni})\|$, so the error will be smaller if the left terms of (13) match.

This means $a_j + b_j = 1$. The formulas for satisfying Verlet equivalence derived in [6, p. 129] are

$$\sum_{k=0}^{N-1} (N-k)\, a_k = N^2,$$

$$\sum_{k=1}^{N-1} k a_k + \sum_{k=0}^{N-1} (N-k)\, b_k = 0, \tag{14}$$

$$\sum_{k=1}^{N-1} k b_k = 0.$$

With the given restriction on a_j and b_j, the equations above reduce to

$$\sum_{k=0}^{N-1} k a_k = \frac{N(N-1)}{2}, \qquad \sum_{k=0}^{N-1} a_k = \frac{3N-1}{2}. \tag{15}$$

If $b_1 = b_2 = \cdots = b_{N-1} = 0$ (which saves storage because $\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N})$ can be discarded before $\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni})$ is computed), then $a_1 = a_2 = \cdots = a_{N-1} = 1$. The condition on the left in (15) is now satisfied, and the right condition gives $a_0 = (N+1)/2$ and $b_0 = (1-N)/2$. The method is then

$$\mathbf{f}_{Ni+j} = \begin{cases} \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j}) + \dfrac{N+1}{2}\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + \dfrac{1-N}{2}\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N}), \\ \qquad j = 0, \\ \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni+j}) + \mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}), \\ \qquad j = 1, 2, \ldots, N-1. \end{cases}$$

This method is not symplectic either.

4.5. *Starting Verlet-II and Verlet-X*

The Verlet-I algorithm is self-starting. Verlet-II and Verlet-X, on the other hand, require $\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N})$, which is not available for the first macro-step.

One possibility, used in [6], is to begin the simulation with all forces in the "near" class, and move "far" forces out to farther classes as the run progresses. The problem with this idea is that Verlet equivalence will be violated when switching methods at time $t = Nh$. This can be fixed by allowing the force used at the end of one time step (in Eq. (11)) and the beginning of the next (in Eq. (9)) to be different as long as the velocity used in (10) works out to be the same. This matters only between macro-steps when switching methods. Let $\mathbf{f}_{Ni}^{+}$ be the force used in (9) when $k = 0$ and $\mathbf{f}_{Ni+N}^{-}$ the force in (11) when $k = N-1$. Using

$$\mathbf{f}_{Ni}^{\pm} = \mathbf{f}^{\text{near}}(\mathbf{x}_{Ni}) + 2a_0^{\pm}\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + 2b_0^{\pm}\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N}),$$

it is then required that

$$a_0^{-} + a_0^{+} = a_0,$$
$$b_0^{-} + b_0^{+} = b_0.$$

To determine $a_0^{\pm}$ and $b_0^{\pm}$ that satisfy Verlet equivalence, Eq. (9) through (11) are used. Assuming $\mathbf{f}^{\text{near}} \equiv \mathbf{0}$, one macro-step of an extrapolation method yields

$$\dot{\mathbf{x}}_{Ni+N} = \dot{\mathbf{x}}_{Ni} + h\left(b_0^{+} + \sum_{k=1}^{N-1} b_k\right)\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N}) + h\left(a_0^{+} + \sum_{k=1}^{N-1} a_k + b_0^{-}\right)\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}) + h a_0^{-}\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni+N}),$$

$$\mathbf{x}_{Ni+N} = \mathbf{x}_{Ni} + Nh\dot{\mathbf{x}}_{Ni} + h^2\left(Nb_0^{+} + \sum_{k=1}^{N-1}(N-k)\, b_k\right)\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N}) + h^2\left(Na_0^{+} + \sum_{k=1}^{N-1}(N-k)\, a_k\right)\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni}).$$

And comparing the above to one step of standard Verlet with step size Nh (to ensure Verlet equivalence) gives

$$b_0^{+} + \sum_{k=1}^{N-1} b_k = 0,$$

$$a_0^{+} + \sum_{k=1}^{N-1} a_k + b_0^{-} = \frac{N}{2},$$

$$a_0^{-} = \frac{N}{2},$$

$$Nb_0^{+} + \sum_{k=1}^{N-1}(N-k)\, b_k = 0,$$

$$Na_0^{+} + \sum_{k=1}^{N-1}(N-k)\, a_k = \frac{N^2}{2}.$$

Equations (14) are used to eliminate $\sum_{k=1}^{N-1} k a_k$, $\sum_{k=0}^{N-1} b_k$, and $\sum_{k=1}^{N-1} k b_k$, to obtain

$$a_0^{-} = \frac{N}{2}, \qquad b_0^{-} = N - \sum_{k=0}^{N-1} a_k.$$

Note that for the Verlet-X method $b_0^{+} = 0$. This is important because otherwise $\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni-N})$ would be needed after $\mathbf{f}^{\text{far}}(\mathbf{x}_{Ni})$, ruining the storage-saving feature of Verlet-X.

5. NUMERICAL EXPERIMENTS: LONG-TIME EFFECTS

The model problems described below are the simplest systems that illustrate the effects of the algorithms tested.

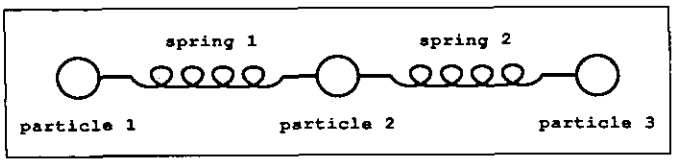

FIG. 1. Spring experiment, system of particles.

The following two sets of experiments show how the nonsymplectic methods Verlet-II and Verlet-X suffer from either instabilities or artificial dissipation, with both linear and nonlinear forces.

The primary metric used here to quantify the accuracy of the results of a simulation is conservation of energy. This measure is convenient, and it has proven to be a useful gauge of accuracy for methods not specifically designed to conserve energy. Very often determining properties of the system as a whole is what is desired, and it is acceptable if particle trajectories are not at all accurate [1, pp. 76–77].

It should be noted that a distance-class algorithm is intended to be run one full macro-step at a time, and positions should not be recorded and statistics sampled in the middle of a macro-step.

5.1. *Experiments with Linear Forces*

The first experiment is a one-dimensional simulation of three particles bound together by two springs of different stiffnesses, with no other forces considered. (This is a way of modeling bondlength forces.) The leftmost particle (particle 1) is bound to the middle particle (particle 2) by spring 1; the middle particle is also bound to the rightmost particle (particle 3) by spring 2 (Fig. 1).

The potential energy of the system is

$$V(r_1, r_2, r_3) = \tfrac{1}{2}k_1(|r_2 - r_1| - l_1)^2 + \tfrac{1}{2}k_2(|r_3 - r_2| - l_2)^2.$$

The parameters and initial conditions for this experiment are given in Table I. The initial total energy of the system is

TABLE I

Parameters and Initial Conditions for Experiment 1

Particle masses	$m_1 = m_2 = m_3 = 1$
Initial positions	$r_1 = -7, r_2 = 0, r_3 = 7$
Initial velocities	All particles at rest
Spring stiffnesses	$k_1 = 16, k_2 = 1$
Equilibrium spring lengths	$l_1 = 6, l_2 = 6$

$E_0 = 8.5$. Since all of the forces in this experiment are linear, the analytic solution can be determined to compare against the approximations. The fundamental frequencies are $\omega_1 = \sqrt{17 + \sqrt{241}} \approx 5.702997$ and $\omega_2 = \sqrt{17 - \sqrt{241}} \approx 1.214836$. The solution is

$$\mathbf{r}(t) = \begin{pmatrix} -6 + \left(-\frac{1}{2} + \frac{1}{2}\frac{\sqrt{241}}{241}\right)\cos(\omega_1 t) + \left(-\frac{1}{2} - \frac{1}{2}\frac{\sqrt{241}}{241}\right)\cos(\omega_2 t) \\ \frac{15}{2}\frac{\sqrt{241}}{241}\cos(\omega_1 t) - \frac{15}{2}\frac{\sqrt{241}}{241}\cos(\omega_2 t) \\ 6 + \left(\frac{1}{2} - 8\frac{\sqrt{241}}{241}\right)\cos(\omega_1 t) + \left(\frac{1}{2} + 8\frac{\sqrt{241}}{241}\right)\cos(\omega_2 t) \end{pmatrix}.$$

Figure 2 shows the analytic solution for a short amount of time. Formulas for the lengths u_k of each spring are

$$\begin{aligned} u_1(t) &= r_2(t) - r_1(t) \\ &\approx 6 + 0.95\cos(5.7t) + 0.05\cos(1.2t), \\ u_2(t) &= r_3(t) - r_2(t) \\ &\approx 6 - 0.50\cos(5.7t) + 1.5\cos(1.2t). \end{aligned}$$

Because spring 1 is stiffer than spring 2, the force it generates varies more rapidly. So the force of spring 1 is evaluated each time step, while the force of spring 2 is evaluated once every N steps. The error in approximating

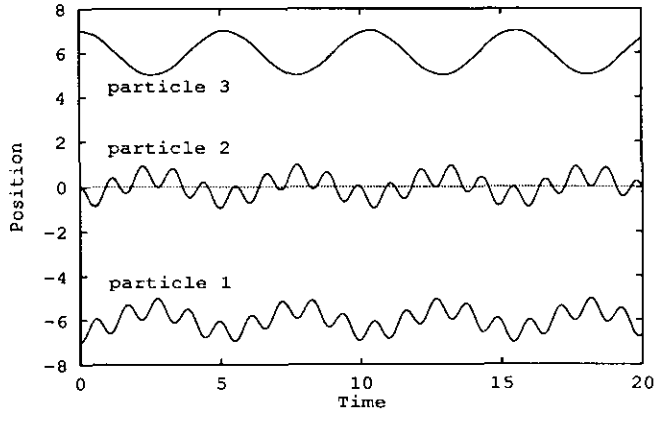

FIG. 2. Spring experiment, analytic solution of particle trajectories.

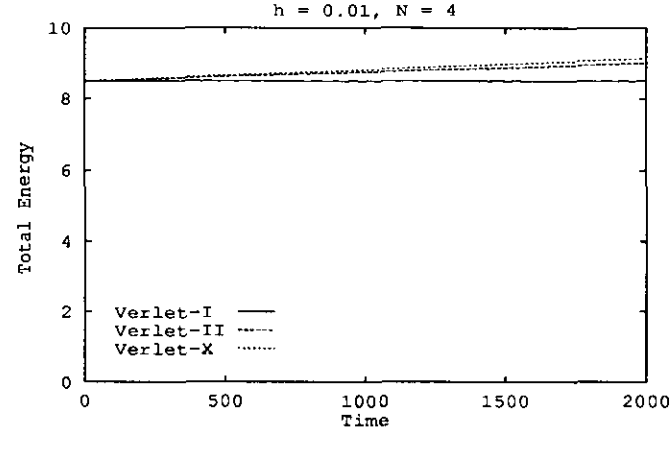

FIG. 3. Spring experiment, stability of the multiple time step algorithms.

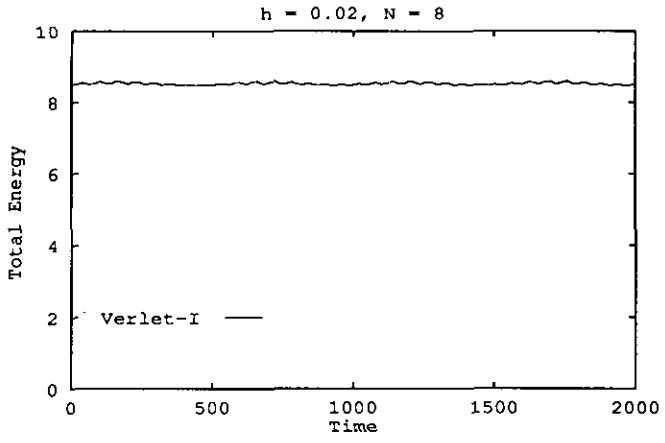

FIG. 4. Spring experiment, stability of Verlet-I with doubled step sizes.

spring 1 with a second-order method will be proportional to $16h^2$ and that from spring 2 will be proportional to $1(Nh)^2$. This suggests that $N=4$ will be most efficient for this problem, because it balances the errors.

Comparisons between the algorithms are made with a total simulation time of $t_{\text{total}} = 2000$. Thirteen values of h are tested, geometrically increasing from 0.001 to 0.112; N is tested at 2, 4, and 8 for each of the h values above.

Figure 3 shows the results of a single run for each method. The nonsymplectic methods Verlet-II and Verlet-X produce substantial energy drift after long-time integration, even when very small time steps are used. The Verlet-I algorithm does not experience this problem and conserves energy well with both N and h doubled (Fig. 4). How rapidly errors accumulate is related to the stability of an algorithm.

Figures 5 and 6 show statistics of all runs together, with two different metrics to quantify the accuracy. Each mark represents a single run with particular N and h values, and statistics were sampled $s = 100$ times for each run. The number of force evaluations done during the entire simulation was compared against the average error in an attempt to quantify the efficiency. The number of force evaluations (η) is equal to the number of macro-steps times the sum of the number of near force evaluations and far force evaluations per macro-step. For this experiment,

$$\eta = \frac{t_{\text{total}}}{Nh}(N+1). \tag{16}$$

For Fig. 5, the average relative error in position (v) is determined using

$$v = \frac{1}{s}\sum_{i=1}^{s}\frac{\|\mathbf{r}_i - \tilde{\mathbf{r}}_i\|_2}{\|\tilde{\mathbf{r}}_i\|_2}, \tag{17}$$

where s is the number of samples taken, and $\mathbf{r}_i$ and $\tilde{\mathbf{r}}_i$ are the computed and analytic solutions, respectively, at the time of the ith sample. Not shown are the results with $N=2$,

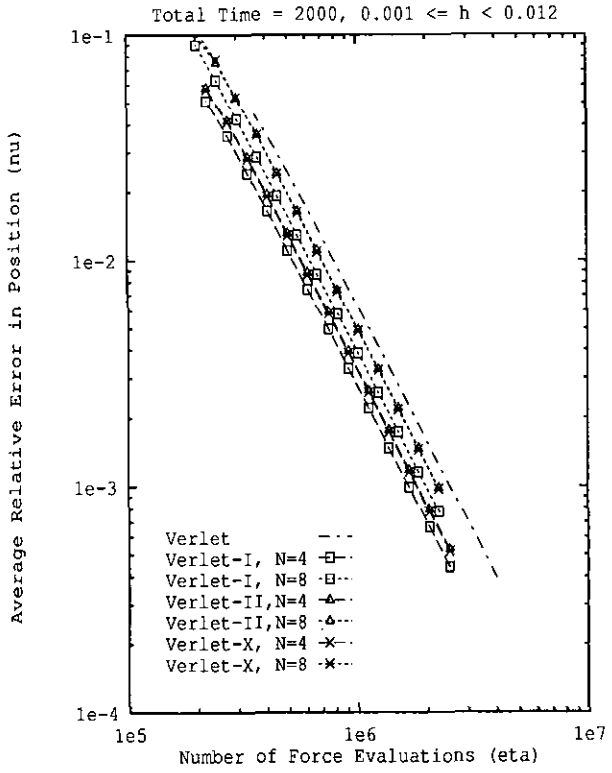

FIG. 5. Spring experiment, accuracy of trajectory.

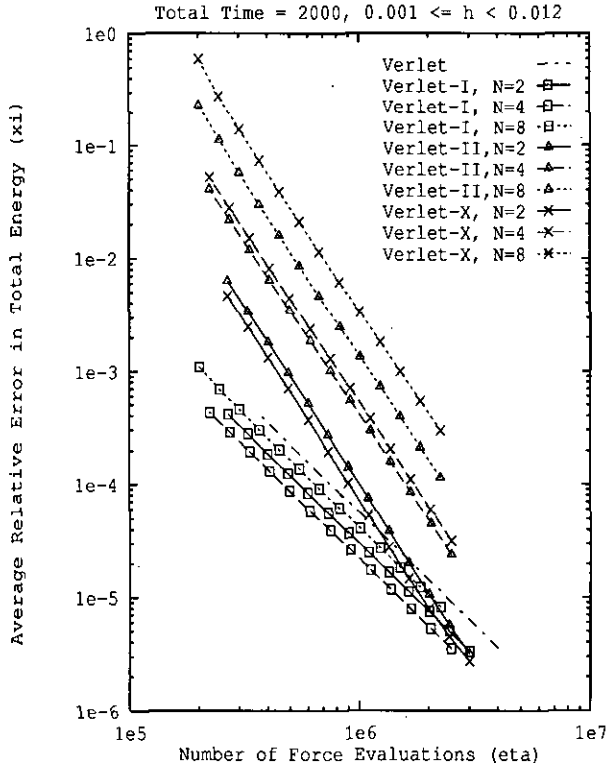

FIG. 6. Spring experiment, accuracy of total energy.

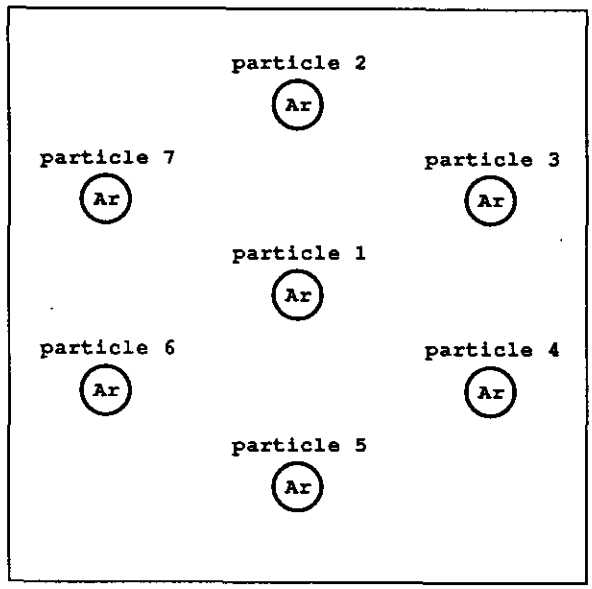

FIG. 7. Argon experiment, system of particles.

because for all methods $N=4$ is the most efficient (needing the fewest number of force evaluations to attain a desired error level) and $N=8$ the least. Verlet-I was the most efficient of the three algorithms for all values of N tested. From the slopes of the lines, each method is second-order accurate (note that h is proportional to $1/\eta$).

For Fig. 6, the average relative error in total energy (ξ) is computed with the formula

$$\xi = \frac{1}{s}\sum_{i=1}^{s}\frac{|E_i - E_0|}{|E_0|}, \tag{18}$$

where s is again the number of samples taken and E_i is the total energy at the time of sample i. Total energy appears to be more sensitive than actual particle positions (perhaps because it involves terms for velocity, which are not taken

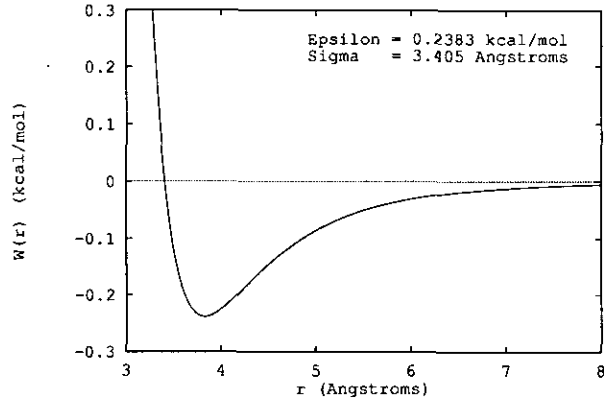

FIG. 8. Lennard–Jones pair potential for argon.

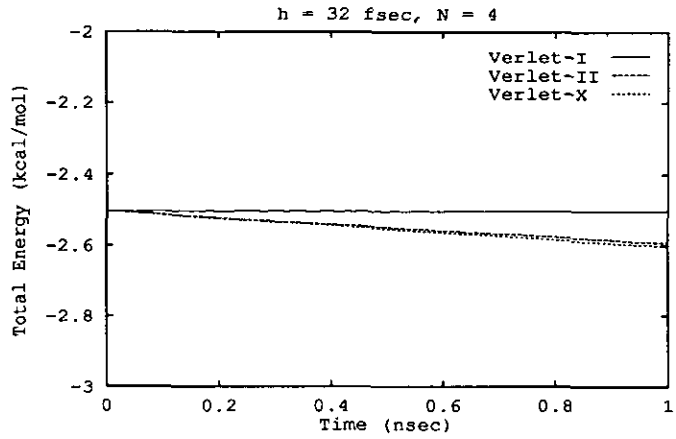

FIG. 9. Argon experiment, stability of the multiple time step algorithms.

into account for Fig. 5). The graph for total energy shows that each of the symplectic methods is far superior to the nonsymplectic methods. Note also that Verlet-I is more efficient than the standard algorithm, even though for this experiment there is not too much room for improvement (the stiffness ratio is only 16 for the system). Again, Verlet-I with $N=4$ is the most efficient. Interestingly, for the other algorithms $N=2$ is most efficient with this metric. The symplectic methods are second-order accurate in energy. The extrapolation methods exhibit a higher order of accuracy for energy (third order) with the range of h tested. The error of the two extrapolation methods is likely to be of the form $h^2e_1(t)+h^3e_2(t)+O(h^4)$ with $e_2 \gg e_1$ so that for fixed t_{total} and small enough h, the second-order effects will dominate. Because they are more accurate, for any fixed t_{total} there may be an h small enough so that an extrapolation method will beat the symplectic methods. This can be seen at the right side of Fig. 6. However, because they are not stable, for any fixed h given sufficiently large t_{total} the extrapolation methods lose to the symplectic methods.

For large enough h and N, Verlet-I also behaves poorly. The results of this experiment, and of other runs with

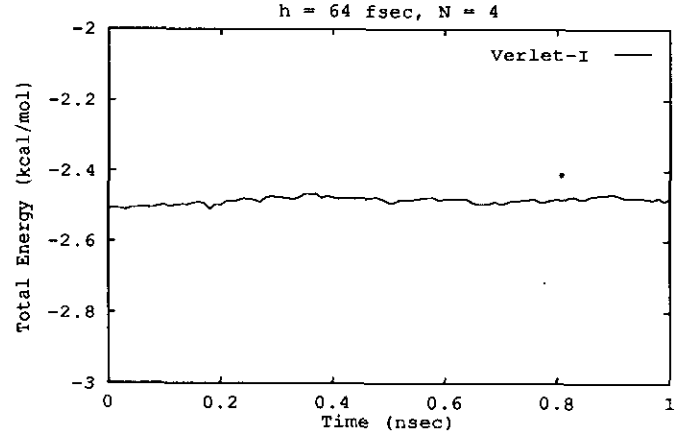

FIG. 10. Argon experiment, stability of Verlet-I with doubled step size h.

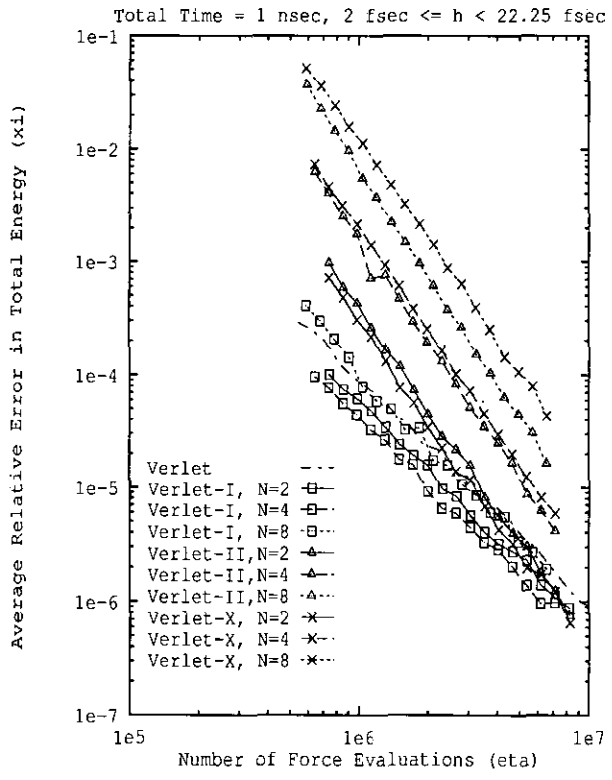

FIG. 11. Argon experiment, accuracy of total energy.

different parameters and initial conditions, indicate that Verlet-II and Verlet-X are unstable for all h but that there exists $h_{\text{threshold}}(N) > 0$ such that Verlet-I is stable for $h < h_{\text{threshold}}(N)$.

5.2. *Experiments with Nonlinear Forces*

This experiment is a simulation of a two-dimensional frozen argon crystal. As shown in Fig. 7, six argon atoms are arranged symmetrically around a center atom with a Lennard–Jones potential between each atom pair. The potential energy of the system is

$$W(r) = 4\varepsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right),$$

$$V(\mathbf{x}) = \sum_{i=1}^{6}\sum_{j=i+1}^{7} W(\|\mathbf{r}_j - \mathbf{r}_i\|_2).$$

Recall that $\mathbf{x} = (\mathbf{r}_1^{\mathrm{T}}, \mathbf{r}_2^{\mathrm{T}}, \ldots, \mathbf{r}_{\mathcal{N}}^{\mathrm{T}})^{\mathrm{T}}$.

The Lennard–Jones potential $W(r)$ is shown in Fig. 8. The position of lowest potential (and no force) is at a distance of $r = \sqrt[6]{2}\,\sigma$, and the potential energy at this point is $-\varepsilon$. The force repulses the particles strongly when they are closer than this and attracts them when they are farther.

TABLE II

Particle masses	39.95 amu $= 66.34 \times 10^{-27}$ kg
Initial positions (Å)	$\mathbf{r}_1 = (0.0, 0.0)$, $\mathbf{r}_2 = (0.2, 3.9)$, $\mathbf{r}_3 = (3.4, 1.7)$, $\mathbf{r}_4 = (3.6, -2.1)$, $\mathbf{r}_5 = (-0.2, -4.0)$, $\mathbf{r}_6 = (-3.5, -1.6)$, $\mathbf{r}_7 = (-3.1, 2.1)$
Initial velocities (m/s)	$\mathbf{v}_1 = (-30, -20)$, $\mathbf{v}_2 = (50, -90)$, $\mathbf{v}_3 = (-70, -60)$, $\mathbf{v}_4 = (90, 40)$, $\mathbf{v}_5 = (80, 90)$, $\mathbf{v}_6 = (-40, 100)$, $\mathbf{v}_7 = (-80, -60)$
ε	120°K $k_{\mathrm{B}} = 120 \times 1.380658 \times 10^{-23}$ joules $= 0.2383$ kcal/mole
σ	3.405 Å

See also [1, pp. 8, 21] for more details and a table of ε and σ for various elements.

The initial positions of the particles are slightly perturbed about those for lowest potential energy (the corners and center of a perfect hexagon with sides of length $\sqrt[6]{2}\,\sigma$). They are given low initial velocities such that the total momentum of the system is zero (see Table II). The initial total energy of the system is -2.5 kcal/mole. Temperature can be calculated with the formula

$$T = \frac{2}{k_{\mathrm{B}}}\frac{1}{d}\sum_{i=1}^{\mathcal{N}}\frac{1}{2}m_i\mathbf{v}_i\cdot\mathbf{v}_i.$$

For this two-dimensional experiment, $d = 2\mathcal{N}$ and $\mathcal{N} = 7$. Using the conditions above, the initial temperature is 22.72°K, which is very low.

This time the analytic solution to this set of differential equations is not known. Error in the energy will be used to measure accuracy here.

The six forces along the outside of the hexagon (between particles 2 and 3, 3 and 4, etc.) are put in the near class; so are the six forces involving the center particle, particle 1. The remaining nine forces are put in the far class. This includes six forces between alternating particles on the outside (particles 2 and 4, 3 and 5, etc.) and the three forces directly across the hexagon (particles 2 and 5, 3 and 6, and 4 and 7). Because the particles lie nearly on the corners and center of a hexagon, distances between far particles are either $\sqrt{3}$ or 2 times as far as distances between near particles.

The simulation is run for a time of $t_{\text{total}} = 1$ ns. Eighteen values of h are tested, geometrically increasing from 2 fs to 22.25 fs. N is tested as before at 2, 4, and 8. Some of these macro-step sizes are fairly large—they work because the temperature is so low.

In these experiments, the non-symplectic methods tend to drift to states of lower energy (Fig. 9). Figure 10 shows that Verlet-I still conserves energy well with h doubled.

Figure 11 shows similar behavior to that of Fig. 6 of the first experiment. Again, statistics were sampled 100 times during each run. Error in total energy ξ is computed using

(18). The formula for the number of force evaluations η is analogous to (16) and is given by

$$\eta = \left\lceil \frac{t_{\text{total}}}{Nh} \right\rceil (12N+9).$$

In this experiment, Verlet-I with $N=4$ was the most efficient. The symplectic methods were again roughly second-order accurate in energy and the nonsymplectic methods were roughly third order for the range of h tested. The same conclusions can be drawn from this experiment as from the previous one—for long-time integration, the nonsymplectic methods perform poorly compared to symplectic methods (even the method that does not use distance classes).

6. NUMERICAL EXPERIMENTS: ARTIFICIAL RESONANCE

In Section 4, we discussed how the different algorithms incorporate the "far" force, with the effects felt as intermittent pulses in Verlet-I and more evenly distributed across the macro-step in the extrapolation methods. This difference could be important if the impulses of Verlet-I occur near a natural frequency of the system, because this could cause resonance in the system. This effect is documented here by experimentation. Another situation where this might occur is hypothesized in [6] and later confirmed in [5].

Figure 12 shows the system of four particles for this experiment. There is a Coulomb interaction between all particle pairs and a bond (spring) between particles 2 and 3. The potential energy of the system is

$$V(r_1, r_2, r_3, r_4) = \frac{1}{2}k(|r_3 - r_2| - l)^2 + \sum_{i=1}^{3}\sum_{j=i+1}^{4} \frac{q_i q_j}{|r_i - r_j|}.$$

The parameters and initial conditions are given in Table III. The initial total energy of the system is 39.05119.

The experiment is set up so that the Coulomb repulsion between particles 1 and 2 is slightly smaller than the sum of attractions between particles 1 and 3 and between 1 and 4. This way particle 1 does not move very much during the simulation, oscillating slightly around its starting position without colliding into the center particles or being flung away from the center. The same is true for particle 4. Since these particles move only a little and since the center particles do not get far past ± 4.5, the Coulomb forces involving particles 1 and 4 do not change much and are put in the "far" class. The spring and Coulomb interaction between particles 2 and 3 are put in the "near" class.

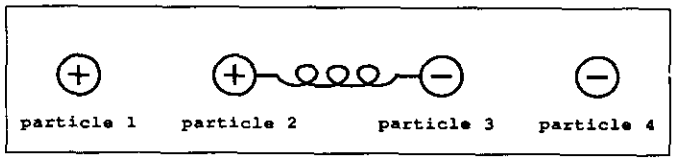

FIG. 12. Resonance experiment, system of particles.

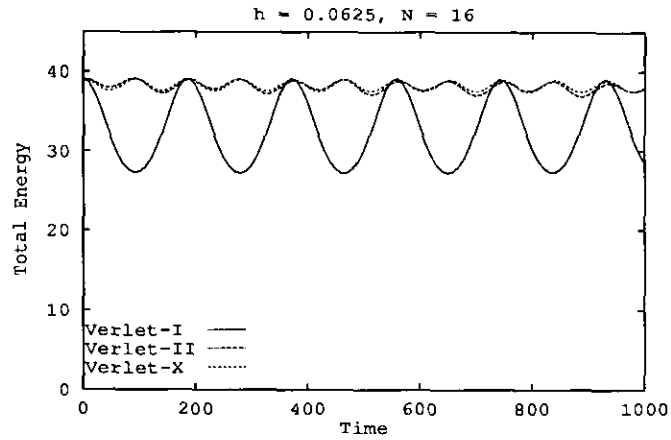

FIG. 13. Resonance experiment, artificial resonance of Verlet-I.

If the system consisted of only particles 2 and 3, they would oscillate with some period $\tau \approx 1.0$ (measured numerically). Now considering the system including particles 1 and 4, suppose the new forces are evaluated with a macro-step size of τ. In this scenario, Verlet-I produces resonance because the far forces add energy to the spring in pulses at its natural frequency. The actual physical system has no resonance because the far forces are not impulses but are continuous. The extrapolation methods approximate this better than Verlet-I, but for long enough time their inherent instabilities may dominate.

Figure 13 shows the results of the experiment with a macro-step size of $1.0 \approx \tau$, and it can be seen that the Verlet-I algorithm is producing larger oscillations of total energy than the other two algorithms. Figure 14 shows that by *increasing* the step size the Verlet-I algorithm actually conserves energy much better, which would normally be

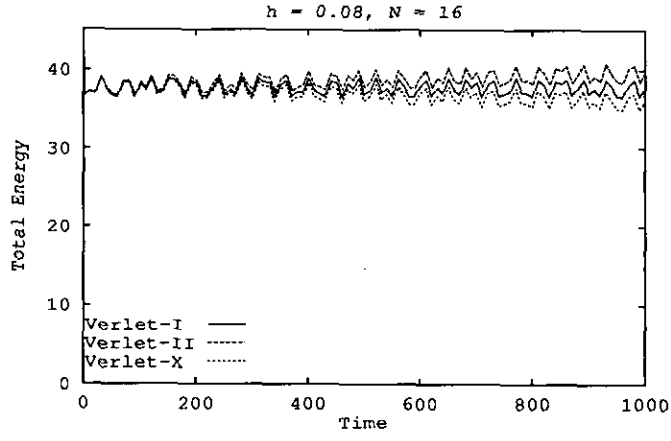

FIG. 14. Resonance experiment, increased step size.

TABLE III

Parameters and Initial Conditions for Experiment 3

Particle masses	$m_1 = 6, m_2 = 1, m_3 = 1, m_4 = 6$
Particle charges	$q_1 = +3, q_2 = +1, q_3 = -1, q = -3$
Initial positions	$r_1 = -20, r_2 = -2.5, r_3 = 2.5, r_4 = 20$
Initial velocities	All particles at rest
Spring stiffness	$k = 19.7$
Equilibrium spring length	$l = 7$

unexpected. Also note that the extrapolation methods are showing signs of the instability seen in Section 5, although the step size of 0.08 is perhaps fairly large compared to $\tau \approx 1.0$.

Clearly this experiment is contrived. In a complex molecule it seems unlikely that an artificial resonance could sustain itself for any length of time.

REFERENCES

1. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon Press, Oxford, 1987).
2. V. I. Arnold, *Mathematical Methods of Classical Mechanics*, 2nd ed. (Springer-Verlag, New York, 1989).
3. P. J. Channell and J. C. Scovel, *Nonlinearity* **3**, 231 (1990).
4. R. D. Skeel and C. W. Gear, *Physica D* **60**, 311 (1992).
5. H. Grubmüller, private correspondence, September 1991.
6. H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, *Mol. Simul.* **6**, 121 (1991).
7. D. Okunbor, *Physica D* **60**, 314 (1992).
8. D. Okunbor and R. D. Skeel, *Math. Comput.* **59**, 439 (1992).
9. J. M. Sanz-Serna, *Acta Numer.* **1**, 243 (1992).
10. W. B. Streett, D. J. Tildesley, and G. Saville, *Mol. Phys.* **35**, 639 (1978).
11. W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, *J. Chem. Phys.* **76**, 637 (1982).
12. L. Verlet, *Phys. Rev.* **159**, 98 (1967).