Parallel Wolff Cluster Algorithm for n-Component Vector Spin Models

J. Kaupuzs, R.V.N. Melnik, and J. Rimšans

Full Bibliographic Reference for this paper

J. Kaupuzs, R.V.N. Melnik, and J. Rimšans, "Parallel Wolff Cluster Algorithm for n-Component Vector Spin Models", in , (Editors), "Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering", Civil-Comp Press, Stirlingshire, UK, Paper 72, 2011.
doi:10.4203/ccp.95.72

Keywords: parallel cluster algorithms, lattice spin models, Monte Carlo simulations, Goldstone mode singularities, correlation functions, power law.

Summary

In Monte Carlo (MC) simulations of the lattice models near criticality, the cluster algorithms provide an indispensable tool because the Metropolis algorithm suffers from the problem of critical slowing down. Among such cluster algorithms, the Swendsen-Wang and Wolff cluster algorithms are well known [1]. First, we generalize the parallel Wolff cluster algorithm for spin models, often called n-vector models or O(n) models, where spins are n-component vectors (the Ising model is a particular case of this consideration where n = 1). We note that for many applications, such algorithms must be developed in the presence of an external field. Therefore, we consider then a modified Wolff algorithm, which can be applied to spin systems in the presence of an external field. A parallel Open MP version of the standard Wolff algorithm for the three-dimensional Ising model has been developed earlier [2]. Here we generalize this parallel algorithm for spin models, often called *n*-vector models or *O(n)* models, where spins are n-component vectors. The Ising model is included here as a particular case of *n=1*. Further on, we elaborate a parallel version of the modified Wolff algorithm in order to simulate *O(n)* models below the critical temperature in the presence of the external field and to study the Goldstone mode effects. A parallel code for this algorithm has been developed and tested, allowing us to speed up the simulation (up to 3 times on 4 processors), as well as to treat larger lattices due to greater operative (shared) memory available. Using this code, we have simulated the *n*-component vector model for *n=2,4,10* and have analysed the Goldstone mode singularities for the transverse and the longitudinal Fourier-transformed two-point correlation functions. The simulation results for *n=10* at several linear lattices sizes *L=192,256,384* and small values of the external field *h=0.000875,0.0004375,0.00021875* are discussed in some detail to test the theoretical predictions for the power-law Goldstone mode singularities of the correlation functions. The results obtained provide good numerical evidence for the theoretically expected power-law divergence of the transverse correlation function at small wave vectors. We have found that by fitting the Monte Carlo data that this divergence is described by the exponent about -1.97 with an uncertainty of order 0.01 in this value. The longitudinal correlation function is analysed, as well. Generally, the simulation results show that the transverse correlation function has remarkably smaller finite-size and finite-*h* effects, as well as a better power-law scaling in certain finite range of the wave vector magnitude *k*, as compared to the longitudinal correlation function.

References

[1] U. Wolff, "Collective Monte Carlo updating for spin systems", Phys. Rev. Lett., 62, 361-364, 1989. doi:10.1103/PhysRevLett.62.361

[2]J. Kaupuzs, R.V.N. Melnik, J. Rimšans, "Parallelization of the Wolff single-cluster algorithm", Phys. Rev. E, 81, 026701, 2010. doi:10.1103/PhysRevE.81.026701

Parallel Wolff cluster algorithm for n-component vector spin models

J. Kaupužs[1,2] R. V. N. Melnik[3], J. Rimšāns[1,2]
[1]Institute of Mathematics and Computer Science,
University of Latvia,
29 Raiņa Boulevard, LV–1459 Riga, Latvia
[2]Institute of Mathematical Sciences and Information Technologies,
University of Liepaja,
14 Liela Street, LV3401 Liepaja, Latvia
[3]Wilfrid Laurier University, Waterloo, Ontario, Canada, N2L 3C5

The cluster algorithms are very useful in Monte Carlo (MC) simulations of the lattice models near criticality, where the usual Metropolis algorithm suffers from the problem of critical slowing down. The Swendsen-Wang and Wolff cluster algorithms [1, 2, 3] are well known. In distinction from the Metropolis algorithm, they ensure reasonably small autocorrelation times for large but finite lattices even at the critical point [4, 5, 6, 7]. A parallel (Open MP) implementation of the Wolff single cluster algorithm has already been developed and tested for the three–dimensional (3D) Ising model [8]. Here we generalize this parallel algorithm for spin models, often called n-vector models or $O(n)$ models, where spins are n-component vectors. Note that the Ising model is a particular case of $n = 1$. A modified Wolff algorithm for n-vector spin models in presence of the external field is of great interest, too (see, e. g., [9]). Here we develop a parallel version also for this case.

According to [1], one step of the Wolff single cluster algorithm for the n-vector model consists of the following substeps:

1. Choose at random a unit vector $\mathbf{r}$ in the n-dimensional space of spin-vectors (this step is not necessary for $n = 1$, where $\mathbf{r} \to r = 1$).

2. Choose a seed spin $\mathbf{s}$ of the new cluster at random and make its mirror-reflection $\mathbf{s} \to \mathbf{s} - 2\mathbf{r}(\mathbf{rs})$ (further called flipping) with respect to the plane, which is orthogonal to $\mathbf{r}$.

3. Look in turn at each of the neighbors of that spin and find the ones, the projection of which on $\mathbf{r}$ has an opposite sign than that of the flipped seed spin. Each of them is added to the cluster and simultaneously flipped with probability $P_{\rm add} = 1 - e^{\beta \Delta E}$, where ΔE is the change in the interaction energy between two considered spins at the flipping.

4. For each spin that was added in the last step, examine each of its neighbors to find the ones for which the projection on $\mathbf{r}$ has opposite sign, adding each of them to the cluster and simultaneously flipping with probability $P_{\rm add} = 1 - e^{\beta \Delta E}$. This step is repeated until there are no new spins added to the cluster.

A new generation of spins is added to the growing Wolff cluster at each iteration of the step 4. Our basic idea is to perform any of such iterations by using parallel threads, provided that the wavefront of the growing cluster, consisting of the spins added in the last step, occupies more than N_{min} lattice sites. We will call these spins the wavefront–spins. If the wavefront contains $\leq N_{min}$ spins, then it is treated serially by the master processor. Here N_{min} is an optimization parameter.

In the case of simple cubic lattice, the parallel treatment of one iteration of step 4 is performed according to the following scheme.

1. Divide the list of the wavefront coordinates (coordinates of those spins added in the last step) between the processors. Three arrays with these x, y, and z coordinates, as well as the array of spin variables are stored in the shared memory.

2. Perform step 4 of the Wolff algorithm in parallel only for the subset of those neighboring spins, which are located in one of the 6 possible directions from each of the considered wavefront–spins. The work of processors is synchronized by putting the OMP BARRIER after this substep. Then the same is performed for the remaining 5 directions, putting the OMP BARRIER after each of them. Each processor treats certain fraction of the cluster wavefront, assigned in step 1 of this scheme. Besides, each processor forms its own lists of x, y and z coordinates of newly added spins and counts the number of elements in these lists. These lists and numbers of elements are shared variables, however, each processor stores them in a separate subdomain of the shared memory.

3. Form the common lists of the x, y, z coordinates of newly added spins. It is done in parallel, in such a way that each processor writes its own

list in a certain place of the common shared arrays, determined according to the values of shared variables (numbers of elements) defined in the previous step. The total number of elements in the common lists is determined by the master processor.

The standard Wolff algorithm is designed for Monte Carlo simulations in zero external field $h = 0$. In the modified Wolff algorithm, the external field is introduced as an auxiliary spin of unit length $\mathbf{e}$ which interacts with all other spins by the coupling constant $\mid h \mid \beta$. This spin is treated on the same grounds as other ones, whereas the longitudinal component of spin $\mathbf{s}$ is its projection on the auxiliary spin $\mathbf{e}$.

The parallelization is performed similarly as described before. The only difference is that the auxiliary spin has to be included as a candidate for inclusion into the cluster. In a fraction of cases, it can appear either as the seed spin, or as one of the wave-front spins at the iteration of step 4. In our formulations of the algorithm, the auxiliary spin is a neighbour of all other lattice spins. Thus, in the above cases, one has to examine all these spins for possible adding to the cluster through the auxiliary spin. Therefore, the parallelization of the modified Wolff algorithm contains a parallel realization of this procedure as an extra step. It is done by dividing the whole lattice in slices treated by different processors.

We have elaborated and tested a parallel code of the modified Wolff algorithm for the n-vector model with n as an input parameter. The simulations have been performed for $n = 2, 4, 10$ in order to study the Goldstone mode effects in these models. The parallel code allows us to reach the speedup about 3 times on 4 processors, as well as to simulate larger lattices due to greater operative (shared) memory available.

References

[1] U. Wolff, Phys. Rev. Lett. **62** (1989) 361

[2] J -S. Wang, R. H. Swendsen, Physica A Vol. 167, pp. 565-679

[3] M. E. J. Newman, G. T. Barkema, Monte Carlo Methods in Statistical Physics, Clarendon Press, Oxford, 1999

[4] S. Gündüç , M. Dilaver, M. Aydin, Y. Gündüç, Computer Physics Communications **166**, 1 (2005)

[5] J. Du, B. Zheng, J.-S. Wang, J. Stat. Mech., P05004 (2006)

[6] D. Ivaneyko, J. Ilnytskyi, B. Berche, Yu. Holovatch, Physica A **370**, 163 (2006)

[7] U. K. Rößler, Phys. Rev. B **59**, 13577 (1999)

[8] J. Kaupužs, R. V. N. Melnik, J. Rimšāns, Phys. Rev. E **81**, 026701 (2010)

[9] J. Kaupužs, R. V. N. Melnik, J. Rimšāns, Physics Letters A **374**, 1943 (2010)