Two- and three-point functions at criticality: Monte Carlo simulations of the improved three-dimensional Blume-Capel model

Martin Hasenbusch*
Institut für Physik, Humboldt-Universität zu Berlin, Newtonstr. 15, 12489 Berlin, Germany

(Received 8 December 2017; published 16 January 2018)

We compute two- and three-point functions at criticality for the three-dimensional Ising universality class. To this end, we simulate the improved Blume-Capel model at the critical temperature on lattices of a linear size up to $L = 1600$. As a check, also simulations of the spin-$\frac{1}{2}$ Ising model are performed. We find $f_{\sigma\sigma\epsilon} = 1.051(1)$ and $f_{\epsilon\epsilon\epsilon} = 1.533(5)$ for operator product expansion coefficients. These results are consistent with but less precise than those recently obtained by using the bootstrap method. An important ingredient in our simulations is a variance reduced estimator of N-point functions. Finite size corrections vanish with $L^{-\Delta_\epsilon}$, where L is the linear size of the lattice and Δ_ϵ is the scaling dimension of the leading Z_2-even scalar ϵ.

DOI: 10.1103/PhysRevE.97.012119

I. INTRODUCTION

Recently, we have seen enormous progress in the understanding of critical phenomena in three dimensions by using the conformal bootstrap approach [1–5], to give only a few references. For a recent lecture note on the subject, see [6]. In particular, precise numbers for scaling dimensions and operator product expansion (OPE) coefficients for the universality class of the three-dimensional Ising model were obtained. For a very detailed account, see Ref. [5].

The functional form of two-point functions of primary operators is fixed by conformal invariance

$$\langle \mathcal{O}_1(x_1)\mathcal{O}_2(x_2)\rangle = \frac{C_1\delta_{\Delta_1,\Delta_2}}{|x_1 - x_2|^{2\Delta_1}}, \tag{1}$$

where $\mathcal{O}_i$ is the operator taken at the site x_i and Δ_i is its scaling dimension. The scaling dimensions of the leading Z_2-odd scalar σ and the leading Z_2-even scalar ϵ for the three-dimensional Ising universality class are $\Delta_\sigma = 0.5181489(10)$ and $\Delta_\epsilon = 1.412625(10)$, respectively [4,5]. The critical exponents that are usually discussed in critical phenomena can be deduced from these two scaling dimensions. Let us consider two examples: The exponent $\eta = d + 2 - 2y_h = d + 2 - 2(d - \Delta_\sigma) = 2\Delta_\sigma - d + 2 = 0.0362978(20)$ governs the decay of the spin-spin correlation function at the critical point, where y_h is the renormalization group (RG) exponent related to the external field and d is the dimension of the system. The critical exponent of the correlation length is given by $\nu = 1/y_t = 1/(d - \Delta_\epsilon) = 0.6299709(40)$, where y_t is the RG exponent related to the temperature. In Table 2 of [5] one finds $\omega = \Delta_{\epsilon'} - 3 = 0.82968(23)$ for the exponent of the leading correction. These results are by far more accurate than previous ones obtained by other methods. For example, using Monte Carlo simulations of the improved Blume-Capel model $\nu = 0.63002(10)$, $\eta = 0.03627(10)$, and $\omega = 0.832(6)$ had been obtained [7]. Improved means that leading corrections to scaling are substantially reduced (see Sec. II below). Results obtained from high-temperature series expansions (see, for example, Refs. [8,9]) are slightly less accurate than those from Monte Carlo simulations. Field theoretic methods (see, for example, Ref. [10]) give even less precise estimates. In our numerical study, corrections caused by the breaking of the rotational symmetry are an important issue. These are governed by the correction exponent ω_{NR}. In Table I of Ref. [8] the authors quote $\omega_{NR} = 2.0208(12)$, which is in reasonable agreement with $\omega_{NR} = 2.022665(28)$ that follows from $\Delta = 5.022665(28)$ for angular momentum $l = 4$ given in Table 2 of Ref. [5]. In the case of improved models, these corrections should be the dominant ones since following Ref. [5] $\Delta_{\epsilon''} = 6.8956(43)$ corresponding to $\omega_2 = 3.8956(43)$. Note that this finding contradicts $\omega_2 = 1.67(11)$ obtained in [11]. For a discussion of various results for ω_2 given in the literature, see Sec. 4.2 of [1].

There is a very rich literature on critical phenomena, which we can not recapitulate here. We refer the reader to reviews of the subject [12–15].

Now, let us turn to the OPE coefficients. Also, the form of three-point functions is fixed by conformal invariance. Normalizing the operators such that $C_i = 1$ [Eq. (1)], one gets [16]

$$\langle \mathcal{O}_1(x_1)\mathcal{O}_2(x_2)\mathcal{O}_3(x_3)\rangle = \frac{f_{123}}{|x_1 - x_2|^{\Delta_1+\Delta_2-\Delta_3}|x_2 - x_3|^{\Delta_2+\Delta_3-\Delta_1}|x_3 - x_1|^{\Delta_3+\Delta_1-\Delta_2}}, \tag{2}$$

where the OPE coefficients f_{123} depend on the universality class. Using the conformal bootstrap method, highly accurate results for OPE coefficients of the Ising universality class in three dimensions were obtained [4,5]:

$$f_{\sigma\sigma\epsilon} = 1.0518537(41) \tag{3}$$

and

$$f_{\epsilon\epsilon\epsilon} = 1.532435(19). \tag{4}$$

*martin.hasenbusch@physik.hu-berlin.de

TABLE I. Statistics of our simulations of the Blume-Capel model at $D = 0.655$ and $\beta = 0.387721735$. For a discussion, see the text.

L	$n_s = 2$	$n_s = 4$
400	239510	74270
600	93150	17810
800	39920	15670
1200	10300	1800
1600	4190	1630

Until recently, there had been no results obtained by other methods that could be compared with Eqs. (3) and (4). In particular, in Monte Carlo simulations of lattice models, it is unclear how an infinite volume at the critical point could be well approximated. Furthermore, one should notice that the conformal invariance is broken by the lattice. Hence, on the lattice, Eqs. (1) and (2) are affected by corrections that decrease with increasing distance between the points.

In Ref. [17], the requirement of an infinite volume was circumvented by studying two-point correlators at the critical temperature applying a finite external field h. The OPE coefficients are obtained from the h dependence of the two-point correlators. For details of the method, we refer the reader to Ref. [17]. Simulating the Ising model on a simple cubic lattice, the authors find

$$f_{\sigma\sigma\epsilon} = 1.07(3),\ f_{\epsilon\epsilon\epsilon} = 1.45(30). \tag{5}$$

These estimates were improved by using a trapping potential in Ref. [18]:

$$f_{\sigma\sigma\epsilon} = 1.051(3),\ f_{\epsilon\epsilon\epsilon} = 1.32(15). \tag{6}$$

Very recently, Herdeiro [19] computed two- and three-point functions for the Ising model on the simple cubic lattice at the critical point, using the so called UV-sampler method. The method had been tested at the example of the Ising model on the square lattice [20]. Fitting his data for the three-point function with a one parameter *Ansatz*, fixing Δ_σ and Δ_ϵ to their bootstrap values, Herdeiro gets

$$f_{\sigma\sigma\epsilon} = 1.05037(152)[398],\ f_{\epsilon\epsilon\epsilon} = 1.5508(62)[176]. \tag{7}$$

The numbers are taken from Table 1 of Ref. [20]. For the meaning of the error bars, see Ref. [20].

Here, we shall follow a different strategy. We simulated the improved Blume-Capel model at the critical point on a simple cubic lattice with periodic boundary conditions. To keep finite size effects small, we simulated lattices of a linear size up to $L = 1600$. Furthermore, an extrapolation in the lattice size is performed. It turns out that finite size effects at the critical point are $\propto L^{-\Delta_\epsilon}$ for all quantities that we study here. We employ variance reduction as proposed in Refs. [21,22]. This is in particular helpful in the case of the $\epsilon\epsilon\epsilon$ function.

The outline of the paper is the following: In the next section we recall the definition of the Blume-Capel model and summarize briefly the results of Ref. [7]. Then, we define the two- and three-point functions that we measure. Then, in Sec. IV we discuss the finite size scaling behavior of the quantities that we study. In Sec. V we discuss the application of the variance reduction method to our problem. Next, we discuss the simulations that we perform. It follows the analysis of the data. Finally, we conclude and give an outlook.

II. MODEL

As in previous work, we study the Blume-Capel model on the simple cubic lattice. For a vanishing external field, it is defined by the reduced Hamiltonian

$$H = -\beta \sum_{\langle xy \rangle} s_x s_y + D \sum_x s_x^2, \tag{8}$$

where the spin might assume the values $s_x \in \{-1,0,1\}$. $x = (x^{(0)},x^{(1)},x^{(2)})$ denotes a site on the simple cubic lattice, where $x^{(i)} \in \{0,1,\ldots,L_i - 1\}$, and $\langle xy \rangle$ denotes a pair of nearest neighbors on the lattice. In this study, we consider $L_0 = L_1 = L_2 = L$ throughout. The inverse temperature is denoted by $\beta = 1/k_B T$. The partition function is given by $Z = \sum_{\{s\}} \exp(-H)$, where the sum runs over all spin configurations. The parameter D controls the density of vacancies $s_x = 0$. In the limit $D \to -\infty$, vacancies are completely suppressed and hence the spin-$\frac{1}{2}$ Ising model is recovered.

In $d \gtrsim 2$ dimensions, the model undergoes a continuous phase transition for $-\infty \leqslant D < D_{\rm tri}$ at a β_c that depends on D, while for $D > D_{\rm tri}$ the model undergoes a first order phase transition, where $D_{\rm tri} = 2.0313(4)$ for $d = 3$ (see Ref. [23]). Numerically, using Monte Carlo simulations it has been shown that there is a point $(D^*,\beta_c(D^*))$ on the line of second order phase transitions, where the amplitude of leading corrections to scaling vanishes. We refer to the Blume-Capel model at values of D that are good numerical approximations of D^* as improved Blume-Capel model. For a more general discussion of improved models, see for example Sec. 3.5 of [24] or Sec. 2.3.1 of [15]. In [7] we simulated the model at $D = 0.655$ close to β_c on lattices of a linear size up to $L = 360$. We obtained $\beta_c(0.655) = 0.387721735(25)$ and $D^* = 0.656(20)$. The amplitude of leading corrections to scaling at $D = 0.655$ is at least by a factor of 30 smaller than for the spin-$\frac{1}{2}$ Ising model.

Here, we simulate the Blume-Capel model at $D = 0.655$. Most of our simulations are performed at $\beta = 0.387721735$. In order to check the sensitivity of the results on β, we performed in addition a few simulations at $\beta = 0.38772$ and 0.38772347. In order to check the effect of leading corrections to scaling, we also simulated the spin-$\frac{1}{2}$ Ising model on the simple cubic lattice at $\beta = 0.22165462$. Note that in Eq. (A2) of Ref. [25] we quote $\beta_c = 0.22165462(2)$.

III. OBSERVABLES

On the lattice we identify

$$\epsilon(x) = s_x^2 - \langle s_x^2 \rangle + \cdots, \tag{9}$$

$$\sigma(x) = s_x + \cdots, \tag{10}$$

where in our numerical study $\langle s_x^2 \rangle$ is replaced by its estimate obtained from the given simulation at finite L. Corrections are caused by fields with the same symmetry properties but higher dimensions.

In the case of the Ising model, Eq. (9) makes no sense. In the literature $s_x s_y$, where x and y are nearest neighbors, is used

instead of s_x^2. Here, motivated by Eq. (33) below, we used S_x^2, where $S_x = \sum_{y.nn.x} s_y$, where $y.nn.x$ means that y is a nearest neighbor of x.

In order to keep the study tractable, we have to single out a few directions for the displacements between the points. In the case of the two-point functions, we consider displacements along the axes

$$x_2 - x_1 = (j,0,0),\ x_2 - x_1 = (0,j,0),\ x_2 - x_1 = (0,0,j), \tag{11}$$

the face diagonals

$$\begin{aligned} &x_2 - x_1 = (j,j,0),\ x_2 - x_1 = (j,0,j),\ x_2 - x_1 = (0,j,j),\\ &x_2 - x_1 = (j,-j,0),\ x_2 - x_1 = (j,0,-j),\\ &x_2 - x_1 = (0,j,-j), \end{aligned} \tag{12}$$

and space diagonals

$$\begin{aligned} &x_2 - x_1 = (j,j,j),\ x_2 - x_1 = (j,j,-j),\\ &x_2 - x_1 = (j,-j,j),\ x_2 - x_1 = (j,-j,-j), \end{aligned} \tag{13}$$

where j is an integer. In the following, we shall indicate these three directions by axis (a), face diagonal (f), and space diagonal (d), respectively. In our simulation program we summed over all choices that are related by symmetry to reduce the statistical error. In particular, we summed over all possible choices of x_1. In the following, we shall denote the two-point function by $g_{r,\mathcal{O}_1,\mathcal{O}_2}(x)$, where $r \in \{a,f,d\}$ gives the direction and $x = |x_1 - x_2|$ is the distance between the two points.

In the discussion of our numerical results, we are a bit sloppy with the notation and use g also for the numerical estimates obtained from the simulation. Hence, there might be also a dependence on the linear lattice size that is not indicated explicitly. In the case of the three-point functions

$$G_{r,\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3}(x) = \langle \mathcal{O}_1(x_1)\mathcal{O}_2(x_2)\mathcal{O}_3(x_3)\rangle, \tag{14}$$

we study the two choices $\mathcal{O}_1 = \mathcal{O}_2 = \sigma$ and $\mathcal{O}_3 = \epsilon$ and $\mathcal{O}_1 = \mathcal{O}_2 = \mathcal{O}_3 = \epsilon$. Furthermore, we consider two different geometries that are indicated by r. For $r = f$ the largest displacement is along a face diagonal. For example,

$$x_3 - x_1 = (j,0,0),\ x_3 - x_2 = (0,j,0). \tag{15}$$

Our second choice is indicated by $r = d$ and the largest displacement is along a space diagonal. For example,

$$x_3 - x_1 = (j,0,0),\ x_3 - x_2 = (0,j,j), \tag{16}$$

where j is integer and also here we sum in our simulation over all choices that are related by symmetry to reduce the statistical error. The argument x of G gives the largest distance between two points $x = |x_1 - x_2|$.

In order to eliminate the constants C_i [Eq. (1)] and the power law behavior from the three-point functions, we directly normalized our estimates of the three-point functions by the corresponding ones of two-point functions. For the direction $r = f$ we get

$$f_{\sigma\sigma\epsilon} \simeq 2^{-\Delta_\epsilon/2} \frac{\langle s_{x_1} s_{x_2} s_{x_3}^2\rangle - \langle s_{x_1} s_{x_2}\rangle \langle s^2\rangle}{\langle s_{x_1} s_{x_2}\rangle \left[\langle s_{x_1}^2 s_{x_3}^2\rangle - \langle s^2\rangle^2\right]^{1/2}} \tag{17}$$

and

$$f_{\epsilon\epsilon\epsilon} \simeq \frac{\langle s_{x_1}^2 s_{x_2}^2 s_{x_3}^2\rangle - \left[\langle s_{x_1}^2 s_{x_2}^2\rangle + \langle s_{x_1}^2 s_{x_3}^2\rangle + \langle s_{x_2}^2 s_{x_3}^2\rangle\right]\langle s^2\rangle + 2\langle s^2\rangle^3}{\left[\langle s_{x_1}^2 s_{x_3}^2\rangle - \langle s^2\rangle^2\right]\left[\langle s_{x_1}^2 s_{x_2}^2\rangle - \langle s^2\rangle^2\right]^{1/2}}, \tag{18}$$

where we used that $|x_1 - x_3| = |x_1 - x_2|$. For the direction $r = d$ we get

$$f_{\sigma\sigma\epsilon} \simeq 3^{-\Delta_\epsilon/2} \frac{\langle s_{x_1} s_{x_2} s_{x_3}^2\rangle - \langle s_{x_1} s_{x_2}\rangle \langle s^2\rangle}{\langle s_{x_1} s_{x_2}\rangle \left[\langle s_{x_2}^2 s_{x_3}^2\rangle - \langle s^2\rangle^2\right]^{1/2}} \tag{19}$$

and

$$f_{\epsilon\epsilon\epsilon} \simeq \frac{\langle s_{x_1}^2 s_{x_2}^2 s_{x_3}^2\rangle - \left[\langle s_{x_1}^2 s_{x_2}^2\rangle + \langle s_{x_1}^2 s_{x_3}^2\rangle + \langle s_{x_2}^2 s_{x_3}^2\rangle\right]\langle s^2\rangle + 2\langle s^2\rangle^3}{\left[\langle s_{x_1}^2 s_{x_3}^2\rangle - \langle s^2\rangle^2\right]^{1/2}\left[\langle s_{x_2}^2 s_{x_3}^2\rangle - \langle s^2\rangle^2\right]^{1/2}\left[\langle s_{x_1}^2 s_{x_2}^2\rangle - \langle s^2\rangle^2\right]^{1/2}}. \tag{20}$$

IV. FINITE SIZE EFFECTS

Compared with the linear size L of the lattice, the distances that we consider for our two- and three-point functions are small. In that respect they can be viewed as local operators in the even channel such as the energy density. The free energy density on a finite lattice, for a vanishing external field behaves as [12–15]

$$f(\beta,L) = L^{-d}\, h(L^{1/\nu} t) + f_{ns}(t), \tag{21}$$

where $t = (\beta_c - \beta)/\beta_c$ is the reduced temperature, h and f_{ns} are analytic functions. Taking the derivative with respect to β we arrive at

$$E(\beta,L) = L^{-d+1/\nu}\, \tilde{h}(L^{1/\nu} t) + E_{ns}(t), \tag{22}$$

where $\tilde{h} = -h'/\beta_c$. Setting $\beta = \beta_c$ we get

$$E(\beta_c,L) = L^{-d+1/\nu}\, \tilde{h}(0) + E_{ns}(0), \tag{23}$$

where $d - 1/\nu = d - y_t = \Delta_\epsilon$.

In the analysis of our data, we assume that the finite size scaling behavior of the two- and three-point functions is given by Eq. (23), where, of course, the values of the constants depend on the quantity that is considered. Given the huge amount of data, we abstain from sophisticated fitting with

Ansätze motivated by Eq. (23). Instead, we consider pairs of linear lattice sizes $L_1 = L$, $L_2 = 2L$ and compute

$$G_{ex}(2L) := G(2L) + \frac{G(2L) - G(L)}{2^{\Delta_\epsilon} - 1}, \tag{24}$$

where G is the quantity under consideration. Equation (24) is derived by inserting L_1 and L_2 into Eq. (23) and solving the system of two equations with respect to the nonsingular (ns) part.

To simplify the analysis, in the case of Eqs. (17)–(20) we computed the estimates of $f_{\sigma\sigma\epsilon}$ and $f_{\epsilon\epsilon\epsilon}$ using the two- and three-point functions obtained for given lattice sizes L. These estimates are then plugged into Eq. (24) to get the results for $L \to \infty$. The validity of the procedure is checked by comparing the results obtained for different linear lattice sizes L. See Fig. 6 below.

V. VARIANCE REDUCTION

Here, we discuss the construction of variance reduced estimators of N-point functions along the lines of Refs. [21,22] in a general setting. Let us consider the N-point function

$$\langle \mathcal{O}_1(x_1)\mathcal{O}_2(x_2)\dots\ \mathcal{O}_N(x_N)\rangle = \frac{\int D[\phi] \exp(-H[\phi])\ \mathcal{O}_1(x_1)\mathcal{O}_2(x_2)\dots\mathcal{O}_N(x_N)}{\int D[\phi] \exp(-H[\phi])}, \tag{25}$$

where $[\phi]$ denotes the collection of all fields. In the case of the Blume-Capel model that we simulate here, it is actually the collection of the spins s_x and the integral becomes the sum over all configurations. Let us consider subsets $x_i \in \mathcal{B}_i \subset \Lambda$ of sites for each i, where Λ is the set of all sites of the lattice. Typically, one takes $x_k \in \mathcal{B}_i$ if $||x_k - x_i|| \leqslant l_{\max}$ for some norm $||\dots||$. The crucial requirement is that for any pair $i \neq j$, no pair of sites $x_k \in \mathcal{B}_i$ and $x_l \in \mathcal{B}_j$ exists such that x_k and x_l are nearest neighbors. Note that we consider a nearest neighbor interaction here. For interactions over a larger distance, the plane of division has to be thickened correspondingly.

In Fig. 1 we sketch an implementation for a square lattice and three points. For simplicity, in the sketch as well as in our simulations, we use the Chebyshev distance $||x_i - x_j|| = \max_\alpha |x_i^{(\alpha)} - x_j^{(\alpha)}|$. Let us denote the collection of fields living on $\mathcal{B}_i$ by $[\phi]_i$ and the collection of fields that are not associated with any of the blocks by $[\phi]_R$. The reduced Hamiltonian can be written as a sum, where one summand depends only on the fields on the remainder and the others on the fields on the remainder and on one of the blocks:

$$H[\phi] = h_R([\phi]_R) + \sum_i h_i([\phi]_i, [\phi]_R). \tag{26}$$

This decomposition allows us to rewrite the expectation value in the following way:

$$\begin{aligned}
\langle \mathcal{O}_1(x_1)\mathcal{O}_2(x_2)\dots\ \mathcal{O}_N(x_N)\rangle &= \frac{\int D[\phi]_R\ \exp[-h_R([\phi]_R)]\ \int(\prod_i D[\phi]_i)(\prod_i \exp[-h_i([\phi]_i,[\phi]_R)])\ (\prod_i\ \mathcal{O}_i(x_i))}{\int D[\phi]_R\ \exp[-h_R([\phi]_R)]\ \int(\prod_i D[\phi]_i)(\prod_i \exp[-h_i([\phi]_i,[\phi]_R)])} \\
&= \frac{\int D[\phi]_R\ \exp[-h_R([\phi]_R)]\ \prod_i(\int D[\phi]_i\ \exp[-h_i([\phi]_i,[\phi]_R)]\ \mathcal{O}_i(x_i))}{\int D[\phi]_R\ \exp[-h_R([\phi]_R)]\ \prod_i(\int D[\phi]_i\ \exp[-h_i([\phi]_i,[\phi]_R)])} \\
&= \frac{\int D[\phi]_R\ \exp[-h_R([\phi]_R)]\ (\prod_i z_i([\phi]_R))\ (\prod_i \langle \mathcal{O}_i(x_i)\rangle_i([\phi]_R))}{\int D[\phi]_R\ \exp[-h_R([\phi]_R)]\ \prod_i z_i([\phi]_R)},
\end{aligned} \tag{27}$$

where we define

$$z_i([\phi]_R) = \int D[\phi]_i \exp[-h_i([\phi]_i,[\phi]_R)] \tag{28}$$

and

$$\langle \mathcal{O}_i(x_i)\rangle_i([\phi]_R) = \frac{\int D[\phi]_i \exp[-h_i([\phi]_i,[\phi]_R)]\mathcal{O}_i(x_i)}{z_i([\phi]_R)}. \tag{29}$$

Note that the variance of $\prod_i \langle \mathcal{O}_i(x_i)\rangle_i$ is smaller than that of $\prod_i \mathcal{O}_i(x_i)$ since degrees of freedom have been integrated out. In our case, we can perform an exact summation over all configurations if $\mathcal{B}_i$ is small. In particular, if $\mathcal{B}_i$ contains only the site x_i we can easily perform the summation over the three possible values of s_i.

However, it is also advantageous to perform the integration over the variables on the blocks $\mathcal{B}_i$ approximately by using a Monte Carlo simulation. Let us denote the simulations restricted to field variables on the blocks $\mathcal{B}_i$ by baby simulations. These baby simulations are started after the whole system is equilibrated. After the baby simulations are finished, updates on the whole system are performed. Then again baby simulations are started. This process is iterated. During the baby simulations, we perform n_B measurements of $\mathcal{O}_i(x_i)$ for all i. Let us define

$$\overline{\mathcal{O}_i(x_i)}_\gamma = \frac{1}{n_B}\sum_{\alpha=1}^{n_B} \mathcal{O}_{i,\gamma,\alpha}(x_i), \tag{30}$$

where α labels the measurements that are performed for a given configuration on the remainder. The configurations on the remainder are labeled by γ. We generate n_R configurations on the remainder after equilibration. An estimate of the N-point function is obtained by

$$\begin{aligned}
&\overline{\overline{\mathcal{O}_1(x_1)}\ \overline{\mathcal{O}_2(x_2)}\dots\overline{\mathcal{O}_N(x_N)}} \\
&\quad = \frac{1}{n_R}\sum_{\gamma=1}^{n_R} \overline{\mathcal{O}_1(x_1)}_\gamma \overline{\mathcal{O}_2(x_2)}_\gamma \dots \overline{\mathcal{O}_N(x_N)}_\gamma.
\end{aligned} \tag{31}$$

Let us discuss the variance of $\prod_i \overline{\mathcal{O}_i(x_i)}$. The baby Monte Carlo simulations are independent of each other. Therefore, actually n_B^N configurations are generated for the N-point function. Hence, at least for small n_b, we expect that the variance of $\prod_i \overline{\mathcal{O}_i(x_i)}$ behaves as n_B^{-N}. As n_B further increases, the variance of $\prod_i \langle \mathcal{O}_i(x_i)\rangle_i$ is approached. Hence, the maximal efficiency is reached at some finite n_B that in general has to be determined numerically.

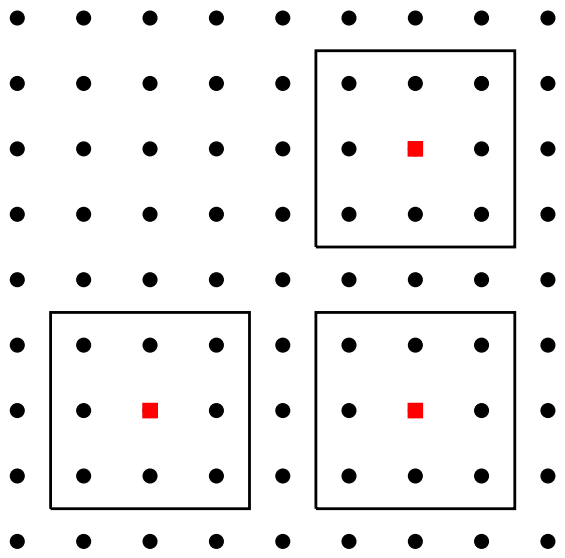

FIG. 1. Two-dimensional sketch of the decomposition of the lattice into blocks and the remainder. The three points x_1, x_2, and x_3 are represented by red squares. The blocks around these three points are bordered by solid black lines. Lattice sites different from x_1, x_2, or x_3 are given by solid black circles.

VI. THE ALGORITHM AND OUR IMPLEMENTATION

As in our previous studies of the Blume-Capel model, for example [26], we simulated the model by using a hybrid of the local heat bath algorithm, the local Todo-Suwa algorithm [27,28], and the single cluster algorithm [29]. The Todo-Suwa algorithm is a local algorithm that does not fulfill detailed balance but only balance. Since ergodicity can not be proved for the Todo-Suwa algorithm, additional local heat bath updates are needed to ensure ergodicity for the update scheme as a whole. The advantage of the Todo-Suwa algorithm is that it reduces autocorrelation times compared with the heat bath algorithm.

A. Implementation

We used the SIMD-oriented Fast Mersenne Twister algorithm [30] as random number generator. Our code is a straightforward implementation of the algorithms in standard C. We abstained from a parallelization of the algorithms since in the case of the single cluster update, a good scaling of the performance with the number of processes is hard to achieve (see, for example, Ref. [31]). We stored the value of the spins as character variable. For simplicity, we abstain from using a more compact storage scheme, using for example two bits for a spin only. With this setup, on the hardware that is available to us, $L = 1600$ is about the largest linear lattice size that can be simulated efficiently. In order to extrapolate in L and also monitor the dependence of our results on the lattice size, we performed simulations for the linear lattice sizes $L = 400, 600$, 800, 1200, and 1600.

A cycle of the update consists of one heat bath sweep followed by n_{cl} single cluster updates. Then, follow n_{ts} sweeps using the Todo-Suwa algorithm. For each of these sweeps, n_{cl} single cluster updates are performed. Mostly, n_{cl} is chosen such that, very roughly, n_{cl} times the average cluster size equals the number of sites L^3. For the range of lattice sizes studied here, $n_{cl} = L$ is a reasonable choice. As we learned from previous work, for example [7,25,26], the performance of the algorithm does not depend sharply on the values of the parameters n_{ts} and n_{cl}. Also skipping the single cluster updates that follow the heat bath sweep likely changes the performance by little. Therefore, we abstained from a fine tuning of the update scheme.

Let us summarize an update cycle with the following pseudo-C code:

```
heat bath sweep
for(icl=0;icl<ncl;icl++) single cluster
for(its=0;its<nts;its++)
  {
  Todo-Suwa sweep
  for(icl=0;icl<ncl;icl++) single cluster
  }
baby simulations for all blocks
measure N-point functions
```

We parallelized trivially by performing several independent runs. The equilibration takes a significant amount of CPU time for the larger lattices. Therefore, in the case of the linear lattice sizes $L = 800$, 1200, and 1600, we first equilibrated a single Markov chain. We started from a disordered configuration. Since in the initial phase the simulation clusters are small, we adapted the number of cluster updates for the first few update cycles, such that the aggregate cluster size per local update sweep is equal to the lattice volume or larger. After this initial phase, we continued with a fixed number of single cluster updates per local update sweep as discussed above. We performed 1000 complete update cycles to equilibrate the system. These update cycles are characterized by $n_{ts} = 3$. Here, we only measured a few quantities like the energy density and the magnetization to monitor the equilibration. We stored the final configuration to disk. Then, we started several runs using this configuration as initial one and different seeds for the random number generator. In these runs we used $n_{ts} = 9$. Since the baby simulations discussed below take a considerable amount of CPU time, we intend to essentially eliminate the correlation between subsequent measurements by using such a large number for n_{ts}. In the case of $L = 800$, 1200, and 1600, starting from the configuration written to disk, we performed four update cycles to achieve a sufficient decorrelation between the branches of our simulation before measuring the N-point functions.

B. The baby Monte Carlos and the measurements

In our simulations we did not attempt to adjust the size of the blocks to the distances between the points in the N-point functions. Instead, we determined the N-point functions for a certain range of distances for a given decomposition into blocks.

In order to save CPU time, we determined $\overline{s_x}$ and $\overline{s_x^2}$ for a subset of the lattice sites only. This subset is characterized by $x^{(i)} = 0, n_s, 2n_s, \ldots, L - n_s$. In our study we performed simulations for the strides $n_s = 2$ and 4. We used blocks around these sites that are defined by $y \in \mathcal{B}_i$ if $|y^{(\alpha)} - x_i^{(\alpha)}| < n_s$ for $\alpha = 0$, 1, and 2. This means that the linear size of these blocks is $l_b = 3$ and 7 for $n_s = 2$ and 4, respectively. Before

starting the baby simulations we copied the spins of the block and its outer boundary to an auxiliary array. Since for our choice nearest neighbor blocks overlap, we can not write all the final configurations of the baby simulations back to the main Markov chain. Only a subset with the larger stride $2n_s$ could be used to this end. However, for simplicity we refrain to do so. At the end of the baby simulations, we only stored the results $\overline{s_x}$ and $\overline{s_x^2}$ to compute the estimates of the N-point functions. We computed the two- and three-point functions for the distances $j = 2n_s, 3n_s, \ldots, 9n_s$. Note that for $j = n_s$ the blocks overlap, and no valid result is obtained.

In the case of $n_s = 2$, we performed n_b sweeps over the block using the Todo-Suwa algorithm. Note that in the case of the baby simulations, ergodicity is not needed. In preliminary simulations we varied n_b. It turns out that the performance maximum is not very sharp and depends on the type of the N-point function and on the distance j. Based on these experiments, we decided to use $n_b = 10$ in our production runs.

We performed a measurement for the starting configuration and after each Todo-Suwa sweep. We reduced the variance by performing the sum over s_x for fixed neighbors exactly:

$$\tilde{s}_x = \frac{\sum_{s_x} \exp\left(\beta s_x S_x - D s_x^2\right) s_x}{\sum_{s_x} \exp\left(\beta s_x S_x - D s_x^2\right)} \tag{32}$$

and

$$\tilde{s}_x^2 = \frac{\sum_{s_x} \exp\left(\beta s_x S_x - D s_x^2\right) s_x^2}{\sum_{s_x} \exp\left(\beta s_x S_x - D s_x^2\right)}, \tag{33}$$

where

$$S_x = \sum_{y.nn.x} s_y, \tag{34}$$

where $y.nn.x$ means that y is a nearest neighbor of x.

In the case of $n_s = 4$ we used two different updating schemes. Since the measurement is performed at the central site only, it might be useful to update the spins at central sites more often than those at the boundary. To this end, we performed one sweep using the Todo-Suwa algorithm over the full 7^3 block, then follows a sweep over the central 5^3, and finally a sweep over the central 3^3 block. This sequence is repeated n_b times. A measurement using Eqs. (32) and (33) is performed for the initial configuration and after each (partial) sweep. This means that $3n_b + 1$ measurements are performed for each baby simulation. After a few preliminary simulations, we decided to take $n_b = 30$. We used this scheme for our simulations of lattices of the linear sizes $L = 400$ and 800.

For $L = 600$, 1200, and 1600, we performed cluster updates in addition to the sweeps with the Todo-Suwa update. The cluster construction is started from the fixed boundary of the block using the standard delete probability $p_d = \min[1, \exp(-2\beta s_x s_y)]$, where x and y are nearest neighbors. All spins that are not frozen to the boundary are flipped. Here, we used $n_b = 20$. It turned out that the cluster update gives little advantage. However, it is likely that going to larger n_s this will change. We made no attempt to construct cluster improved estimators of s_x and s_x^2.

We performed simulations of the Ising model for $L = 400$ and 800 and $n_s = 2$. In the baby Monte Carlos, we replaced the Todo-Suwa update by the Metropolis one. In the main Markov chain, we replaced the Todo-Suwa sweeps by heat bath sweeps. Note that in the case of the Ising model, the single cluster algorithm is ergodic and it is not necessary to add heat bath sweeps.

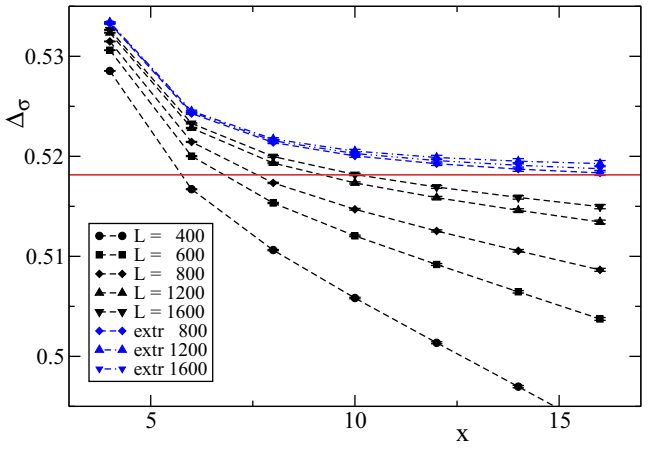

FIG. 2. We plot the effective dimension $\Delta_{\sigma,\mathrm{eff}}(x,s)$ of the field as defined by Eq. (35) for various linear lattice sizes L and extrapolated estimates. The displacement between the points is in direction $r = a$. The dashed lines should only guide the eye. We give only results obtained from our simulations with stride $n_s = 2$ to keep the figure readable. The horizontal solid line gives $\Delta_\sigma = 0.5181489$ obtained by the bootstrap method.

C. Statistics of our simulations

For a given set of parameters n_s and L, we performed 8 up to 20 independent runs. Each running for a month on a single core of the CPU. In Table I we summarize the total number of update and measurement cycles performed for each stride n_s and linear lattice size L. We performed preliminary simulations without using the variance reduction method. We abstain from discussing these simulations in detail. In total, our study took about 10 years on a single core of a recent CPU.

VII. NUMERICAL RESULTS

Throughout we used the jackknife method to compute statistical errors.

A. Two-point functions

First, we extracted the dimension of the operators from the two-point functions $g(x)$. To this end, we define an effective exponent by

$$\Delta_{\mathrm{eff}}(x,\Delta x) = -\frac{1}{2}\frac{\ln(g(x+\Delta x)/g(x))}{\ln((x+\Delta x)/x)}, \tag{35}$$

where $\Delta x = n_s$, $\Delta x = \sqrt{2}n_s$, and $\Delta x = \sqrt{3}n_s$ for $r = a$, f, and d, respectively. Let us first discuss our results obtained for $g_{r,\sigma\sigma}$. In Fig. 2, we plot results obtained for displacements along the axis and the stride $n_s = 2$. We give results for all linear lattice sizes that we have simulated. Up to our largest linear lattice size $L = 1600$ we see a dependence on the lattice size. In addition, we give extrapolated results using the pairs of lattice sizes (400,800), (600,1200), and (800,1600).

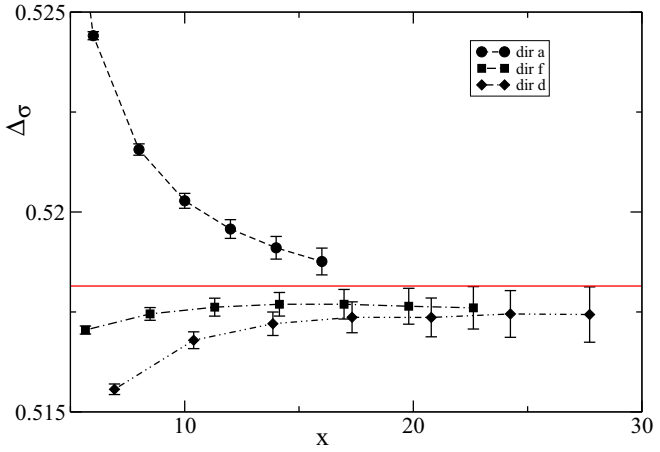

FIG. 3. We plot the effective dimension $\Delta_{\sigma,\mathrm{eff}}(x,s)$ of the field as defined by Eq. (35) for the extrapolation obtained for $L = 800$ and 1600. All results are obtained for the stride $n_s = 2$. We give results for all three directions of the displacement that we have studied. The dashed lines should only guide the eye. The horizontal solid line gives $\Delta_\sigma = 0.5181489$ obtained by the bootstrap method.

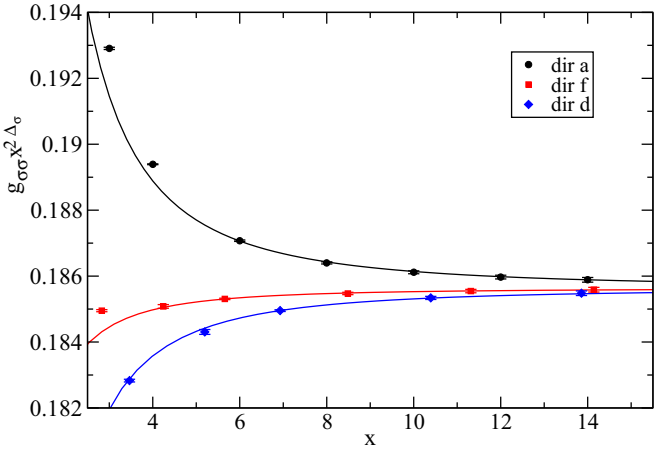

FIG. 4. We plot $g_{\sigma\sigma}x^{2\Delta_\sigma}$ with $\Delta_\sigma = 0.5181489$ for the three directions studied. Most of the data are taken from the simulations with the stride $n_s = 2$. A few data at small distances are taken from our preliminary simulations without variance reduction. The solid lines give the result of the fit with the *Ansatz* (36).

The extrapolated results are essentially consistent among each other, confirming the validity of the extrapolation. It seems plausible that the extrapolated result approaches the bootstrap value as x increases.

In Fig. 3, we give results for the extrapolation of the pair $(L,2L) = (800,1600)$ for all three directions that we have studied. The small x deviations are the largest for $r = a$. For $r = f$ and d they have the opposite sign as for $r = a$. The amplitude of corrections is the smallest for $r = f$. Since the corrections depend strongly on r, it is likely that they are dominantly caused by the breaking of the rotational symmetry and fall off like $x^{-\omega_{NR}}$.

Here, we make no effort to extract an optimal estimate for Δ_σ from our data. Just looking at the figure, one might take the result for direction f at distance $x = 10\sqrt{2}$ as final estimate: $\Delta_\sigma = 0.5177(3)$. Using $\eta = 0.03627(10)$ [7], we get $\Delta_\sigma = (1+\eta)/2 = 0.51814(5)$, which is consistent, but clearly more accurate. In the following analysis, we shall use the bootstrap value $\Delta_\sigma = 0.5181489(10)$, which outpaces the Monte Carlo results by far.

Let us look in more detail at the corrections at small distances. Since we study an improved model, we expect that the numerically dominant corrections are due to the breaking of the Galilean invariance by the lattice. In Fig. 4, we plot our extrapolated results for $g_{\sigma\sigma}x^{2\Delta_\sigma}$ with $\Delta_\sigma = 0.5181489$ for the three directions studied. We have fitted these data with

$$x^{2\Delta_\sigma} g_{r,\sigma\sigma}(x) = c + a_r x^{-\omega_{NR}} \tag{36}$$

with $\omega_{NR} = 2.022665$. For simplicity, we performed a naive fit of the data, not taking into account the statistical correlation of the data for different distances and directions. We fitted jointly the data for the three different directions, requiring that the constant c is the same for all three directions. The result is represented by the solid lines in Fig. 4. Given the shortcomings of the fit, we do not give final results for the constants c and a_r. Our aim is to demonstrate that the *Ansatz* (36) indeed describes the data well as can be seen in the plot.

Next, we analyzed the $\epsilon\epsilon$ function. The findings are analogous to those of the $\sigma\sigma$ function. Therefore, we refrain from a detailed discussion. A main difference is that the relative statistical error increases faster with increasing distance as for the $\sigma\sigma$ function. In Fig. 5, we have plotted the analog of Fig. 4.

B. Three-point functions

Let us first check that the extrapolation in the lattice size works. To this end, we plot in Fig. 6 our estimates of $f_{\sigma\sigma\epsilon}$ for $r = f$ [Eq. (17)] as a function of the distance x for different linear lattice sizes L. In addition, we give the results obtained from the extrapolation (24). Note that we applied Eq. (24) to our estimates of $f_{\sigma\sigma\epsilon}$ obtained for given L. For the results obtained for a given lattice size, we see a clear dependence on the lattice size. In contrast, the extrapolated results are consistent among each other.

In Fig. 7, we plot our results for $f_{\sigma\sigma\epsilon}$ obtained from the extrapolation using the lattice sizes $(L,2L) = (800,1600)$. We

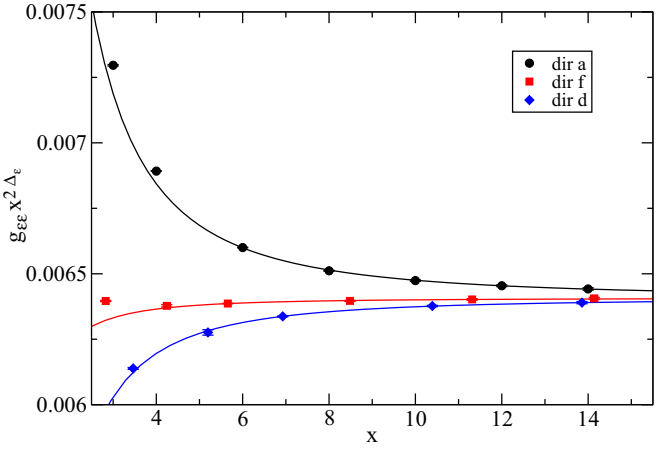

FIG. 5. We plot $g_{\epsilon\epsilon}x^{2\Delta_\epsilon}$ with $\Delta_\epsilon = 1.412625$ for the three directions studied. Analogous to Fig. 4.

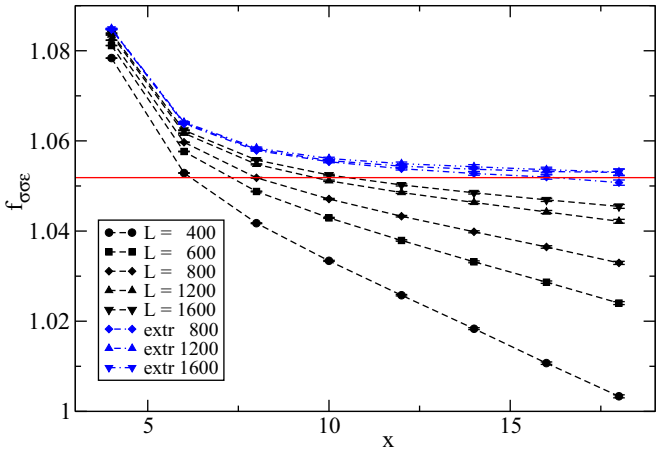

FIG. 6. We plot our estimates of $f_{\sigma\sigma\epsilon}$ for the direction $r = f$. We give results obtained for the linear lattice sizes $L = 400$, 600, 800, 1200, and 1600 and the extrapolations for the pairs $(L,2L) = (400,800)$, $(600,1200)$, and $(800,1600)$. All data shown are based on our simulations with $n_s = 2$. The straight line gives the bootstrap result.

give results for both geometries that we have implemented. Here, we use both the simulations with stride $n_s = 2$ and 4. For distances, where data from both strides are available, we give the weighted average. Throughout, the estimates from $n_s = 2$ are more accurate than those obtained from $n_s = 4$. For example, for $r = f$ we get for the distance $j = 16$ the estimates 1.05322(59) and 1.05242(88) from the simulations with stride $n_s = 2$ and 4, respectively. Also here, we see corrections at short distances. We fitted with the *Ansatz* (36). For simplicity, we did not take into account the cross correlations for different distances. We performed independent fits for both directions. Good fits are obtained starting from $j_{\min} = 6$, where all $j \geqslant j_{\min}$ are included. For example, for $j_{\min} = 8$ we find $c = 1.0513(3)$ and 1.0506(5) for $r = f$ and d, respectively. The errors give only an indication since cross correlations of the

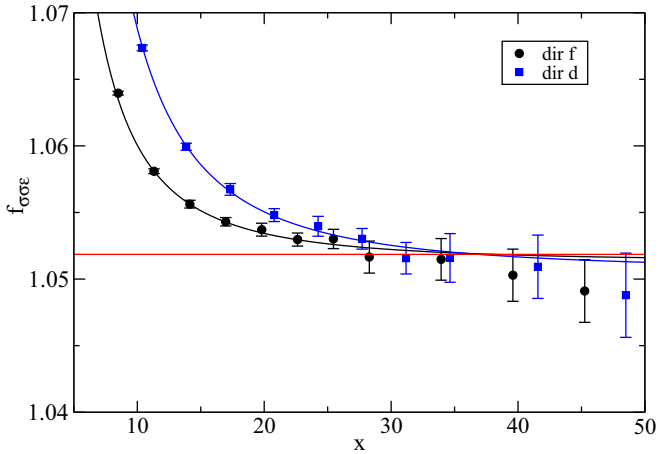

FIG. 7. We plot our estimates of $f_{\sigma\sigma\epsilon}$ as a function of the distance x. These estimates are obtained by the extrapolation of the pair of lattice sizes $(L,2L) = (800,1600)$. The solid lines represent the results of fits with the *Ansatz* (36).

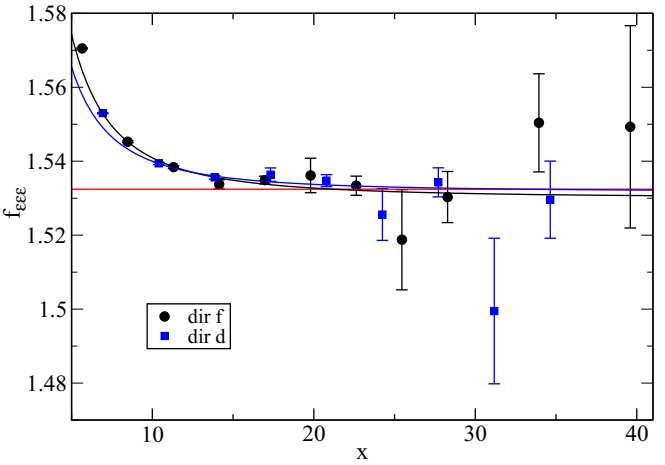

FIG. 8. We plot our estimates of $f_{\epsilon\epsilon\epsilon}$ as a function of the distance x. The solid lines give the results of fits with the *Ansatz* (36).

data are not taken into account in the fit. In Fig. 7, results of these fits are given as solid lines.

It is hard to quote a final value that is not biased by the knowledge of the bootstrap result. But, it is quite clear that our numerical data are consistent with the bootstrap result. Looking at Fig. 7 we might read off

$$f_{\sigma\sigma\epsilon} = 1.051(1) \tag{37}$$

from the estimate of $f_{\sigma\sigma\epsilon}$ at distances $x \approx 30$, without relying on the fits discussed above.

Next, we discuss our results obtained for $f_{\epsilon\epsilon\epsilon}$. Here, the statistical error increases faster with the distance than for $f_{\sigma\sigma\epsilon}$. Furthermore, we find that our results obtained for the simulations with the stride $n_s = 4$ are more accurate than those with $n_s = 2$. For example, for $r = f$ and $j = 16$ we get $f_{\epsilon\epsilon\epsilon} = 1.53089(85)$ and 1.53363(27) from our simulations with $n_s = 2$ and 4, respectively. Performing fits with the *Ansatz* (36) we get for $j_{\min} = 8$ the results $c = 1.5312(14)$ and 1.5333(22) for $r = f$ and d, respectively.

In Fig. 8, the results of these fits are given as solid lines. As final result we read off from distances $x \approx 25$

$$f_{\epsilon\epsilon\epsilon} = 1.533(5) \tag{38}$$

not relying on the fits with *Ansatz* (36).

C. Statistical errors

We find that the relative statistical error of the two- and three-point functions increases with the distance x following a power law. The exponent is the same for $n_s = 2$ and 4 and for the simulations without variance reduction. The smallest exponent $\approx$1.36 is found for the $\sigma\sigma$ function and the largest $\approx$4 for the estimate of $f_{\epsilon\epsilon\epsilon}$. This corresponds to the fact that we see the largest gain by using the variance reduction method in the case of $f_{\epsilon\epsilon\epsilon}$.

D. Sensitivity on the value taken as estimate of β_c

We performed simulations for $L = 800$ with stride $n_s = 2$ at $\beta = 0.38772$ and 0.38772347 in order to estimate the dependence of the N-point functions on β. For $f_{\sigma\sigma\epsilon}$ and $f_{\epsilon\epsilon\epsilon}$,

we find that the slope $\frac{\Delta f(x,\beta)}{\Delta\beta}$ increases with increasing distance x of the points, following a power law. Here, $f(x,\beta)$ is the numerical estimate of $f_{\sigma\sigma\epsilon}$ or $f_{\epsilon\epsilon\epsilon}$ as a function of x and β. In the case of $f_{\sigma\sigma\epsilon}$ we find an exponent $\approx$1.25, while for $f_{\epsilon\epsilon\epsilon}$ we get an exponent $\approx$1.38. From the data it is not completely clear whether the exponents are different. We estimate that the uncertainty of the estimate $\beta_c = 0.387721735(25)$ used here results in an uncertainty in the fourth digit of our final results for $f_{\sigma\sigma\epsilon}$ and $f_{\epsilon\epsilon\epsilon}$ which is negligible compared with the errors quoted above.

E. Sensitivity on leading corrections

Since we do not know the value of D^* exactly, our results might be affected by residual leading corrections to scaling. In Ref. [7] we have demonstrated that the amplitude of leading corrections to scaling should be at least reduced by a factor of 30 in the Blume-Capel model at $D = 0.655$ compared with the Ising model. Therefore, we performed simulations of the Ising model at β_c, the pair of lattice sizes $L = 400$ and 800, and the stride $n_s = 2$. A bit to our surprise, it is hard to see a difference between the results obtained from the Ising model and the improved Blume-Capel model. We conclude that the effects of leading corrections to scaling can be ignored at the level of our accuracy of our final results for $f_{\sigma\sigma\epsilon}$ and $f_{\epsilon\epsilon\epsilon}$.

VIII. SUMMARY AND CONCLUSIONS

We computed the operator product expansion coefficients $f_{\sigma\sigma\epsilon}$ and $f_{\epsilon\epsilon\epsilon}$ by using Monte Carlo simulations of the improved Blume-Capel model at the critical point. We simulated lattices with periodic boundary conditions up to a linear size $L = 1600$. We extrapolated our results assuming that finite size corrections vanish $\propto L^{-\Delta_\epsilon}$, where L is the linear size of the lattice. An important ingredient of our study is the variance reduced estimator based on [21,22]. In particular, $f_{\epsilon\epsilon\epsilon}$ could be determined accurately due to the variance reduction. The error of our results is smaller than that of previous estimates obtained by Monte Carlo simulations of the three-dimensional Ising model.

Our results are fully consistent with the predictions obtained by the conformal bootstrap method. These results are, however, by far more accurate than ours. Our study still can be understood as preliminary. Easily more CPU time could be used and the parameters of the algorithm could be better tuned. However, one would hardly reach the accuracy obtained by using the conformal bootstrap method. The fact that our results agree with those of the conformal bootstrap method confirm the theoretical expectation that the RG fixed point that governs the three-dimensional Ising universality class is indeed invariant under conformal transformations.

The strategy used here seems to be applicable to other universality classes. However, in the case of $O(N)$-symmetric model with $N \geqslant 2$, more memory per site is needed and therefore the linear lattice sizes that can be reached are smaller. It is unclear whether these lattice sizes allow for a reliable extrapolation to the infinitely large system.

An interesting idea would be to combine the UV sampler [19,20] or some similar method with the variance reduced estimator of N-point functions discussed here.

ACKNOWLEDGMENT

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under the Grant No. HA 3150/4-1.

[1] S. El-Showk, M. F. Paulos, D. Poland, S. Rychkov, D. Simmons-Duffin, and A. Vichi, Solving the 3d Ising model with the conformal bootstrap II. c-minimization and precise critical exponents, J. Stat. Phys. **157**, 869 (2014).

[2] F. Kos, D. Poland, and D. Simmons-Duffin, Bootstrapping mixed correlators in the 3D Ising model, J. High Energy Phys. 11 (2014) 109.

[3] F. Gliozzi and A. Rago, Critical exponents of the 3d Ising and related models from conformal bootstrap, J. High Energy Phys. 10 (2014) 042.

[4] F. Kos, D. Poland, D. Simmons-Duffin, and A. Vichi, Precision islands in the Ising and $O(N)$ models, J. High Energy Phys. 08 (2016) 036.

[5] D. Simmons-Duffin, The lightcone bootstrap and the spectrum of the 3d Ising, J. High Energy Phys. 03 (2017) 086.

[6] D. Simmons-Duffin, The conformal bootstrap, in *Proceedings of the 2015 Theoretical Advanced Study Institute in Elementary Particle Physics: New Frontiers in Fields and Strings (TASI 2015): Boulder, CO, USA, June 1-26, 2015* (World Scientific, Singapore, 2017), pp. 1–74.

[7] M. Hasenbusch, Finite size scaling study of lattice models in the three-dimensional Ising universality class, Phys. Rev. B **82**, 174433 (2010).

[8] M. Campostrini, A. Pelissetto, P. Rossi, and E. Vicari, 25th order high temperature expansion results for three-dimensional Ising like systems on the simple cubic lattice, Phys. Rev. E **65**, 066127 (2002).

[9] P. Butera and M. Comi, The ϕ_3^4 lattice field theory viewed from the high-temperature side, Phys. Rev. B **72**, 014442 (2005).

[10] R. Guida and J. Zinn-Justin, Critical exponents of the N vector model, J. Phys. A: Math. Gen. **31**, 8103 (1998).

[11] K. E. Newman and E. K. Riedel, Critical exponents by the scaling-field method: The isotropic N-vector model in three dimensions, Phys. Rev. B **30**, 6615 (1984).

[12] K. G. Wilson and J. B. Kogut, The Renormalization group and the epsilon expansion, Phys. Rep. **12**, 75 (1974).

[13] M. E. Fisher, The renormalization group in the theory of critical behavior, Rev. Mod. Phys. **46**, 597 (1974); The renormalization group in the theory of critical behavior, **47**, 543(E) (1975).

[14] M. E. Fisher, Renormalization Group Theory: Its Basis and Formulation in Statistical Physics, in Conceptual Foundations of Quantum Field Theory, Proceedings, Symposium and Workshop, Boston, USA, March 1–3, 1996, Rev. Mod. Phys. **70**, 653 (1998).

[15] A. Pelissetto and E. Vicari, Critical phenomena and renormalization group theory, Phys. Rep. **368**, 549 (2002).

[16] A. M. Polyakov, Conformal symmetry of critical fluctuations, JETP Lett. **12**, 381 (1970) [Pisma Zh. Eksp. Teor. Fiz. **12**, 538 (1970)].

[17] M. Caselle, G. Costagliola, and N. Magnoli, Numerical determination of the operator-product-expansion coefficients in the 3D Ising model from off-critical correlators, Phys. Rev. D **91**, 061901 (2015).

[18] G. Costagliola, Operator product expansion coefficients of the 3D Ising model with a trapping potential, Phys. Rev. D **93**, 066008 (2016).

[19] V. Herdeiro, Numerical estimation of structure constants in the three-dimensional Ising conformal field theory through Markov chain uv sampler, Phys. Rev. E **96**, 033301 (2017).

[20] V. Herdeiro and B. Doyon, Monte Carlo method for critical systems in infinite volume: The planar Ising model, Phys. Rev. E **94**, 043322 (2016).

[21] G. Parisi, R. Petronzio, and F. Rapuano, A measurement of the string tension near the continuum limit, Phys. Lett. B **128**, 418 (1983).

[22] M. Lüscher and P. Weisz, Locality and exponential error reduction in numerical lattice gauge theory, J. High Enery Phys. 09 (2001) 010.

[23] Y. Deng and H. W. J. Blöte, Constraint tricritical Blume-Capel model in three dimensions, Phys. Rev. E **70**, 046111 (2004).

[24] M. Hasenbusch, Monte Carlo studies of the three-dimensional Ising model in equilibrium, Int. J. Mod. Phys. C **12**, 911 (2001).

[25] M. Hasenbusch, Thermodynamic Casimir effect: Universality and corrections to scaling, Phys. Rev. B **85**, 174421 (2012).

[26] M. Hasenbusch, Variance-reduced estimator of the connected two-point function in the presence of a broken Z_2-symmetry, Phys. Rev. E **93**, 032140 (2016).

[27] S. Todo and H. Suwa, Geometric allocation approaches in Markov chain Monte Carlo, J. Phys.: Conf. Ser. **473**, 012013 (2013).

[28] F. Gutsch, *Markov-Ketten ohne detailliertes Gleichgewicht*, Bachelor thesis, Humboldt-Universität zu Berlin, 2014.

[29] U. Wolff, Collective Monte Carlo Updating for Spin Systems, Phys. Rev. Lett. **62**, 361 (1989).

[30] M. Saito and M. Matsumoto, SIMD-oriented Fast Mersenne Twister: a 128-bit Pseudorandom Number Generator, in *Monte Carlo and Quasi-Monte Carlo Methods 2006*, edited by A. Keller, S. Heinrich, and H. Niederreiter (Springer, Berlin, 2008); M. Saito, Masters' thesis, Math. Dept., Graduate School of Science, Hiroshima University, 2007. The source code of the program is provided at "http://www.math.sci.hiroshima-u.ac.jp/∼m-mat/MT/SFMT/index.html"

[31] J. Kaupužs, J. Rimšāns, and R. V. N. Melnik, Parallelization of the Wolff single-cluster algorithm, Phys. Rev. E **81**, 026701 (2010).