



Computer Simulation of Complex, Strongly Coupled Nanometer-Scale Systems: Breaking the Billion Atom Barrier

R. E. TUZUN, D. W. NOID AND B. G. SUMPTER

Chemical and Analytical Sciences Division

Oak Ridge National Laboratory

Oak Ridge, TN 37831-6197, U.S.A.

tuzunre@ornl.gov

Abstract—The capabilities of polymer science and computational chemistry are reaching a point of convergence. New computer hardware and novel computational methods have created opportunities to investigate new polymeric materials, as well as to model and predict their properties. The recent arrival of massively parallel computers and new algorithms for sharing computational tasks among many processors now bring simulation sizes on the order of 10^9 atoms to within reasonable time limits and will allow for new studies in emerging fields such as molecular nanotechnology.

Keywords—Molecular dynamics, Symplectic integration, Geometric statement function method.

1. INTRODUCTION

Enormous improvements in simulation methodology and in computer hardware have recently created opportunities to investigate new polymeric materials, as well as to model and predict their properties. Polymer simulations involving hundreds of thousands of atoms may now be accomplished within reasonable time limits on standard workstations; the advent of parallel computing has extended this limit into the billions [1]. Because many details of polymer dynamics are not directly observable, the molecular dynamics [2,3] (MD) method is crucial for the understanding of polymers.

Recent progress toward miniaturization has demanded new ways of thinking about mechanical devices. This is particularly true in the technologies of sensors-on-a-chip and information storage, where micro-electromechanical systems (MEMS) are recognized as major, new areas for development. The logical extension of this technology is into the area of nano-scale devices such as bearings and gears, in which the whole component is comprised of only a few hundreds or thousands of atoms. This area of study, often referred to as molecular nanotechnology, has the potential to revolutionize chemistry, materials science, biology, and many other fields by creating an entirely new set of atomically precise mechanical devices and molecular machines. Molecular

Research sponsored by the Division of Materials Sciences, Office of Basic Energy Sciences, U.S. Department of Energy under Contract DE-AC05-84OR21400 with Lockheed Martin Energy Systems, Inc. Also supported in part by an appointment to the Oak Ridge Postdoctoral Research Associates Program administered jointly by Oak Ridge National Laboratory and the Oak Ridge Institute for Science and Education. The authors acknowledge the use of Intel Paragon supercomputers located in the Center for Computational Sciences at ORNL, funded by the DOE's Mathematical, Information, and Computational Sciences (MICS) Division of the Office of Computational and Technology Research.

nanotechnology proposes to exploit a bottom-up approach to constructing nano-scaled objects, that is, assembling components atom-by-atom. In principle, any arrangement of atoms that is consistent with the laws of chemistry and physics can be made. Based on this premise, Drexler [4] and Merkle have proposed a number of possible designs that represent essential components of machines that perform specific operations.

While molecular nanotechnology is not a present reality, the surprising capabilities of the Atomic Force Microscope, laser lithography and the optical tweezers give credibility to predictions of molecular control of complex structures near its atomic limit. Simulation of the internal dynamics and performance of molecular-based materials will provide much of the needed data for developing and testing fundamental concepts and designs of nanomachines or components. Many proposed nanodevices have several million atoms and are thus too large to conveniently simulate on a standard workstation. In addition, in order to examine details pertaining to the operation of such devices, the simulation time must be comparable to their expected operation time. Thus, it is clear that new avenues to the molecular dynamics (MD) method, which has traditionally been plagued by a size-time scale bottleneck, are needed to make the fundamental foundation of the method useful.

At long simulation times, there is often an additional problem. Most integration algorithms exhibit small drift rates in energy and other conserved quantities, which at large times would lead to unacceptably large total drifts in energy. Although velocity rescaling would remove this drift, there is no guarantee that the resulting dynamics would be realistic. This issue could be especially important in treating, for example, an interface between two polymer surfaces, where errors in the dynamics could cause disproportionately large surface effects.

In this paper, we will discuss two enhancements to traditional MD methods: geometric state-function functions [5,6] (for producing faster MD codes) and symplectic integration [7] (for generating stable long time trajectories). An important aspect of these techniques is that the simulations are very efficient (orders of magnitude faster than previous methods) and exact (classical constants of the motion, such as energy and momentum, are conserved). The use of these techniques is illustrated for nanotechnology and polymer science applications, and issues involving parallel computation discussed.

2. MOLECULAR DYNAMICS METHOD

MD essentially consists of integrating, over small time steps, Hamilton's equations of motion [8,9]:

$$\dot{p}_i = -\frac{\partial H}{\partial q_i}, \quad \dot{q}_i = \frac{\partial H}{\partial p_i}, \quad (1)$$

where q_i refers to a component of position, p_i its conjugate momentum, the dot to a time derivative, and H to the classical Hamiltonian consisting of the sum of kinetic and potential energies. Although the above equations are valid for any set of conjugate positions and momenta, MD simulations are usually performed in Cartesian coordinates because this greatly simplifies (in fact, completely decouples) the kinetic energy term. In polymer systems, the potential energy is usually written as a sum of interactions:

$$V = \sum_i V(r_i) + \sum_j V(\theta_j) + \sum_k V(\cos \tau_k) + \sum_{m < n} V_{NB}(r_{mn}), \quad (2)$$

where i , j , and k denote sums over stretch, bend, and torsion interactions, and m and n nonbonded interactions between atoms m and n . Stretch potential energy terms are typically of the Morse functional form:

$$V(r) = D \left[1 - e^{-\alpha(r-r_e)} \right]^2, \quad (3)$$

where D , α , and r_e are constants. This is a reasonable approximation for a chemical bond with equilibrium length r_e which may dissociate. Bend and torsion potential energy terms are typically

polynomials in the bend or torsion angles. Cross terms (for example, bend-stretch interactions) may also be included.

To perform an MD simulation requires initial conditions (positions and momenta). Positions are often initially set to an equilibrium conformation. Momenta are set so as to yield an initial temperature or some other desired initial condition. If momenta are set randomly, center of mass translational (and sometimes rotational) motion are usually subtracted out to fix the frame of reference. If the system under study has more than one independent bond network (for example, two polymer crystals in close proximity), translational and rotational motion is usually removed from each piece separately.

At each time step, depending on the algorithm, one or more force evaluations are required. The forces on particle i for a potential energy term $V(\phi)$, where ϕ is some internal coordinate, may be written

$$F_{qi} = -\frac{\partial V}{\partial \phi} \frac{\partial \phi}{\partial q_i}, \quad (4)$$

where F_{qi} refers to the q (x , y , or z) component of the force acting on particle i . Force calculations are the most expensive and difficult to program part of an MD simulation, with derivatives of internal coordinates taking most of the computational effort.

3. OPTIMIZATION OF FORCE CALCULATIONS

The computational effort for an MD simulation is directly proportional to the simulation time and, if nonbonded interaction cutoff distances are implemented, to the number of particles. For simulations in the nanosecond range (typically millions of time steps), this can prove impractical. The size-time scale bottleneck can be especially severe for polymers and other systems with highly connected bond networks, because the number of three- and four-body interactions is large and three- and four-body forces can be time-consuming to calculate. The geometric statement function method [5,6], which has been continually developed over the past five years, minimizes the effort for force evaluations and is eminently suitable for systems with highly connected bond networks. This method is now illustrated for a fully atomistic 100 monomer unit polyethylene chain, $(\text{CH}_2)_{100}$.

In the polyethylene chain $(\text{CH}_2)_{100}$, there are 299 CC and CH bond stretches, 396 CCC, HCH, and CCH bends, 97 CCCC torsions, and nonbonded interactions. To calculate the forces for a stretch interaction between atoms 1 and 2 requires the bond distance r_{12} , which requires a square root. Derivatives of r_{12} may be calculated from

$$\frac{\partial r_{12}}{\partial q_1} = \frac{q_1 - q_2}{r_{12}}, \quad (5)$$

for which it is convenient to store $1/r_{12}$ as an intermediate.

Each bond may be a part of as many as six bends. It therefore makes sense to use bond distances and their derivatives as intermediates for bend angles and their derivatives. For example, the usual dot product expression for an angle θ for a bond sequence 123 can be rewritten

$$\cos \theta = \frac{\partial r_{12}}{\partial x_1} \frac{\partial r_{23}}{\partial x_3} + \frac{\partial r_{12}}{\partial y_1} \frac{\partial r_{23}}{\partial y_3} + \frac{\partial r_{12}}{\partial z_1} \frac{\partial r_{23}}{\partial z_3}, \quad (6)$$

and thus, to calculate a bend angle from stretch intermediates requires only a few multiplications and divisions. Similarly, derivatives of $\cos \theta$ may be calculated as follows:

$$\frac{\partial \cos \theta}{\partial q_1} = -\frac{1}{r_{12}} \left(\frac{\partial r_{12}}{\partial q_1} \cos \theta + \frac{\partial r_{23}}{\partial q_2} \right), \quad (7)$$

$$\frac{\partial \cos \theta}{\partial q_3} = \frac{1}{r_{23}} \left(\frac{\partial r_{23}}{\partial q_2} \cos \theta + \frac{\partial r_{12}}{\partial q_1} \right). \quad (8)$$

In some potential energy surfaces, bend potential energy terms are parameterized by θ rather than $\cos \theta$. Derivatives of θ may be simply calculated from derivatives of $\cos \theta$

$$\frac{\partial \theta}{\partial q_i} = -\frac{1}{\sin \theta} \frac{\partial \cos \theta}{\partial q_i}, \quad (9)$$

which requires a square root for the factor of $\sin \theta$.

We now turn to torsion angles. A torsion angle for a bond sequence 1234 may be simply calculated from

$$\cos \tau = \frac{1}{\sin \theta_{123} \sin \theta_{234}} \left(\cos \theta_{123} \cos \theta_{234} - \frac{\partial r_{12}}{\partial x_1} \frac{\partial r_{34}}{\partial x_3} - \frac{\partial r_{12}}{\partial y_1} \frac{\partial r_{34}}{\partial y_3} - \frac{\partial r_{12}}{\partial z_1} \frac{\partial r_{34}}{\partial z_3} \right). \quad (10)$$

The first derivatives for the end atoms are

$$\frac{\partial \cos \tau}{\partial q_1} = a_{123} \left(\cos \tau \frac{\partial \theta_{123}}{\partial q_3} - \frac{\partial \theta_{234}}{\partial q_2} \right), \quad (11)$$

$$\frac{\partial \cos \tau}{\partial q_4} = a_{432} \left(\cos \tau \frac{\partial \theta_{234}}{\partial q_2} - \frac{\partial \theta_{123}}{\partial q_3} \right), \quad (12)$$

where

$$a_{123} = \frac{r_{23}}{r_{12} \sin \theta_{123}}, \quad a_{432} = \frac{r_{23}}{r_{34} \sin \theta_{234}}. \quad (13)$$

Middle atom derivatives can be written as linear combinations of bend angle derivatives. In addition to being up to 60 times faster, these formulas are more accurate than the corresponding Cartesian formulas because of the smaller number of operations. Savings are even greater for second derivatives, whose formulas are more complex and which are used in molecular mechanics and normal coordinate analysis calculations.

Further effort is saved by noting that internal forces are translationally invariant, i.e., for an internal force acting on atoms 1 through n , we have

$$\mathbf{F}_1 + \mathbf{F}_2 + \cdots + \mathbf{F}_n = \mathbf{0}. \quad (14)$$

For a bend interaction, this means that forces on the center atom may be calculated from those on the two end atoms. We choose the center atom because the formulas for bend forces on the center atom are more complicated. This strategy cuts the effort for calculating bend forces by an additional 50%. All of these methods are conveniently implemented as FORTRAN statement functions, which is why we call them the geometric statement function method.

4. LONG SIMULATION TIMES

The current practical limits on simulation times are beginning to approach the microsecond range. At such long simulation times, many integration algorithms introduce unacceptably large drifts in energy and other conserved quantities. Over short time scales, for example, high-order predictor-corrector methods [9] can be made to conserve energy as accurately as desired. However, the energy (and angular momentum) tend to have some steady linear drift rate with time. At long times, even a small drift rate can yield an unacceptably large energy or angular momentum drifts. As discussed in the Introduction, velocity or other rescaling does not guarantee correct dynamics. As a side issue, high-order predictor-corrector calculations are expensive.

A common numerical integration algorithm in MD simulations is the Verlet method. It has the advantage of being easy to program and is very robust, i.e., it often generates stable, long time trajectories. The Verlet method is a second-order symplectic integrator (SI), and its reliability is now understood to be in part due to this. SIs are special algorithms for solving the classical equations of motion that are guaranteed to preserve certain dynamical properties that the exact

trajectories are known to exhibit. Over the past ten years, SIs have been demonstrated to provide reliable long time trajectories which exhibit small deviations of conserved quantities about a steady average rather than linear drift.

The basic symplectic integration algorithm is to divide a single time step into several intermediate time steps as follows:

```

do k = 1, M,
    p(k) = p(k-1) + bk dtF (q(k-1)),
    q(k) = q(k-1) + ak dtG (p(k)),
end do,

```

where dt is the time step, a_k and b_k are numbers corresponding to fractions of a time step, q and p are coordinates and their conjugate momenta, and G and F are, respectively, their derivatives. Judiciously choosing (a_k, b_k) guarantees conservation of energy and angular momentum to some desired order. Our simulations typically use a fifth-order symplectic integrator which requires 4 force evaluations per 1 fs time step; this usually leads to energy and angular momentum conservation to better than five figures for simulation times exceeding 2 ns (two million time steps), with undetectable drift.

As increasing computer power and improved simulation techniques increase the practical simulation time range into the microsecond region and beyond, long time stability will become an increasingly important issue. To keep the drift in energy and other conserved quantities within reasonable limits, and also, at such long times will probably require only increasing the order of symplectic integrators rather than the development of any new integration algorithms.

In some cases, an integration algorithm that is nonoptimal for serial machines may prove optimal on parallel machines. For example, using a high-order symplectic integrator enables using a larger time step to obtain accurate dynamics. On a serial machine, a point of diminishing returns will rapidly be reached (doubling the order of symplectic integration, and therefore, the number of force evaluations, will not double the time step allowable for equally accurate dynamics). On a parallel machine, however, lengthening the time step would drastically reduce the total communication traffic for a given total simulation time. Although this may increase the total number of operations, depending on the communication speed, it could greatly decrease the real (wall clock) simulation time.

5. NANOTECHNOLOGY APPLICATIONS

In order to understand the dynamics of nanomachine operation, we first seek to understand the dynamics of individual components. To this end, we have undertaken several studies involving individual carbon nanotubes [10], rotational friction of carbon nanotubes [11], the introduction of controlled motion [12], and fluid flow [13,14]. Issues involving parallel computing are discussed in the context of several nanotechnology applications.

Various types of molecular bearings have recently been proposed in the growing nanotechnology literature. Using novel molecular dynamics methods, we have recently simulated several model carbon nanotube bearings. This system was chosen in part because carbon nanotubes have already been synthesized in a variety of sizes and shapes and manipulating them into bearings seems plausible. The bearings varied in size from an inner shaft of between 4 and 16 Å in diameter and up to 120 Å in length, and an outer cylinder of between 10 and 23 Å in diameter and up to 40 Å in length. Figure 1 shows the basic construction of a nanobearing at the beginning of a simulation. Both the shaft (inner cylinder) and sleeve (outer cylinder) are centered about a common rotational axis, denoted the z axis, and the shaft extends symmetrically out of the sleeve.

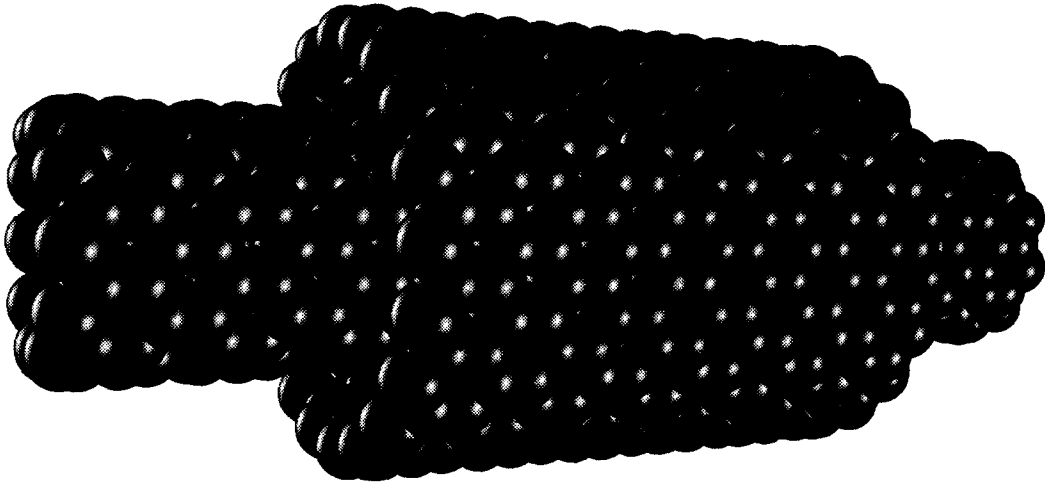


Figure 1. Basic shaft-sleeve construction of a nanobearing at the beginning of a simulation.

Typical results are shown in Figure 2. The shaft and sleeve are given equal and opposite initial rotational velocities corresponding to $10000 \hbar$ (1054 maJ ps); no external torque is applied. Shaft-sleeve nonbonded interactions excite vibrational modes in each nanotube and lead to rotational friction. Drag coefficients are obtained from a linear fit to the angular momentum during the first 2 ps of simulations; these drag coefficients usually follow expected trends in velocity, temperature, and bearing clearance (difference between shaft and sleeve diameters).

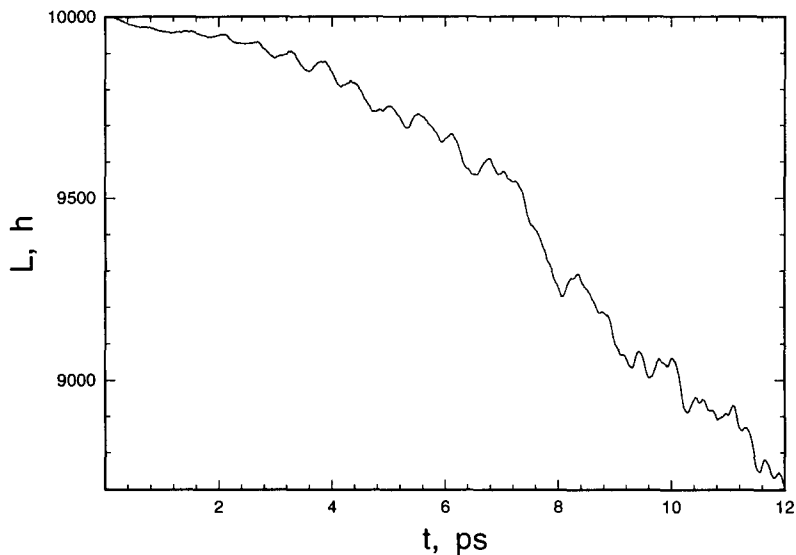


Figure 2. Rotational drag in a stable model graphite molecular bearing.

Results of one particularly interesting simulation are shown in Figure 3. This run is identical to Figure 2 except that the sleeve is shorter. From the shaft angular momentum in this figure, the first 15 ps of the simulation are similar to those of most of the other simulations in this study: slow, steady loss of angular momentum with fairly small oscillations. Between about 15 ps and about 60 ps, however, there are three beats with oscillations of several thousand \hbar (several hundred maJ ps). After 60 ps, the shaft once again steadily loses angular momentum; however, the oscillations are much larger than in the first 15 ps. At times, the shaft (and therefore, the sleeve, since in this case the total angular momentum is zero) briefly reverse direction. Computer

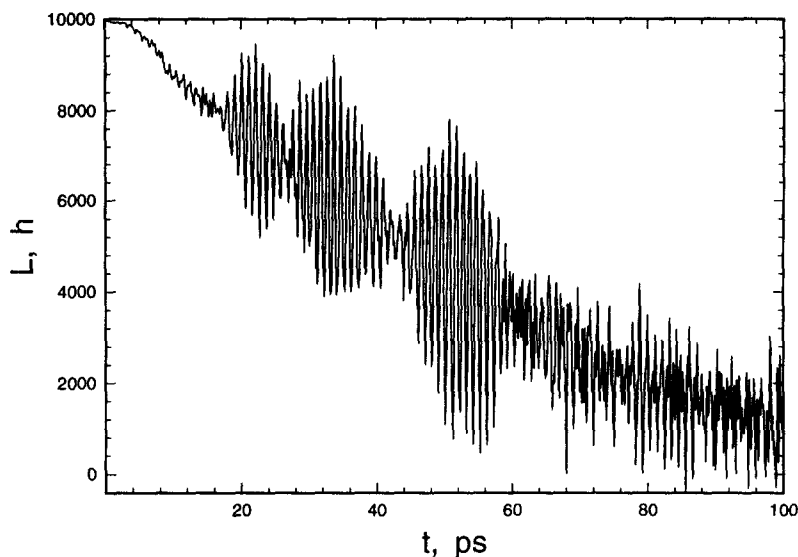


Figure 3. Rotational motion (angular momentum) beat patterns in an unstable model graphite molecular bearing.

animations and detailed analysis of the simulation data show that the beats occur because a shaft S mode resonates with a sleeve breathing mode, and so the shaft shimmies up and down the sleeve several times. The large magnitude of the shaft-sleeve interaction is related to a dynamic instability in the individual nanotubes, which in turn is controlled by the nanotube geometry (length/diameter ratio). Changing this ratio even by a small amount makes individual nanotubes more stable and the bearing performance more ideal.

While a few simulations are sufficient to study the most essential features of carbon nanotube bearings, designing a nanomachine using these bearings would require ensembles of simulations covering a large parameter space (nanotube sizes and temperatures, rotational velocities, and external load). Such a study would be an excellent candidate for the so-called master-worker calculation, where a master controls several workers, in this case a set of independent trajectories typically run one per processor. Since the trajectories are independent, there is no need for interprocessor communication.

Both memory and processor speed limit the size of systems that may be simulated on standard workstations. To perform a single simulation on a large nanomachine (millions or hundreds of millions of atoms) would today require a massively parallel computer. Although issues such as load balancing and data distribution/redistribution are well studied for regularly distributed data (such as solids or Lennard-Jones liquids), treating irregularly spaced data or complex interactions between chemically bonded atoms, especially different species, introduces additional complexities. For example, it is fairly easy to find a spatial decomposition for a regular polymer crystal for which the computations are optimally efficient. To simulate the laser ablation of a spot on the crystal would require allowing bonds to break. This would require not only complicated potential energy terms, which would slow down force calculations, but also an irregular and rapidly changing spatial decomposition in the vicinity of the laser beam. Similar problems could occur at liquid/solid phase boundaries or in nanomachines with large numbers of moving parts.

6. CONCLUSIONS

Programming an MD simulation on a massively parallel computer requires considering a number of issues related to data distribution across and communication between processors. Right now, much of this programming must be done by hand. The development of High Performance

Fortran and other automatically parallelizing compilers will simplify and speed code development and increase code portability across parallel platforms. To optimize a serial MD simulation code for use on a parallel machine may also entail changing details of the MD algorithm such as the order of symplectic integration; this issue merits further study.

The last decade has seen enormous improvements in both the speed and long-time stability of MD techniques. Our geometric statement function method greatly speeds up force calculations, the most expensive part of an MD simulation, and is eminently suitable for polymers and other systems with highly interconnected bond networks. Symplectic integration ensures long-time stability for integration. These techniques allow simulation of materials with complex bond networks and hundreds of thousands of atoms to be performed on standard workstations. The advent of parallel computing allows this limit to be extended into the billions of atoms, making possible the simulation of materials on the size scale of 0.1 micron and fully atomistic design and modeling of complex nanomachines.

REFERENCES

1. E.F. D'Azevedo, C.H. Romine and D.H. Walker, *SIAM News* **28**, (May/June 1995).
2. W.G. Hoover, Nonequilibrium molecular dynamics, *Annu. Rev. Phys. Chem.* **34**, 103, (1983).
3. M.L. Klein, Computer simulation studies of solids, *Annu. Rev. Phys. Chem.* **36**, 525, (1985).
4. K.E. Drexler, *Nanosystems*, (1992).
5. D.W. Noid, B.G. Sumpter, B. Wunderlich and G.A. Pfeffer, Molecular dynamics simulations of polymers: Methods for optimal Fortran programming, *J. Comput. Chem.* **11**, 236, (1990).
6. T.E. Tuzun, D.W. Noid and B.G. Sumpter, Efficient computation of potential energy first and second derivatives for molecular dynamics, normal coordinate analysis, and molecular mechanics calculations, *Macromol. Theory Simul.* **5**, 771, (1996).
7. S.K. Gray, D.W. Noid and B.G. Sumpter, Symplectic integrators for large scale molecular dynamics simulations: A comparison of several explicit methods, *J. Chem. Phys.* **101**, 4062, (1994).
8. H. Goldstein, *Classical Mechanics*, Addison-Wesley, (1980).
9. M.P. Allen and D. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, (1987).
10. B.G. Sumpter and D.W. Noid, The onset of instability in nanostructures: The role of nonlinear resonance, *J. Chem. Phys.* **102**, 6619, (1995).
11. R.E. Tuzun, D.W. Noid and B.G. Sumpter, The dynamics of molecular bearings, *Nanotechnology* **6**, 64–74, (1995).
12. R.E. Tuzun, D.W. Noid and B.G. Sumpter, Dynamics of a laser driven molecular motor, *Nanotechnology* **6**, 52–63, (1995).
13. R.E. Tuzun, D.W. Noid and B.G. Sumpter, Dynamics of fluid flow inside carbon nanotubes, *Nanotechnology*, (1995) (to appear).
14. R.E. Tuzun, D.W. Noid and B.G. Sumpter, Dynamics of He/C₆₀ flow inside carbon nanotubes, *Nanotechnology*, (1995) (to appear).