Appendix E

Long-range interactions

Most examples discussed in this book are models with on-site or nearest-neighbor interactions. Physical systems, however, often have interactions reaching beyond one lattice constant, and there are cases where such long-range interactions alter the essential properties of the system. A classic example is the dipolar interaction in magnets. Because of the long-range dipolar interaction, magnetic domains form, reducing the total magnetization. Another more recent example is the effect of dipolar interactions among ultra-cold atoms in an optical lattice. Because of the much larger lattice constant in such systems, the magnitude of the dipolar interaction relative to the nearest-neighbor interaction is enhanced compared with a conventional solid. Frustrated systems are other good examples, since for these the short-range interactions sometimes cancel, making the middle/long-range interactions more relevant.

From the computational point of view, long-range interactions generally increase the computational complexity of Monte Carlo simulations. Consider, for example, the Ising model

$$H = -\sum_{ij} J(R_{ij}) s_i s_j,$$

where $J(R_{ij})$ is some decreasing function of the distance between two sites R_{ij}. If we adopt the single-spin update, every time we attempt flipping a spin we must compute the molecular field produced by all the other spins. This requires $\mathcal{O}(N)$ operations for a lattice of N sites. Therefore, every Monte Carlo sweep takes a computational time proportional to N^2 instead of N, as in the case of the short-range interaction models. A neat trick exists to reduce this N^2 dependence to NM, where M is the effective number of interacting spins defined by

$$M \equiv a^{-d} \int d^d R\, J(R).$$

https://doi.org/10.1017/CBO9780511902581.019 Published online by Cambridge University Press

If M is bounded from above by some constant that is independent of the system size, this trick reduces the computational complexity back to $\mathcal{O}(N)$. Even if M diverges in the thermodynamic limit, reducing the complexity from N^2 to NM can still be a major gain. For example, if the coupling constant $J(R)$ is inversely proportional to R, M is proportional to L^{d-1} if L is the linear size and the dimension is $d \geq 3$. In this case, $NM \propto L^{2d-1}$, which is less than N^2 by a factor L.

As a first method and specific example, let us discuss the approach originally proposed by Luijten and Blöte (1995), which is based on a look-up table. We consider the Swendsen-Wang cluster algorithm for an Ising model with long-range interactions. The system has N sites. In the following, we first compute the probability for placing a bond assuming that all the spins are parallel, and then, before we actually place a bond, we examine whether the spins are parallel. (If not, we do not place a bond, of course.) The probability of having a bond between the p-th ($p = 1, 2, \ldots, N(N-1)/2$) pair of spins can be written as

$$P_{\text{bond}}(p) = 1 - e^{-2K_p}, \tag{E.1}$$

where K_p describes the interaction strength for the pair p. For example, if p is the pair (i, j), $K_p \equiv \beta J_{ij}$. Now consider a sequence of $N_{\text{pair}} \equiv N(N-1)/2$ boxes, each representing a pair of sites, and suppose that a ball is placed in each box with the corresponding probability (E.1). The probability that no ball is put in any box between the m-th and the n-th becomes

$$P_{\text{no bond}}(m; n) = \prod_{p=m+1}^{n-1} e^{-2K_p} = \exp\left[-2 \sum_{p=m+1}^{n-1} K_p\right]. \tag{E.2}$$

If we define an array ("look-up table") A of length N_{pair} with elements

$$A_0 = 1, \quad A_n = \prod_{p=1,\ldots,n} e^{-2K_p} \quad \left(n = 1, \ldots, N_{\text{pair}}\right), \tag{E.3}$$

then (E.2) can be written as

$$P_{\text{no bond}}(m; n) = \frac{A_{n-1}}{A_m}.$$

The pairs that are to be connected can then be calculated as follows: A random number $\zeta_1 \in [0, 1)$ is chosen and the array A is searched for the first index, say, p_1, such that $A_{p_1} \leq \zeta_1$. For the calculation of the next bond, the values of A must be divided by A_{p_1}, or equivalently, the random number $\zeta_2 \in [0, 1)$ multiplied by A_{p_1}. The site number p_2 of the second connected spin is the first index, such that $A_{p_2} \leq A_{p_1}\zeta_2$. This process is continued until $A_{p_{k-1}}\zeta_k < A_{N_{\text{pair}}}$. As was already mentioned, the outlined procedure assumes that all spins are aligned. Before actually inserting

https://doi.org/10.1017/CBO9780511902581.019 Published online by Cambridge University Press

Algorithm 49 Cluster algorithm with long-range interactions: look-up table approach.

Input: N_{pair}, array A defined in (E.3). l , an initially empty list of connected pairs, $p = 0$.
loop
Draw random number $\zeta \in [0, 1)$;
$r = A_p \zeta$;
if $r < A_{N_{\text{pair}}}$ **then**
break ;
end if
Search the lowest index p such that $A_p \leq \zeta$;
If the spins connected by pair p are parallel, add p to the list l ;
end loop
return the list l of connected pairs.

a proposed bond, we have to check whether the two spins are parallel. Because $A_0 = 1$, the above algorithm corresponds to the pseudocode in Algorithm 49.

If the bisection method is used to search for the lowest index such that $A_p \leq \zeta$, the calculation time is of order $\mathcal{O}(\log N_{\text{pair}}) = \mathcal{O}(\log N)$ for each proposed bond. Since the number of bonds connecting to a given site is of order M, and we have N sites, the total calculation time for the bond placing is $O(NM \log N)$.

We can even get rid of the $\log N$ factor, although it may not matter in practical applications. To do this, let us regard a missing bond on a pair of interacting sites as a "pair survival." If the survival probability of a pair p is $e^{-\Gamma_p}$, we can interpret it as surviving the decay rate of unity for the duration Γ_p. ($\Gamma_p = 2\beta J_p$ in the Swendsen-Wang algorithm.) In the case of many pairs, we have many intervals with various durations Γ_p. Let us now consider a line segment of length $\Gamma \equiv \sum_{p=1}^{N_{\text{pairs}}} \Gamma_p$. Instead of judging for each individual interval whether pairs survive or not, we distribute uniformly and randomly decay events over this line segment with unit density. Specifically, we generate the total number of decay events by drawing an integer n from the Poisson distribution of mean Γ and distribute n events uniformly and randomly on the line. This procedure is like throwing n darts onto the line. As a result, some darts may or may not fall into the interval Γ_p. If we have no dart in Γ_p, the pair p survives (no bond assigned); otherwise, it is killed (a bond assigned). Given that the expectation value of n is proportional to Γ, the computational cost for generating n bonds is of order Γ. In the case of the Swendsen-Wang algorithm this cost is proportional to $\sum_{ij} J_{ij}$ and hence $O(NM)$.

Finding the intervals hit by any dart is less trivial. We carry out the task n times (for each of the randomly distributed darts). Generating a uniform random number

https://doi.org/10.1017/CBO9780511902581.019 Published online by Cambridge University Press

is easy and costs $\mathcal{O}(1)$ for each dart thrown. The difficult part is finding the index p of the interval hit. Using the technique discussed in Appendix A we can do this in a time $\mathcal{O}(1)$ (Knuth, 1997; Walker, 1977; Fukui and Todo, 2009). As a result, throwing darts and judging survival takes a computational time of $\mathcal{O}(\Gamma) \sim \mathcal{O}(NM)$ in total.

The application of these ideas to quantum simulations is rather straightforward. For example, in the directed-loop algorithm for a long-range interaction model, we generate the total number of vertices n by the Poisson distribution with mean Γ, defined in this case as $\Gamma = \sum_l \Gamma_l$ with

$$\Gamma_l \equiv \beta \max_{\psi} \langle \psi | - H_l | \psi \rangle. \tag{E.4}$$

In the placement of the vertices, the algorithm, however, differs from the general procedure described above. There, the number of darts n_l on the interval Γ_l did not matter as long as $n_l \neq 0$. Here, it matters since n_l is the number of the vertices we assign to the interacting spins involved in H_l. For each of the n_l vertices, we generate an imaginary time in the interval $\tau \in [0, \beta]$ and with probability

$$P_{\text{accept}} = \Gamma_l^{-1} \beta \langle \psi(\tau) | - H_l | \psi(\tau) \rangle$$

decide whether we place the vertex there. In this way, we compensate for the overestimate of the graph assignment density made in (E.4) by using the maximum value.

https://doi.org/10.1017/CBO9780511902581.019 Published online by Cambridge University Press