Cluster Monte Carlo algorithm for the quantum rotor model

Fabien Alet and Erik S. Sørensen

Laboratoire de Physique Quantique and UMR 5626, Université Paul Sabatier, 31062 Toulouse, France
Department of Physics and Astronomy, McMaster University, Hamilton, Ontario, Canada L8S 4M1

We propose a highly efficient “worm”-like cluster Monte Carlo algorithm for the quantum rotor model in the link-current representation. We explicitly prove detailed balance for the algorithm even in the presence of disorder. For the pure quantum rotor model with \( \mu = 0 \), the algorithm yields high-precision estimates for the critical point \( K_c = 0.333 \pm 0.005 \) and the correlation length exponent \( \nu = 0.670(3) \). For the disordered case, \( \mu = \frac{1}{2} \pm \frac{1}{2} \), we find \( \nu = 1.15(10) \).

DOI: 10.1103/PhysRevE.67.015701

PACS number(s): 74.20.Mn, 02.70.-c, 05.45.Mt, 73.43.Nq

What types of insulating, conducting, superconducting, and more exotic phases occur in two-dimensional systems at \( T = 0 \) is a topic of considerable current interest. A significant amount of theoretical [1–4] and experimental [5,6] effort has focused on bosonic systems where a superconductor to insulator transition is known to occur. In agreement with most experiments [5], it was shown under quite general conditions [1,2] that a transition can occur directly between the superconducting and insulating states. However, more recently, it has been suggested that an exotic metallic phase is also possible [4,6]. In this context, precise numerical results would be very valuable, and in the present paper we propose a very efficient cluster Monte Carlo algorithm for this purpose, allowing us to significantly improve previous results. In particular, we show that the inequality [7] \( \nu \geq 2d/4 \) is not violated in the presence of disorder, resolving contradictions in previous work. The high precision of the algorithm should allow for precise calculations of transport properties of quantum rotor models studied theoretically in [1,4]. The ideas presented here could also be useful for the study of classical spin systems [19].

Low-dimensional bosonic systems are often described in terms of the (disordered) boson Hubbard model: \( H_{\text{BH}} = \sum_r \left[ \left( U/2 \right) \hat{n}_{r}^2 - \mu \hat{n}_{r} - t_0 \sum_{\langle \mathbf{r}, \mathbf{r}' \rangle} \hat{c}_{\mathbf{r}} \hat{c}_{\mathbf{r}'} + \hat{c}_{\mathbf{c}} \hat{c}_{\mathbf{c}} \right] \). Here \( U \) is the on-site repulsion, \( t_0 \) is the hopping strength, \( \mu \) is the chemical potential varying uniformly in space between \( \mu \pm \Delta \), and \( \hat{n}_{r} = \hat{c}_{\mathbf{r}} \hat{c}_{\mathbf{r}}^\dagger \) is the number operator. If we set \( \hat{c}_{\mathbf{c}}^\dagger = \hat{c}_{\mathbf{c}} e^{i \phi_{\mathbf{c}}} \) and integrate out amplitude fluctuations, \( H_{\text{BH}} \) becomes equivalent to the quantum rotor model [8]:

\[
H_{\text{rot}} = \frac{1}{2} \sum_{\mathbf{r}} \left( \frac{1}{i} \frac{\partial}{\partial \theta_{\mathbf{r}}} \right)^2 + i \sum_{\mathbf{r}} \mu_{\mathbf{r}} \frac{\partial}{\partial \theta_{\mathbf{r}}} - t \sum_{\langle \mathbf{r}, \mathbf{r}' \rangle} \cos(\theta_{\mathbf{r}} - \theta_{\mathbf{r}'}) - \frac{t_0}{i} \sum_{\langle \mathbf{r}, \mathbf{r}' \rangle} \left( \frac{1}{i} \frac{\partial}{\partial \phi_{\mathbf{r}}} \right) \left( \frac{1}{i} \frac{\partial}{\partial \phi_{\mathbf{r}'}} \right).
\]

Here, \( \theta_{\mathbf{r}} \) the phase of the quantum rotor, \( t \) the renormalized hopping strength, and \( \langle 1/\mathbf{i} \rangle (\partial / \partial \theta_{\mathbf{r}} - \partial / \partial \phi_{\mathbf{r}}) \) is the phase. The quantum rotor model describes a wide range of phase transitions dominated by phase fluctuations, and it is well known [8] that an equivalent classical model exists where the Hamiltonian is written in terms of currents defined on the links of a lattice, \( \mathbf{J} = (J^x, J^y, J^z) \). These link-current variables describe the “relativistic” bosonic current which should be divergenceless, \( \nabla \cdot \mathbf{J} = 0 \). In terms of these variables, the classical (2 + 1)D Hamiltonian can be written as follows [8]:

\[
H = \frac{1}{K} \sum_{\langle \mathbf{r}, \mathbf{r}' \rangle} \left[ \frac{1}{2} J^2_{\mathbf{r}, \mathbf{r}'} - \mu_r J_{\mathbf{r}, \mathbf{r}'} \right].
\]

\( \Sigma' \) denotes a summation over configurations with \( \nabla \cdot \mathbf{J} = 0 \). Varying \( K \) corresponds to changing the ratio \( t/U \) in the quantum rotor model. The quantum rotor model has been extensively studied [8–10] in this representation, but a number of conclusions can be questioned due to severe finite-size effects. For notational convenience it is useful to slightly enlarge the definition of the link currents at a given site in the following way: At each site \( (\mathbf{r}, \tau) \) in the lattice we define six surrounding link variables \( J^\sigma_{\mathbf{r}, \tau} \), where \( \sigma \) runs over \( x, y, \tau \). Note that, with this notation \( J^- = J^x \tau^y - J^y \tau^x \) and \( J^x = J^x \tau^y + J^y \tau^x \), so that the sums of the incoming and outgoing currents are equal. Conventional Monte Carlo updates [8] on this model consist of updating simultaneously four link variables. Global moves, updating a whole line of link variables thus allowing particle and winding numbers to fluctuate, are added to ensure ergodicity, but the acceptance ratio for these moves becomes exponentially small for large lattice sizes. Here we will describe a way to construct a wormlike algorithm to perform nonlocal moves for this model.

The cluster algorithm [11–13] we propose is similar in spirit to worm algorithms [14,15] in the sense that we update the link currents by moving a “worm” through the lattice visiting the sites \( s_i = (\mathbf{r}, \tau) \). The links through which the worm passes are updated during its construction. At a given site, the links with \( \sigma \) equal to \( x, y, \tau \) are called outgoing links and those with \( \sigma \) equal to \( \tau^x, \tau^y, \tau \) are incoming links. When the worm is moving through the lattice, the currents \( J^\sigma_{s_i} \) are updated in the following manner: if the worm is leaving the site \( s_i \) along an outgoing link, we increment the corresponding current,

\[
J^\sigma_{s_i} \rightarrow J^\sigma_{s_i} = J^\sigma_{s_i} + 1, \quad \sigma = x, y, \tau.
\]
If the worm is leaving the site \(s_i\) along an incoming link, we decrement the corresponding current, 
\[
J^\sigma_{s_i} \rightarrow J^\sigma_{s_i} = J^\sigma_{s_i} - 1, \quad \sigma = -x, -y, -\tau. \tag{4}
\]

The construction of the worm starts with the choice of a random initial site \(s_1 \equiv (r_1, \tau_1)\) in the space-time lattice. Then the algorithm can be decomposed in two steps. (i) The worm moves to one of the six neighboring sites. To decide which direction to go from a site \(s_i \equiv (r_i, \tau_i)\), we calculate for all directions \(\sigma = \pm x, \pm y, \pm \tau\) weights, \(A^\sigma_{s_i}\), according to local detailed balance. A good choice is 
\[
A^\sigma_{s_i} = \min (1, \exp (-\Delta E^\sigma_{s_i}/K)), \quad \Delta E^\sigma_{s_i} = E^\sigma_{s_i} - E^\sigma_{s_i}. \tag{5}
\]

Here \(E^\sigma_{s_i} \equiv \frac{1}{2} (J^\sigma_{s_i} - \mu)^2 - \mu J^\sigma_{s_i} \delta_{\sigma, \pm 1}\) is the contribution to the total energy from the link \(J^\sigma_{s_i}\), before the worm moves through it. \(E^\sigma_{s_i}\) is the energy contribution with \(J^\sigma_{s_i}\) replaced by \(J^\sigma_{s_i}\). By normalizing the \(A^\sigma_{s_i}\)'s, we define the probabilities \(p^\sigma_{s_i} \equiv A^\sigma_{s_i}/N_{s_i}\), where \(N_{s_i} \equiv \sum_{\sigma} A^\sigma_{s_i}\). A direction \(\sigma\) is then chosen according to these probabilities. (ii) Once \(\sigma\) is chosen, we update the corresponding \(J^\sigma_{s_i}\) according to the above rules, Eqs. (3) and (4), and extend the worm to the new lattice site \(s_{i+1}\). (i) and (ii) are then repeated until the worm passes through the initial site where \(s_{i+1} = s_1\). Finally, in order to satisfy detailed balance, we have to erase the worm with a probability determined in the following way. If \(N(\text{worm})\) and \(N(\text{no worm})\) are the normalization of the probabilities at the initial site \(s_1\) with and without the worm present, then we erase the constructed worm with a probability 
\[
P^e = 1 - \min \left(1, \frac{N(\text{no worm})}{N(\text{worm})} \right). \tag{6}
\]

Under most conditions, this probability is very small. Several points are noteworthy about this algorithm. First of all, the configurations generated during the construction of the worm are not valid (\(\nabla \cdot J \neq 0\)). However, once the construction of the worm is finished and the path of the worm closed, the divergenceless constraint is satisfied. Secondly, when the worm moves through the lattice it may pass many times through the same link and cross itself before it reaches the initial site where the construction terminates. Hence, it is crucial that the current variables are updated during the construction of the worm. Finally, at each step \(i\) in the construction of the worm it is likely that the worm at the site \(s_i\) will partially “erase” itself by choosing to go back to the site \(s_{i-1}\) visited immediately before, thereby “bouncing” off the site \(s_i\).

Now we turn to the proof of detailed balance for the algorithm. Let us consider the case where the worm, \(w\), visits the sites \(\{s_1, \ldots, s_N\}\), where \(s_1\) is the initial site. The worm then goes through the corresponding link variables \(\{l_1, \ldots, l_N\}\), with \(l_i\) connecting \(s_i\) and \(s_{i+1}\). Note that \(s_N\) is the last site visited before the worm reaches \(s_1\). Hence, \(s_N\) and \(s_1\) are connected by the link \(l_N\). The total probability for constructing the worm \(w\) is then given by 
\[
P_w = P_{s_1}(1 - P^e_{s_1} N_{s_1}) \prod_{i=1}^{N-1} \left(1 - P^e_{s_i} N_{s_i} \right) \exp(-\Delta E_\text{tot}/K), \tag{7}
\]

where \(\Delta E_\text{tot}\) is the total energy difference between a configuration with and without the worm \(w\) present. Now we consider 
\[
P^e = 1 - \min(1, N_{s_1}(\text{no worm})/N_{s_1}(\text{worm})).
\]

Here, \(N_{s_1}(\text{no worm})\) is equal to \(N_{s_1}(\text{antiworm})\), and \(N_{s_1}(\text{no antiworm})=N_{s_1}\) is equal to \(N_{s_1}(\text{worm})\). Hence, we find for the
probability to erase the worm \( P_w^c = 1 - \min(1, N_{s_j}/\bar{N}_{s_j}) \), and
\( P_w^e = 1 - \min(1, \bar{N}_{s_j}/N_{s_j}) \) for erasing the antiworm. With this choice of \( P_w^c \), we satisfy detailed balance since \( P_w/P_w^c = \exp(-\Delta E_{\text{tot}}/K) \). Ergodicity is simply proven as the worm can perform local loops and wind around the lattice in any direction, as in the conventional algorithm.

To demonstrate the efficiency of the proposed algorithm, we have calculated autocorrelation times for different lattice sizes for the worm algorithm and the conventional algorithm. For an observable \( \mathcal{O} \) we define the autocorrelation function and the autocorrelation time \( \tau_{\mathcal{O}} \) in the usual manner [16].

\[
\frac{\langle \mathcal{O}(0)\mathcal{O}(t) \rangle - \langle \mathcal{O} \rangle^2}{\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2} = a e^{-\tau_{\mathcal{O}}} + b e^{-\tau_{\mathcal{O}}} + \cdots. \tag{8}
\]

Here, \( t \) is the Monte Carlo time measured in Monte Carlo sweeps (MCS), with 1 MCS corresponding to \( L^d \) attempted updates. The autocorrelation function is calculated from simulations with \( 10^6 \) MCS, and to obtain the best estimate of \( \tau_{\mathcal{O}} \) we always fit to the indicated double-exponential form with \( \tau_{\mathcal{O}} \approx \tau_{\mathcal{O}} \). To make a fair comparison of \( \tau_{\mathcal{O}} \) for the two algorithms, one customarily [12,16] multiplies \( \tau_{\mathcal{O}} \) for the worm algorithm by \( N/<l> \), with \( <l> \) the mean number of links in a worm and \( N = 3L^3 \). With this rescaling we show in Fig. 1 the autocorrelation times, \( \tau_{\rho} \), for the stiffness (see the exact definition below) at \( \mu = 0 \) for both algorithms. The calculations have been performed on cubic lattices at \( K = 0.333 \), very near previous estimates of the critical point [9]. For the worm algorithm we also show the autocorrelation time for the energy, \( \tau_{E_k} \), which is almost identical to \( \tau_{\rho} \). The autocorrelation times increase dramatically with system size for the conventional algorithm, whereas they remain very small (of the order of 2–3 MCS per link) for the worm algorithm. If we fit the \( L \) dependence of \( \tau_{\rho} \sim L^{z_{MC}} \) with a power law, we obtain an autocorrelation exponent \( z_{MC} \) larger than 4 for the conventional algorithm. For the conventional algorithm, it is likely that \( \tau_{\rho} \) is diverging exponentially with \( L \) since \( \rho \) is solely determined by global updates for which the acceptance probability decreases exponentially with \( L \) for the worm algorithm, we find a very small \( z_{MC} \approx 0.3 \).

We now present results for the model Eq. (2) at \( \mu = 0 \). There, the model is expected to undergo a transition in the \((2+1)D\) XY universality class [1,9] from a superfluid into a Mott insulating phase with a dynamical critical exponent \( z = 1 \). The different phases can be distinguished by calculating the stiffness defined as [8]

\[
\rho = \frac{1}{L/L^2} \left( \sum_{\tau} J_{\tau,\tau} \right)^2. \tag{9}
\]

Since we expect \( z = 1 \), we use \( L_{x} \), the system size in the third direction, equal to \( L \). To obtain the \( K \) dependence of \( \rho \), we have used reweighting techniques [17] on large runs (of the order of \( 10^8 \) MCS) at \( K = 0.333 \). The error bars are determined using jackknife techniques [16]. Using finite-size scaling relations, the quantity \( \rho L^z \) is expected to be independent of system size at the critical point [8], \( K_c \). Moreover, \( L^2 d\rho/dK \) is expected to diverge at \( K_c \) as \( L^{1/v} \) where \( v \) is the correlation length exponent. We have explicitly calculated this quantity by evaluating the thermodynamic derivative of
We thank M. Troyer and H. Rieger for useful discussions and IDRIS (Paris), CALMIP (Toulouse), and SHARCNET (Hamilton) for time allocations on their computer systems.