Symplectic Integration Methods in Molecular and Spin Dynamics Simulations

Shan-Ho Tsai[a,b], M. Krech[c], and D.P. Landau[a]

[a] Center for Simulational Physics, University of Georgia, Athens, GA 30602, USA

[b] Enterprise Information Technology Services, University of Georgia, Athens, GA 30602, USA

[c] Max-Planck-Institut für Metallforschung, Stuttgart 70569, Germany

Received on 2 September, 2003

We review recently developed decomposition algorithms for molecular dynamics and spin dynamics simulations of many-body systems. These methods are time reversible, symplectic, and the error in the total energy thus generated is bounded. In general, these techniques are accurate for much larger time steps than more standard integration methods. Illustrations of decomposition algorithms performance are shown for spin dynamics simulations of a Heisenberg ferromagnet.

1 Introduction

Molecular dynamics[1] and spin dynamics[2] simulations are powerful tools for investigating the time evolution of physical quantities and hence enhancing our understanding of dynamic properties of many-body systems. In these simulations the classical equations of motion governing the dynamical properties of the systems are solved numerically, with restrictions given by some initial conditions. Typically the time scale of the phenomena of interest is much longer than the time steps that can be used in standard finite time difference methods. Therefore, a large number of total integration steps is required, and this generates large truncation errors unless the time step used is very small. Small time steps often lead to forbiddingly long integrations.

Progress was accelerated with the advent of new, symplectic methods for integrating coupled equations of motion [3-6]. These numerical algorithms are based on decompositions of exponential operators. They are time reversible, symplectic (i.e they conserve exactly the invariant phase-space volume) and the error in the total energy of the system is bounded[7, 8]. (In some cases the total energy of the system is conserved exactly[9, 10].) The effectiveness and efficiency of these symplectic methods have been illustrated with spin dynamics simulations of the magnetic excitations in $RbMnF_3$ using a fourth-order Suzuki-Trotter decomposition method[6, 9]. The improved integration method has been used to obtain high-resolution results, which could, for the first time, be compared directly and quantitatively with neutron-scattering data, yielding good agreement and shedding light onto controversies between theory and experiment[11].

In this paper we review decomposition algorithms applied to classical molecular dynamics and spin dynamics simulations. In Section II we express the classical equations of motion used in molecular dynamics in the Liouville formulation and in Section III we introduce spin dynamics simulations. We discuss criteria for good integration methods in Section IV and we briefly review a few standard integration algorithms in Section V. In Section VI we describe the new decomposition algorithms and show some numerical tests and in Section VII we present spin dynamics results for $RbMnF_3$. Finally, a summary is provided in Section VIII.

2 Molecular Dynamics

Let us consider a system of N particles with masses m_i described by their positions $\mathbf{r}_i$ and velocities $\mathbf{v}_i$, interacting via a potential $u(r_{ij})$, where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$. The Hamiltonian function of the system can be written as

$$\mathcal{H} = \sum_{i=1}^{N} \frac{1}{2} m_i v_i^2 + \sum_{i,j,\, j\neq i} u(r_{ij}), \tag{1}$$

and the force on particle i due to particle j is given by

$$\mathbf{f}_{ij} = -\nabla_{\mathbf{r}_i} u(r_{ij}) = -\frac{\partial u(r_{ij})}{\partial r_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}}. \tag{2}$$

The equations of motion are given by

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \sum_{j,\, j\neq i} \mathbf{f}_{ij} \equiv \mathbf{f}_i, \quad i = 1, .., N. \tag{3}$$

The time evolution of the system can be studied by integrating the equations of motion to obtain $\mathbf{r}_i(t)$ and $\mathbf{v}_i(t)$, for $i = 1, 2, \cdots, N$, and by expressing other physical quantities in terms of $\mathbf{r}_i(t)$ and $\mathbf{v}_i(t)$.

2.1 Liouville Formulation

The equations of motion (3) can be rewritten as

$$\frac{dy}{dt} = \hat{L} y(t) \tag{4}$$

where $y(t) = \{\mathbf{r}_i(t), \mathbf{v}_i(t)\}$ denotes a configuration of the N particles, and $\hat{L}$ is the Liouville operator defined as

$$\hat{L} \equiv \sum_{i=1}^{N} \left(\mathbf{v}_i \cdot \frac{\partial}{\partial \mathbf{r}_i} + \frac{\mathbf{f}_i}{m_i} \cdot \frac{\partial}{\partial \mathbf{v}_i} \right) \equiv A + B \qquad (5)$$

The term $\sum_{i=1}^{N} \mathbf{v}_i \cdot \frac{\partial}{\partial \mathbf{r}_i} \equiv A$ in Eq.(5) corresponds to the free motion of the particles (kinetic part), whereas the potential part is given by the term $\sum_{i=1}^{N} \frac{\mathbf{f}_i}{m_i} \cdot \frac{\partial}{\partial \mathbf{v}_i} \equiv B$. With these definitions of operators A and B, the equations of motion (4) can be written as

$$\frac{dy}{dt} = (A + B)y(t), \qquad (6)$$

which have the formal solution

$$y(t + \Delta) = \mathrm{e}^{(A+B)\Delta} y(t), \qquad (7)$$

where Δ represents a time step. For a general many-body system the combined operation $\mathrm{e}^{(A+B)\Delta} y(t)$ cannot be easily performed. However, the separate operators $\mathrm{e}^{A\Delta} y$ and $\mathrm{e}^{B\Delta} y$ can be written as

$$\mathrm{e}^{A\Delta} y = \exp\left(\Delta \sum_{i=1}^{N} \mathbf{v}_i \cdot \frac{\partial}{\partial \mathbf{r}_i} \right) \{\mathbf{r}_i, \mathbf{v}_i\} = \{\mathbf{r}_i + \mathbf{v}_i \, \Delta, \mathbf{v}_i\} \qquad (8)$$

$$\mathrm{e}^{B\Delta} y = \exp\left(\Delta \sum_{i=1}^{N} \frac{\mathbf{f}_i}{m_i} \cdot \frac{\partial}{\partial \mathbf{v}_i} \right) \{\mathbf{r}_i, \mathbf{v}_i\} = \{\mathbf{r}_i, \mathbf{v}_i + \frac{\mathbf{f}_i}{m_i} \, \Delta\} \qquad (9)$$

and they represent shifts in the positions and in the velocities, respectively. Moreover, the shift in the positions (velocities) generated by $\mathrm{e}^{A\Delta} y$ ($\mathrm{e}^{B\Delta} y$) only depends on the velocities (positions) and can be easily computed. However, note that in general $\mathrm{e}^{(A+B)\Delta} \neq \mathrm{e}^{A\Delta} \mathrm{e}^{B\Delta}$!

3 Spin Dynamics

For simplicity, let us discuss spin dynamics simulations for a specific spin model, namely the classical isotropic Heisenberg model, described by the Hamiltonian

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j \qquad (10)$$

where $\mathbf{S}_i$ is a unit vector located on a lattice site i, and nearest-neighbor pairs of spins are coupled with an interaction parameter J, which can be ferromagnetic ($J > 0$) or antiferromagnetic ($J < 0$). One of the best physical realizations of this model is $RbMnF_3$, where the magnetic ions Mn^{+2} have spin $S = 5/2$, and are located on the sites of a simple cubic lattice. Nearest-neighbor interactions are antiferromagnetic, with $J^{\mathrm{exp}} = -(0.58 \pm 0.06)\mathrm{meV}$. Magnetic ordering with antiferromagnetic alignment of spins occurs below the critical temperature $T_c = 83K$. For the discussions in this section we will consider simple cubic lattices with periodic boundary conditions.

Unlike the Ising model ($S_i = \pm 1$), Heisenberg models have true dynamics governed by the equations of motion

$$\frac{d\mathbf{S}_i}{dt} = -\mathbf{S}_i \times \frac{\partial \mathcal{H}}{\partial \mathbf{S}_i} \qquad (11)$$

which can be rewritten as

$$\frac{d\mathbf{S}_i}{dt} = -\mathbf{S}_i \times \mathbf{H}_{\mathrm{eff}\,i} \qquad (12)$$

where the effective field $\mathbf{H}_{\mathrm{eff}\,i}$ is defined as

$$H^k_{\mathrm{eff}\,i} = -J \sum_{j=NN(i)} S^k_j, \qquad k = x, y, z \qquad (13)$$

with the sum performed over all nearest-neighbor sites of i. If we denote

$$\mathbf{S}_i = \begin{pmatrix} S^x_i \\ S^y_i \\ S^z_i \end{pmatrix} \qquad (14)$$

we can then write the equations of motion as

$$\frac{d\mathbf{S}_i}{dt} = \begin{bmatrix} 0 & -H^z_{\mathrm{eff}\,i} & H^y_{\mathrm{eff}\,i} \\ H^z_{\mathrm{eff}\,i} & 0 & -H^x_{\mathrm{eff}\,i} \\ -H^y_{\mathrm{eff}\,i} & H^x_{\mathrm{eff}\,i} & 0 \end{bmatrix} \mathbf{S}_i \equiv R\mathbf{S}_i \qquad (15)$$

for which the formal solution is

$$\mathbf{S}_i(t + \Delta) = \mathrm{e}^{R\Delta} \mathbf{S}_i(t), \qquad (16)$$

where Δ represents a time step.

Spin dynamics simulations can be used to study several dynamic properties of classical spin systems[2]. Of particular interest are the dynamic structure factors, which are Fourier transforms of space- and time-displaced spin-spin correlation functions given by

$$C^k(\mathbf{r} - \mathbf{r}', t) = \langle S^k_{\mathbf{r}}(t) S^k_{\mathbf{r}'}(0) \rangle - \langle S^k_{\mathbf{r}}(t) \rangle \langle S^k_{\mathbf{r}'}(0) \rangle \qquad (17)$$

where $k = x, y, z$.

The simple cubic lattice can be divided into two interpenetrating sublattices denoted here as sublattices $\mathcal{A}$ and $\mathcal{B}$. Let us denote as y_A and y_B the spin configurations on sublattices $\mathcal{A}$ and $\mathcal{B}$, respectively. In this notation, the formal solution of the equations of motion can be written as $y(t + \Delta) = \mathrm{e}^{(A+B)\Delta} y(t)$, where $y = \{y_A, y_B\}$. The operator $\mathrm{e}^{A\Delta}$ rotates y_A by an angle $|H_{\mathrm{eff}_A}|\Delta$ at fixed y_B, and, similarly, the operator $\mathrm{e}^{B\Delta}$ rotates y_B by an angle $|H_{\mathrm{eff}_B}|\Delta$ at fixed y_A. Here H_{eff_A} and H_{eff_B} denote the effective field acting on the spins of sublattices $\mathcal{A}$ and $\mathcal{B}$, generated by the spins on the other sublattice, namely sublattices $\mathcal{B}$ and $\mathcal{A}$, respectively. Because these are spin rotation operators, the scalar products of spins are preserved in each of these operations; therefore the spin length and the energy are conserved exactly (within machine precision) for this system. These separate operators $\mathrm{e}^{A\Delta} y$ and $\mathrm{e}^{B\Delta} y$ also have a simple explicit form[9].

In order to obtain the dynamic properties of the spin model at fixed temperature T, rather than at fixed energy, we use equilibrium configurations obtained from Monte Carlo simulations[12] of the model at a given T as initial configurations for the integration of the equations of motion. Solutions for different initial configurations are then averaged to yield results in the Canonical Ensemble.

As mentioned earlier molecular dynamics simulations determine the time evolution of particle positions and velocities, and a system configuration is denoted as $y(t) = \{\mathbf{r}_i(t), \mathbf{v}_i(t)\}$. In comparison, spin dynamics simulations determine temporal evolution of the spin orientations, and a system configuration is denoted as $y(t) = \{y_A(t), y_B(t)\}$. In both cases the equations of motion can be written as $\frac{dy}{dt} = (A + B)y(t)$, for which the formal solution is

$$y(t + \Delta) = \mathrm{e}^{(A+B)\Delta} y(t), \tag{18}$$

where Δ represents a time step.

4 Criteria for a good integration algorithm

Given the limited computer resources and the interest in long time evolutions of the equations of motion, the overall speed of the integration algorithm is very important. Because each integration step in general involves force (MD) or spin derivative (SD) computations, which are very time consuming, it is desirable that an integration algorithm be accurate for large time steps thus reducing the total number of force or spin derivative recalculations per unit time. The speed of a single integration step itself is not such an important factor because more complex and hence slower integration steps may allow the usage of much larger time steps, generating a faster algorithm without larger truncation errors.

Another criterion for a good integration method is that it reproduces conservation laws and properties of the classical equations of motion. Of particular importance is that it conserves energy and the phase-space volume, and that it be time reversible. These properties are closely related to the stability of the algorithm and its accuracy for large time steps.

5 Standard integration methods

Ordinary differential equations, such as the equations of motion in MD and SD, are often solved numerically using finite difference methods[13]. The procedure is to use the variables such as positions in MD and spins in SD and their time derivatives at time t to compute the values of these quantities at a later time $t + \Delta$. The accuracy of this procedure is often proportional to a power of Δ. An integration method is referred to as an n-th order algorithm when the local (per time step) truncation error is of $\mathcal{O}(\Delta^{n+1})$. Here we briefly review some of the most commonly used finite difference methods. To simplify the notation, we will omit the particle label in the variables.

5.1 Truncated Taylor expansion

The simplest integration method is to write Taylor expansions such as

$$r(t + \Delta) = r(t) + v(t)\Delta + \frac{f(t)}{2m}\Delta^2 + ... \quad (MD) \tag{19}$$

$$S(t + \Delta) = S(t) + \frac{dS}{dt}\Delta + \frac{1}{2}\frac{d^2S}{dt^2}\Delta^2 + ... \quad (SD) \tag{20}$$

and then truncate them to $\mathcal{O}(\Delta^3)$. However, this algorithm is not time reversible, it does not conserve the phase-space volume and it gives rise to very large energy drift.

A better implementation based on Taylor expansions that avoids the large energy drift is the very popular Verlet algorithm. For brevity we will discuss this algorithm in the next two sections referring to molecular dynamics only. All equations can be easily converted to spin dynamics variables.

5.2 Position-Verlet algorithm

Let us consider now the Taylor expansions for the positions at times $t + \Delta$ and $t - \Delta$, which are written as

$$r(t + \Delta) = r(t) + v(t)\Delta + \frac{f(t)}{2m}\Delta^2 + \frac{d^3r}{dt^3}\frac{\Delta^3}{3!} + \mathcal{O}(\Delta^4) \tag{21}$$

$$r(t - \Delta) = r(t) - v(t)\Delta + \frac{f(t)}{2m}\Delta^2 - \frac{d^3r}{dt^3}\frac{\Delta^3}{3!} + \mathcal{O}(\Delta^4) \tag{22}$$

Adding Eqs.(21) and (22) we have

$$r(t + \Delta) = 2r(t) - r(t - \Delta) + \frac{f(t)}{m}\Delta^2 + \mathcal{O}(\Delta^4), \tag{23}$$

which is then used to obtain the time evolution of $r(t)$. Note that the computation of positions at a later time does not use any velocities, but the velocities will be needed for computing the kinetic energy and thus the total energy of the system. Subtracting Eq.(22) from Eq.(21) we have

$$r(t + \Delta) - r(t - \Delta) = 2v(t)\Delta + \mathcal{O}(\Delta^3) \tag{24}$$

and the time evolution of velocities can be determined by

$$v(t) = \frac{r(t + \Delta) - r(t - \Delta)}{2\Delta} + \mathcal{O}(\Delta^2) \tag{25}$$

The Verlet algorithm is a second-order method that has been widely used in molecular dynamics simulations[1]. It is time reversible, it preserves phase-space volume and, although the total energy is not conserved, the long-term energy drift is not very large provided a small time step is used. Since this is a second-order method, it is not very accurate for long time steps.

5.3 Velocity-Verlet algorithm

Another implementation of the Verlet algorithm, denoted as velocity-Verlet algorithm, computes the time evolution of the position and velocity with

$$r(t+\Delta) = r(t) + v(t)\Delta + \frac{f(t)}{2m}\Delta^2 \qquad (26)$$

and

$$v(t+\Delta) = v(t) + \frac{f(t+\Delta)+f(t)}{2m}\Delta, \qquad (27)$$

respectively. This corresponds to first computing $r(t+\Delta)$ using Eq.(26), then from these new positions the forces $f(t+\Delta)$ can be determined. Finally, the velocities $v(t+\Delta)$ are computed from Eq.(27).

To show the equivalence between the position- and the velocity-Verlet algorithms, we write Eq.(26) for time $t+2\Delta$, namely

$$r(t+2\Delta) = r(t+\Delta)+v(t+\Delta)\Delta+\frac{f(t+\Delta)}{2m}\Delta^2\ , \quad (28)$$

and we then subtract Eq.(26) from Eq.(28) to obtain

$$\begin{aligned} r(t+2\Delta) \quad &= 2r(t+\Delta) - r(t) + [v(t+\Delta) - v(t)]\Delta \\ &\quad + \tfrac{f(t+\Delta)-f(t)}{2m}\Delta^2 \end{aligned} \qquad (29)$$

Substituting Eq.(27) in Eq.(29), we get

$$r(t+2\Delta) = 2r(t+\Delta) - r(t) + \frac{f(t+\Delta)}{m}\Delta^2 \qquad (30)$$

which is the position-Verlet algorithm derived before.

5.4 Predictor-Corrector method

Predictor-corrector algorithms are very popular and versatile higher-order methods. Before introducing one such method, let us write the equations of motion in the general form $\frac{dy}{dt} = F(y)$. One common implementation of a predictor-corrector method is a fourth-order algorithm that uses the explicit Adams-Bashforth four-step method given by

$$\begin{aligned} y(t+\Delta) \quad &= y(t) + \tfrac{\Delta}{24}[55F(y(t)) - 59F(y(t-\Delta)) \\ &\quad +37F(y(t-2\Delta)) - 9F(y(t-3\Delta))] \end{aligned} \qquad (31)$$

as the predictor step and the implicit Adams-Moulton three-step method given by

$$\begin{aligned} y(t+\Delta) \quad &= y(t) + \tfrac{\Delta}{24}[9F(y(t+\Delta)) + 19F(y(t)) \\ &\quad -5F(y(t-\Delta)) + F(y(t-2\Delta))] \end{aligned} \qquad (32)$$

as the corrector step. Both the predictor and the corrector steps have a local truncation error of $\mathcal{O}(\Delta^5)$. The values of $y(t)$ for the initial three time steps, namely $y(\Delta)$, $y(2\Delta)$, and $y(3\Delta)$, can be provided by three successive integrations of the equation of motion by other methods.

One great advantage of this method is that it is easy to apply for a general set of equations. However, it is in general not time reversible, it does not preserve the phase-space volume and it yields large energy drifts unless very small time steps are used.

6 Decomposition algorithms

A different finite time difference approach to determine the temporal evolution described by the equations of motion is to expand the exponential operator in Eq.(18) as follows[3, 4, 5, 6]

$$\mathrm{e}^{(A+B)\Delta} = \prod_{j=1}^{P} \mathrm{e}^{Aa_j\Delta}\mathrm{e}^{Bb_j\Delta} + \mathcal{O}(\Delta^{K+1}) \qquad (33)$$

where a_j and b_j are chosen to provide the highest $K \geq 1$ for a given $P \geq 1$.

The simplest approximations are the lowest-order decompositions, namely

$$\mathrm{e}^{(A+B)\Delta} = \mathrm{e}^{A\Delta}\mathrm{e}^{B\Delta} + \mathcal{O}(\Delta^2) \qquad (34)$$

to first order and

$$\mathrm{e}^{(A+B)\Delta} = \mathrm{e}^{B\frac{\Delta}{2}}\mathrm{e}^{A\Delta}\mathrm{e}^{B\frac{\Delta}{2}} + \mathcal{O}(\Delta^3) \qquad (35)$$

to second order. Eq.(35) is equivalent to the Suzuki-Trotter decomposition used in Quantum Monte Carlo simulations[14, 15] in which A and B represent two non-commuting parts of the Hamiltonian. Note that Eq.(35) is also equivalent to the velocity-Verlet algorithm discussed above, and the position-Verlet algorithm is equivalent to using Eq.(35) with A and B interchanged[16, 8].

Higher-order integration algorithms can be implemented using higher-order decompositions, such as the fourth-order Suzuki-Trotter decomposition[6] given by

$$\mathrm{e}^{(A+B)\Delta} = \prod_{i=1}^{5} \mathrm{e}^{p_i A\Delta/2}\mathrm{e}^{p_i B\Delta}\mathrm{e}^{p_i A\Delta/2} + \mathcal{O}(\Delta^5) \qquad (36)$$

where $p_1 = p_2 = p_4 = p_5 \equiv p = \frac{1}{4-4^{\frac{1}{3}}}$, $p_3 = 1-4p$. Eq.(36) shows that the fourth-order Suzuki-Trotter decomposition is composed by a product of 15 exponential operators; however, consecutive operators with A in the exponent can be combined into a single operation, yielding a total of 11 operators for this decomposition. Several other fourth-order and higher-order decompositions of the exponential operators have been derived in the literature [6, 3, 5, 4, 17, 8]. For example, the fourth-order Forest-Ruth decomposition is comprised of 7 operators, and it can be written as

$$\begin{aligned} \mathrm{e}^{(A+B)\Delta} \quad &= \mathrm{e}^{A\theta\Delta/2}\mathrm{e}^{B\theta\Delta}\mathrm{e}^{A(1-\theta)\Delta/2}\mathrm{e}^{B(1-2\theta)\Delta} \\ &\quad \times\mathrm{e}^{A(1-\theta)\Delta/2}\mathrm{e}^{B\theta\Delta}\mathrm{e}^{A\theta\Delta/2} + \mathcal{O}(\Delta^5) \end{aligned} \qquad (37)$$

where $\theta = 1/(2-2^{1/3}) \approx 1.3512$. Because this decomposition involves only 7 single exponential operators instead of 11 as in the case of Eq.(36), the cpu time required for one integration step using Eq.(37) is less than using Eq.(36). However, truncation errors arising from using Eq.(37) are much

larger than when Eq.(36) is used, as will be illustrated below. Recently, Omelyan and collaborators[17] have obtained optimized decomposition algorithms in which new operators are introduced and the parameters in the exponents of these new operators are chosen to minimize the truncation error. An optimized Forest-Ruth decomposition is given by

$$\mathrm{e}^{(A+B)\Delta} = \mathrm{e}^{B\zeta\Delta}\mathrm{e}^{A(1-2\lambda)\Delta/2}\mathrm{e}^{B\chi\Delta}\mathrm{e}^{A\lambda\Delta}\mathrm{e}^{B(1-2(\chi+\zeta))\Delta} \times\mathrm{e}^{A\lambda\Delta}\mathrm{e}^{B\chi\Delta}\mathrm{e}^{A(1-2\lambda)\Delta/2}\mathrm{e}^{B\zeta\Delta} + \mathcal{O}(\Delta^5) \quad (38)$$

where[17] $\zeta = 0.17208656$, $\lambda = -0.09156203$, and $\chi = -0.16162176$.

6.1 Properties of the decomposition algorithms

The decomposition algorithms described above preserve the phase space volume element in molecular and spin dynamics simulations. In the molecular dynamics case the operator $\mathrm{e}^{A\,\Delta}$ ($\mathrm{e}^{B\,\Delta}$) acting on y only shifts $\mathbf{r}_i$ ($\mathbf{v}_i$) and the shift only depends on the $\mathbf{v}_i$ ($\mathbf{r}_i$), as shown in Eqs.(8) and (9). In the spin dynamics case the operator $\mathrm{e}^{A\,\Delta}$ ($\mathrm{e}^{B\,\Delta}$) acting on y rotates y_A (y_B) at fixed y_B (y_A). Therefore, in both cases the Jacobian of the transformation from $y(t)$ to $y(t+\Delta)$ is equal to one, hence the preservation of the phase-space volume.

The condition of being time reversible requires that only even-ordered decompositions be used and that the operators $\mathrm{e}^{A\,\Delta}$ and $\mathrm{e}^{B\,\Delta}$ enter symmetrically in the decomposition. As an example, we show the time reversibility property of the second-order decomposition given in Eq.(35). We first define

$$\hat{U}(\Delta) \equiv \mathrm{e}^{B\Delta/2}\mathrm{e}^{A\Delta}\mathrm{e}^{B\Delta/2} \quad (39)$$

and obtain

$$\hat{U}(\Delta)\hat{U}(-\Delta) = \mathrm{e}^{B\frac{\Delta}{2}}\mathrm{e}^{A\Delta}\mathrm{e}^{B\frac{\Delta}{2}}\mathrm{e}^{-B\frac{\Delta}{2}}\mathrm{e}^{-A\Delta}\mathrm{e}^{-B\frac{\Delta}{2}} = 1 \quad (40)$$

Similarly, $\hat{U}(-\Delta)\hat{U}(\Delta) = 1$; hence each time step in the temporal evolution is reversible, leading to a time reversible trajectory. This proof can be easily extended to higher-order decompositions.

Although the decomposition algorithm does not conserve the total energy of a general system, the long-term energy deviation as function of time is characterized by a *random walk*, i.e., it does *not* display a systematic drift in any direction. This is due to the time reversibility property shown in Eq.(40) and applies to other non-conserved quantities as well as we will illustrate below. In addition, the higher-order integration methods are accurate for larger time steps, allowing time evolutions to longer times without generating large truncation errors.

In general, higher-order decomposition algorithms require more operations, and hence more cpu time, per integration step. However, they are accurate for larger time steps, and they are more efficient methods if the increase in time steps compensate for the slower integration step. Higher-order decompositions are also more advantageous if the physical system studied requires that the conservation laws be obeyed more strictly.

6.2 Algorithm performance

The performance of some of the algorithms discussed here will be illustrated with spin dynamics simulations of a Heisenberg ferromagnet on a 10x10x10 simple cubic lattice at temperature $T = 0.8T_c$, where T_c is the critical temperature of the model. The equations of motion conserve both the total energy per site $E(t)$ and the uniform magnetization per site $M(t)$ of the system. The fourth-order predictor-corrector method described above conserves the uniform magnetization exactly; however, the total energy drifts systematically and considerably, even for relatively small time steps, as shown in Fig. 1.

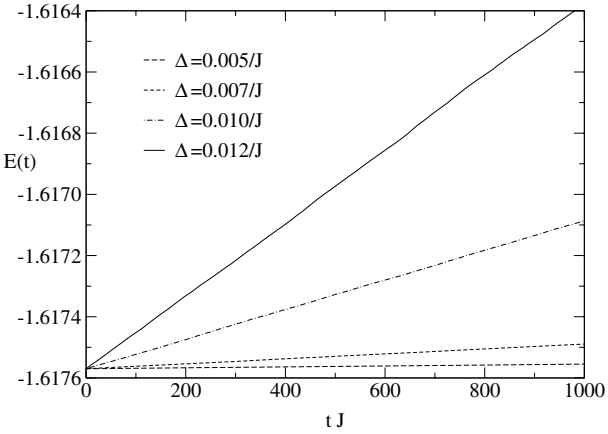

Figure 1. Energy per site versus time obtained with the fourth-order predictor-corrector method for a single initial configuration using different time steps Δ.

In contrast the decomposition algorithms conserve both energy and spin length exactly, because the scalar product of nearest-neighbor spins is preserved during the rotation of a spin around its effective field. The uniform magnetization is not exactly conserved by the decomposition algorithms; however, there is no long time drift in the magnetization, as illustrated in Fig. 2 for the fourth- and an eighth-order[6, 18] Suzuki-Trotter decomposition with $\Delta = 0.10/J$. Note that the size of the integration step used here is almost an order of magnitude larger than the maximum Δ in Fig. 1. Moreover, the maximum integration time here is $t_{max} = 5000/J$, which is a factor of 5 larger than in Fig. 1. Fig. 2 also shows that for the same time step, higher-order methods yield smaller magnetization fluctuations. The total fluctuation of the uniform magnetization per site $M(t)$ for the fourth- and the eighth-order method shown in Fig. 2 are $\sim 2 \times 10^{-5}$ and $\sim 2 \times 10^{-7}$, respectively. Fig. 3 shows the fluctuation in the uniform magnetization from integrations using an eighth-order Suzuki-Trotter decomposition with different time steps. In this figure, the magnetization fluctuations for $\Delta = 0.10/J$, $0.20/J$, and $0.25/J$ are $\sim 2 \times 10^{-7}$, $\sim 2 \times 10^{-5}$, and $\sim 1 \times 10^{-4}$, respectively. In Fig. 4 we compare the fluctuations in the uniform magnetization per site obtained with three different fourth-order decomposition methods, namely the Suzuki-Trotter (SZT), the Forest-Ruth (FR), and the optimized Forest-Ruth (OFR) decompositions given in Eqs.(36), (37), and (38), respectively. In all

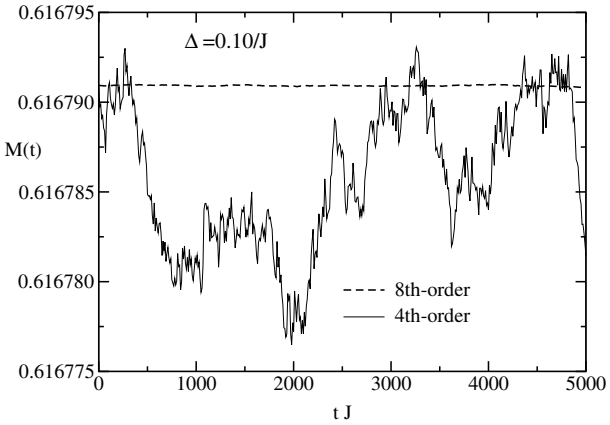

Figure 2. Comparison of fluctuations in the uniform magnetization per site for the fourth- and an eighth-order Suzuki-Trotter decomposition with $\Delta = 0.10/J$ for a single initial configuration.

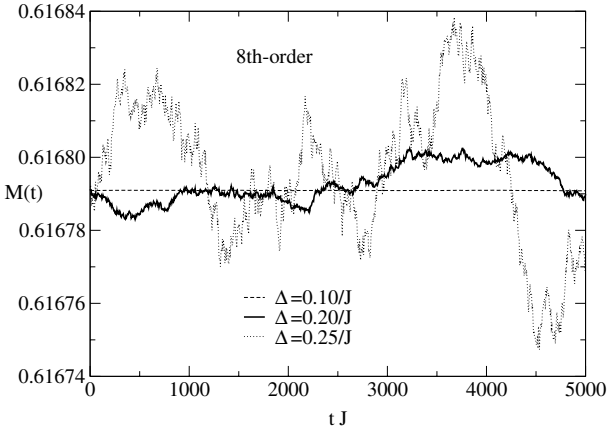

Figure 3. Comparison of fluctuations in the uniform magnetization per site for an eighth-order Suzuki-Trotter decomposition with different time steps Δ for a single initial configuration.

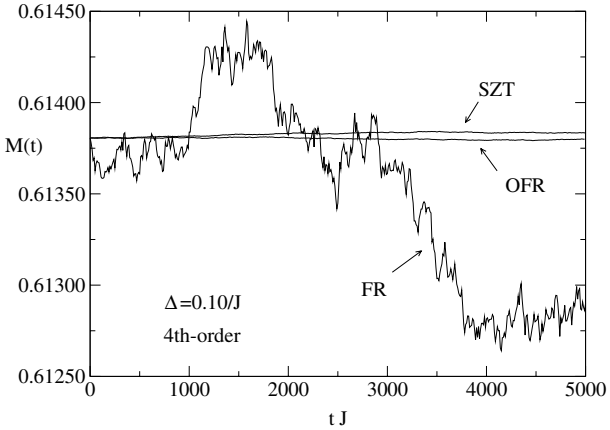

Figure 4. Comparison of fluctuations in the uniform magnetization per site for three different 4th-order decomposition algorithms, namely Forest-Ruth (FR), Suzuki-Trotter (SZT) and an optimized Forest-Ruth (OFR) [see text], as a function of the integration time. A time step of $\Delta = 0.10/J$ have been used in all cases.

three cases the equations of motion were integrated out to $t_{max} = 5000/J$, using a step of $\Delta = 0.10/J$. The FR method requires the fewer operations (rotations) per time step; however, it also yields the largest magnetization fluctuation ($\sim 2 \times 10^{-3}$). In contrast, much smaller fluctuations were observed with the SZT and the OFR methods ($\sim 2 \times 10^{-5}$ in both cases).

Each integration step of the predictor-corrector method used here is approximately 2.5 times faster than each step using the fourth-order Suzuki-Trotter decomposition. However, the latter generates results that are accurate for much larger time steps, and thus constitutes a much faster algorithm. Although the eighth-order method used here provides better magnetization conservation, it is not a very competitive algorithm because it requires a large number, namely 31, rotations per time step.

6.3 Further developments

Decompositions of exponential operators involving higher-order derivatives of the variables in the equations of motion, such as force gradients in the case of MD, have been implemented and shown to be more advantageous for some applications[19, 20, 21]. One such force-gradient decomposition is given by[21]

$$\mathrm{e}^{(A+B)\Delta} = \prod_{i=1}^{P} \mathrm{e}^{a_i A \Delta} \mathrm{e}^{b_i B \Delta + c_i C \Delta^3} + \mathcal{O}(\Delta^{K+1}) \quad (41)$$

where

$$C \equiv [B,[A,B]] = \sum_{i=1}^{N} \frac{\mathbf{g}_i}{m_i} \cdot \frac{\partial}{\partial \mathbf{v}_i}, \quad (42)$$

$$\mathbf{g}_i = 2 \sum_{k,k\neq i} \sum_{jp,j\neq p} \frac{\mathbf{f}_{jp}}{m_j} \frac{\partial \mathbf{f}_{ik}}{\partial \mathbf{r}_{jp}}, \quad (43)$$

and a_i, b_i, and c_i are chosen to minimize the truncation errors.

For systems involving different time scales, decomposition methods can be used to integrate the slow varying components of the system with a larger time step than the rapidly varying components[22, 23]. As a simple example, let us consider an MD simulation where the Liouville operator can be separated into a slow and a rapidly varying part denoted as L_s and L_f, respectively. The second-order decomposition given by Eq.(35) can be further decomposed into

$$\mathrm{e}^{(L_f+L_s)\Delta} = \mathrm{e}^{L_s \frac{\Delta}{2}} [\mathrm{e}^{L_f \delta}]^n \mathrm{e}^{L_s \frac{\Delta}{2}} + \mathcal{O}(\Delta^3) \quad (44)$$

where $\delta = \Delta/n$ is a smaller time step used to evolve the fast dynamics of the system.

Depending on the dynamics and types of interactions in the systems, it may be necessary to decompose the exponential operator into more than two individual operators. For example, spin dynamics simulations of a spin system on a two-dimensional triangular lattice require a three-sublattice decomposition, and a second-order decomposition can be written as

$$\mathrm{e}^{(A+B+C)\Delta} = \mathrm{e}^{A\frac{\Delta}{2}} \mathrm{e}^{B\frac{\Delta}{2}} \mathrm{e}^{C\Delta} \mathrm{e}^{B\frac{\Delta}{2}} \mathrm{e}^{A\frac{\Delta}{2}} + \mathcal{O}(\Delta^3), \quad (45)$$

where each of the three separate operators $\mathrm{e}^{A\Delta}$, $\mathrm{e}^{B\Delta}$, and $\mathrm{e}^{C\Delta}$ rotates the spins on one sublattice with the spins

on the other two sublattices fixed. Spin dynamics simulations of an antiferromagnetic XY model on the triangular lattice have been done using a second-order decomposition algorithm[24]. The dynamic behavior of the model was studied for a range of temperatures, including around the Kosterlitz-Thouless transition and the Ising transition, where long-range order appears in the staggered chirality[24]. There are several other applications studied in the literature that require decompositions involving multiple operators[25].

Finally, we remark that the sublattice decomposition required for the implementation of decomposition algorithms allows a direct parallelization according to the shared memory model (OpenMP) which results in essentially 100% parallel code for, e.g., a spin dynamics integrator. This is particularly interesting for many commercially available parallel clusters in which each node is often equipped with two processors on the main board sharing the installed main memory. However, in practice one observes a severe reduction in parallel efficiency on many commercially available systems, e.g., the Intel Xeon, if the storage required to hold the lattice or the numerical grid of the problem exceeds the cache size of the processor. The reason for this is a lack of memory bandwidth in particular if CPU1 has to access the memory share of CPU2. In this case an additional chip set is invoked which performs a time consuming memory mapping operation. We had the opportunity to make a few tests on an AMD Opteron node with two CPUs and the new Hypertransport architecture which makes one CPU essentially transparent for access to its memory share requested by the other CPU. It turns out that the parallel efficiency on this system remains on a high level independent of the lattice size which makes shared memory parallelism an efficient and easy-to-use tool to increase the turnaround also for simulations on memory consuming lattices in the future [26].

7 Spin dynamics results for $RbMnF_3$

Spin dynamics of $RbMnF_3$ have been performed on simple cubic lattices with linear sizes up to $L = 72$; this corresponds to solving a system of $72^3 = 373248$ equations[27]. Direct comparisons of dynamic structure factors $S(q,\omega)$ for momentum q and frequency ω obtained from spin dynamics simulations and neutron scattering data[28] yielded good quantitative agreement, with no adjustable parameters[11]. An illustration of this comparison at $T = 0.894T_c$ for momentum transfer $\mathbf{q}$ in the [111] direction is shown in Fig. 5.

Integrations of the equations of motion were done up to $t_{max} = 1000/J$ using the fourth-order Suzuki-Trotter decomposition given in Eq.(36) with a time step of $\Delta = 0.2/J$. The experimental energy resolution width was 0.25 meV, which is shown as a horizontal line segment in Fig. 5. For the direct comparison, dynamic structure factors from the simulations were convoluted with the experimental resolution function and the T- and ω-dependent population factor was removed from the neutron scattering data. The normalization of the intensities of $S(q,\omega)$ between simulation and experiment was done at one T and q, the same factor was then used to normalize the curves for all values of q.

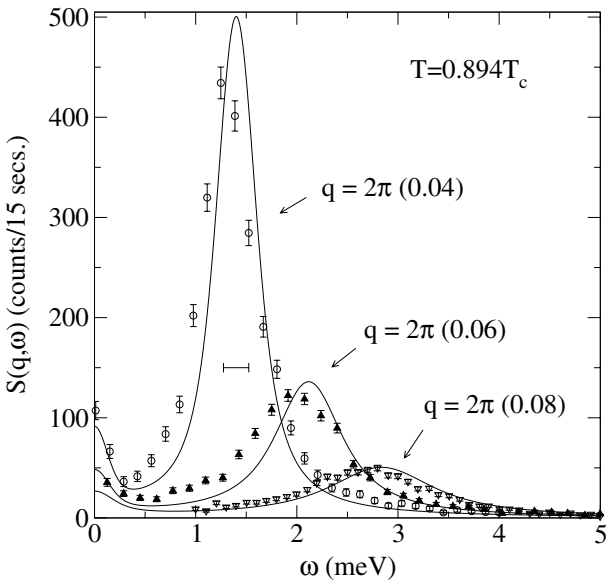

Figure 5. Comparison of dynamic structure factor as a function of frequency from simulation and experiment, for $T = 0.894T_c$ and $\mathbf{q} = (q,q,q)$. The symbols represent neutron scattering data [the circles, triangles, and inverted triangles correspond to $q = 2\pi(0.04)$, $2\pi(0.06)$, and $2\pi(0.08)$, respectively] while the solid lines are simulation results for $L = 60$.

8 Summary

Molecular dynamics and spin dynamics simulations require good algorithms for the time integration of the equations of motion. Desirable properties of integration algorithms include accuracy for long time steps, time reversibility, good conservation of energy, and being symplectic (conserve phase-space volume).

Standard integration algorithms in applied mathematics are, in general, neither time reversible nor symplectic, and they yield large long-term energy drift, unless very small time steps are used. In contrast, algorithms based on decomposition of exponential operators are time reversible, symplectic, and the energy fluctuations are bounded. In some cases the energy can be conserved exactly (within machine precision). In general, decomposition algorithms are also accurate for larger time steps and allow integration to much longer times thus allowing study of low frequency modes. These methods are broadly applicable and may be straightforwardly applied to more complicated systems although more sublattices may be needed with a resultant increase in complexity.

Acknowledgments

We are indebted to H.-K. Lee, K. Nho, and A. Bunker for very fruitful discussions. We also thank R. Coldea and R.A. Cowley for very helpful discussions and for sending us their data. This work is partially supported by NSF grants number DMR-0094422 and number DMR-0307082.

References

[1] See e.g. M.P. Allen and D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press 1987; D. Frenkel and B. Smit, *Understanding Molecular Simulation*, Academic Press 1996.

[2] D.P. Landau and M. Krech, J. Phys.: Condens. Matter **11**, R179 (1999).

[3] H. Yoshida, Phys. Lett. A **150**, 262 (1990)

[4] E. Forest and R.D. Ruth, Phys. D **43**, 105 (1990)

[5] M. Suzuki, Phys. Lett. A **165**, 387 (1992).

[6] M. Suzuki and K. Umeno in *Computer Simulation Studies in Condensed Matter Physics VI*, edited by D.P. Landau, K.K. Mon, and H.-B. Schüttler (Springer, Berlin, 1993).

[7] M.E. Tuckerman and G.J. Martyna, J. Phys. Chem. B **104**, 159 (2000).

[8] I.P. Omelyan, I.M. Mryglod, and R. Folk, Comput. Phys. Commun. **151**, 272 (2003).

[9] M. Krech, A. Bunker and D.P. Landau, Comput. Phys. Commun. **111**, 1 (1998).

[10] J. Frank, W. Huang, and B. Leimkuhler, J. Comput. Phys. **133**, 160 (1997).

[11] S.-H. Tsai, A. Bunker, and D.P. Landau, Phys. Rev. B **61**, 333 (2000).

[12] See e.g., D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, 2000).

[13] See e.g. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes, the Art of Scientific Computing*, second edition, Cambridge University Press, 1992.

[14] M. Suzuki, Prog. Theor. Phys. **56**, 1454 (1976).

[15] M. Suzuki, S. Miyashita, and A. Kuroda, Prog. Theor. Phys. **58**, 1377 (1977).

[16] I.P. Omelyan, I.M. Mryglod, and R. Folk, Phys. Rev. E **65**, 056706 (2002).

[17] I.P. Omelyan, I.M. Mryglod, and R. Folk, Comput. Phys. Commun. **146**, 188 (2002).

[18] D.P. Landau, S.-H. Tsai, M. Krech, and A. Bunker, Int. J. Mod. Phys. C **10**, 1541 (1999).

[19] M. Suzuki, Phys. Lett. A **201**, 425 (1995).

[20] S.A. Chin, Phys. Lett. A **226**, 344 (1997).

[21] I.P. Omelyan, I.M. Mryglod, and R. Folk, Phys. Rev. E **66**, 026701 (2002).

[22] M. Tuckerman, B.J. Berne, and G.J. Martyna, J. Chem. Phys. **97**, 1990 (1992).

[23] S.J. Stuart, R. Zhou, and B.J. Berne, J. Chem. Phys. **105**, 1426 (1996).

[24] K. Nho and D.P. Landau, Phys. Rev. B **66**, 174403 (2002).

[25] See e.g. I.P. Omelyan, I.M. Mryglod, and R. Folk, Phys. Rev. Lett. **86**, 898 (2001).

[26] M. Krech, unpublished.

[27] S.-H. Tsai and D.P. Landau, Phys. Rev. B **67**, 104411 (2003).

[28] R. Coldea, R.A. Cowley, T.G. Perring, D.F. McMorrow, and B. Roessli, Phys. Rev. B **57**, 5281 (1998).