JOURNAL OF COMPUTATIONAL PHYSICS 113, 320-335 (1994)

An Evaluation of the Gauss–Radau Algorithm for the Simulation of Chemical Dynamics

KIM BOLTON AND STURE NORDHOLM

Department of Physical Chemistry, University of Göteborg and Chalmers University of Technology, S-412 96 Göteborg, Sweden

Received August 17, 1993; revised January 18, 1994

The Verlet, Verlet leap frog, Gear fixed time step, Gear variable time step, Runge–Kutta, and Gauss-Radau algorithms have been compared using trajectory data obtained from the integration of a one-dimensional diatomic chain under constant pressure. Investigation into the times of local and normal mode relaxation and conservation of the constants of the motion facilitated comparison of the integration techniques. It has been found that the Gauss-Radau algorithm, which is not widely used in the simulation of chemical dynamics, generally affords a higher accuracy at an improved efficiency. © 1994 Academic Press, Inc.

1. INTRODUCTION

Molecular dynamics (MD) and Monte Carlo (MC) methods are the two most widely used simulation techniques currently employed in computer analysis of chemical and physical systems. These techniques are linked via statistical mechanics and the assumption of ergodicity. Thus, if equilibrium properties of an ergodic system are desired then either of these methods may be employed. However, if time dependent (dynamical) properties are required then MD simulation is the sole option.

Although much work has been done on the improvement of MD algorithms [1, 2], it is clear that there is no single algorithm available that is superior to all others for all systems studied. This is made evident by the large number of algorithms that are in use at present [3–5]. Although this is partly attributable to the different accuracy and efficiency requirements of the various users, it is also due to the inapplicability of certain algorithms for the study of specific systems or properties.

This paper presents the results obtained from the comparison of the Verlet [1], Verlet leap frog [1] (VLPF), Gear variable time step (GVTS), Gear fixed time step [1, 6] (GFTS), Gauss-Radau [7] (Radau), and Runge–Kutta [2] algorithms for the system presented below. These algorithms represent a wide range of integrators from those which are subject to errors of order δt^4 in the position and δt^2 in the velocity coordinates (where δt is the micro time step used in the Taylor expansion of the position and velocity coordinates), as is the case for the Verlet algorithm, to those integrators which have adjustable parameters for energy conservation, notably the Radau and GVTS algorithms. Our focus is on the Radau algorithm which has found favour within our own group in recent years. Pettersson and Marković have employed the Radau algorithm to simulate van der Waals cluster scattering [8], it has also been used by Marković *et al.* in the study of binary collisions [9] and by Davidsson *et al.* to simulate the energy transfer properties of Br_2 diatomics [10]. In all of these studies the Radau algorithm was found to be efficient and to conserve energy to at least the tenth significant figure. Apart from the Radau integrator, the various algorithms investigated will not be discussed in detail as this is done in the references cited. Since it is the aim of this article to help introduce the Radau integrator into the field of chemical simulations where it is not widely employed, it is briefly introduced in Appendix A. Although we do not offer an analytical proof of the superior efficiency and accuracy of the Radau technique, the work presented in this paper, together with the results already obtained by our group (see references cited above), give strong support to the suitability of this method for a wide range of chemical and physical dynamics applications.

The system investigated was a one-dimensional chain of Morse oscillators under a constant external pressure. An important difference between this system and other studied previously is that the constant external pressure restricts the bond lengths in such a way as to force the bonds to spend most of their time on the repulsive part of the local mode potential energy surface. This leads to frequent collisions and interparticle forces that are, on average, larger than in a system that is free from external pressure. In terms of the integrator used, this means that the larger time steps associated with the efficient, sophisticated algorithms may be undesirable for the positive pressure system.

A second relevant characteristic of the system studied

0021-9991/94 $6.00

is that it is essentially nonergodic at the internal energy considered. Any fluctuations in the trajectory that result from integrator inaccuracies may have a qualitative influence on the path followed by the trajectory. In this way the local equilibrium that is approached by the trajectory will be influenced by integrator inaccuracies. It is also possible that ergodicity may be induced by stochastic inaccuracy. This is not the case for ergodic systems where small fluctuations in the trajectory path is unlikely to lead the trajectory away from global equilibrium. The chosen system, therefore, provides a particularly severe test of integrator accuracy.

The tests employed in the comparison of the integration techniques can be divided into two groups. The first includes the traditional tests of back integrability, algorithm efficiency, and algorithm conservation properties [1, 2]. The second group includes the comparison of data relevant to chemical dynamics. Local equilibrium mode energies and the time taken for the system to relax to local equilibrium were chosen for this purpose.

2. THE SYSTEM AND CALCULATIONS

The system studied is a one-dimensional chain of 10 bromine molecules (Br_2), where the alternating weak intermolecular and strong intramolecular bonds are both described by Morse potentials. At either end of the chain there is a particle (with the mass of a Br atom) whose function it is to convey the constant pressure to the system. The system is clearly bounded due to the presence of the pressure. The Hamiltonian for this system is

$$H=\tfrac{1}{2}\sum_{i=1}^{22} mv_i^2+\sum_{i=1}^{11} D_{e,a}(1-e^{-\beta_a(ra_i-rea_i)})^2$$
$$+\sum_{i=1}^{10} D_{e,b}(1-e^{-\beta_b(rb_i-reb_i)})^2+kr_{22-1},$$

where m is the atomic mass of Br, v_i is the atomic velocity, ra_i are the intermolecular bond lengths, and $D_{e,a}$, β_a, and rea are the intermolecular Morse parameters. The identical terms with the subscript b are the corresponding intramolecular Morse parameters. The force constant, k, acts over the chain length r_{22-1}. The values of the Morse parameters [10] are listed in Table I. The value of the force constant used is 2.97 kcal mol^{-1}Å^{-1}.

The above Hamiltonian was studied since, together with the Morse parameters listed in Table I, it is representative of Br_2 gas [10, 11] and is typical of most diatomic gases (although the dimensionality has been reduced from three to one). In general, diatomic frequencies depend on the masses of the bond atoms and the bond force constant. They range from below 41.99 cm^{-1} (corresponding to the heavy Cs_2 diatomic) to 4400 cm^{-1} (the frequency of H_2) [11].

TABLE I

Inter- and Intra-Molecular Bond Morse Parameters Used in the Diatomic Chain Hamiltonian

	Bond type	
Morse parameter	Intermolecular	Intramolecular
D_e	0.3370	45.889
β	1.4848	1.9400
re	4.0410	2.2800

The intramolecular bond frequency of 398 cm^{-1}, obtained using the tabulated parameters (when the effect of the constant pressure is excluded and low energy harmonic oscillations are assumed), lies within this range, whereas the intermolecular frequency of 28.15cm^{-1} describe slower oscillations typical of van der Waals bonds.

Although Lennard-Jones functions are generally used to describe intermolecular bonds, Morse functions have been employed in this investigation. The Br_2–Br_2 intermolecular bonds spend as much as 98% of the trajectory time on the repulsive part of the bond potential energy surface due to the positive pressure. The dynamics is thus better described by employing the more accurate exponential repulsion offered by the Morse potential than the r^{-12} repulsion used in the Lennard-Jones description.

The six algorithms listed in the Introduction were used to integrate the system described above. Comparison of the performance of these algorithms proceeded as follows:

1. Comparison of the energy and linear momentum conservation and back integrability of each algorithm over 1, 2, and 3000 ps and, in the case of the energy and linear momentum studies, over 60 ns. The back integrability was not compared over this longer period since none of the algorithms showed back integrability over the shorter period of 3000 ps.

2. Algorithm efficiency comparisons.

3. Comparison of the instantaneous bond lengths of bonds 10 and 11 and atomic velocity of atom 15 as a function of time. Bonds 10 and 11 and atom 15 were chosen arbitrarily. Bond 10 is the fifth intermolecular bond and bond 11 is the fifth intramolecular bond. This shows the divergence of the trajectories as a funcion of the algorithm used.

4. Comparison of the long (60 ns) time-average energies of bonds 10 and 11 and of normal modes 5 and 19. These normal modes were chosen arbitrary—mode 5 having low frequency (39.0 cm^{-1}) and mode 19 having high frequency (331.1 cm^{-1}). Clearly these modes are not normal in the strict sense of the word since these simulations were performed at finite energy in the presence of anhar-

monicities. This comparison was performed because the time-average properties are often calculated to obtain a direct determination of the ergodicity of the system, i.e., the long time-average of any property (obtained from MD) equals the corresponding ensemble-average (obtained from microcanonical MC) if the system is ergodic.

5. Comparison of the time taken for the system to relax to the time-average values determined. In the case of an ergodic system this is the time taken to relax to global equilibrium and for a nonergodic system, such as the one studied in this paper, it yields the time taken to relax to local equilibrium.

6. Another quantity that is often of interest is the ensemble-average generated from short MD simulations. By reducing the time of the simulation it is hoped that the round-off errors inherent in computational calculations and the inaccuracies inherent in the integrator will be reduced. In the case of the nonergodic system studied, 4000 ps trajectories were required to obtain meaningful relaxation values. The Boltzmann H-function method used by Henry and Grindlay [12] to determine the time taken by each mode to reach local equilibrium was deemed suitable for this application. Thus, the equilibrium energy and plateau value of the ensemble-generated H-functions for bonds 10 and 11 and normal modes 5 and 19 were compared as a function of algorithm used.

7. The time taken to reach the plateau region (i.e., relaxation to local equilibrium) was determined for the various algorithms.

Thus, apart from the traditional comparisons of back integrability and algorithm conservation properties and efficiencies, comparisons were also made using the data relevant to the understanding of the system dynamics from a chemical perspective.

As mentioned above, both the time-average and ensemble-average mode energies have been obtained. The time-averaging is performed over a single long trajectory whose initial configuration is obtained from MC simulation. That is, equilibrium coordinates are used for the initial configuration and the time taken for the chain to relax is the time taken for the average mode energies (determined by $E^{\text{av}}_{\text{mode}} = \sum_{n=1}^{N} E^{n}_{\text{mode}}/N$, where E^{n}_{mode} is the instantaneous mode energy at time step n and N is a specific number of time steps along the trajectory) to converge. Thus, at each step along the trajectory the average mode energy is calculated and compared to the corresponding energies of the other similar modes. After effects of the fluctuations have been minimised by the averaging procedure, these values converge. The convergence of these average mode energies indicates that the system has relaxed to local equilibrium.

The reason for using this method to determine the time of relaxation instead of determining the time taken to relax to the MC calculated equilibrium mode energy is that, due to the nonergodic nature of the system, global equilibrium is not generally approached by an arbitrarily chosen trajectory. The relaxation time is therefore the time taken to reach a *local* equilibrium, where the average mode energies of all similar modes are equal.

The ensemble-averaging is performed over a swarm of trajectories using $E^{\text{av}}_{\text{mode}} = \sum_{n=1}^{N} E^{n}_{\text{mode}}/N$, where N is the number of trajectories in the ensemble and E^{n}_{mode} is the mode energy at a particular time step observed along trajectory n. Clearly, if the initial configurations for the individual trajectories are chosen to be uncorrelated equilibrium coordinates (obtained from MC methods), then the ensemble-average mode energy will be equal to the equilibrium value at all times. In order to obtain a relaxation time it is therefore necessary to choose another set of initial conditions when performing ensemble-averaging. In this work the starting configurations were obtained by exciting a specific mode from the uncorrelated MC coordinates. The decay of the ensemble-average energy out of this mode yielded the relaxation time.

The initial conditions and methods of determining the time-average and ensemble-average energies and times of relaxation are, therefore, very different. As a result, the average mode energies and relaxation times obtained from these two techniques are not comparable.

When undertaking a comparison of this nature it is obviously extremely important to minimise all extraneous influences. In an effort to do this the following conditions were adhered to:

1. The initial internal energy, $E^{\text{initial}}_{\text{internal}}$, of the system was set equal to 200.000000 kcal mol^{-1} for all trajectories.

2. Identical starting coordinates:

   The same microcanonical MC-determined coordinates were used as the initial coordinates for all algorithms. This set of coordinates could be used directly by the self-starting integrators. These integrators require one instantaneous system configuration in the input; that is, just a single set of particle positions and velocities (which is valid at a single time step) is required in the input for these integration methods.

   Two of the integrators studied are non-self-starting in the sense that, in order to initialise the integration, they require the system's coordinates at more than one time step along the trajectory path. The Verlet algorithm requires particle positions at two neighbouring time steps while the VLPF requires the intial particle velocities at a time equal to half the step size away from the initial particle positions.

   In order to begin the trajectories propagated by the non-self-starting integrators in the same sub-volume of phase space, the starting coordinates for the Verlet algorithm were determined by propagating the original coordinates (i.e., those used as input for the self-starting integrators) over one time step using the GFTS algorithm. This yielded the two sets of particle positions required as input by the Verlet integration procedure.

The starting coordinates of the trajectory propagated by the VLPF algorithm were determined in a similar manner. The GFTS algorithm was used to propagate the trajectory over two time steps, each equivalent to one-half of the micro time step. The first step yielded particle velocities one-half of a time step before the particle positions—which were obtained by the second step—as required by the VLPF integration scheme.

The same random numbers were used to determine the 100 MC starting coordinates for the ensemble-average tests.

In the case of the normal mode ensemble-average tests, the normal mode excitation was performed with the same random numbers (for each algorithm) to ensure the same initial phase and velocity corresponding to the excitation. The initial configuration was obtained by putting virtually all[1] of the excess energy (i.e., the energy above classical zero) into normal mode 5.

In order to study the ensemble-averages of local modes 10 and 11, local mode 10 was excited by compressing bond 10 (the fifth intermolecular bond) to 2.67 Å while all the other local modes were placed at their equilibrium positions. This gave a total potential energy of 199.5 kcal mol^{-1}. The remaining energy (0.5 kcal mol^{-1}) was randomly distributed among the atomic velocities (using the same random numbers for all the algorithms).

3. Comparisons were made using the same fixed step length. In the case of the two variable time step methods, the time step was chosen as described below.

4. All other variables (e.g., Morse parameters) remained fixed.

The concepts of macro and micro time steps are needed to describe the method used to ensure comparable step lengths for both variables and fixed time step integration schemes: The macro time step, which pertains to the variable time step integrators, is the step length used for each call to the integration subroutine. This time step is adjusted to allow one to exit from this subroutine in order to obtain trajectory data at specific times throughout the length of the trajectory. Within the subroutine the time steps (used in the Taylor expansion, for example) are varied to maintain the required accuracy. These variable time steps are the micro time steps. Fixed time step integrators integrate the trajectory over one time step for each call to the integration subroutine. Thus, in this case, the macro time step, micro time step and fixed time step are synonymous.

The fixed time step used was determined as follows: The GVTS algorithm was used with a macro time step of 6.356 fs. Since an average of 4.56 steps were performed in each macro time step (determined over a 60 ns trajectory), the time step used for the fixed step integrators was $6.356/4.56 = 1.394$ fs. This gave a direct comparison between the performance of the GVTS and the fixed time step algorithms.

It was not as simple to determine a similar macro time step for the second variable time step algorithm—the Radau integrator. The main reasons for this were that the efficiency of the integrator improves with larger macro time steps and thus for optimal efficiency the macro time step should be the trajectory length, and, that this integrator allows one to set the accuracy desired. The time step and the efficiency decrease as the accuracy requirement becomes more stringent. In keeping with the notation of Everhart [7], a large LL parameter, which implies a high accuracy, reduces the step size (see Appendix A for a detailed description of LL).

Since it was expected that the Radau integrator would give accurate simulations, the value of LL was initially set equal to 16 (this value has been used to conserve energy to better than one part in 10^{12} [8]). The energy conservation obtained using this extremely strict criterion ($LL = 16$) is marked appropriately (see the next section). This stringent accuracy requirement made the integrator extremely inefficient compared to the other algorithms. Whereas the GVTS algorithm required 0.305 cpu seconds to propagate a 1 ps trajectory, it took the Radau algorithm 8.42 cpu seconds to integrate the same trajectory. When this same comparison was made over 60 ns (giving the Radau integrator a macro step size of 60 ns) the Radau integrator was still at least six times less efficient than the GVTS algorithm. As a result of this inefficiency, no further comparisons were made using the Radau integrator under these conditions.

The energy conservation criterion was thus relaxed in an effort to improve the efficiency. A macro step size of 6.35 fs, together with a value of $LL = 4$ meant that the integrator performed a micro step equivalent in $0.8371 * 10^{-2}$ cpu seconds (whereas the GVTS algorithm required $0.4245 * 10^{-3}$ cpu seconds). A further reduction in the value of LL did not improve the efficiency significantly. The efficiency was, however, still too poor to perform long integration runs.

By increasing the macro step to 1 ps, the time per micro step was reduced to $0.3939 * 10^{-3}$ s. The Radau integrator is, therefore, more efficient than the GVTS when this step

[1] The remaining modes were given a minimal amount ($0.01\,\hbar\omega$) of energy.

size is employed. As this macro step size is increased the efficiency is further improved. Since the LL value remained unaltered, this improvement in efficiency was obtained without decreasing the integrator accuracy. In order to obtain mode energies at the same time steps as the other algorithms—in order to accommodate comparisons of mode relaxation times—the Radau integrator with an LL value of 4 and a macro time step of 348.40 fs was used for the long time and ensemble integrations.

It should be noted that an LL value of 4, while being suitable for the one-dimensional chain, may not be applicable for all systems and accuracy requirements. It is suggested by Everhart [7] that LL should be assigned an initial value of 8, and that this value should be altered depending on the accuracy required. Although this suggestion was made for the integration of planetary motion, we have found that it holds for chemical systems and, for common accuracy requirements (energy conservation to 7 or 8 significant figures), LL values lower than this are suitable. The improved performance of the Radau algorithm shown in this work (using LL = 4) is thus relevant to a wide range of chemical and physical systems.

3. RESULTS AND DISCUSSION

3.1. *Conservation Properties and Back Integration*

It is common to assign a certain accuracy to a particular algorithm based on its conservation and back integrability properties [1, 2, 7]. This is usually done for a 1 or 2 ps trajectory. The same was done in this comparison but, due to the lower dimensionality of the system studied and the effects of the constant pressure, direct comparison with other systems is not feasible. The trend in the accuracy of the algorithms studied would, however, be expected to resemble that found for other systems.

Table II lists the fractional deviation in the total internal energy as a function of the integrator employed over trajectories of 1 and 2ps. The fractional deviation in the energy is the ratio of the maximum deviation in the energy over the complete trajectory (1 or 2 ps) to the $E^{\text{initial}}_{\text{internal}}$. The trend of improved energy conservation (smaller deviations) obtained for higher order algorithms is as expected with the lower order integrators (e.g., the VLPF and Verlet) showing worse conservation (larger deviations) than the higher order integrators. The variable time step methods have superior energy conservation properties since this is kept at a set level throughout the algorithm.[2]

TABLE II

Fractional Energy Deviation (Maximum Deviation/Initial Energy) of the Internal Energy over 1 and 2 ps trajectories

Algorithm name	1-ps trajectory	2-ps trajectory
GVTS	$-7.50 * 10^{-8}$	$-1.35 * 10^{-7}$
GFTS	$-1.20 * 10^{-7}$	$-3.27 * 10^{-7}$
VLPF	$5.31 * 10^{-5}$	$5.31 * 10^{-5}$
Verlet	$-4.02 * 10^{-5}$	$-4.02 * 10^{-5}$
Runge–Kutta	$-1.00 * 10^{-7}$	$-1.90 * 10^{-7}$
Radau (LL = 4)	10^{-11}	10^{-11}
Radau (LL = 16)	10^{-12}	10^{-12}

Note. Negative deviations indicate a loss in energy.

It is noticeable that the fractional deviations in energy of the trajectories propagated by the Verlet and VLPF procedures are independent of whether the trajectory time is 1 or 2 ps. The reason for this is that the maximum deviation is a combination of the initialisation procedure of the non-self-starting integrators and the fact that the largest subsequent deviation occurs within the first half of the 2-ps trajectory. For example, the Verlet algorithm requires two consecutive sets of particle positions in order to propagate a trajectory. As described earlier, the GFTS procedure was used to integrate the original coordinates $\{x_i^0, \dot{x}_i^0\}$ over one time step to yield the second set of coordinates $\{x_i^1, \dot{x}_i^1\}$. Using these two sets of positions, the Verlet integrator predicts a third set of positions, $\{x_i^2\}$, with errors of order 4 and then uses these new positions together with $\{x_i^0\}$ to generate the velocities $\{\dot{x}_i^1\}$ with error of order 2. These velocities, $\{\dot{x}_i^1\}$, together with the positions at this time step, $\{x_i^1\}$, are used to obtain the energy at the first step of the Verlet-propagated trajectory. This energy was calculated to be 199.995350 kcal mol^{-1}. Thus, the initialisation procedure required by the Verlet algorithm leads to a fractional deviation in the internal energy of $2.325 * 10^{-5}$. Whereas the coordinates at this step were determined from velocities and positions which were partially obtained from the non-time-reversible GFTS algorithm, all subsequent steps are determined by the Verlet algorithm which *is* time reversible. The energy of the trajectory therefore fluctuates around 199.995350 kcal mol^{-1}. The maximum deviation is thus the sum of the initial deviation and the maximum fluctuation ($1.69 * 10^{-5}$ kcal mol^{-1}). Since the maximum fluctuation occurs within the first picosecond of the 2-ps trajectory, the maximum deviation is the same for the 1- and 2-ps trajectories as shown in Table II.

Figures 1 and 2 show the conservation of internal energy obtained using the various algorithms over trajectories of 3000 ps and 60 ns, respectively. It is clear that the energy of the trajectory propagated by the time-reversible Verlet algorithm fluctuates around 199.995350 kcal mol^{-1} and the energy of the trajectory propagated by the VLPF algorithm fluctuates around 200.008204 kcal mol^{-1}. Since there is no

[2] This is not precisely true for the GVTS algorithm since the conservation limit is ignored if the time step becomes too small. It is, however, the general idea behind variable time step algorithms and is applicable to the Radau integrator.

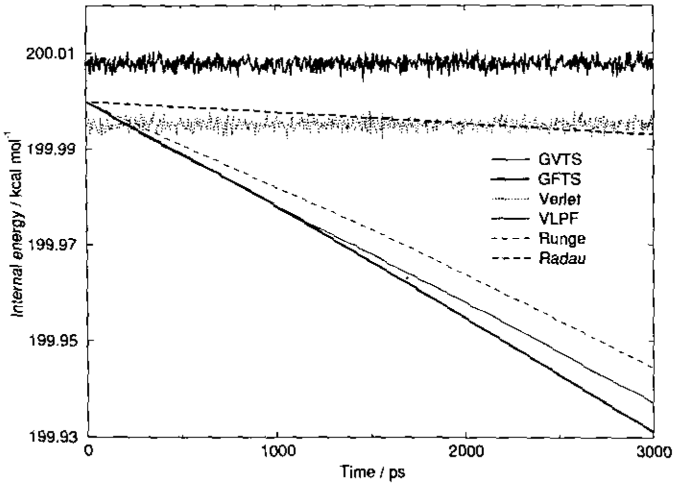

FIG. 1. Time dependence of the system internal energy over 3000 ps. The energy conservation of the Radau algorithm is compared with those of the time invariant (Verlet and VLPF) and lower order self-starting algorithms. Due to the integration methods of the non-selfstarting algorithms (which satisfy time reversal invariance), they do not show a continual decrease in energy.

continuous decrease in energy over time as there is for the higher order algorithms (which are not time reversible), it may be preferable to use the lower order Verlet and VLPF algorithms for the propagation of long trajectories. However, the noisy time dependence of the energy suggests that the trajectory may not be so accurate by other measures.

The figures also show that the Radau algorithm offers superior energy conservation compared to the other higher order algorithms for all times. The Radau algorithm used here had the low energy conservation parameter of LL = 4 (and a macro step of 348.4 fs). The energy conservation and back integrability over 1 and 2 ps were not tested at this macro step size since 348.4 fs is not a factor of 1000 or 2000 fs.

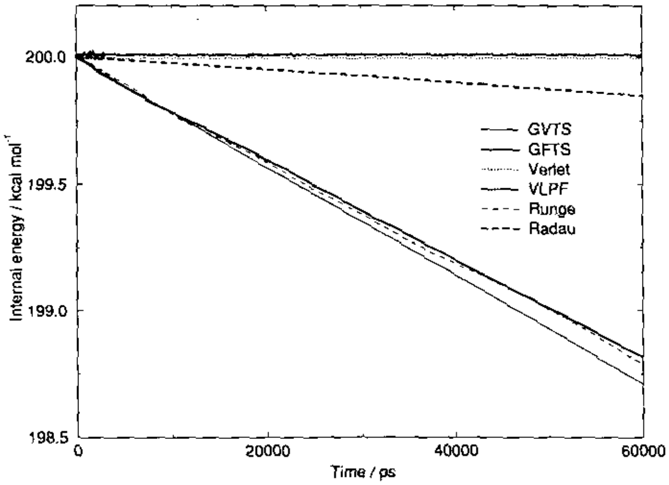

FIG. 2. Time dependence of the system internal energy over the 60 ns trajectory.

Due to the multiple crossings of the remaining three trajectories, it is impossible to designate a particular algorithm as being superior to the others over all times. It is clear, however, that the energy conservation of the Runge–Kutta algorithm is almost always better than that of the GVTS. The GFTS is worse than these two integrators at shorter times (see Fig. 1) but is comparatively better at longer times (see Fig. 2). The good long time conservation of the predictor–corrector algorithm is expected due to the stability of this method of integration. Clearly, introducing an error-limiting condition into this algorithm partially removes the stability associated with the integrator.

The linear momentum conservation over 1 and 2 ps was excellent for all integrators with the GVTS conserving to the 11th significant figure, the GFTS, VLPF, and Runge–Kutta integrators conserving to the 12th significant figure and the Radau integrator conserving to the 13th significant figure. The Verlet algorithm conserved to the eighth significant figure. The energy conservation requirements are clearly far more difficult to fulfil than momentum conservation requirements.

The back integrability is the most stringent test that is traditionally used to ascertain algorithm accuracy. Table III shows the number of significant figures recovered when back integrating a trajectory over 1 and 2 ps. The back integrability is tabulated as a function of the integration scheme employed. The position and velocity coordinates are shown separately since the slower varying position coordinates usually show superior back integrability. All values are in reduced units.

The trend in the back integrability of the algorithms is the same as that for the energy conservation (it should be remembered that the poor back integrability of the time-reversible Verlet and VLPF procedures is due to the initialisation procedure). This is expected since both good back integrability and good energy conservation are

TABLE III

Back Integrability over 1 and 2 ps Shown as the Number of Significant Figures Recovered

	1 ps trajectory		2 ps trajectory	
Algorithm name	Position	Velocity	Position	Velocity
GVTS	7	4	7	4
GFTS	7	3	7	3
VLPF	2	0	2	0
Verlet	2	0	2	0
Runge–Kutta	7	4	7	4
Radau (LL = 4)	12	11	11	10
Radau (LL = 16)	12	11	11	11

associated with algorithm accuracy. When comparison between the integrators is made in terms of their back integrability, it is clear that the Radau algorithm is far superior to the other algorithms studied.

The 3000-ps back integrated trajectories showed no back integrability for any of the algorithms tested; that is, all coordinates changed in the first significant figure.

The tests described above show that all integrators have good energy conservation, but good back integrability is only obtained for the higher order algorithms. Thus, based on conservation properties, the higher order algorithms would be preferred to the lower order ones. However, the algorithm chosen will be dependent on the conservation requirements *and* the efficiency of the algorithm.

3.2. *Algorithm Efficiency*

Table IV gives the efficiency comparison between the different integrators investigated. The time tabulated is the time required to integrate a single trajectory over a period of 1 ps. The values were obtained by propagating 20 ns trajectories and dividing the total cpu time by $2 * 10^4$ (the starting configuration was the same for all trajectories). The step size for the fixed time step integrators was 1.394 fs (see the previous section), for the GVTS it was 6.356 fs and for the Radau it was the trajectory length. It should be noted that no concerted effort was made to improve the integrator code in order to obtain maximum efficiency. The results shown are, however, obtained from compiler-optimised code.

The low order VLPF algorithm is the most efficient of all the algorithms compared. The Verlet integrator is also a low order algorithm but it is less efficient because the positions of two consecutive steps need to be available in each integration (which necessitates an extra do-loop), the new positions need to be calculated and a division is required to calculate the velocity (these inefficient steps can be removed if a concerted effort is made to maximise the integrator efficiency).

The GVTS is more efficient than the GFTS since fewer force field evaluations are required for this trajectory when using the GVTS technique (see Table IV). (It has been observed that other trajectories require more force field evaluations by the GVTS scheme in order to maintain a specific accuracy. In these cases the GVTS will become less efficient.) The Runge–Kutta integrator is less efficient than the GVTS despite the fact that it is a fixed time step method. This decrease in efficiency is expected since it is an iterative procedure.

It is clear that the Radaù integrator is very efficient even though it is a higher order algorithm and has excellent conservation properties. However, it must be remembered that this efficiency is only achieved by using large macro step sizes (see the previous section) and thus if output is required at short, regular (fixed-time) intervals this procedure is unsuitable. Since regular output is not generally required for the analysis of data obtained from MD simulations, the Radau integrator will be preferred for most investigations of a chemical or physical nature.

The greater efficiency of the Radau technique is due, in part, to the reduced number of force field evaluations (see Table IV) afforded by the variable time step procedure. However, it is possible that for larger systems, where the interactions are larger in number and more complex, a larger number of force field evaluations will be performed in order to maintain the desired accuracy. This would reduce the efficiency of this variable time step technique. In order to investigate the effect of the system size, we have considered the case where all the atoms in the chain interact via the weak Morse potentials listed in Table I (i.e., the limitation to nearest neighbour interaction has been removed to include all possible interactions). This change will lead to a significant increase in the time spent on the force field evaluation. The results listed in Table V were obtained in an identical manner to those of Table IV, but they pertain to the system with global interactions. For this system, the GVTS procedure is less efficient than the GFTS technique since more force field evaluations are required to maintain the required accuracy (which is the same as it was for the nearest neighbour interaction chain). The Radau integra-

TABLE IV

Comparison of the CPU Time Required to Integrate the System over 1 ps

Algorithm name	CPU time/s	Force field evaluations
GVTS	0.132	14112940
GFTS	0.150	14351235
VLPF	0.101	14351235
Verlet	0.123	14351235
Runge–Kutta	0.397	57404937
Radau	0.134	11906579

Note. The second column lists the number of force field evaluations performed during the 20 ns trajectory. Integration was performed on an IBM RISC System/6000 320.

TABLE V

The Same as for Table IV for the System Characterised by Global Interactions (See Text for Details)

Algorithm name	CPU time/s	Force field evaluations
GVTS	0.947	15268099
GFTS	0.908	14351235
VLPF	0.864	14351235
Verlet	0.889	14351235
Runge–Kutta	3.416	57404937
Radau	0.787	12356219

tion method is, once again, very efficient since it requires fewer force field evaluations than the other methods. Due to the increase in time taken to evaluate the force field, the saving in cpu time when using the Radau algorithm is substantial.

Although the Radau integrator is superior for the one-dimensional chain systems investigated (and, as has been discussed, this system is a severe test for higher order algorithms), we cannot exclude the possibility that for certain systems—or certain trajectories in a particular system—this integration scheme may require a large number of force field evaluations to maintain a high accuracy and thereby be less efficient than the best low order integrators.

3.3. *Comparison of Time-Averages*

Although the instantaneous values of the cartesian and internal coordinates are not expected to be similar for trajectories propagated by different algorithms over chemically significant times, one would hope that their time- and ensemble-averages will be approximately the same, so that correct conclusions regarding the properties and dynamics of the system (e.g., ergodicity of the system and rate of vibrational relaxation) can be drawn regardless of the algorithm used. The tests discussed below focus on this point.

Figure 3 shows the instantaneous velocity of atom 15 between 100 and 102 ps. Although these trajectories have identical initial coordinates, it is clear that the discrepancy in the instantaneous particle velocities occurs within a short period. Similar plots for the bond lengths yield similar results, thereby showing that different instantaneous coordinate values are obtained when different integration procedures are employed. Comparison of instantaneous coordinate values which are obtained from different integration techniques is not feasible except within very short periods of integration.

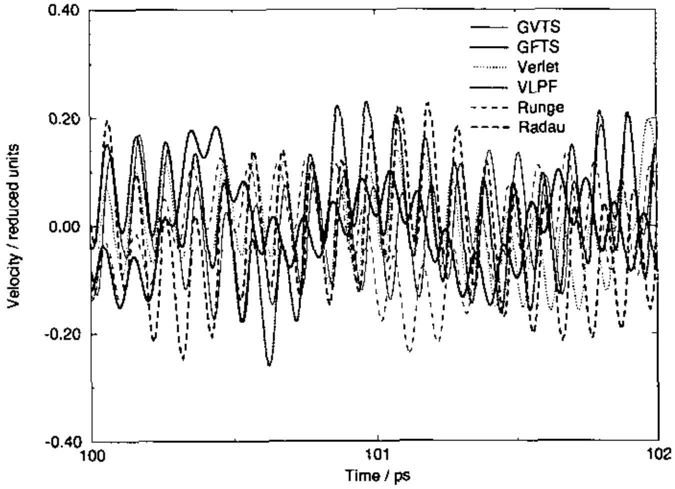

FIG. 3. Time dependence of the instantaneous velocity of atom 15 illustrating the influence of the integration scheme employed.

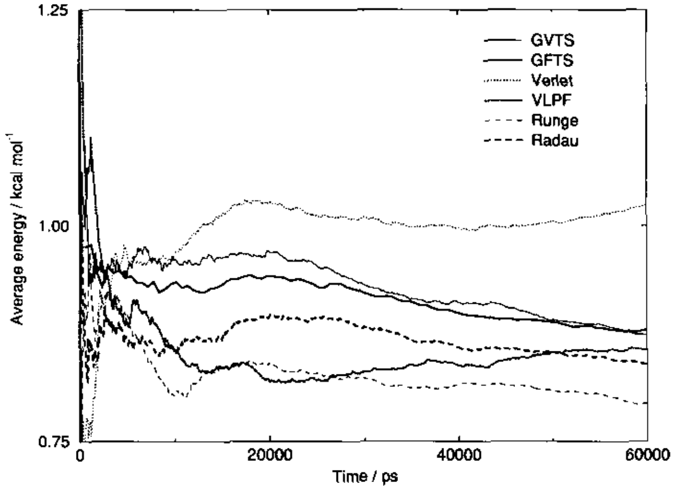

FIG. 4. Time-average energy of normal mode 5, obtained when trajectories with identical initial coordinates are propagated using different integration schemes.

The instantaneous coordinates generated by MD simulation are often used to generate data regarding the ergodic properties of a system and to generate the internal rate of relaxation. This can be obtained, for example, by studying the time-average bond energy over a single long trajectory or an ensemble-average of the bond energy over time. Both methods were investigated.

Since the system studied is apparently nonergodic, there is no relaxation to a global equilibrium. In fact, the local equilibrium that is obtained within a set of similar bonds is strongly dependent on the starting configuration. The same starting configuration (determined from MC sampling) was therefore used for all the algorithms studied. Equilibrium was attained once the long time-averages of all similar bonds or all similar normal modes were equal.[3] The system studied shows two distinct sets of bonds (hence its nonergodicity); the weak Morse bonds (of which bond 10 is an element) and the strong Morse bonds (of which bond 11 is an element). This gives rise to two separable sets of normal modes, a low frequency set (of which normal mode 5 is an element) and a high frequency set (of which normal mode 19 is an element).

The time-average energies—determined from a single trajectory—of normal modes 5 and 19 are shown in Figs. 4 and 5 while the time-average energies of bonds 10 and 11 are plotted in Figs. 6 and 7. It is clear from these graphs that the time-average mode energies are influence by the integration method employed. This dependence of the time-average

[3] Since absolute equality is not computationally obtainable, a small deviation is allowed. These were 0.032 kcal mol^{-1} for the strong Morse bonds, 0.015 kcal mol^{-1} for the weak Morse bonds, 0.15 kcal mol^{-1} for the low frequency normal modes and 0.05 kcal mol^{-1} for the high frequency normal modes. Equality of mode energies was assumed if the energies were within this error deviation for 500 ps.

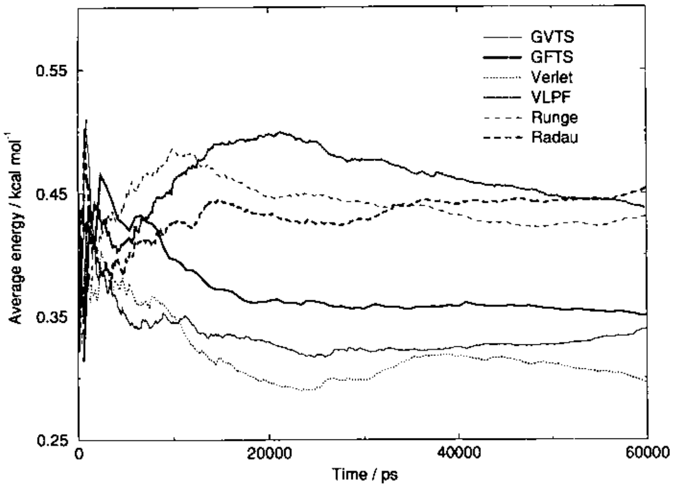

FIG. 5. The same as for Fig. 4 but for normal mode 19. The dependence of the time-average mode energy on the integration method employed is apparent.

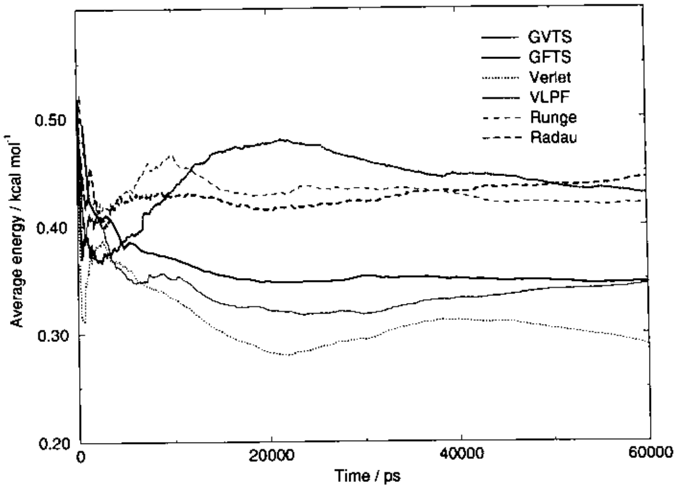

FIG. 7. The same as for Fig. 4 but for bond 11. This bond is representative of the strong bonds in the diatomic chain.

mode energies on the integration procedure is explained with the aid of Fig. 8.

Figure 8 shows the time-average energy of bond 11 (determined in an identical manner to that illustrated in Fig. 7), where the different curves in this figure are calculated from time-averaging over trajectories with different initial coordinates but the same initial energy (200.000000 kcal mol^{-1}) and propagated by the same integrator (GVTS). Thus, whereas Fig. 7 shows the influence of the integration scheme on the time-average mode energies, Fig. 8 illustrates the influence of the initial coordinates on these energies. It is evident from Fig. 8 that the time-average mode energies are influenced by the initial configuration (as is expected for a nonergodic system) and that the range of these energies is large (i.e., at 60 ns the range in energies of the 10 trajectories investigated is approximately 0.25 kcal mol^{-1}). The 10 trajectories propagated to generate the data shown in Fig. 8 are just 10 of many such trajectories determined by different starting configurations. In the discussion below the trajectories used to generate Fig. 8 will be referred to as traj1$_{\mathrm{GVTS}}$, traj2$_{\mathrm{GVTS}}$, ..., traj10$_{\mathrm{GVTS}}$.

Suppose that $\{x_1, ..., x_{22}, \dot{x}_1, ..., \dot{x}_{22}\}$ are the initial coordinates (positions and velocities, respectively) of the trajectory used to generate the data in Fig. 7. These are also the coordinates used to generate traj1$_{\mathrm{GVTS}}$ shown in Fig. 8. The set of coordinates $\{x'_1, ..., x'_{22}, \dot{x}'_1, ..., \dot{x}'_{22}\}$ determined by one GVTS integration step clearly lie on traj1$_{\mathrm{GVTS}}$. It is clear that the corresponding set of coordinates

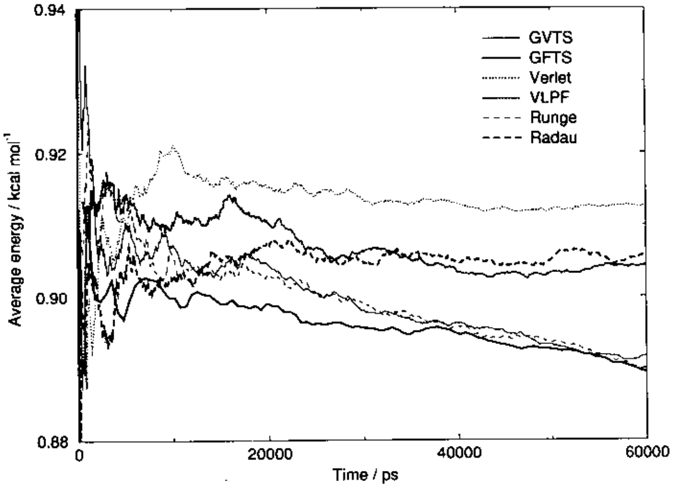

FIG. 6. The same as for Fig. 4 but for bond 10. This bond is representative of the weak Morse bonds in the diatomic chain.

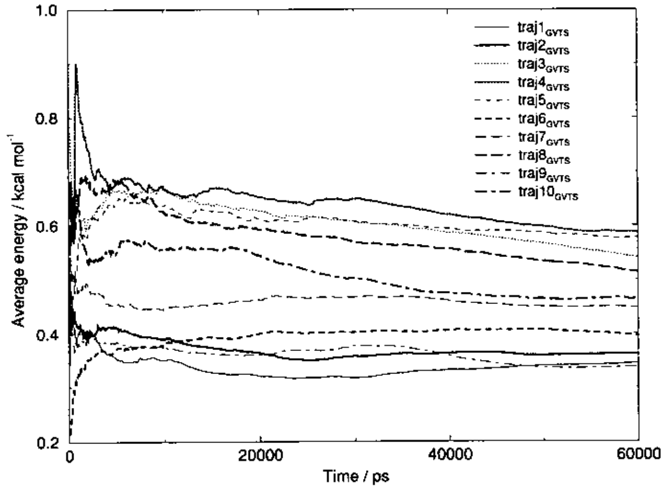

FIG. 8. Time-average energy of bond 11 obtained when trajectories with different MC determined initial coordinates are propagated by the GVTS algorithm. The dependence of the time-average mode energies on the initial coordinates is illustrated over a 60 ns trajectory.

$\{x_1'', ..., x_{22}'', \dot{x}_1'', ..., \dot{x}_{22}''\}$ determined by a single integration step using the Radau integrator (for example) will differ from $\{x_1', ..., x_{22}', \dot{x}_1', ..., \dot{x}_{22}'\}$ due to the differences in accuracy and method of the two integration schemes. In general, the configuration defined by $\{x_1'', ..., x_{22}'', \dot{x}_1'', ..., \dot{x}_{22}''\}$ will not lie on traj1$_{\mathrm{GVTS}}$. Hence, even though the starting coordinates are identical, different integrators will propagate different trajectories. Thus, the effect of maintaining identical starting coordinates but varying the integration scheme used to propagate these coordinates is similar to the effect of propagating different initial coordinates using a single integration method. That is, since the configuration defined by $\{x_1'', ..., x_{22}'', \dot{x}_1'', ..., \dot{x}_{22}''\}$ is in the phase space accessible to this system at 200.000000 kcal mol^{-1}, it will lie on one of the trajectories propagated by the GVTS method (i.e., one of the traj2$_{\mathrm{GVTS}}$, ...). This results in a range of time-average mode energies (e.g., approximately 0.15 kcal mol^{-1} at 60 ns in Fig. 7) due to the different integration schemes in the same way that there are a range of energies due to different initial coordinates (see Fig. 8).

Therefore, it is clear that time-average mode energy comparisons based on single trajectories propagated by different algorithms is not feasible even when the initial coordinates are identical.

When a system is nonergodic the difference in initial coordinates of two trajectories, e.g., traj1$_{\mathrm{GVTS}}$ and traj2$_{\mathrm{GVTS}}$, may result in their exploring different regions of phase space, thereby yielding different time-average mode energies. The rate at which a specific trajectory covers its available phase space will depend on properties such as the size of phase space accessible to that trajectory and the way in which the trajectory propagates through phase space. This may be different for different trajectories and hence the rate of relaxation to local equilibrium (which is governed by the time it takes for a trajectory to cover a representative part of its accessible phase space) will also depend on the starting coordinates of the trajectory. It is, therefore, expected that the trajectories traj1$_{\mathrm{GVTS}}$, traj2$_{\mathrm{GVTS}}$, ..., traj10$_{\mathrm{GVTS}}$ will have different relaxation times. This is the case as is shown in Table VI.

Since the effect of propagating trajectories (with identical starting coordinates) using different integrators is akin to the effect of varying the initial coordinates (and using the same method of integration), one would expect that time-average mode relaxation times would also depend on the integration method. This is the case as is shown in Table VII. Thus, in a similar way to that described for the time-average mode energies, there is a range of relaxation times that can be sampled. The particular relaxation time determined depends on the initial coordinates chosen and on the integration scheme employed.

Hence, in addition to the fact that the comparison of time-average properties obtained using different integrators is not feasible, the corresponding relaxation rates can also only be compared if they are obtained using the same integration scheme.

As was discussed in Section 2, the initial conditions used when determining time- and ensemble-average mode energies were different. When ensemble-average mode energies were calculated it was necessary to use a starting configuration where one of the modes was excited (see Section 2 for details), whereas equilibrium starting coordinates obtained by MC sampling were used when determining time-average mode energies. The choice of equilibrium starting coordinates allowed one to observe the large influence that the integration method has on the time-average mode energies and relaxation rates. This influence of the integrator limits our ability to draw chemical and dynamical conclusions from individual trajectories representing an equilibrium fluctuation. If, instead, initial coordiates similar to those used in the ensemble-averaging method were used (i.e., initially exciting one of the modes) then the "error

TABLE VI

Time-Average Relaxation Times (See Text for Details) for the Trajectories Shown in Fig. 8

Trajectory	Relaxation time/ps
traj1$_{\mathrm{GVTS}}$	12100
traj2$_{\mathrm{GVTS}}$	14800
traj3$_{\mathrm{GVTS}}$	33600
traj4$_{\mathrm{GVTS}}$	50900
traj5$_{\mathrm{GVTS}}$	> 60 ns
traj6$_{\mathrm{GVTS}}$	13100
traj7$_{\mathrm{GVTS}}$	20600
traj8$_{\mathrm{GVTS}}$	42000
traj9$_{\mathrm{GVTS}}$	7660
traj10$_{\mathrm{GVTS}}$	> 60 ns

Note. The dependence of the relaxation time on the initial trajectory coordinates is illustrated.

TABLE VII

Time-Averaged Relaxation Times (ps) as a Function of the Algorithm Used

Name of algorithm	Relaxation time for bond 10	Relaxation time for bond 11	Relaxation time for mode 5	Relaxation time for mode 19
GVTS	11600	12100	8530	23200
GFTS	7150	6070	14900	8540
VLPF	3110	19100	> 60 ns	> 60 ns
Verlet	6640	4150	> 60 ns	> 60 ns
Runge–Kutta	30300	54800	1700	10200
Radau	7830	25800	709	11000

Note. These relaxation times are obtained from a single trajectory as described in the text.

margin" that is allowed on either side of the equilibrium mode energy value could be increased. As the initial mode excitation is increased, this error margin could also be increased. This would obviously decrease the dependence of the time-average mode energy (which would be more coarse grained) and the corresponding relaxation time on the integration scheme employed.

Before discussing the ensemble-average results, it is of interest to compare the above results with those that would be observed for an ergodic system. The fact that the system studied in this report is apparently nonergodic meant that the starting coordinates (and integration method) determined which trajectory (e.g., traj$1_{\rm GVTS}$, ..., traj$10_{\rm GVTS}$) would be propagated. In the case of an ergodic system one trajectory covers, in essence, the total phase space. This means essentially that all initial conditions (determined from microcanonical MC sampling) will lie on this trajectory and that the same time-average mode energies (the *global* equilibrium values) will be determined independently of the starting coordinates chosen. Also, since the effect of the integrator is just to predict a new configuration on essentially the same trajectory (i.e., $\{x'_1, ..., x'_{22}, \dot{x}'_1, ..., \dot{x}'_{22}\}$ and $\{x''_1, ..., x''_{22}, \dot{x}''_1, ..., \dot{x}''_{22}\}$ lie on the same and only trajectory), the time-average mode energies will be independent of the integration scheme (apart from the relatively minor differences due to the overall loss in energy of the system). It is expected, however, that the relaxation times will be weakly integrator dependent with the more inaccurate integrators (which introduce stochasticities into the system) propagating trajectories that would, in general, yield shorter relaxation times.

These predictions were validated using the one-dimensional chain Hamiltonian described in the previous section where all the bonds were strong; that is, the Morse parameters of all bonds were the intramolecular Morse parameters listed in Table I. This system is apparently ergodic at an internal energy of 200.000000 kcal mol^{-1}. Figure 9 shows the time-average energy of bond 11 for the ergodic system and Table VIII lists the corresponding relaxation times. It is clear from Fig. 9 that the time-average mode energies converge over 75 ps, thus showing that the time-average mode energy is independent of the integration scheme employed. This constrasts with the result obtained for the non-ergodic system (see Fig. 7). The relaxation times listed in Table VIII show that these "ergodic" relaxation times are only weakly influenced by the integration scheme employed (when compared to those of the non-ergodic system shown in Table VII). The stochastic trajectory propagated by the Verlet algorithm—which employs first-order velocity predictions—leads to a relaxation time that is much shorter than those yielded by the other integration techniques.

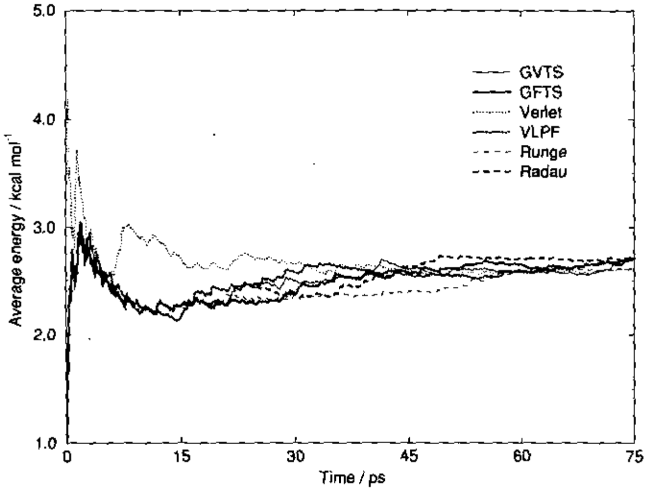

FIG. 9. Time-average energy of bond 11 obtained when trajectories of an ergodic system are propagated using different integration schemes.

TABLE VIII

Time-Average Relaxation Times Obtained when Propagating Trajectories of an Ergodic System

algorithm	Relaxation time/ps
GVTS	45.8
GVFS	49.8
VLPF	47.4
Verlet	21.4
Runge-Kutta	47.8
Radau	38.4

Note. The averaging technique employed here is identical to that used to generate the data of Table VII.

3.4. *Comparison of Ensemble-Averages*

The ensemble-average energy of a mode is defined as $E^{\rm av}_{\rm mode} = \sum_{n=1}^{N} E^{n}_{\rm mode}/N$, where $\{E^{n}_{\rm mode}\}$ are the energies obtained from the N different trajectories at the same time step. In contrast to the method of determining time-average mode energies, the ensemble-averaging is not based on a single trajectory and thus the effects of integrator inaccuracies in determining $E^{n}_{\rm mode}$ are averaged out at each time step. This fact, together with the fact that the simulations are performed over shorter trajectories, leads one to expect that the ensemble-average equilibrium values will be less sensitive to the algorithm used, as will be the time taken for mode relaxation to equilibrium.

The Boltzmann H-function [12] was used to determine the time taken for mode relaxation to local equilibrium. If this function falls to a plateau region (i.e., a minimum that is maintained over a period of time) then the system is taken to be in equilibrium [12] and the relaxation time is the time taken to reach this plateau region. The coarse-grained H-function for mode m was determined as follows: One hundred trajectories were propagated under the conditions

TABLE IX

Plateau Value of the H-Function Obtained from the Normal Mode Ensemble Averages and Related Relaxation Times

Name of algorithm	Plateau value for mode 5	Decay time for mode 5/ps	Plateau value for mode 19	Decay time for mode 19/ps
GVTS	166	7.94	152	3370
GFTS	166	8.36	149	3400
VLPF	166	8.36	151	3135
Verlet	166	8.36	149	3880
Runge–Kutta	166	8.36	156	3170
Radau	165	8.36	150	3780

TABLE X

Same as Table IX, but for Local Modes 10 and 11

Name of algorithm	Plateau value for bond 10	Decay time for bond 10/ps	Plateau value for bond 11	Decay time for bond 11/ps
GVTS	150	25.4	169	45.3
GFTS	150	30.0	167	43.9
VLPF	150	30.0	165	45.3
Verlet	151	30.0	169	44.6
Runge–Kutta	150	30.0	167	43.9
Radau	150	26.5	168	41.1

described in the previous section and the energy of mode m was monitored over the length of these trajectories. The maximum and minimum energy, $E_m^{\max}$ and $E_m^{\min}$, that mode m obtained over any of these trajectories was used to define an energy range, $[E_m^{\max}, E_m^{\min}]$, for mode m. This range was subsequently divided into 100 equal bins each having a range of $[E_m^{\max}, E_m^{\min}]/100$. Since 100 trajectories were run there are 100 energies, $\{\varepsilon_m\}$, associated with mode *m at any specific time step*. These energies were partitioned into the bins defined above. If $n_{i,m}$ is the number of occasions that a mode energy, ε_m, was inserted into bin i ($i = 1, ..., 100$), then the H-function is given by

$$H_m = \sum_{i=1}^{100} n_{i,m} \ln n_{i,m}.$$

This function is, therefore, determined at each time step and can thus be evaluated over the length of the trajectory.

A typical ensemble-generated H-function plot is shown in Fig. 10. This is the plot obtained for normal mode 19 using the GVTS algorithm and shows the decay to a plateau region; that is, it shows the decay to local equilibrium.

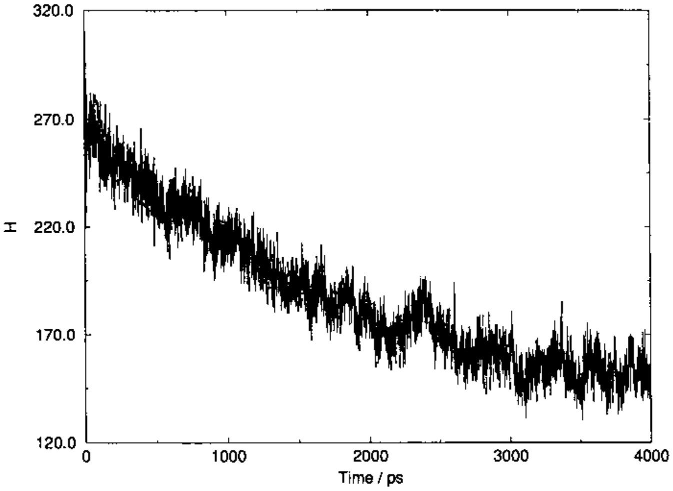

FIG. 10. The ensemble generated Boltzmann H-function for normal mode 19 illustrating the typical decay to a plateau region. This plot was generated using the GVTS algorithm.

The comparison of the H-function plateau value (i.e., the value of the H-function where the plateau value is reached—see, for example, Fig. 10 where the plateau value is 152) and the time taken to reach equilibrium is shown in Tables IX and X, while Table XI lists the equilibrium energies of the various modes. It is apparent that all algorithms give comparable relaxation energies and times of relaxation when the ensemble-average method is employed. It is therefore more feasible to compare the ensemble-generated relaxation energies and times obtained from different algorithms than the corresponding time-average values.

4. CONCLUSIONS

Algorithms representing a wide range of molecular dynamics integrators have been compared. The system used to conduct the comparison is considered to be severe for two reasons: First, the system is apparently nonergodic at the internal energy considered and thus the local equilibrium attained is dependent on the starting coordinates. If the system had been ergodic then equilibrium may have been obtained by all algorithms, thereby minimising the influences of integrator related fluctuations in the trajectory. Second, the system is under a high pressure. This restricts

TABLE XI

Ensemble-Average Equilibrium Mode Energies (kcal mol^{-1})

Algorithm name	Normal mode 5	Normal mode 19	Bond 10	Bond 11
GVTS	1.30	0.0744	0.909	0.207
GFTS	1.31	0.108	0.907	0.208
VLPF	1.31	0.105	0.908	0.215
Verlet	1.31	0.105	0.907	0.189
Runge–Kutta	1.31	0.0999	0.913	0.201
Radau	1.31	0.108	0.909	0.201

the bond motion to the repulsive region of the bond potentials for the majority of the time. The forces are thus always large, thereby decreasing the efficiency of the higher order variable time step algorithms which rely on large step lengths in regions of low force to increase their efficiency.

The accuracy of the six different integrators studied has been compared by investigating their energy conservation ability and back integrability. The results show the expected trend that the higher order algorithms are more accurate than the low order algorithms and that the more sophisticated variable time step procedures are more accurate than the fixed time step procedures. The lowest order Verlet and VLPF algorithms show the best energy conservation for long time trajectories (due to the method of integration which satisfies time reversal invariance). In general, the increase in accuracy is accompanied by a loss of efficiency. It has been shown, however, that the Radau integrator is highly accurate and efficient, provided that output is not required at frequent and specific times.

Inter-algorithm comparison between ensemble-average equilibrium values and time taken for mode relaxation is far more feasible than the corresponding time-average comparison. The results show that the mode relaxation times and average energies are insensitive to the algorithm used for the simulation when ensemble-averaging techniques are employed.

The comparisons made in this report show that the Radau algorithm has superior conservation properties to the other self-starting integrators and is also more efficient in general applications where output is not required at regular intervals. This observation holds for the apparently nonergodic, rapid collision system studied. It is in full agreement with the evidence from earlier applications of the Radau algorithm made by Marković *et al.* to binary molecular collisions [9] and large inert gas clusters [8].

A general conclusion from these results relates to the importance of selecting the algorithm that is best suited to the type and accuracy of output required from an MD simulation in addition to the characteristics of the MD trajectory (e.g., length of run). For example, if the ergodicity of a system is to be ascertained over a μs period or longer, it is imperative to have good energy conservation (since a system's ergodic behaviour is strongly energy dependent). The low order Verlet and VLPF algorithms *may* be best suited to this type of MD simulation because of their long time energy conservation characteristics. However, this is achieved at the cost of poorer short time accuracy. Thus, timescales shorter than 100 ps may be better probed by higher order algorithms in the case of the present system. It is, of course, common to improve energy conservation by rescaling of velocities at regular intervals during the run, but this device may be associated with errors of another type. Higher order algorithms are thus not always superior to their low order counterparts.

APPENDIX: THE IMPLICIT GAUSS–RADAU INTEGRATION TECHNIQUE

The description of the Radau algorithm presented here is merely a brief overview of the integration technique and is included in this report for the sake of completeness. The concepts presented here have been extracted from a more detailed description which has been presented, together with the relevant computer code, by Everhart [7, 14].

The force (acceleration) experienced by a particle at time t is a function of the positions of all the particles in that system at t. Since the force is separable in the x, y, and z directions, one has

$$\ddot{x} = F_x = f(x, y, z, \alpha, \beta, \gamma, ..., t),$$

where F_x is the force on the particle in the x direction; x, y, z are the position coordinates of particle 1; α, β, γ are the position coordinates of particle 2; etc. In many of the common integration techniques, this force is included in the Taylor expansion of the position (and velocity) which, when no higher order terms are added, yields an integration of third-order accuracy. The force used in these integration techniques is the force on the particle at the beginning of the time step. It is assumed that the time step chosen is sufficiently small so that the force is essentially constant over the integration step. Clearly, the assumption of a constant force leads to inaccuracies in the integration. Integration schemes such as the Runge–Kutta and the Gauss–Radau techniques thus attempt to employ a force which is more representative of the force over the complete step length. In the present application the force is expanded in a time series of the form

$$\ddot{x} = F_x = F_1 + A_1 t + A_2 t^2 + A_3 t^3 + \cdots + A_N t^N, \quad (1)$$

where F_1 is the force at $t = 0$; that is, F_1 is the force on the particle in the x direction at the beginning of the time step. The velocity and position of the particle at t can be obtained by integrating Eq. (1). Thus

$$\dot{x} = \dot{x}_1 + F_1 t + A_1 t^2 + A_2 t^3/3 + A_3 t^4/4 + \cdots + A_N t^{N+1}/(N+1) \quad (2)$$

and

$$x = x_1 + \dot{x}_1 t + F_1 t^2/2 + A_1 t^3/6 + A_2 t^4/12 + A_3 t^5/20 + \cdots + A_N t^{N+2}/(N+1)(N+2). \quad (3)$$

Clearly, if the coefficients $A_1, ..., A_N$ can be determined accurately then the value calculated for the position and velocity at time t will be accurate. This is the aim of trajectory integration.

There are many sets of coefficients satisfying Eqs. (1) to (3). Some will exceed the accuracy implied by the Taylor series expansion. Such sets will be used here. The form of Eq. (1) is thus not the usual Taylor expansion since the coefficients $A_1, ..., A_N$ are altered when integrating over subsequent steps (called sequences) in order to obtain an accurate position and velocity at the end of each step (sequence).

The procedure outlined below is followed in order to determine the coefficients in Eq. (1) at the end of a sequence of length T. Time steps $t_1 = 0$, t_2, t_3, ..., $t_N = T$ are chosen within the sequence length T and F_x is calculated at each of these steps. If $F_x(t_1), ..., F_x(t_N)$ are known—and F_x is analytical—then it is possible to express F_x in the form

$$F_x = F_1 + \alpha_1 t + \alpha_2 t(t - t_2) + \alpha_3 t(t - t_2)(t - t_3) + \cdots, \qquad (4)$$

where the coefficients $\alpha_1, ...$ can readily be determined by noting that

$$\begin{aligned} &F_x(t_1) = F_1 \qquad \text{(since } t_1 = 0\text{)} \\ &F_x(t_2) = F_1 + \alpha_1 t_2 \\ &F_x(t_3) = F_1 + \alpha_1 t_3 + \alpha_2 t_3(t_3 - t_2) \\ &\quad\vdots \end{aligned} \qquad (5)$$

Solving for $\{\alpha_i\}$ yields,[4]

$$\begin{aligned} &\alpha_1 = (F_2 - F_1)/t_2 \\ &\alpha_2 = ((F_3 - F_1)/t_3 - \alpha_1)/(t_3 - t_2) \\ &\quad\vdots \end{aligned} \qquad (6)$$

The coefficients $A_1, ..., A_N$ in Eq. (1) can be determined from these values of α_i. By equating like powers of t in Eqs. (1) and (4) one obtains

$$\begin{aligned} A_1 &= \alpha_1 + (-t_2)\,\alpha_2 + (t_2 t_3)\,\alpha_3 + \cdots \\ &= c_{11}\alpha_1 + c_{21}\alpha_2 + c_{31}\alpha_3 + \cdots \\ A_2 &= \alpha_2 + (-t_2 - t_3)\,\alpha_3 + \cdots \\ &= c_{22}\alpha_2 + c_{32}\alpha_3 + \cdots \\ A_3 &= \alpha_3 + \cdots = c_{33}\alpha_3 + \cdots \\ &\ \vdots \end{aligned} \qquad (7)$$

where the coefficients c_{ij} have the recursion relationships given by

$$\begin{aligned} &c_{ii} = 1, \\ &c_{i1} = -t_i c_{i-1,1}, && i > 1, \\ &c_{ij} = c_{i-1,j-1} - t_i c_{i-1,j}, && 1 < j < i. \end{aligned}$$

Thus, by determining the values of α_i from Eqs. (6), accurate values of $A_1, ..., A_N$ can be determined from Eqs. (7) and hence, accurate positions and velocities can be predicted from Eqs. (2) and (3). Explicitly, if the force on a particle in the x direction at $t = t_1 = 0$ (the beginning of the sequence) is F_1 and the first time step is t_2, then the force at t_2 is given by $F_x(t_2) = f(x, y, z, x_1, y_1, z_1, ..., t_2)$. If this is the first sequence of the trajectory then $A_1 \cdots A_N$ are set equal to zero so that (from Eqs. (2) and (3))

$$\dot{x}(t_2) = \dot{x}_1 + F_1 t_2$$

and

$$x(t_2) = x_1 + \dot{x}_1 t_2 + F_1 t_2^2/2.$$

These positions (and velocities) are used to calculate $F_x(t_2)$. α_1 is then determined from Eqs. (6), i.e.,

$$\alpha_1 = (F_x(t_2) - F_1)/t_2.$$

Finally, from Eqs. (7),

$$A_1 = \alpha_1.$$

Similarly, if the next time step is t_3, then the velocities and positions at t_3 are obtained from Eqs. (2) and (3) as

$$\dot{x}(t_3) = \dot{x}_1 + F_1 t_3 + A_1 t_3^2/2$$

and

$$x(t_3) = x_1 + \dot{x}_1 t_3 + F_1 t_3^2/2 + A_1 t_3^3/6.$$

These coordinates are used to calculate the force on the particle in the x direction at t_3, i.e., $F_x(t_3)$. Equation (6) is then used to determine α_2, i.e.,

$$\alpha_2 = ((F(t_3) - F_1)/t_3 - \alpha_1)/(t_3 - t_2),$$

and, hence, A_1 is improved *and* A_2 is determined from Eq. (7). That is,

$$A_1 = \alpha_1 + (-t_2)\,\alpha_2$$

[4] Each new time step adds another α_i to Eq. (4) (and another equation to Eqs. (5)) without altering the preceding $\{\alpha_i\}$. This very important point will be referred to later.

and

$$A_2 = \alpha_2.$$

Thus, at each new step t_i the previous values of A_j, $j < i-1$ are improved and A_{i-1} is determined. In order to improve the values of $A_1, ..., A_N$ even further, the sequence may be repeated a number of times where the initial $A_1, ..., A_N$ of a subsequent sequence are the final $A_1, ..., A_N$ of the preceding sequence.

It is found that the first sequence of a trajectory—where $A_1, ..., A_N$ are set equal to zero—requires six sequence repetitions to obtain $A_1, ..., A_N$ of sufficient accuracy. However, in subsequent sequences the previous $A_1, ..., A_N$ (or an extrapolated value of the previous $A_1, ..., A_N$) are used as the initial $A_1, ..., A_N$ and, hence, just two sequence repetitions are required for accurate prediction of the velocity and positions. In this manner the computational efficiency is dramatically improved.

The preceding description is valid for any (fixed or variable) time steps $t_1 = 0$, $t_2, t_3, ..., t_N = T$ within the sequence. Hence, if it is decided to solve F_x over three time steps within the sequence, i.e., $t_1 = 0$, t_2, t_3, $t_4 = T$, Eqs. (2) and (3) have the form

$$\dot{x} = \dot{x}_1 + F_1 t + A_1 t^2/2 + A_2 t^3/3 + A_3 t^4/4 \tag{8}$$

and

$$x = x_1 + \dot{x}_1 t + F_1 t^2/2 + A_1 t^3/6 + A_2 t^4/12 + A_3 t^5/20. \tag{9}$$

Thus the position is accurate to order 5 in t and the velocity is accurate to order 4 in t. It is possible to choose specific step sizes so as to increase the order in position accuracy by 2 and in the velocity accuracy by 3 without any additional steps (i.e., without decreasing the computational efficiency). This technique is explained below.

If seventh-order accuracy in the position is desired then an expression of the form

$$x = x_1 + \dot{x}_1 t + F_1 t^2/2 + A'_1 t^3/6 + A'_2 t^4/12 + A'_3 t^5/20 + A'_4 t^6/30 + A'_5 t^7/42$$

is required. The difference between the seventh- and fifth-order prediction of x at the end of the sequence T is

$$\Delta x = (A'_1 - A_1)\, T^3/6 + (A'_2 - A_2)\, T^4/12 + (A'_3 - A_3)\, T^5/20 + A'_4 T^6/30 + A'_5 T^7/42. \tag{10}$$

The aim is to choose t_2, t_3, t_4 in such a way that $\Delta x = 0$. If this can be achieved then, by using just three time steps within each sequence, seventh-order accuracy will be obtained.

As was specified in the footnote earlier in the Appendix, incorporating additional values of α_i in Eq. (4) does not alter the preceding α_i. Thus, from Eqs. (7) it is clear that

$$A'_1 - A_1 = c_{41}\alpha_4 + c_{51}\alpha_5,$$
$$A'_2 - A_2 = c_{42}\alpha_4 + c_{52}\alpha_5,$$
$$A'_3 - A_3 = c_{43}\alpha_3 + c_{53}\alpha_5,$$
$$A'_4 = c_{44}\alpha_4 + c_{54}\alpha_5,$$
$$A_5 = c_{55}\alpha_5.$$

Using the recursion relationships described previously, the c_{5i} coefficients can be expressed in terms of $c_{41}, ..., c_{44}$. That is,

$$c_{55} = 1,$$
$$c_{54} = c_{43} - t_5 c_{44},$$
$$c_{53} = c_{42} - t_5 c_{43},$$
$$c_{52} = c_{41} - t_5 c_{42},$$
$$c_{51} = -t_5 c_{41}.$$

Substituting these changes into Eq. (10) and introducing the variable $h_i = t_i/T$ which leads to the recursion relationships being expressed in terms of h_i (e.g., $c_{41} = -t_2 t_3 t_4 \to c'_{41} = -h_2 h_3 h_4$) yields

$$\Delta x = (\alpha_4 - t_5\alpha_5)\, T^6 [c'_{41}/6 + c'_{42}/12 + c'_{43}/20 + 1/30] + \alpha_5 T^7 [c'_{41}/12 + c'_{42}/20 + c'_{43}/30 + 1/42].$$

Setting the expressions in the square brackets to zero will set Δx to zero (and hence seventh-order integration accuracy will be obtained) independently of α_4, α_5, and t_5. To achieve this one must solve

$$c'_{41}/6 + c'_{42}/12 + c'_{43}/20 + 1/30 = 0,$$
$$c'_{41}/12 + c'_{42}/20 + c'_{43}/30 + 1/42 = 0,$$

and also

$$c'_{41}/2 + c'_{42}/3 + c'_{43}/4 + 1/5 = 0,$$
$$c'_{41}/3 + c'_{42}/4 + c'_{43}/5 + 1/5 = 0,$$
$$c'_{41}/4 + c'_{42}/5 + c'_{43}/6 + 1/7 = 0,$$

which are obtained from a similar treatment of the velocity coordinates. Solving these equations simultaneously (two of them are redundant) yields

$$c'_{41} = -h_2 h_3 h_4 = -\tfrac{4}{35},$$
$$c'_{42} = h_2 h_3 + h_3 h_4 + h_2 h_4 = \tfrac{6}{7},$$
$$c'_{43} = -h_2 - h_3 - h_4 = -\tfrac{12}{7},$$

or, expressed in another form,

$$h_2 = t_2/T = 0.2123,$$
$$h_3 = t_3/T = 0.5905,$$
$$h_4 = t_4 T = 0.9114.$$

These are the Gauss–Radau spacings.

Thus, if the sequence length T is known then h_i (or t_i) can be chosen to be the Gauss–Radau spacings so that $\Delta x = 0$. Seventh-order accuracy in the velocity and the position is thereby obtained from using just three time steps within the sequence. That is, the integration accuracy is greatly enhanced without reducing the computational efficiency.

As has been described, an improved accuracy with good efficiency is obtained by using Gauss–Radau spacings and by using initial values of $A_1, ..., A_N$ that are (extrapolated from) the final values of the peceding sequence. It is possible to increase the efficiency of the algorithm still further by allowing the sequence length, T, to vary. In this way the sequence length can be maximised—within an accuracy constraint—thereby improving the efficiency. This is achieved as follows.

The value of A_N (the last coefficient in the force expansion) which is used as an initial value in a new sequence, denoted A'_N, is determined from the preceding A_N by

$$A'_N = \frac{T'}{T} A_N, \qquad (11)$$

where T' is the length of the new sequence. This relation is valid for all extrapolation methods used. The last term in the position expansion—which is a function of A_N (see Eq. (3))—gives a good indication of the accuracy of the integration; that is, the smaller the value of this term the more accurate the expansion since it implies that the higher order terms make insignificant contributions to the expansion. This term is thus used to control the accuracy of integration in the following manner: An accuracy constraint is entered into the integration scheme as an input parameter (LL). This parameter is used by setting

$$10^{-\mathrm{LL}} = A'_N T'^{N+2}/(N+1)(N+2).$$

Thus LL governs the magnitude of the final term in the next sequence. From Eq. (11) one has

$$10^{-\mathrm{LL}} = (T'/T)\, A_N T'^{N+2}/(N+1)(N+2).$$

Rearranging yields

$$T'^{N+3} = T(N+1)(N+2)\, 10^{-\mathrm{LL}}/A_N.$$

In this way T', the new sequence length, is determined to comply to the desired accuracy constraint.

In summary, the Gauss–Radau integration offers an extremely accurate method of integration while maintaining good efficiency. This is achieved by performing the force evaluations at Gauss–Radau spacings, using previous coefficients $A_1, ..., A_N$ which are merely improved upon in subsequent sequences and by maximising the sequence length while still maintaining a predetermined accuracy. This algorithm has therefore been employed in the field of astronomy [7] and to a far smaller extent in simulations of chemical and physical systems [8, 9].

ACKNOWLEDGMENTS

K. B. extends his gratitude to N. Marković and T. D. Sewell for their interest and illuminating discussions and to the Swedish Natural Science Research Council and the Foundation for Research Development, South Africa, for financial support.

REFERENCES

1. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon, Oxford, 1990).
2. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes:The Art of Scientific Computing* (Cambridge Univ. Press, Cambridge, 1987).
3. J. Marro and J. Masoliver, *J. Phys. C: Solid State Phys.* **18**, 4691 (1985).
4. A. Wallqvist and O. Teleman, Properties of flexible water molecules, *Mol. Phys.* **74**, 515 (1991).
5. H. W. Schranz, L. M. Raff, and D. L. Thompson, Intramolecular energy transfer and mode-specific effects in unimolecular reactions of disilane, *J. Chem. Phys.* **95**, 106 (1991).
6. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice–Hall, Englewood Cliffs, NJ, 1971).
7. E. Everhart, "An Efficient Integrator That Uses Gauss–Radau Spacings," in *Dynamics of Comets: Their Origin and Evolution*, edited by A. Carusi *et al.* (Reidel, Dordrecht, 1985), p. 185.
8. J. B. C. Pettersson and N. Marković, Dynamics of cluster scattering from surfaces, *Chem. Phys. Lett.* **201**, 421 (1993).
9. N. Marković, G. Nyman, and S. Nordholm, Complex formation in O + OH collisions—A two-step mechanism, *Chem. Phys. Lett.* **159**, 435 (1989).
10. J. Davidsson, S. Nordholm, and L. L. Anderson, The interaction strength dependence of diatomic collisional energy transfer: A molecular dynamics study, *Chem. Phys. Lett.* **191**, 489 (1992).
11. R. S. Berry, S. A. Rice, and J. Ross, *Physical Chemistry* (Wiley, New York, 1987), p. 267.
12. B. I. Henry and J. Grindlay, Reversible approach to statistical equilibrium in a nonlinear chain: An ensemble study, *Physica D* **28**, 49 (1987).
13. P. Brumer and M. Shapiro, Chaos and reaction dynamics, *Adv. Chem. Phys.* **70**, 365 (1988).
14. E. Everhart, Implicit single-sequence methods for integrating orbits, *Celest. Mech.* **10**, 35 (1974).