Abstract

We outline how the wavelet transform, a hierarchical averaging scheme, can be used to perform both spatial and topological coarse-graining in systems with multiscale physical behavior, such as Ising lattices and polymer models. We illustrate how to create sampling mechanisms, which we call wavelet-accelerated Monte Carlo (WAMC), to study these systems and obtain qualitatively and quantitatively accurate results in orders of magnitude less time than using atomistic simulations.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: Wavelets; Molecular simulation; Ising model; Lattice gas; Self-avoiding walks

1. Introduction

One of the major new paradigms for simulations in recent years has been the rise of multiscale approaches to molecular simulations. Multiscale approaches have typically involved the use of several simulations of quite different types, each operating at different length or time scales. The various simulations are usually connected by "hand-shaking," the passing of parameters among the multiple levels of simulation. For example, a Monte Carlo simulation at the molecular scale might be used to derive the constitutive relationships needed for a finite element calculation, or a quantum mechanical calculation could be performed to estimate the rate constants required by a kinetic Monte Carlo simulation. While such approaches have been quite effective, there are also some substantial shortcomings associated with them. Most of the individual steps in the simulation are designed to operate at a specific length or time scale; adjusting these characteristic scales can be challenging, or even impossible. In addition, the existence of distinct regimes in which the assumptions underlying each calculation are valid means that there are often length and time scales over which we cannot effectively simulate the relevant physical processes.

One solution to these problems has been the development of multiscale approaches: that is, simulations which can operate on multiple physical scales (or resolutions). In these simulations, the specific level of resolution incorporated either is determined before simulation begins or is adjusted on-the-fly, which allows the performance of the simulation to be optimized according to the specific details of the physical system in question. The multiscale modeling framework that we shall outline in this paper exploits the behavior of self-similar systems such as lattice gases and polymer chains whose physical behaviors are often only weakly dependent on physical scale.

Our method is based upon the wavelet transform (Daubechies, 1992; Strang & Nguyen, 1996; Sweldens, 1995). Although the wavelet transform was originally applied to problems in signal processing and image analysis (Chou, 1991; Irving, 1995; Luettgen, Karl, Willsky, & Tenney, 1993), we have chosen the wavelet transform because of its data compression abilities. The basic principle in our application of the wavelet transform is that we use a characteristic property of our objects – such as the spin of a magnetic particle, or the position of an atom along the backbone of a chain molecule – as the basis for our sampling. The wavelet transform is then used to develop coarse-grained representations of "block" variables, as well as effective potentials, describing the behavior of the system over successively larger length scales.
This approach has several advantages, including general- 
ity with respect to the range of systems to which it can be 
and, extensibility, as it can be used as the basis for a hi-
erarchical simulation scheme; and efficiency, as the resulting 
algorithm is capable of yielding qualitatively accurate pre-
dictions of behavior, over a wide range of parameters, orders 
of magnitude faster than is capable with traditional atomistic 
simulations. We discuss our development of this paradigm in 
the form of a new simulation technique, wavelet-accelerated 
Monte Carlo (WAMC), through its application to lattice sys-
tems and to polymer random walks.

Very little work to date has been done in applying the 
wavelet transform to problems in statistical mechanics. The 
closest approaches to the work presented in this paper are 
multiscale analyses of the Landau-Ginzburg and XY models 
by Best (1999), Best and Schäfer (1994) and Best, Schäfer, 
and Greiner (1994), the heterogeneous coarse-graining 
scheme of Curtarolo and Ceder (2002), and the development 
of multiscale dynamic simulations by Katsoulakis, Majda, 

2. The wavelet transform

2.1. Definition of the wavelet transform

The wavelet transform takes a mathematical object and 
transforms it into another:

\[ \tilde{u} = W[u]; \quad (1) \]

the specific form of W depends both on the type of wavelet 
we have selected, and the object u to be transformed. All 
versions of the wavelet transform W, however, are derived 
from the same source: a set of coefficients which define the 
transform. For the methods considered in this paper, u will 
be a discrete set of data, so that it will be convenient to define 
W as a matrix.

Like the Fourier transform, the wavelet transform decom-
poses the object u into two separate components since two dif-
f erent functions, a scaling function \( \phi \) and a wavelet function 
\( \psi \), both operate on \( u \). However, the two functions separate its 
components not into cosines and sines, but into "averages" 
and "details." In addition, the wavelet transform is re-
cursive, so that it can be applied in succession to any set of 
averages calculated using the same wavelet transform to con-
struct another level of averages and another level of details.

If \( u \) is defined to be a set of samples \( u = (u(1), u(2), \ldots, u(n)) \), 
then applying the scaling and wavelet functions \( \phi \) and \( \psi \) to \( u \) 
creates a set of averages \( s(i) \) and a set of differences \( \delta(i) \):

\[ s(i) = \sum_{k=0}^{r-1} \phi(k)u(i + k), \quad (2) \]

\[ \delta(i) = \sum_{k=0}^{r-1} \psi(k)u(i + k), \quad (3) \]

where \( r \) is a finite integer which defines the length scale, often 
referred to as the "size of the support," over which \( \phi \) and \( \psi \) 
are nonzero. The index \( i \) runs from 1 to \( n \); generally the data 
set is padded with zeros to ensure that all sums in (2) and 
(3) are well-defined, although periodicity is sometimes used 
instead (Strang & Nguyen, 1996). The coefficients \( \phi(k) \) and 
\( \psi(k) \) in (2) and (3) are closely related (Daubechies, 1992; 
Strang & Nguyen, 1996), and are central in controlling the 
features of the wavelet transform.

The wavelet transform is inherently redundant: for every 
sample in the original set \( u \), we now have two values, a local 
average \( s(i) \) and a local difference \( \delta(i) \). Since the new data are sim-
ply linear combinations of the original values, it is superfluous 
to retain both sets. To keep the number of data points in the 
system constant, we keep only the odd-numbered \( s(i) \)’s and 
\( \delta(i) \)’s: this process is called downsampling (Strang & Nguyen, 
1996). Now we are left with \( n \) data: \( s(1), s(3), \ldots, s(n-1) \) 
and \( \delta(1), \delta(3), \ldots, \delta(n-1) \). These \( n \) points can be stored 
as the level-one wavelet transform \( \tilde{u} \) of \( u \), by assigning 
\( s(1), s(3), \ldots, s(n-1) \) to \( u^{(1)}(1), u^{(1)}(2), \ldots, u^{(1)}(n/2) \), and 
the corresponding \( \delta(i) \)’s to \( u^{(1)}(n/2 + 1), \ldots, u^{(1)}(n) \). The 
superscript denotes that the wavelet transform has been ap-
plied once to this data set. We can either stop at this level 
of description, or continue by further decomposing the av-
erages: then the new object \( u^{(1)} \) to be transformed is \( u^{(1)} = 
(\tilde{u}^{(1)}(1), \ldots, \tilde{u}^{(1)}(n/2)) \), and so on. Note that although \( \tilde{u}^{(1)} \) 
contains the averages \( s(i) \)’s and the differences \( \delta(i) \)’s obtained 
in the previous step, successive transforms only apply to av-
erages obtained in the previous step. This process can be 
repeated until we have reduced our set of averages to a single 
point; no further averaging is then possible.

The Haar wavelet pair, graphically represented in Fig. 1, 
is the oldest and simplest set of wavelets (Haar, 1910): the 
coefficients of the scaling function are \( \phi = (\phi(0), \phi(1)) = 
(1, 1)/\sqrt{2} \), while the wavelet function has coefficients 
\( \psi = (\psi(0), \psi(1)) = (-1, 1)/\sqrt{2} \). All other pairs require three 
or more coefficients per function, which means that no other pair 
has a support as compact as the Haar wavelet pair. The scaling 
function \( \phi \) simply averages the values stored at neighboring 
points, while \( \psi \) finds the difference between those values;

![Fig. 1](image-url)
the factor in the denominator is \( \sqrt{2} \) to ensure orthonormality between overlapping \( \psi(k) \) and \( \psi(l) \).

### 2.2. Matrix formulation

A simple method for implementing the wavelet transform is to construct a matrix equation. The vectors \( s \) and \( \delta \), which contain the wavelet-transformed coefficients of the decomposition, can be obtained by forming the matrices \( H \) and \( L \) and right-multiplying by the input vector \( u \):

\[
s = Hu \quad \text{and} \quad \delta = Lu,
\]

where the rows of \( H \) and \( L \) are given by \((h(1), \ldots, h(n))\) and \((l(1), \ldots, l(n))\), respectively:

\[
h(i) = (0, \ldots, 0, \phi(0), \phi(1), \ldots, \phi(r-1), 0, \ldots, 0),
\]

\[
l(i) = (0, \ldots, 0, \phi(0), \psi(1), \ldots, \psi(r-1), 0, \ldots, 0).
\]

The vectors \( h(i) \) and \( l(i) \) are padded so that \( \phi(0) \) and \( \psi(0) \) occur at index \( i \). To take advantage of downsampling, we can combine all the necessary coefficients into a single matrix \( W(1) \):

\[
(h(1), h(3), \ldots, h(n-1), l(1), l(3), \ldots, l(n-1)) = (H^T L^T)^T.
\]

Thus, the wavelet transformation \((1)\) can be written as

\[
\begin{bmatrix}
    s \\
    \delta
\end{bmatrix} = 
\begin{bmatrix}
    H^T \\
    L^T
\end{bmatrix} u.
\]

We will denote the product on the left-hand side of \((6)\) as \( \tilde{u} \). To apply the wavelets recursively, the set of averages \((u(1), u(3), \ldots, u(n-1))\)

\[
(6)
\]

produced by \((6)\) can be treated as a new data sample \( \tilde{u}^{(1)} = (u(3)/1, \ldots, u(n/2)/1) \), and operated on by \( W^{(1)} \), an \( N/2 \times N/2 \) reduction of \( W^{(1)} \), to produce a set of \( n/4 \) averages \((\tilde{\delta}^{(1)}(1), \tilde{\delta}^{(2)}(3), \ldots, \tilde{\delta}^{(n/2-1)}(n/2-1))\)

\[
(8)
\]

and a set of \( n/4 \) differences \((\tilde{\delta}^{(1)}(1), \tilde{\delta}^{(2)}(3), \ldots, \tilde{\delta}^{(n/2-1)}(n/2-1))\).

\[
(9)
\]

This process can be repeated as many times as desired, dividing the \( m \) averages \( s^{(k)} \) into \( m/2 \) averages \( \tilde{\delta}^{(k+1)} \) and \( m/2 \) differences \( \tilde{\delta}^{(k+1)} \).

However, since at each iteration the matrix \( W^{(k)} \) only operates on selected elements of the vector \( \tilde{u}^{(k)} = W^{(k-1)} \tilde{u}^{(k-1)} \), we can construct a single matrix corresponding to multiple consecutive iterations of the transform. Thus, to apply the wavelet transform \( K \) times, we can write the extended matrix product (Gines, Beylkin, & Dunn, 1998):

\[
\tilde{u}^{(0)} = Wu = \prod_{k=1}^{K} Q^{(k)} u.
\]

where \( Q^{(k)} \) is a block matrix of the form

\[
Q^{(k)} = \begin{bmatrix}
    W^{(k)} & 0 \\
    0 & 1
\end{bmatrix}.
\]

\[
(11)
\]

In \((11)\), \( Q^{(k)} \) is always an \( N \times N \) matrix, while the matrix \( W^{(k)} \) has size \((N/2^k-1) \times (N/2^k-1)\).

Since virtually all thermodynamic systems of interest are in two or three physical dimensions, it will be helpful to have an extension of the wavelet transform which can manipulate multivariate data sets. A particularly convenient basis function for use with the WAMC framework is the generalized orthogonal Haar wavelets outlined by Sweldens (1997). Orthogonal Haar wavelets can be constructed in any number of dimensions, and satisfy the same orthonormality properties as the one-dimensional Haar functions, although the orthonormality constant becomes \( 2^{-d/2} \), where \( d \) is the dimensionality of the physical system.

### 2.3. Using the wavelet transform as a coarse-graining scheme

There are two principal means for carrying out coarse-graining of a physical system: we can call these approaches spatial and topological. Spatial approaches operate on the simulation space, dividing it into regions, and then combining the regions from one scale to another. Topological approaches operate on the particles inserted into the simulation space, so that the rules for coarse-graining the system do not depend upon any specific physical structure of the simulation space, such as a lattice, but instead upon the structure of the particles. Ising lattices and polymer chains typify each of these individual approaches. We show how to develop coarse-grained simulations for each of these models below.

3. Spatial coarse-graining: lattice gases and the Ising model

The lattice gas is a useful model for studying surface processes, such as adsorption or catalytic reactions. A lattice gas consists of a regular lattice, each site of which can be either occupied or unoccupied; we represent the occupation number \( n_i \) of lattice site \( i \) by \( n_i = 1 \) for an occupied site and \( n_i = 0 \) for an unoccupied site. There is an interaction of strength \( \epsilon_0 \) between nearest-neighbor occupied sites. The energy of the lattice gas is therefore given by the Hamiltonian

\[
-\beta H = \epsilon_0 \sum_{\langle ij \rangle} n_i n_j,
\]

\[
(12)
\]

where the notation \( \langle ij \rangle \) indicates that sites \( i \) and \( j \) are nearest neighbors on the lattice, and \( \beta = (k_B T)^{-1} \) is the inverse temperature. These units are selected to make the interaction parameter \( \epsilon_0 \) dimensionless. In addition, we can specify a chemical potential \( \mu \) indicating the energy required to introduce an additional particle onto the lattice. Since the number
of particles contained on the lattice usually varies during the course of the simulation, we would need to use the grand canonical ensemble to study the typical lattice gas.

While it is possible to simulate a lattice gas directly, it is usually easier to exploit the relationship between the lattice gas and the Ising model, which was originally used to study magnetic lattices (Lavis & Bell, 1999). The Hamiltonian for the Ising model, which contains both nearest-neighbor pair-wise interactions as well as interactions between lattice sites and an external field, can be generally written in the form

$$-\beta H = \sum_i h_i \sigma_i + \frac{1}{2} \sum_i \sum_j J_{ij} \sigma_i \sigma_j,$$

(13)

where $\sigma_i = \pm 1$ is the spin of lattice site $i$, $h_i$ is the strength of the external field in the direction of the spins $\sigma_i$ at site $i$, and $J_{ij}$ is the strength of the interaction between the pair of sites $i$ and $j$. As in (12), the units in (13) are chosen to make the coefficients $h_i$ and $J_{ij}$ dimensionless, where the constant $1/2$ is included in Eq. (13) to avoid double-counting. The spin $-1/2$ Ising model, the most frequently encountered, has $\sigma_i = +1$ or $-1$ and $J_{ij} = 0$, unless $i$ and $j$ are nearest neighbors on the lattice, in which case $J_{ij}$ is a constant.

Although at first appearance the Hamiltonians (12) and (13) do not appear compatible, we note that $n_i$ and $\sigma_i$ are related via the transformation

$$n_i = \frac{1}{2} (\sigma_i + 1),$$

(14)

and that the number of spins $N$ in the magnetic lattice is equivalent to the total volume (or area) of the lattice gas. The other correspondences that are necessary are

$$e_0 = 4f,$$

(15)

$$\hbar = kT \ln z = \mu + kT \ln Q_1,$$

(16)

where in (16) $z$ is the chemical activity, and $Q_1$ is the partition function of a single lattice particle. All other details of the system—such as the mean occupation number or the pressure of the lattice gas—can be computed using these relations. For instance, the mean occupation number of the lattice gas is

$$\bar{n}_h = \frac{1}{N} \left( \frac{m}{N_m} + 1 \right),$$

(17)

where $\bar{n}_h$ is the number of occupied sites in the lattice gas, and $m$ is the mean magnetization of the Ising model.

Eq. (13) is a compact representation of the Ising model Hamiltonian. However, the challenge of expanding the lattice variables $\sigma_i$ and $\sigma_j$ in terms of the appropriate wavelet coefficients means that these equations are impractical for use in a simulation. However, because the Ising model has a lattice with regular structure, we can construct a matrix formulation of the Hamiltonian. To apply the wavelet transformation. Since the system is described on a regular lattice, we can use discrete wavelets (also known as filter banks), so a matrix formulation of the Hamiltonian would be convenient. The interaction $J_{ij}$ between sites can be converted to a matrix $J$ using graph theory (Cormen, Leiserson, & Rivest, 1990). Similarly, the external field and spin variables can be converted into vectors: $u = (\sigma_1, \sigma_2, \ldots, \sigma_N)$ and $h = (h_1, \ldots, h_N)$, respectively. We can then convert (13) into the matrix equation

$$-\beta H = h^T u + u^T J u.$$

(18)

We can represent the discrete wavelet transform as a self-adjoint matrix $W$ (Ismail, Rutledge, & Stephanopoulos, 2003); applying the wavelet transform to (18) is accomplished by inserting the identity matrix $I = W^T W$ between each pair of terms in (18). The resulting expression,

$$-\beta H = h^T W^T W u + u^T W^T W J W u,$$

(19)

can then be written in the same form as (18) by choosing

$$\bar{h} = W h, \quad \bar{u} = W u, \quad \bar{J} = W J W^T,$$

which makes (19)

$$-\beta \bar{H} = \bar{h}^T \bar{u} + \bar{u}^T \bar{J} \bar{u}.$$

(20)

The process which coarse-grains (18)–(20) can be repeated many times; we indicate the number of iterations $K$ of this procedure as a superscript in front of each of the transformed terms.

It should be noted that if we keep track of all of the scaling and wavelet coefficients produced using the wavelet transform, Eq. (20) is identical to (18). On the other hand, the structures of the matrix $J$ and the spin vector $u$ in (20) are much less sparse than (18), which does not makes implementing a simulation any simpler. To reduce the complexity of the simulation, we assume that we can neglect terms in the Hamiltonian (20) which are dependent on the wavelet coefficients. Thus, we can restrict the scope of Hamiltonian to those terms which contain only the scaling coefficients produced by the final iteration of the wavelet transform. Consequently, each iteration of the wavelet transform reduces the dimensions of the matrix $J$ by a factor of $2^d$, where $d$ is the dimensionality of the lattice.

One further point should be made at this time: although some have interpreted (20) as excluding self-interaction terms (Katsoulakis & Vlachos, 2003), it is straightforward to show that self-interaction terms are almost certain to arise in (20). Consider neighboring spins $I_{nm}^{(s)}$ and $I_{np}^{(s)}$ on a particular lattice which will be mapped into the same averaging coefficient $I_{nm}^{(s+1)}$ at the next iteration of the wavelet transform. Then, inserting the wavelet expansions for $I_{nm}^{(0)}$ and $I_{np}^{(0)}$ into the Hamiltonian (18) yields a self-interaction term of the form $I_{nm}^{(1)} I_{nm}^{(1)}$ which will survive the elimination of terms containing differencing coefficients.
3.1. Effect of the wavelet transform on the configuration space

By using the wavelet transform method, we are attempting to merge two different approaches to simplifying molecular simulations: renormalization-group approaches and mean-field approaches (Kadanoff, 2000; Ma, 1976a; Pathria, 1996). The wavelet-accelerated Monte Carlo algorithm attempts to capture the effects of local fluctuations at a given scale by modifying the set of distributions passed from one scale to the next. That is, by performing a simulation at a given scale, it computes the average properties on a finer-scaled lattice, which is in turn passed to a coarser-scale simulation. It is expected that the microscopic effect of fluctuations will be seen in the distributions of the average properties of the system, which will in turn affect the coarser-scale simulations. However, the interactions that are incorporated into the coarse-grained model do not explicitly include terms describing events occurring on the finer length scales studied at the previous scale of the simulation. In this respect, the wavelet transform method is similar to renormalization group approaches, in which a specific cutoff length is increased with each iteration of the renormalization transformation.

From a conceptual standpoint, the wavelet transform simplifies the configuration space. For example, consider a $2 \times 2$ block of a lattice gas, as illustrated in Fig. 2. There are 16 possible unique configurations, as shown in the left-hand box. If the wavelet transform replaces each of these $2 \times 2$ blocks with a single site whose occupation is the average of the four sites, then we see that there are five possible configurations of the $1 \times 1$ block: fully occupied, 3/4 occupied, 1/2 occupied, 1/4 occupied, and fully unoccupied, as indicated in the right-hand box in Fig. 2. Thus, the configuration space of the original block, which has sixteen possible states, is reduced via the wavelet transform to a space with five states. This makes the configuration space much smaller, and allows us to sample it more rapidly.

Although (13) and (20) are structurally the same, we cannot impose a one-to-one correspondence between the states in the configuration space of (13) and the states in the configuration space of (20). Since multiple states in the original configuration space are mapped to a single state in the coarse-grained configuration space, we introduce a weighting factor $g$ for each state in the coarse-grained configuration space. While in principle it is possible to determine the exact weighting factors needed to ensure exact agreement between the original and coarse-grained ensembles, computing those factors would require analyzing every state in the system. As an approximation, we assume that $g$ can be estimated as the degeneracy of a coarse-grained state: that is, the number of states in the original configuration space which are mapped to a particular coarse-grained state.

The effect of this procedure is shown schematically in Fig. 3, which represents a lattice gas. At the original scale, there are only two options for each site—occupied or unoccupied. Applying the wavelet transform to a $2 \times 2$ block on this lattice leads to a new “block site,” which can have occupations ranging from fully empty to fully occupied, with one-quarter, half, and three-quarters occupied as the other options. If we choose, we can then re-block each $5 \times 2$ segment of the new model to produce an even coarser model, where each site can take one of 17 different values (corresponding to occupation fractions of $0, \frac{1}{16}, \frac{1}{8}, \ldots, 1$). The Hamiltonian which describes each of these systems is structurally equivalent, but the numeric constants $h_i$ and $J_{ij}$ in 13 will be different in each case.

The main challenge is to devise an approach for generating and accepting these coarse-grained configurations in a manner which is thermodynamically consistent with the results obtained for the original model. Depending on the size of the coarse-grained system, we can do this either analytically or numerically.

3.2. Analytic treatment

For simple systems, we can enumerate all of the configurations in the coarse-grained model, which allows us to determine the partition function, and thus all of the thermodynamic properties without molecular simulation. As an example, we can compute the free energy surface,

$$A(h, J) = -k_BT \ln Z(h, J),$$

where

$$Z(h, J) = \sum \exp \left( \frac{-h_i + \sum J_{ij} \sigma_i \sigma_j}{k_B} \right).$$
where $Z(h, J)$ is the partition function determined by enumerating all states according to the Hamiltonian (20). For a coarse-grained system, we can still use (21) to compute the free energy of the system; however, the partition $Z(h, J)$ needs to be modified to account for the weight of each state:

$$Z_{cg}(h, J) = \sum_i g_i e^{-βh(0, i)},$$

(22)

where the summation is taken over all states in the coarse-grained configuration space and $g_i$ is the relative “weight” of state $i$ in the coarse-grained ensemble. Thus, for systems where we can compute (21) exactly, we can use (22) to compare the results obtained from the original and coarse-grained models.

To illustrate this, let us consider a $4 \times 4$ Ising model with periodic boundary conditions. There are $2^{16}$ possible configurations, which we can enumerate and evaluate relatively easily. Then, let us compute explicitly the free energy curve for the case of zero external field and temperatures in the range $0 \leq (βJ)^{-1} \leq 10$. $Δ(βJ)^{-1} = 0.1$; this is shown as the solid curve in Fig. 4. We can also repeat this calculation for a $2 \times 2$ and a $1 \times 1$ coarse-grained system produced by one and two iterations of the wavelet transform procedure. The free energies for these cases are shown as the dashed curves in Fig. 4. The relative agreement of all three curves can be seen by the difficulty of distinguishing the curves in the graph.

While it is possible to compute the exact statistics for all three cases, the computational time of the coarse-grained models is significantly smaller than the computational time of the original model. For example, in the time required to perform the free energy calculations in Fig. 4 for the $4 \times 4$ model, we can perform the same calculation for a $2 \times 2$ coarse-grained model for the entire surface $0 \leq (βJ)^{-1} < 10$, $Δ(βJ)^{-1} = 0.1$ and $-5 \leq h \leq 5$, $Δh = 0.025$. This surface is plotted in Fig. 5. We can show that the entire surface agrees with the $4 \times 4$ case within a few percent error. Thus, we have effectively expanded the parameter space we are able to study, as we have reduced the phase space from $2^{16}$ configurations to $5^4$ configurations. In addition, the results we obtain from the coarse-grained model are in excellent agreement with the original model.

3.3. Numerical treatment

3.3.1. Wavelet-accelerated Monte Carlo (WAMC) algorithm

The effect of applying the wavelet transform in (20) is to create a representation in which the elements of $u^K$ represent “block spins” whose values are determined by wavelet averaging over a well-defined region of the original system. The Hamiltonians (13) and (20) have the same formal structure so, from a procedural standpoint, Monte Carlo simulations of the two systems will be essentially identical. The only modifications needed for the coarse-grained system are the ability to select new microstates $\tilde{u}^{(K)}$ which are consistent with wavelet transformations of the original microstates $u$, and the elimination of unwanted degrees of freedom from (20). One other modification is an obvious consequence of the wavelet transform procedure: the set of allowed values of the elements of $\tilde{u}^{(K)}$ in (20) is larger than the corresponding set of allowed values of the elements of $u$.

To construct a new microstate $\tilde{u}^{(K)}$, we need to have a probability distribution $p(\tilde{u}^{(K)})$ for individual sites in the new, coarse-grained lattice. To determine the distribution which yields the same results as the original lattice, we would need to determine the energy of every state in the original configuration space or perform a detailed simulation of the original system. Since we want to avoid both of these options, we must derive an alternative method to estimate $p(\tilde{u}^{(K)})$.

For simple systems, we can compute the degeneracy “weighting factor” described above. For larger systems, for which this computation would be impractical, we simulate a sublattice with the same dimensions as $\tilde{u}^{(K)}$, ignoring phys-
particular, the blocking approach of Kadanoff (2000). How-
together, and the multigrid MC methods can only be applied spa-
chains, while renormalization-group Monte Carlo methods
W AMC algorithm can be applied topologically to polymer
complete all calculations at one scale before proceeding to
scales much more frequently—typically, one lattice pass at
point, this method attempts to work on a series of progres-
form defines a single block spin $\tilde{u}_K$ as a linear function of
the individual spins at level $K = 1$ which comprise the block
spin, we can use the probability distribution of the magneti-
zation $m$ as an estimate for the probability distribution of the
block spin $\tilde{u}_K$. Finally, using the distribution for $\tilde{u}_K$ as a starting point, we perform a Monte Carlo simulation on the
system of block spins defined by the Hamiltonian (20). The
detailed balance condition is satisfied for the coarse-grained
simulation:
$$\frac{\alpha(\tilde{u}_K^m \to \tilde{u}_K^n)}{\alpha(\tilde{u}_K^n \to \tilde{u}_K^m)} \frac{p(\tilde{u}_K^n)}{p(\tilde{u}_K^m)} e^{-\beta(h(\tilde{u}_K^n) - h(\tilde{u}_K^m))},$$
(23)
where $\alpha(m \to n)$ is the probability of accepting a move from
microstate $m$ to microstate $n$, and $\alpha(m)$ is the probability of
selecting microstate $m$ as determined from simulations on finer-grained lattices at lower scales. The basic steps above define the wavelet-accelerated Monte Carlo (W AMC) algo-
rithm; a further discussion of the details and implementa-
tion of this algorithm are described in a recent paper (Ismail,
Stephanopoulous, & Rutledge, 2003).

The spirit of this method is close to real-space renomal-
ization methods (Ma, 1976a; Wilson, 1971a, 1971b), and in
particular the blocking approach of Kadanoff (2000). How-
ever, the wavelet-transform method is substantially differ-
ent from renormalization group methods, as well as the
renormalization-group Monte Carlo methods developed by Ma (1976b), Swendsen (1979) and Pawley, Swendsen, Wal-
lace, and Wilson (1984). The wavelet-transform approach to
course-graining does not require that the allowed values that
a spin can take remain fixed from one coarse-graining itera-
tion to the next; instead, the spin can take any value that is
a permissible average value for the block spin, thereby ob-
vviating the need for the "tie-breakers" sometimes required in renormalization theories (Goldenfeld, 1992). In addition,
the W AMC framework does not require a posteriori calcu-
lation of the coupling constants, as is found in the Wilson
recursion approach or in Migdal's "bound-moving" frame-
work (Migdal, 1976). Comparison can also be made to the
multigrid Monte Carlo work of Goodman and Sokal (1986,
1989). Introduced to address the critical slowing down phe-
nomenon observed in lattice simulations near the critical
point, this method attempts to work on a series of progress-
ively coarser meshes, but moves back and forth between scales much more frequently—typically, one lattice pass at
each scale per cycle. The W AMC algorithm attempts to
complete all calculations at one scale before proceeding to
the next, coarser scale. In addition, as we shall show, the
W AMC algorithm can be applied topologically to polymer
chains, while renormalization-group Monte Carlo methods and the multigrid MC methods can only be applied spa-
tially.

3.3.2. Simulation results

Because of the mean-field like characteristics of the algo-
rithm, W AMC tends to produce overestimates for the critical
point of the system. Therefore, if we start with the high-
temperature limit of our algorithm and slowly reduce the
temperature in our simulation, we can observe the movement
toward the critical point by watching various fluctuation pa-
rameters, such as the heat capacity at constant external field,
$C_b = \langle (E^2) \rangle - \langle E \rangle^2 / k_B T^2$. Near the critical point, we ex-
ppect to see a rapid increase in the value of $C_b$, consistent with the
expected logarithmic divergence observed in the limit of
finite-size systems (Frenkel & Smit, 1996; Pathria, 1996).

Since the indicator is the rapid increase in the heat ca-
pacity, we can monitor changes in $\Delta C_b/\Delta T$ as we decrease
the temperature. Once the magnitude of $\Delta C_b/\Delta T$ increases
above some threshold, it indicates that we are too close to
the critical point of the system for the resolution we have
chosen. At this point, we choose to refine our model, and
include fewer degrees of spins in our coarse-grained blocks.
This process of stepwise refinement continues until the origi-
nal atomistic model has been reached. We will need to use
the atomistic model until we have moved past the critical
point. However, the region of parameter space where this
is necessary is relatively small compared to the complete
parameter space. After the critical point has been reached,
we can reverse the process; as the magnitude of $\Delta C_b/\Delta T$
decreases, we can coarse-grain our model to accelerate
our calculations.

As an example, we compute the spontaneous magneti-
ization curve for a $64 \times 64$ Ising lattice in the temperature
range $0.5 \leq T \leq 10.0$, with $\Delta T = 0.05$, and choosing as
our refinement criterion $\Delta C_b/\Delta T \leq -0.5$, until we reach
the finest scale, corresponding to the original problem. We
begin by using the wavelet transform to reduce the system
to an $8 \times 8$ lattice, each block of which represents an $8 \times 8$
sublattice. [For convenience, we will call a simulation of a
$b \times b$ lattice representing $a \times a$ sublattices an $(a, b)$-model.] Starting from $T = 10.0$, we find that $\Delta C_b/\Delta T \leq -0.5$ from
$T = 10.0$ down to $T = 5.1$. At this point, we study a $(4, 16)$-
model until we reach $T = 4.0$, at which point the refinement
criterion is exceeded. Refining once more, we proceed with
a $(2, 32)$-model until we reach $T = 3.4$, at which point the
refinement criterion is again crossed.

The next refinement is the original $64 \times 64$ model; there-
fore, we simulate the original model until we have passed
the critical point, so that $\Delta C_b/\Delta T$ is positive. As a coars-
ing criterion, we select for simplicity the opposite of the
refinement criterion, $\Delta C_b/\Delta T \leq 0.5$. Using this criterion,
we find that we coarsen the model to the $(2, 32)$, $(4, 16)$- and
$(8, 8)$-models at temperatures of $T = 1.75$, $T = 1.65$, and
$T = 1.55$, respectively. The rapid coarsening of the model
results from the higher estimates of the critical point in the
coarsened models. Since we are well past the critical point,
we expect changes in the heat capacity as a function of tem-
perature to be relatively small, and thus it is possible to obtain
accurate results from a relatively coarse model. Computation-
ity, the time required to create this diagram was only 28% that required to perform a standard Metropolis Monte Carlo simulation with the same number of steps. Moreover, in the
regions that were not simulated using MMC, the computation time required was just 8% of the time required for MMC. The resulting plot of magnetization versus temperature, shown as
Fig. 6, compares favorably to the analytical solution of Onsager’s, which is also shown (Onsager, 1944).

Similarly, we can construct a phase diagram for a two-dimensional Ising model of size 64 × 64, as shown in
Fig. 7. We see that the essential structure of the ferromagnetic phase diagram is preserved by the WAMC algorithm. However, as mentioned above, the principal change is the location of the critical point, which is deflected upwards as a result of the mean-field like effect of the WAMC algorithm. However, the details of the phase diagram far away from the critical point are both qualitatively and quantitatively correct: average magnetization goes to unity for low temperatures and high external fields, and vanishes in the high temperature limit.

The accuracy of the results produced by the WAMC algorithm are dependent largely on the fluctuation properties of the system, and not the particular details of the potential describing the system at the atomistic scale. That is, our results depend on the relative accuracy of the assumption of setting all of the δi terms equal to zero in the coarse-grained Hamiltonian Eq. (20). For regions of parameter space where fluctuation quantities like the heat capacity are small, such as in the low- and high-temperature regimes, and for large absolute values of the external field or chemical potential, there is very good quantitative and qualitative agreement between WAMC simulations and traditional Metropolis Monte Carlo results. As fluctuations become correlated and the magnitude of fluctuation properties increase, our algorithm, which effectively imposes a cutoff on fluctuations by setting interaction coefficients involving wavelet differencing terms to zero, has increasingly greater difficulty in capturing the behavior of the atomistic system.

4. Topological coarse-graining: simple polymer lattice models

4.1. Coarse-graining a polymer chain

As mentioned above, significant challenges arise once topological constraints, such as molecular connectivity, are introduced into a system to be coarse-grained. Simply coarse-graining the physical space of the system, without paying attention to the connectivity of individual particles, can introduce non-physical behavior. Similarly, arbitrarily assigning molecules to coarse-grained particles, as can be done in dissipative particle dynamics (Español, 1995; Schläpfer, Hoogerbrugge, & Manke, 1995), can make it difficult to preserve the identity of individual chains during a simulation.

Fig. 8. Illustration of “adaptive” coarse-graining scheme for a 64 × 64 Ising model. The simulation sizes are as indicated; the model spends approximately 90% of its computational time in the region marked MMC (Metropolis Monte Carlo). Reprinted from (Ismail, Stephanopoulos, et al., 2003).

Thus, the fundamental concept behind the application of the wavelet transform to systems with topological structure, such as simple polymer chains, is that we apply the wavelet transform along the chain backbone to the positions of the individual “atoms” (or segments). Thus, we take as our input data the set of positions R = {r1, ..., rN}, and define the unnormalized Haar wavelet transform as the mapping

\[ r_k^{(i)} = \frac{1}{2} (r_{2k-1}^{(i)} + r_{2k}^{(i)}), \]  

(24)

\[ r_k^{(i)} = \frac{1}{2} (r_{2k-1}^{(i)} - r_{2k}^{(i)}). \]  

(25)

The output is the set of averages \( r_1^{(1)}, r_2^{(1)}, \ldots, r_N^{(1)} \) and the set of differences \( r_1^{(2)}, r_2^{(2)}, \ldots, r_N^{(2)} \), where the superscript denotes the number of times we have repeated the process (on the set of scaling coefficients). The effect of the averaging operator in (24) is to create a new coarse-grained bead \( r_k^{(i)} \) at the center of mass of the beads at \( r_{2k-1}^{(i)} \) and

Fig. 7. Phase diagram plotting average magnetization versus temperature and external field strength for a 64 × 64 ferromagnetic Ising lattice, computed using an (8, 8)-model via WAMC. The general features correspond to those that would be produced with standard MMC, but require less than 3% of the computational time. Reprinted from (Ismail, Stephanopoulos, et al., 2003).
The different operator (25) returns the distance between the original particles’ positions and their center of mass. A schematic representation of this operation is shown in Fig. 8. The methods most similar to the approach presented here are the “soft colloid” method of Louis, Bolhuis, Hansen, and Meijer (2000) and the “bond fluctuation” model of Carmesin and Kremer (1988).

4.2. Self-avoiding chains

We perform our simulations on self-avoiding random walks. These walks have a pairwise interaction potential $U(r_i, r_j)$.

When we coarse-grain the chain, we can write the total number of beads $N$ in the detailed chain as $N = N_b \times N_e$, where $N_b$ is the number of beads actually being simulated in the chain, and $N_e$ is the “effective size” of a bead; that is, $N_e$ represents the number of beads on the original representation of the chain that are combined to form a single coarse-grained bead. By definition, an atomistic simulation has $N_e = 1$ and therefore $N = N_b$. If we proceed through multiple stages of a hierarchical simulation, we write the vector $N_e = (N_{b,1}, N_{b,2}, \ldots, N_{b,m})$, where $m$ is the number of stages in the simulation so far. The effective bead size $N_{b,m}$ is then given by $N_{b,m} = \prod_{j=1}^{m} N_{b,j}$. In general, each $N_{b,j}$ is of the form $2^k$, where $k$ is the number of iterations of averaging that have been performed using (24) and (25).

4.3. Applying the WAMC algorithm to polymer chains

Our hierarchical algorithm proceeds as follows. We start by performing an atomistic simulation (O(10⁶) attempted MC moves) on a chain containing $N_{b,1}$ beads, with $N_{b,1} = 1$. For simplicity, we assume that the polymer exists on a three-dimensional cubic lattice and use the pivot algorithm (Lal, 1969; Madras & Sokal, 1988) to generate new configurations of the chain. Then, every $N_{b,1}$ steps, we use (24) and (25) to determine the centers-of-mass $r^{(K)}$ of the coarse-grained beads at a given length scale $N_{b,2} = 2^2$, and the wavelet coefficients $w^{(1)}, \ldots, w^{(K)}$.

From the locations of the centers of mass, we can compute the coarse-grained bond lengths, bond angles, and torsion angles. Like the weighting factors used in the Ising model, we keep track of the distributions of the bond lengths and angles and use them as weighting factors at the next stage of the simulation. The next step is a coarse-grained simulation of a chain of length $N_{b,2}$ beads, where each coarse-grained bead represents $N_{b,2}$ beads of the original chain, where $N_{b,2} = 2^2 \leq N_{b,1}/4$. The upper bound on $N_{b,2}$ is created by the need to have four segments to compute the distribution of coarse-grained torsion angles. If the chain length $N = N_{b,2} \times N_{b,2}$ is the desired length, we can terminate the simulation; otherwise, we can proceed to a third stage simulation of $N_{b,3}$ beads, each with effective size $N_{b,3} = (N_{b,2} \times N_{b,2})/4$. In order to reach the desired length $N$ of the actual chain, we can perform as many stages as we like. The coarse-grained algorithm runs independently of the source of the input distributions; the distributions can be derived from experimental data, from an atomistic simulation, or from a previous iteration of the coarse-grained algorithm.

A move on the coarse-grained lattice consists of replacing a selected bond with a new bond whose internal coordinates are drawn from the bond-length, bond-angle, and torsion-angle distributions derived in the previous stage of
the simulation. Following the work of Dautenhahn and Hall (1994), the remainder of one side of the chain is displaced correspondingly to preserve the connectivity of the chain. Once the new configuration is obtained, it is checked for the course of the simulation, we could reconstruct the chain and the distribution and behavior of the \( r_{k}^{(i)} \)'s. The distribution and behavior of the \( r_{k}^{(i)} \)'s are usually much more complicated than that of the \( r_{e}^{(i)} \)'s. Consequently, we will almost always find it advantageous to use the \( r_{k}^{(i)} \)'s instead of the combination of \( r_{k}^{(i)} \)'s and \( r_{e}^{(i)} \)'s.

### 4.4. Results

A comparison of the results obtained for a detailed atomistic simulation and for two-scale coarse-grained simulations of chains up to length \( 2^{13} \approx 8192 \) beads are shown in Fig. 9. Two different coarse-grained simulations are plotted, showing \( N_{e,2} = 32 \) and 64, respectively; the mean end-to-end distances of the three simulations show excellent agreement with one another, indicating that for relatively large values of \( N_{e,2} \), there is little effect on the end-to-end distance. Additionally, the Flory exponent for the scaling of the end-to-end distance, \( \langle R \rangle \sim N^{\nu} \), is shown to be \( \nu_{t} = 0.578 \pm 0.006 \) for \( N_{e,2} = 32 \) and \( \nu_{t} = 0.581 \pm 0.006 \) for \( N_{e,2} = 64 \), both of which are within 1% of the best available estimate \( \nu \approx 0.577 \) (Hughes, 1995).

To show that this hierarchical simulation strategy is effective for studying large systems, we can measure its performance over a variety of chain sizes, as shown in Fig. 10, which shows the running times for three different algorithms. The topmost line in the graph corresponds to a detailed atomistic simulation using an algorithm comparable to the WAMC algorithm; the running time of this algorithm is approximately \( O(N^{4/5}) \). If we optimize the atomistic algorithm by using the pivot algorithm, along with some of the additional improvements suggested by Kennedy (2002), we obtain the “optimized atomistic” plot shown. The running time has been reduced to roughly \( O(N^{3/5}) \); still further changes can reduce the running time to \( O(N) \) or below; however, the resulting algorithm is impractical for use in models with interparticle potentials other than self-avoidance.

The lower line plotted in Fig. 10 represents the running time of the WAMC algorithm with \( N_{e} = 32 \). We note that the algorithm has a running time of \( O(N^{7/5}) \), which is in between the results for the atomistic and optimized atomistic algorithms. However, since by definition we have \( N_{e} \ll N \), we expect that the WAMC algorithm will be faster on a per-move basis than either atomistic algorithm, unless the associated prefactor is much larger for the WAMC algorithm. However, since both methods are executing variations on the same basic process, this does not occur, and the WAMC algorithm is generally faster than either atomistic technique.

Our work suggests that the choice of the optimal effective bead size depends upon the user’s requirements. There is, of course, a slight increase in the error of the algorithm associated with increasing amounts of coarse-graining. At the same time, the relative performance will increase, since there are fewer beads in the system whose interactions need to be computed for a given MC move. The relative importance of accuracy and speed in simulation results will therefore drive the selection of effective bead size. If both are equally important, then selecting an effective bead size on the order of...
\[ N^{1/2} \] would be optimal; more accurate results favor smaller values, and faster results favor larger values.

5. Conclusions

The wavelet-accelerated Monte Carlo scheme is a robust technique for performing coarse-grained molecular systems with multiscale behavior. The simulations, although requiring more structural information than the initial atomistic simulation, have many fewer degrees of freedom, and therefore execute much more quickly. In addition, the adaptable nature of wavelets allows us to adjust our coarse-graining strategy without knowing a priori any information about the system to be studied. Moreover, this gives us the ability to explore physics at any scale between those of the atomistic models and the model with the maximum allowable coarse-graining, a feature which other "handshaking" coarse-graining methodologies cannot offer.

The goal of developing WAMC has been to create a framework for coarse-graining which is more generally applicable than currently existing simulation methodologies, which tend to address a single type of system. In general, we do not restrict the moves that can be constructed within a stage of the simulation, in either the lattice model or the polymer model. Any set of moves compatible with detailed balance within the simulation stage can be incorporated into our method. Thus, for example, by restricting movement to states in which the average spin is one higher or lower than the present value, and assigning appropriate rate constants for these moves on the basis of finer-scaled simulations, one could produce a "kinetic WAMC" algorithm for lattice systems.

The WAMC paradigm is predicated on the existence of self-similarity in the system. Our method systematically constructs representations of larger particles by studying the interactions between and behavior of smaller particles. Thus, the existence of self-similarity should be sufficient to permit the use of WAMC in other systems. For systems where self-similarity is not exhibited, such as in systems with variable external fields or interaction strengths, our method would still be applicable, but it would be necessary to use multiple distributions describing the various parameter conditions used in the system.

Acknowledgments

The authors would like to thank the DOE/Krell Institute Computational Sciences Graduate Fellowship for their financial support. This work was also supported by the Engineering Research Centers Program of the National Science Foundation under NSF Award Number EEC-9731680. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

References


