Abstract

The linked-cluster expansion technique for the high-temperature expansion of spin model is reviewed. A new algorithm for the computation of three-point and higher Green's functions is presented. Series are computed for all components of two-point Green's functions for a generalized 3D Ising model, to 25th order on the bcc lattice and to 23rd order on the sc lattice. Series for zero-momentum four-, six-, and eight-point functions are computed to 21st, 19th, and 17th order respectively on the bcc lattice.

1 Introduction

The high-temperature (strong-coupling) series expansion is one of the most successful tools for the study of physical systems near a critical point.

High-temperature series are analytic; the radius of convergence is usually quite large, often reaching the boundary of the high-temperature phase. This property allows the application of powerful techniques of resummation and analytical continuation, which can yield very precise and reliable results, provided that long series are available. It is therefore worthwhile to push the computation of high-temperature series as far as our algorithms and computers allow.

The most successful technique for the computation of high-temperature series of 3D spin models is the linked-cluster expansion (LCE), which is well suited for the fully computerized approach required to reach very high orders of the expansion.

Several detailed discussion of the LCE appeared in the literature. Wortis' review covers most of the basic topics, and provides many graphical rules fit for algorithmic implementation. Nickel performed a remarkable computation for the 3D Ising model on the bcc lattice, which had not been surpassed until the present work; in a more detailed paper, Nickel and Rehr also present several clever algorithms which we found very useful. Lüscher and Weisz, describing their application of the LCE to lattice field theory, also provide several important implementation hints.

Unfortunately, the notation found in the literature is by no means uniform. Therefore we will review the relevant aspects of the LCE, which can be found in Ref. [2], not only to make our paper more self-contained, but also to explain notations carefully and to remark the correspondence with Refs. [2, 4, 5]. We will follow the notations of Ref. [4] whenever possible.

The paper is organized as follows:

Sect. 2 introduces the relevant graph theory concepts and definitions.

Sect. 3 presents the generalized Ising model we focus on.
Sects. 4, 5, and 6 review the LCE, with special focus on the two-point Green’s functions.
Sect. 7 describes our algorithm for the computation of three-point and higher Green’s functions.
Sect. 8 is devoted to programming details.
Sect. 9 displays a (small) selection of the series generated.

Forthcoming papers will be devoted to the analysis of the series, using the techniques presented in Ref. [6], and to the generation and analysis of the series for $XY$ systems.

We will not give proofs of our formulae. The only nontrivial step in the proofs of Sects. 5–7 is to show that the symmetry factors compensate exactly the different number of contributions that may appear on the two sides of the equations; it is typically a straightforward, if tedious, exercise in combinatorics. The proofs of Sects. 4–6 are given or sketched in Ref. [2]. The proofs of Sect. 7 are especially easy, since the symmetry factor of a 1-irreducible tree graph is always 1.

2 Graphology

In this section we introduce a number of graph theory concepts relevant for the LCE. We refer the reader to Ref. [7] for a comprehensive introduction to the subject.

A graph is a set of vertices and edges (also named links or bonds in the literature). Each edge $l$ is incident with two distinct vertices, its extrema (we do not allow the extrema to coincide); the set of extrema will be denoted by $\partial l$; we write $\partial l = \{i(l), f(l)\}$; the choice of an “initial” and a “final” vertex is arbitrary. Two vertices are adjacent if they are the extrema of the same edge. We will denote the number of vertices and edges of a graph by $v$ and $e$ respectively. We also consider arcs or oriented edges, incident out of the initial vertex $i(l)$ into the final vertex $f(l)$.

The valence $n(i)$ of a vertex $i$ is the number of edges incident with $i$.

An $r$-rooted graph is a graph with $v$ vertices, $v \geq r$: $r$ roots or external vertices and $v - r$ internal vertices. We will assign the indices 1, ..., $r$ to the roots and the indices $r + 1$, ..., $v$ to the internal vertices. In the drawings, roots will be denoted by open dots and internal vertices by filled dots.

Two $r$-rooted graphs are isomorphic if there exists a one-to-one correspondence $\pi$ of their internal vertices and edges such that the incidence relations are preserved, i.e., $\partial(\pi(l)) = \{\pi(i(l)), \pi(f(l))\}$ (\(\pi(i) = i\) for the roots). From now on, we will identify isomorphic graph, and silently assume that all sets of graphs we define contain only non-isomorphic graphs.

The symmetry factor $S(G)$ of an $r$-rooted graph $G$ is the number of isomorphisms of $G$ into itself, i.e. the number of of permutations of internal vertices and edges preserving the incidence relations.

We will also consider $p$-ordered $r$-rooted graphs, $p \leq r$, i.e. the classes of rooted graphs isomorphic up to permutations of $r - p$ roots. We will assign indices 1, ..., $p$ to the fixed roots and indices $p + 1$, ..., $r$ to the roots which can be permuted. The symmetry factor $S(G)$ is defined as the number of isomorphisms of $G$ into itself, including permutation of the roots $p + 1$, ..., $r$. The symmetry factor divided by $(r - p)!$ is called the modified symmetry factor $S_E(G)$ (cfr. Ref. [5]); it need not be an integer. The $r$-rooted graphs defined above are ordered ($r$-ordered); unless otherwise specified, we will assume that graphs are ordered. 0-ordered graphs are unordered.

We will also discuss ($p$-ordered) $r$-rooted graphs whose edges and/or vertices are assigned a label; let us consider e.g. the case of an edge label $a(l)$ and vertex label $b(i)$. Two labelled
graphs \((G, a, b)\) and \((G', a', b')\) are equivalent (isomorphic) if there exists an isomorphism of \(G\) into \(G'\) such that \(a\) is mapped into \(a'\) and \(b\) is mapped into \(b'\). The symmetry factor \(S(G, a, b)\) is the number of isomorphisms of \((G, a, b)\) into itself.

A pair of vertices \(i\) and \(j\) is connected if there exists a sequence of vertices \(k_1, \ldots, k_n\), with \(k_1 = i\) and \(k_n = j\), and a sequence of edges \(l_1, \ldots, l_{n-1}\) such that \(\partial l_a = \{k_a, k_{a+1}\}, a = 1, \ldots, n - 1\). A graph is connected if every pair of its vertices is connected. In the following, unless otherwise noted, we will assume that every graph is connected.

A sequence of distinct vertices \(k_1, \ldots, k_n\) and distinct edges \(l_1, \ldots, l_n\) is called a loop of length \(n\) if \(\partial l_a = \{k_a, k_{a+1}\}, a = 1, \ldots n - 1\), and \(\partial l_n = \{k_n, k_1\}\). The number of independent loops (also known as cyclomatic number) of a connected graph is \(e - v + 1\).

A connected graph is called a tree graph if it contains no loop. A tree graph has \(v - 1\) edges.

### 3 The model

We wish to compute the high-temperature (HT) expansion of the \(q\)-point functions of a generalized Ising model on a \(D\)-dimensional Bravais lattice \(\Lambda\); notice that Bravais lattices enjoy inversion symmetry at each lattice site. The model is defined by the generating functional

\[
\exp(W[h]) = \frac{1}{Z} \prod_i \left[ \int d\phi_i f(\phi_i) \exp(h_i \phi_i) \right] \exp \left( K \sum_{\langle ij \rangle} \phi_i \phi_j \right),
\]

where \(\phi_i\) is a scalar field, \(f\) is an even non-negative function or distribution decreasing faster than \(\exp(-\phi^2)\) as \(\phi \to \infty\), normalized by the condition

\[
\int d\phi f(\phi) = 1,
\]

\(K = \beta J\), the sum runs over all pairs of nearest neighbours, and the normalization \(Z\) is fixed by requiring \(W[0] = 0\).

The connected \(q\)-point function (denoted by \(\mathcal{M}\) in Ref. [2]) at zero magnetic field is defined by

\[
G_q(x_1, \ldots, x_q) \equiv \langle \phi_{i_1} \cdots \phi_{i_q} \rangle^\text{con}\big|_{h=0} = \left. \frac{\partial^n W[h]}{\partial h_{i_1} \cdots \partial h_{i_q}} \right|_{h=0},
\]

where \(x_i \equiv (x_1(i), \ldots, x_D(i))\) is the coordinate vector of the lattice site \(i\). \(G_q\) is invariant under the lattice symmetry group, including (discrete) translations, and under permutation of its arguments; it is customary to write \(G_2(x_i, x_j)\) in the form \(G_2(x_i - x_j)\). We will apply the LCE to the computation of \(G_q\).

### 4 Unrenormalized expansion

Let us parametrize the potential \(f\) in terms of the bare vertices \(\mu_0(2n)\), defined by the generating function

\[
\exp \left[ \sum_n \frac{\mu_0(2n)}{(2n)!} h^{2n} \right] = \int d\phi f(\phi) \exp(h \phi).
\]

These quantities are named bare semi-invariants and denoted by \(M_n^0\) in Ref. [2]; they are named cumulant moments and denoted by \(\mu_{2n}\) in Ref. [1]; \(\mu_0(2n) = (2n - 1)!! \tilde{m}_{2n}^\text{con}\) in the notations of Ref. [1]. Without loss of generality, we can rescale \(\phi\) and \(K\) to fix \(\mu_0(2) = 1\).
For a generic $r$-rooted graph $G$, we define the bare external vertex factor

$$V_0^{(e)}(n_1, ..., n_r; G) = \prod_{i=1}^{r} \mu_0(n(i) + n_i),$$

(4)

the bare internal vertex factor

$$V_0^{(i)}(G) = \prod_{i=r+1}^{v} \mu_0(n(i)),$$

(5)

and the bare edge factor

$$L_0(x_1, ..., x_v; G) = \prod_{l=1}^{e} [K\theta(x_i(l) - x_f(l))],$$

(6)

where $\theta(x) = \delta(\|x\| - 1)$ and $\|x\|$ is the lattice distance between 0 and $x$.

It is convenient to focus on the contributions of $r$-rooted graphs to $q$-point functions. To this purpose we introduce the auxiliary $r$-point functions $X$, whose unrenormalized LCE is

$$X_r(x_1, ..., x_r; n_1, ..., n_r) = \sum_{G \in \mathcal{I}^{(r,0)}} \sum_{x_{r+1}, ..., x_v} V_0^{(e)}(n_1, ..., n_r; G) V_0^{(i)}(G) L_0(x_1, ..., x_v; G) S(G),$$

(7)

where $\mathcal{I}^{(r,0)}$ is the set of all $r$-rooted connected graphs.

$X$ is invariant under simultaneous permutation of coordinates and valences:

$$X_r(x_1, ..., x_r; n_1, ..., n_r) = X_r(x_{\pi(1)}, ..., x_{\pi(r)}; n_{\pi(1)}, ..., n_{\pi(r)}),$$

but not over independent permutation of coordinates and valences. Furthermore, $X$ is invariant over the lattice symmetry group, e.g., it is translation-invariant:

$$X_r(x_1, ..., x_r; n_1, ..., n_r) = X_r(x_1 + x, ..., x_r + x; n_1, ..., n_r).$$

$X_1(x; n)$ is independent of $x$, and it will be denoted by $X_1(n)$; it will also be denoted by $\mu(n)$ in its role of renormalized vertex. $X_2(x_1, x_2; n_1, n_2)$ only depends on the difference $x_2 - x_1$, and it will be denoted by $X_2(x_2 - x_1; n_1, n_2)$. Since, by invariance under space inversion, $X_2(x; n_1, n_2) = X_2(-x; n_1, n_2)$, we also have $X_2(x; n_1, n_2) = X_2(x; n_2, n_1)$.

The sum over the location of internal vertices of the $\theta$ functions is by definition the (free) lattice embedding number of $G$ with fixed roots:

$$\sum_{x_{r+1}, ..., x_v} \prod_{l=1}^{e} \theta(x_i(l) - x_f(l)) = E(x_1, ..., x_r; G).$$

(8)

Therefore

$$X_r(x_1, ..., x_r; n_1, ..., n_r) = \sum_{G \in \mathcal{I}^{(r,0)}} \frac{V_0^{(e)}(n_1, ..., n_r; G) V_0^{(i)}(G) K^{(e)}(G) E(x_1, ..., x_q; G)}{S(G)}. $$

(9)

Finally, the $q$-point functions are computed as:

$$G_q(x_1, ..., x_q) = \sum_{\text{partitions}} X_r(x_{i_11}, ..., x_{i_1r}; u_1, ..., u_r) \prod_{l=1}^{r} \delta_{u_l}(x_{i_1l}, ..., x_{i_ul}),$$

(10)
Figure 1: Example of decomposition of a graph into its 1-skeleton and 1-decorations. Open dots are roots, filled dots are internal vertices.

where the $q$-point delta function is

$$
\delta_1(x_1) = 1, \quad \delta_2(x_1, x_2) = \delta(x_1 - x_2), \quad \ldots, \quad \delta_q(x_1, \ldots, x_q) = \prod_{l=2}^{q} \delta(x_1 - x_l)
$$

and $\{\{i_{11}, \ldots, i_{1u_1}\}, \ldots, \{i_{r1}, \ldots, i_{ru_r}\}\}$ is a generic partition of $\{1, \ldots, q\}$ into $r$ sets of size $u_1, \ldots, u_r$. We will call a root bearing a factor of $\delta_u$ a $u$-th order root.

The two-point function is simply

$$
G_2(x) = X_2(x; 1, 1) + \delta(x) X_1(2).
$$

5 Vertex-renormalized expansion

The graph $G\setminus i$ is obtained by deleting from $G$ the vertex $i$, i.e. by removing $i$ and all edges incident with $i$. A vertex $i$ of a rooted graph $G$ is called an articulation point if there exist vertices of $G\setminus i$ not connected to a root. A rooted graph is called 1-irreducible if it does not contain any articulation point.

Any $r$-rooted ($r > 1$) connected graph $G$ can be decomposed in a unique way into a 1-irreducible $r$-rooted 1-skeleton $S$ and a 1-rooted 1-decoration for each vertex; $G$ is reconstructed by decorating each vertex, identifying the root of its decoration with the vertex; an example is presented in Fig. 1.

Since the only 1-irreducible 1-rooted graph is the single-vertex graph, we use a different definition for 1-rooted graphs. A 1-rooted graph is called a 1-skeleton if it has no articulation points except the root. Any 1-rooted connected graph $G$ can be decomposed in a unique way into a 1-rooted 1-skeleton $S$ and a 1-rooted 1-decoration for each vertex except the root, which is left undecorated.

A 1-rooted connected graph is called a 1-insertion if $G\setminus 1$ is connected.

The LCE can be reorganized by summing together all contributions from graphs having the same 1-skeleton, incorporating 1-decorations into renormalized vertices $\mu(n) = X_1(n)$ (named semi-invariants and denoted by $M_n$ in Ref. [2]; $\mu(n) = (n-1)!!m_n$ in the notations of Ref. [5]). The unrenormalized LCE of $\mu(n)$ is given by Eq. (6).

The $r$-point function can be computed restricting the sum in Eq. (7) or (9) to the (much smaller) set $I^{(r,1)}$ of 1-irreducible $r$-rooted graphs:

$$
X_r(x_1, \ldots, x_r; n_1, \ldots, n_r) = \sum_{G \in I^{(r,1)}} \sum_{x_{r+1}, \ldots, x_v} \frac{V^{(e)}_1(n_1, \ldots, n_r; G) V^{(i)}_1(G) L_0(x_1, \ldots, x_v; G)}{S(G)}
$$
\[
\sum_{G \in \mathcal{I}^{(1,1)}} \frac{V_1^{(e)}(n_1, \ldots, n_r; G) V_1^{(i)}(G) K_{e(G)} E(x_1, \ldots, x_r; G)}{S(G)},
\]

where the internal and external renormalized vertex factors are

\[
V_1^{(e)}(n_1, \ldots, n_r; G) = \prod_{i=1}^{r} \mu(n(i) + n_i)
\]

and

\[
V_1^{(i)}(G) = \prod_{i=r+1}^{v} \mu(n(i)).
\]

Eq. (11) requires a sum over all connected graphs, and therefore it is impractical for the computation of \( \mu(n) \) at large orders of the LCE. We introduce the renormalized moments \( q(n) \) (named self-fields and denoted by \( G_n \) in Ref. [2]), defined by

\[
q(n) = \sum_{G \in \mathcal{I}^{(1,0)}_{n,in}} \sum_{x_2, \ldots, x_v} \frac{V_0^{(i)}(G) L_0(x_1, \ldots, x_v; G)}{S(G)},
\]

where \( \mathcal{I}^{(1,0)}_{n,in} \) is the set of 1-insertions with root of valence \( n \). The following equations hold:

\[
q(n) = \sum_{G \in \mathcal{I}^{(1,1)}_{n,in}} \sum_{x_2, \ldots, x_v} \frac{V_1^{(i)}(G) L_0(x_1, \ldots, x_v; G)}{S(G)},
\]

where \( \mathcal{I}^{(1,1)}_{n,in} = \mathcal{I}^{(1,0)}_{n,in} \cap \mathcal{I}^{(1,1)} \), and \( \mathcal{I}^{(1,1)} \) is the set of 1-rooted 1-skeletons;

\[
\mu(n) = \mu_0(n) + \sum_{s=1}^{\infty} \frac{1}{s!} \sum_{l_1=1}^{\infty} \ldots \sum_{l_s=1}^{\infty} q(l_1) \ldots q(l_s) \mu_0(n + l_1 + \ldots + l_s).
\]

Since \( q(2n - 1) = 0 \), and odd values of \( n \) and \( l_i \) do not contribute to Eq. (11), \( q \) and \( \mu \) can now be computed recursively in parallel order by order in \( K \), since, once Eq. (11) is expanded in powers of \( K \) and truncated, the coefficient of the highest power of \( K \) of \( q \) in the l.h.s. depends only on coefficients of lower powers of \( K \) of \( \mu \) in the r.h.s.

6 Edge-renormalized expansion

A pair of distinct vertices \( i \) and \( j \) of a rooted graph \( G \) is called an articulation pair if there exist vertices of \( G \setminus i \setminus j \) not connected to a root, or if \( i \) and \( j \) are joined by more than one edge. A rooted graph is called 2-irreducible if it does not contain any articulation pair.

Any 1-irreducible \( r \)-rooted \( (r > 2) \) graph \( G \) can be decomposed in a unique way into a 2-irreducible \( r \)-rooted 2-skeleton \( S \) and a 1-irreducible 2-rooted 2-decoration for each edge (oriented in a canonical way, e.g. by choosing \( r(l) < f(l) \)); \( G \) is reconstructed by replacing each edge with its decoration, identifying the first and second decoration root with the initial and final vertex of the edge respectively. An example is shown in Fig. 2.

We use a different definition for 2-rooted graphs, since the only 2-irreducible 2-rooted graph is the bond graph (no internal vertices and only one edge); the roots of all other 1-irreducible 2-rooted graphs are an articulation pair. We call a 1-irreducible 2-rooted graph a 2-skeleton.
if it does not contain any articulation pair except the pair consisting of the two roots. Any 1-irreducible 2-rooted graph can be decomposed in a unique way into a 2-rooted 2-skeleton $S$ and a 1-irreducible 2-rooted 2-decoration for each edge except edges connecting the roots, which are left undecorated.

The LCE can be reorganized by summing together all contributions from graphs having the same 2-skeleton, incorporating all 2-decorations into renormalized edges.

We start by decomposing Eq. (11) for $r > 1$ into

$$W_r(x_1, ..., x_r; n_1, ..., n_r) = \sum_{G \in I(r, 1)} \sum_{n_{r+1}, ..., n_{v}} \frac{V_1^{(i)}(G) L_0(x_1, ..., x_v; G)}{S(G)} ,$$

$$X_r(x_1, ..., x_r; n_1, ..., n_r) = \sum_{s_1, ..., s_r} \left[ \prod_{i=1}^{r} \mu(n_i + s_i) \right] W_r(x_1, ..., x_r; s_1, ..., s_r),$$

where $I(r, 1)$ is the set of 1-irreducible $r$-rooted graphs with roots of valence $n_1, ..., n_r$. $W_r$ enjoys the same symmetry properties of $X_r$. $W_1(x, n)$ is undefined.

$W_r$ can be computed by assigning an initial and a final valence $i_l, f_l$ to each oriented edge of a 2-rooted graph; the valence $i_l$ is incident with $i(l)$ and $f_l$ is incident with $f(l)$:

$$W_r(x_1, ..., x_r; n_1, ..., n_r) = \sum_{G \in I(r, 2)} \sum_{x_{r+1}, ..., x_v; i_1, ..., i_e, f_1, ..., f_e} \prod_{i=1}^{r} \delta(n_i - \nu_i(i_1, ..., i_e, f_1, ..., f_e; G))$$

$$\times \frac{V_2^{(i)}(i_1, ..., i_e, f_1, ..., f_e; G) L_2(x_1, ..., x_v; i_1, ..., i_e, f_1, ..., f_e; G)}{S(G)},$$

$$L_2(x_1, ..., x_v; i_1, ..., i_e, f_1, ..., f_e; G) = \prod_{l=1}^{e} W_2(x_{i(l)} - x_{f(l)}; i_l, f_l).$$
For $r = 2$ Eqs. ([19] and [24]) require a slight modification: we define $I^{(2,2)}$ as the set of 2-rooted 2-skeletons; for edges incident with at least one root, we replace $W_2(x_1 - x_2; i, f)$ with $K\theta(x_1 - x_2)\delta_{i1}\delta_{f1}$.

The sum over graphs can be restricted to a subset of $I^{(r,2)}$; we will discuss here the case $r = 2$; the next Section will be devoted to the case $r \geq 3$. Let us start by classifying 1-irreducible 2-rooted graphs into several classes.

An internal vertex $i$ of a 1-irreducible 2-rooted graph $\mathcal{G}$ is called a nodal point if the roots of $\mathcal{G}\setminus i$ are not connected. A 1-irreducible 2-rooted graph is nodal (also named articulated or separable in the literature) if it contains one or more nodal points; otherwise it is non-nodal.

A 1-irreducible 2-rooted graph is simple if $\mathcal{G}\setminus 1\setminus 2$ is connected, and 1 is not adjacent to 2. By definition, all nodal graphs are simple. A 1-irreducible 2-rooted graph is a ladder graph if it is not simple and it is not the bond graph.

A 1-irreducible 2-rooted graph is elementary if it is both simple and non-nodal.

We have divided 1-irreducible 2-rooted graphs into four disjoint classes: bond, nodal, ladder, and elementary graphs. Let us separate the contributions to $W_2$ according to the four classes:

$$W_2(x; n_1, n_2) = W_2^{bo}(x; n_1, n_2) + W_2^{no}(x; n_1, n_2) + W_2^{la}(x; n_1, n_2) + W_2^{el}(x; n_1, n_2),$$

i.e. bond, nodal, ladder, and elementary contributions respectively.

The bond contribution is trivial. Nodal contributions can be factorized into a product of non-nodal contributions:

$$W_2^{no}(x; n_1, n_2) = \sum_{x_3, i_1, i_2} W_2^{mn}(x_3; n_1, i_1) \mu(i_1 + i_2) W_2^{mn}(x - x_3; i_2, n_2) + \sum_{x_3, x_4; i_1, i_2, i_3, i_4} W_2^{mn}(x_3; n_1, i_1) \mu(i_1 + i_2) \times W_2^{mn}(x - x_3; i_2, i_3) \mu(i_3 + i_4) W_2^{mn}(x - x_4; i_4, n_2) + \ldots, \quad (22)$$

which can be written recursively as

$$W_2^{no}(x; n_1, n_2) = \sum_{x_3; i_1, i_2} W_2^{mn}(x_3; n_1, i_1) \mu(i_1 + i_2) W_2(x - x_3; i_2, n_2). \quad (23)$$

Likewise, ladder contributions can be factorized into a product of non-ladder contributions:

$$W_2^{la}(x; n_1, n_2) = \sum_{s=2}^{\infty} \frac{1}{s!} \sum_{i_1, \ldots, i_s, f_1, \ldots, f_s} \delta(n_1 - \sum_{t=1}^{s} i_t) \delta(n_2 - \sum_{t=1}^{s} f_t) \prod_{t=1}^{s} W_2^{nl}(x, i_t, f_t). \quad (24)$$

Elementary contributions can be computed by setting $r = 2$ into Eq. ([19]), and restricting the sum to $I^{(2,2)}_{el}$, the set of elementary 2-rooted 2-skeletons. The sum can be further restricted to $I^{(2,2)}_{0,el}$, the set of unordered elementary 2-rooted 2-skeletons, provided that we replace $S(\mathcal{G})$ with $S_E(\mathcal{G})$ and we symmetrize the result:

$$W_2^{el}(x; n_1, n_2) = \frac{1}{2} \sum_{\mathcal{G} \in I^{(2,2)}_{0,el}} \sum_{x_3, \ldots, x_v; i_1, \ldots, i_e, f_1, \ldots, f_e} \prod_{l=1}^{2} \delta(n_i - v_l(i_1, \ldots, i_e, f_1, \ldots, f_e; \mathcal{G})) \times \frac{V_2^{(l)}(i_1, \ldots, i_e; \mathcal{G}) L_2(x_1, \ldots, x_v; i_1, \ldots, i_e, f_1, \ldots, f_e; \mathcal{G})}{S_E(\mathcal{G})} + n_1 \leftrightarrow n_2. \quad (25)$$
The last ingredient we need for a fully edge-renormalized expansion are the renormalized vertices $\mu(n)$; they can be computed by combining Eq. (16) with

$$q(n_1+n_2) = \frac{n_1! n_2!}{(n_1+n_2)!} (W_{2}^{\text{no}}(0; n_1, n_2) + W_{2}^{\text{el}}(0; n_1, n_2)), \quad (26)$$

reflecting the fact that the contributions to $q(n_1+n_2)$ in Eq. (15) can be obtained from the contributions to $W_{2}(0; n_1, n_2)$ in Eq. (17) by identifying the roots of the 2-rooted graph, i.e. by suppressing the second root and reattaching all the edges incident with it to the first root, provided that the roots are not adjacent.

Expanding in powers of $K$ Eqs. (23), (24), (25), (26), and (16), we can compute $W_{\text{el}}(x; n_1, n_2)$, $W_{\text{no}}(x; n_1, n_2)$, $W_{\text{la}}(x; n_1, n_2)$, $q(n)$, and $\mu(n)$ in parallel order by order in $K$. The only step which involves a summation over graphs is Eq. (25), where the sum only runs over $I_{(2,2)}^{0,\text{el}}$, a relatively small set.

7 Three-point and higher functions

A vertex (internal or external) $i$ of a 1-irreducible $r$-rooted graph $G$ is called a nodal point if $G\setminus i$ is not connected. By definition of 1-irreducibility, each connected component of $G\setminus i$ must contain at least one root. A 1-irreducible $r$-rooted graph $G$ is nodal if it contains one or more nodal points; otherwise it is non-nodal.

In the rest of this section, we will assume that every graph is 2-irreducible.

A nodal point $j$ of a 2-irreducible $r$-rooted graph $G$ is called a tree-insertion point if at least one of the connected components of $G\setminus j$ is a tree graph. The order $t$ of $j$ is the number of roots of $G$ contained in all tree graph components of $G\setminus j$, plus 1 if $j$ itself is a root. A 2-irreducible $r$-rooted graph is called compact if it contains no tree-insertion points.

Let us consider a 2-irreducible $r$-rooted graph $G$. We generate a compact 2-irreducible graph $G'$, called the compact kernel of $G$, by removing all the tree graphs attached to every tree-insertion point, and promoting all internal tree-insertion points to root. $G$ is obtained by attaching a 2-irreducible tree graph to each root of $G'$. If $G'$ is not a tree graph (and therefore it has at least 3 roots), the decomposition is unique; otherwise, $G$ itself is a tree graph. An example is shown in Fig. 3.

By summing all contribution of 2-irreducible graphs with the same compact kernel, we can write $W_q$ as

$$W_q(x_1, ..., x_q; n_1, ..., n_q) = W_q^{\text{tr}}(x_1, ..., x_q; n_1, ..., n_q)$$

$$+ \sum_{\text{partitions } y_1, ..., y_r; i_1, ..., i_r} \sum_{r>2} W_{r}^{\text{co}}(y_1, ..., y_r; i_1, ..., i_r) \prod_{l=1}^{r} Y_{u_l+1}(y_l; x_{i_{u_l}}, ..., x_{i_{u_l}}; i_l, n_{i_{u_l}}, ..., n_{i_{u_l}}),$$
where $W_{tr}^q(x_1, \ldots, x_q; n_1, \ldots, n_q)$ is the tree graph contribution to $W_q$, $W_{co}^r(y_1, \ldots, y_r; i_1, \ldots, i_r)$ is the compact graph contribution to $W_r$, and

\[
Y_{t+1}(y, x_1, \ldots, x_t; i, n_1, \ldots, n_t) = \sum_j \mu(i + j) W_{tr}^{t+1}(y, x_1, \ldots, x_t; j, n_1, \ldots, n_t) + \sum_j \delta(y - x_1) \delta(n_1 - i - j) W_t^{tr}(x_1, \ldots, x_t; j, n_2, \ldots, n_t),
\]

\[
Y_2(y, x; i, n) = \sum_j \mu(i + j) W_2^{tr}(y, j, n) + \delta(y - x) \delta(n - i)
\]

(28)

(the first and second term correspond to an internal and external $t$-tree-insertion point respectively); notice that $W_2^{tr} = W_2$. Eq. (27) can be combined with Eq. (10) to give

\[
G_q(x_1, \ldots, x_q) = G_{tr}^q(x_1, \ldots, x_q)
\]

\[
+ \sum_{\text{partitions } y_1, \ldots, y_r, i_1, \ldots, i_r} \sum_{r > 2} W_{co}^r(y_1, \ldots, y_r; i_1, \ldots, i_r) \prod_{l=1}^r Z_{u_l+1}(y_l, x_{i_l1}, \ldots, x_{i_{ul}}; i_l),
\]

(29)

where

\[
Z_{t+1}(y, x_1, \ldots, x_t; i) = \sum_{\text{partitions } s_1, \ldots, s_r} \sum_{\{1, \ldots, t\}} Y_{r+1}(y, x_{i_11}, \ldots, x_{i_{r1}}; i, s_1, \ldots, s_r)
\]

\[
\times \prod_{l=1}^r \mu(s_l + u_l) \delta_{u_l}(x_{i_11}, \ldots, x_{i_{ul}}),
\]

(30)

$Z_{t+1}(y, x_1, \ldots, x_t; i)$ is symmetric under permutations of $x_1, \ldots, x_t$ and lattice symmetries, e.g. simultaneous translation of $y$ and $x_1, \ldots, x_t$.

These formulae can be written graphically, according to the rules presented in Table 1. A sum over all dummy coordinates $y$ and all dummy valences $i, f$ and a sum over all inequivalent permutation of external coordinates $x$ or coordinate-valence pairs $x, n$ are understood. Notice that, despite the graphical notation, all pairs of roots of $W_{nn}^q$ and $W_{co}^q$ are equivalent.

Eq. (29) can be expressed by writing all polygons with a letter “c” having 3 to $q$ vertices, placing a crossed dot with a positive integer label at each vertex in all inequivalent ways, the sum of the labels being $q$, and adding the tree contribution. The case $q = 5$ is shown in Fig. 4.

$G_q$ and $Z_q$ can be computed by adding the contributions of all 2-irreducible $r$-rooted tree graphs with roots labelled by positive integers with sum $q$; for $Z_q$, the first root must be drawn as a square. The case $Z_3$ is shown in Fig. 4.
Table 1: Graphical rules. \( \nu \) is the sum of all the valences incident with the vertex. The argument \( z_i \) of \( W_{nn}^q \) is the coordinate \( x \) or \( y \) associated with the symbol placed at the vertex \( i \) of the polygon; the argument \( y_i \) of \( W_{co}^q \) is the variable \( y \) of the function \( Z_{q+1} \) placed at the vertex \( i \).

| symbol | comment | variables | factor |
|--------|---------|-----------|--------|
| ●      | internal vertex | \( y \) | \( \mu(\nu) \) |
| ○      | unlabelled root | \( x, n \) | \( \delta(n - \nu) \) |
| ○      | labelled root | \( x_1, ..., x_q \) | \( \mu(q + \nu + n - 1) \delta_q(x_1, ..., x_q) \) |
| ●      | first root of \( Z \) | \( y, x_1, ..., x_q \) | \( Z_{q+1}(y, x_1, ..., x_q; \nu) \) |
| ●      | root of \( G \) | \( x_1, ..., x_q \) | \( G^r_q(x_1, ..., x_q) \) |
| ○      | \( q \)-sided polygon | \( i, f \) | \( W_2(x_{i(l)} - x_{f(l)}; i, f) \) |
| ○      | \( q \)-sided polygon with a letter “c” | \( z_1, ..., z_q; \nu_1, ..., \nu_q \) | \( W_{nn}^q(z_1, ..., z_q; \nu_1, ..., \nu_q) \) |
| ○      | \( W_{co}^q \) | | |
| ○      | \( G_{tr}^q \) | | |

Figure 5: Contributions to \( Z_3 \).

Figure 6: Compact contributions to \( W_{co}^q \) and corresponding nodal skeletons.

The next step is to write an expression of \( W_{co}^q \) in terms of \( W_{nn}^r \). For \( q = 3 \) we have simply \( W_3^{co} = W_3^{nn} \). The case \( q = 4 \) is shown in Fig. 6. For larger values of \( q \), the number of non-nodal contributions to \( W_{co}^q \) grows rapidly, and a systematic approach is needed.

Let us define for a (connected or non-connected) graph \( G \) and a vertex \( i \) the graph \( G/i \): let \( G^* \) be the connected component of \( G \) containing \( i \); if \( G^* \backslash i \) is connected, set \( G/i = G \); otherwise, for each connected component \( G^* \backslash i \) of \( G^* \backslash i \) generate the graph \( \overline{G}_{k} \) by adding a new vertex, internal or external like \( i \), and joining it to all the vertices adjacent to \( i \) in \( G^* \), by the same number of edges; replace \( G^* \) with the connected components \( \overline{G}_{k} \). The edges and vertices of \( G/i \) are in one-to-one correspondence with the edges and vertices of \( i \), except for the new vertices which all correspond to \( i \) (a nodal point of \( G \)). Notice that \( G/i/j = G/j/i \).

Let \( G \) be a compact 2-irreducible \( r \)-rooted graph (\( r \geq 3 \)) with \( t \) nodal points \( i_1, ..., i_t \). Observe that all the connected components \( G^*_l \) of \( G \equiv G/i_1/.../i_t \) are non-nodal. Generate a new graph \( T \), the nodal skeleton of \( G \), in the following way: for each \( G^*_l \) with \( v \geq 3 \), containing \( n \) roots corresponding to non-nodal roots of \( G \), write a root \( l \) of \( T \) with a label \( n \geq 0 \); for
Figure 7: A compact graph, its non-nodal components, its nodal skeleton, and the contribution to $W^{co}$ of the set of graphs sharing the nodal skeleton.

Figure 8: Nodal skeletons contributing to $W^{co}$.

each node $i_k$ of $G$ write an unlabelled vertex $i_k$ of $T$, internal or external like $i_k$; join $i_k$ to all the labelled roots $l$ such that $G_l$ contain a vertex corresponding to $i_k$, and with all unlabelled vertices which are adjacent to $i_k$ in $G$.

A nodal skeleton is a connected 1-irreducible tree graph, but it is not in general 2-irreducible (it may contain 2-valent internal vertices). A nodal skeleton enjoys the following properties: each 2-valent internal vertex is adjacent to a labelled root; labelled roots are never adjacent; unlabelled roots are at least 2-valent; $m$-valent roots with label $n$ satisfy $n + m \geq 3$. Every nodal skeleton can be generated by adding labels to some of the roots and by splicing 2-valent internal vertices into a 2-irreducible tree graph.

Every 1-irreducible tree graph, with some roots carrying a non-negative integer label, satisfying the above properties, is the nodal skeleton of a set of $q$-rooted 2-irreducible compact graphs, with $q$ equal to the sum of the labels plus the number of unlabelled roots. The contribution of this set to $W_q^{co}$ can be computed by replacing each $m$-valent root of $T$ carrying a label $n$ with an $(n + m)$-sided polygon whose vertices are the $m$ vertices adjacent to the root and $n$ new (unlabelled) roots, and applying the rules of Table 1.

An example of the construction of the nodal skeleton and its evaluation is presented in Fig. 7. The set of all nodal skeletons contributing to $W_5^{co}$ is shown in Fig. 8; see also Fig. 6.
We could carry further the reduction of the set of graphs to be summed over, e.g. by identifying ladder graphs along the lines of Sect. 6. This is rather complicated for arbitrary \( r \), and goes beyond the scope of the present work. Moreover, the zero-momentum projection described below is not applicable to the ladder graph reduction. Therefore we compute \( W_{nn}^r \) by restricting the sum of Eq. (19) to \( \mathcal{I}^{(r,2)}_{nn} \), the set of non-nodal 2-irreducible \( r \)-rooted graphs.

The above considerations can be simplified considerably if we are only interested in moments of the \( q \)-point functions, e.g.

\[
\chi_q = \sum_{x_2, \ldots, x_q} G_q(x_1, \ldots, x_q),
\]

\[
M^{(q)}_2 = \sum_{x_2, \ldots, x_q} (x_1 - x_2)^2 G_q(x_1, \ldots, x_q).
\]

(31)

Let us introduce the moments of \( Z_q, W_q^{co} \) and \( W_q^{nn} \):

\[
\zeta_q(n) = \sum_{x_2, \ldots, x_q} Z_q(x_1, \ldots, x_q; n),
\]

\[
\omega_q^{co}(n_1, \ldots, n_q) = \sum_{x_2, \ldots, x_q} W_q^{co}(x_1, \ldots, x_q; n_1, \ldots, n_q),
\]

(32)

the corresponding definition for \( \chi_q^{tr} \) and \( \omega_q^{nn}(n_1, \ldots, n_q) \), all independent second moments, etc.

Eq. (29) can be projected over zero momentum to give

\[
\chi_q = \chi_q^{tr} + \sum_{r=3}^{q} \frac{q!}{\prod_{l=1}^{r} u_l!} \sum_{i_1, \ldots, i_r} \omega_q^{co}(i_1, \ldots, i_r) \prod_{l=1}^{r} \zeta_u(i_l).
\]

(33)

The computation of \( \zeta_q \), \( \chi_q^{tr} \), and \( \omega_q^{co} \) is also easy; the graphical rules can be immediately projected over zero momentum, suppressing all coordinates \( x \) and \( y \) and removing all space delta functions; the only nontrivial part is the counting of the number of inequivalent permutations of roots.

\( \zeta_q(n) \) can be computed by summing over all inequivalent 2-irreducible 1-ordered \( r \)-rooted tree graphs, with the roots labelled by positive integers \( u_1, \ldots, u_r \) with sum \( q \). The number of inequivalent permutations of the roots 2, \ldots, \( r \) is

\[
\frac{(q - 1)!}{S(u_1 - 1)! \prod_{l=2}^{r} u_l!},
\]

where \( S \) is the symmetry factor of the labelled graph.

The computation of \( \chi_q^{tr}(n) \) is very similar, but we sum over unordered graphs, and the number of inequivalent permutations of the roots is

\[
\frac{q!}{S \prod u_l!}.
\]

\( \omega_q^{co}(n_1, \ldots, n_q) \) can be computed by summing over all inequivalent unordered \( r \)-rooted nodal skeletons, with \( p \) roots labelled by non-negative integers \( u_1, \ldots, u_p \) with \( r - p + u_1 + \ldots + u_p = q \), with a weight \( 1/S \), where \( S \) is the symmetry factor of the labelled graph, with unlabelled roots assigned an arbitrary distinct label (e.g. \(-1\)).

Finally, we can compute \( \omega_q^{nn}(n_1, \ldots, n_q) \) by summing over unordered non-nodal 2-irreducible \( q \)-rooted graphs, provided that we use the modified symmetry factor and we symmetrize the result under permutation of the valences.
The computation of the second moment of the above quantities proceeds along the same lines. The factor \((x_i - x_j)^2\) in Eq. (31) is dealt with in the following way: the two roots are connected by a chain of terms with a space structure of the form

\[
\sum_{x,y_1,...,y_n} (x_i - x_j)^2 f_1(x_i - y_1) f_2(y_1 - y_2) ... f_{n+1}(y_n - x_j)
\]

(we have dropped the dependency on coordinates lying outside the branch connecting \(i\) with \(j\)). Let us write

\[
(x_i - x_j)^2 = (x_i - y_1)^2 + (y_1 - y_2)^2 + ... + (y_n - x_j)^2 + \text{cross terms}.
\]

The cross terms do not contribute to the sum, and the result is

\[
f^{(2)}_1 f^{(0)}_2 ... f^{(0)}_{n+1} + f^{(0)}_1 f^{(2)}_2 ... f^{(0)}_{n+1} + ... + f^{(0)}_1 f^{(0)}_2 ... f^{(2)}_{n+1},
\]

where

\[
f^{(0)}_i = \sum_y f_i(y), \quad f^{(2)}_i = \sum_y y^2 f_i(y).
\]

Therefore we can compute the second moment by taking each contribution to the zero-momentum quantity, promoting one of the zero-momentum factors along the branch connecting the roots \(i\) and \(j\) to second moment, and summing over all possible choices.

By dealing with moments, we avoid the need of storing all the values of \(Z, W^{co},\) and \(W^{nn}\), which can rapidly exhaust all available memory. The extension to higher moments is straightforward but cumbersome.

8 Programming details

We wrote a set of computer programs to implement the automatic evaluation of the edge-renormalized LCE on the simple cubic lattice (sc) and on the body-centered cubic lattice (bcc) (3D); the same programs evaluate the LCE on two different representations of the square lattice (2D) and on the 1D lattice.

The computation of \(q\)-point functions is performed for a generic potential, keeping \(\mu_0(2n)\) symbolic; each term of the series is a polynomial in \(\mu_0(2n)\) with rational coefficients. We also implemented the same computation for a specific potential; this requires much less memory and is somewhat faster (up to 30%), but not enough to give up the flexibility of a generic potential.

To speed up search and insertion into ordered sets of data, graph sets and polynomials in \(\mu_0(2n)\) are implemented as AVL trees (height-balanced binary trees) (cfr. e.g. Ref. 8, Chapt. 6.2.3), using the ubiqx library. Rational numbers and (potentially) large integers are handled by the GNU multiprecision (gmp) library.

Given the complexity of the procedure, it is crucial to perform a number of checks in order to flush out all algorithm and program errors. In 1D our series are compared with exact results for the spin-1/2 and the spin-1 Ising model; this is already a very stringent check, especially of the graph sets (cfr. Ref. 3). In 2D, our results are compared with the series for \(\chi\) and \(M_2\) for spin-1/2 published in Ref. 3. In 3D, our results are compared with the lower-order series already available, for \(\chi\) and \(M_2\) for specific potentials in Refs. 3, 4, 11, and for \(\chi, M_2,\) and \(\chi_q\) for a generic potential in Ref. 6. \(q(n)\) can be computed from different combinations of \(n_1\) and \(n_2\) in Eq. (24); their agreement is non-trivial. Finally, for the spin-1/2 Ising model on any lattice, the series for \(G_q,\) rewritten in terms of \(v = \tanh K,\) must have integer coefficients.
8.1 Graph generation

A program generates the table of all unordered elementary 2-rooted 2-skeletons contributing to the desired order; the algorithm follows Ref. [4]. Starting from the graph drawn in Fig. 9, we apply recursively the following modified Heap rules [12]:

(a) join any two distinct vertices by a new edge, provided the two vertices are not already adjacent;
(b) insert a new internal vertex on any edge and join it to any vertex, excluding the edge extrema, by a new edge;
(c) insert two new internal vertices on any two distinct edges, and join them by a new edge;
(d) do not join the roots by a new edge.

The modifications to the original Heap rules (a), (b), and (c), marked in italics, prevent the generation of 2-reducible graphs. Rule (d) prevents the generation of ladder graphs.

The reduction of graphs to canonical form is performed using a generalization of the algorithm of Ref. [4]. Graphs are stored in a compact form similar to the one of Ref. [4].

To reduce the proliferation of graphs at higher orders, it is extremely important to know the order ("strict bound") \( \alpha_s \) at which a given graph will enter in the expansion (it is not trivially \( \epsilon \), since we require even valence of all internal vertices, and, being interested in bipartite lattices, even length of all loops).

We must also keep in mind that some graphs don’t contribute at the desired order, but graphs generated from them might contribute. We define the “Heap bound” \( \alpha_H(\mathcal{G}) \) as the minimum of \( \alpha_s \) on the set of graphs including \( \mathcal{G} \) and all graphs generated from it. We also define the two bounds for \((\mathcal{G}, \eta)\), i.e. the minimal order when the vertex \( i \) is forced to be embedded in a lattice site of parity \( \eta(i) \).

We apply the modified Heap rules to the graph \( \mathcal{G} \) in the following way: assign a parity label \( \eta(i) \) to each vertex, in all the ways compatible with the Heap bound; apply the modified Heap rules assigning all possible parity labels to the new vertices; discard immediately the generated graph-parity pairs not satisfying the Heap bound; discard vertex parity information and store the generated graphs not isomorph to previously generated graphs. Finally, save into a file only the graphs satisfying the strict bound.

The generation of the elementary 2-rooted 2-skeletons contributing to the 25th order required 41 hours of computation on one CPU of a Compaq ES-40, and ca. 300 Mbytes of RAM. The number of inequivalent elementary 2-skeletons for each order of \( \alpha_s \) is reported in Table 2.

A similar program generates all unordered non-nodal 2-irreducible \( r \)-rooted graphs for \( r \geq 3 \). The table is initialized by applying rules (a) and (b) recursively, starting from each 2-irreducible \( r \)-rooted tree graph, until the result is non-nodal. Rules (a), (b), and (c) are then applied recursively. The number of inequivalent non-nodal 2-irreducible \( r \)-rooted graphs for each order of \( \alpha_s \) is reported in Table 2.

We remark that this graph tables can be used for the LCE of any spin model with \( \phi \rightarrow -\phi \) symmetry on a bipartite lattice in any dimension.

The generation of all the required families of tree graphs is straightforward.
Table 2: Number of inequivalent unordered elementary 2-rooted 2-skeletons ($r = 2$), or unordered non-nodal 2-irreducible $r$-rooted graphs ($r > 2$), for each order of the strict bound $o_s$. The number of graphs satisfying the Heap bound is typically 20% to 30% higher. No graph in any of these sets has a bound lower than 4.

| $o_s$ | $r = 2$ | $r = 3$ | $r = 4$ | $r = 5$ | $r = 6$ | $r = 7$ | $r = 8$ |
|-------|---------|---------|---------|---------|---------|---------|---------|
| 4     | 0       | 1       | 1       | 0       | 0       | 0       | 0       |
| 5     | 0       | 0       | 0       | 0       | 0       | 0       | 0       |
| 6     | 0       | 0       | 1       | 2       | 1       | 0       | 0       |
| 7     | 0       | 1       | 1       | 1       | 1       | 0       | 0       |
| 8     | 1       | 3       | 5       | 4       | 4       | 2       | 1       |
| 9     | 0       | 2       | 4       | 6       | 6       | 5       | 2       |
| 10    | 3       | 7       | 19      | 26      | 27      | 22      | 12      |
| 11    | 0       | 9       | 23      | 47      | 63      | 48      | 33      |
| 12    | 13      | 46      | 111     | 175     | 229     | 228     | 159     |
| 13    | 6       | 54      | 168     | 378     | 603     | 661     | 575     |
| 14    | 59      | 263     | 737     | 1436    | 2224    | 2691    | 2465    |
| 15    | 29      | 367     | 1364    | 3473    | 6404    | 8694    | 9216    |
| 16    | 367     | 1855    | 5824    | 13190   | 23766   | 34106   | 38239   |
| 17    | 197     | 2898    | 12088   | 34726   | 72900   | 116210  | 146284  |
| 18    | 2589    | 14937   | 51801   | 133739  | 275031  | 436976  | 558511  |
| 19    | 1547    | 25332   | 118225  | 375859  | 884317  | 139610  | 173282  |
| 20    | 21682   | 135325  | 514319  | 139610  | 239610  | 448510  | 594317  |
| 21    | 13933   | 245306  | 1251818 | 312207  | 624414  | 116210  | 1516576 |

8.2 Computation of the $q$-point functions

A separate program reads the table of elementary 2-rooted 2-skeletons and computes all components of $W_2$.

The evaluation of $\mu$ and of bond, nodal, and ladder contribution to $W_2$ is a straightforward application of the formulae of Sect. 6.

The evaluation of elementary contribution dominates the computation time, and must be optimized as much as possible. Assume that all lower-order contributions to $W_2$ have been computed. For each unordered elementary 2-skeleton $\mathcal{G}$ with $o_s$ not larger than the desired order, all inequivalent assignations $\mathcal{G}, (n, l)$ of edge valence parity $n$ and length parity $l$ compatible with the desired order, with even length of all loops, and with even valence of all internal vertices, are generated. We have implemented two different algorithms for the computation of the contribution of $\mathcal{G}, (n, l)$ to the two-point function.

The first algorithm is essentially the one used by Nickel and Rehr in Ref. [4]: all inequivalent 1-irreducible 2-rooted graphs with a 2-skeleton compatible with $\mathcal{G}, (n, l)$ are generated, and their contributions are computed according to Eq. (11). In the second algorithm, the contribution is computed according to Eq. (25), and 1-irreducible graphs are not needed. The first algorithm is more efficient for 2-skeletons with large $o_s$, while the second algorithm is more efficient for
small $o_s$; for each value of $o_s$ we select the algorithm which is (presumably) more efficient. On the sc lattice, the speed-up obtained over the use of either algorithm for all skeletons grows with the order, and is about a factor of 4 at order 23. On the bcc lattice, the first algorithm is very efficient, since the embedding number factorizes into a product of 1-dimensional embedding numbers \[4\]; we still use the second algorithm for the simplest 2-skeletons ($o_s \leq 10$), since the computation of the corresponding 1-irreducible 2-rooted graphs contributing to orders higher than 21 is extremely time- and memory-consuming.

Keeping in RAM all components of $W_2$ for a generic potential would be problematic. Most of these components are needed only to compute nodal and ladder contributions, and can be kept on disk; keeping in RAM just the components needed to compute elementary contribution is manageable.

The computation of the 25th-order LCE for the two-point function on the bcc lattice required ca. 400 hours of computation, and ca. 700 Mbytes of RAM.

A similar program reads the table of non-nodal 2-irreducible $r$-rooted graphs and the components of $W_2$, and computes $\omega^m_q$. The computation of $\chi_q$ is then straightforward. We computed $\chi_4$, $\chi_6$, and $\chi_8$ to 21st, 19th, and 17th order respectively on the bcc lattice.

The computation of the same quantities on the sc lattice is much slower (but does not requires more RAM); so far, we obtained $W_2$ to 23th order, with an effort not much smaller than the 25th order on the bcc lattice. The computation of $W_2$ and $\chi_q$ to the same orders as on the bcc lattice is in progress, but it will require a non-trivial amount of time.

9 Selected results

All high-temperature series computed in the present work are available for the most general potential, in the form of polynomials in the bare vertices $\mu_0(2n)$. The general results are extremely lengthy, and are only useful for further computer processing.

We present here a selection of high-temperature series for the spin-1/2 Ising model, i.e. for

$$f(\phi) = \frac{1}{2}(\delta(\phi + 1) + \delta(\phi - 1)).$$

For sake of compactness, all the series are written in terms of $v = \tanh K$. Series for other specific potentials are available upon request from the author.

Although we computed all components of $G_2(x, y)$, we report here only $\chi \equiv \chi_2$ and $M_2 \equiv M_2(2)$ (cfr. Eq. \[(\text{32})\]). For the $q$-point functions, we only computed $\chi_q$.

On the bcc lattice, we obtained

$$\chi = 1 + 8v + 56v^2 + 392v^3 + 2648v^4 + 17864v^5 + 118760v^6 + 789032v^7 + 5201048v^8 + 34268104v^9 + 224679864v^{10} + 1472595144v^{11} + 9619740648v^{12} + 6282314192v^{13} + 409297617672v^{14} + 2665987056200v^{15} + 17333875251192v^{16} + 112680746646856v^{17} + 731466943653464v^{18} + 4747546469665832v^{19} + 3077916675700312v^{20} + 199518218638233896v^{21} + 1292141318087690824v^{22} + 836730042426139624v^{23} + 5414125229349325768v^{24} + 350288350314921653160v^{25} + O(v^{26}); \tag{34}$$
\[ M_2 = 8v + 128v^2 + 1416v^3 + 13568v^4 + 119240v^5 + 992768v^6 + 7948840v^7 \\
+ 61865216v^8 + 470875848v^9 + 3521954816v^{10} + 25965652936v^{11} \\
+ 189180221184v^{12} + 136448929005454632v^{13} + 9757802417152v^{14} \\
+ 69262083278152v^{15} + 488463065172736v^{16} + 342531086090312v^{17} \\
+ 2389602058539312v^{18} + 1659047942692960384v^{19} + 11479047942692960384v^{20} \\
+ 791057961767454248v^{21} + 54332551216709931904v^{22} \\
+ 37203390516123712392v^{23} + 2540342425838560175616v^{24} \\
+ 1730145720711072027840v^{25} + O(v^{26}); (35) \]

\[- \chi_4 = 2 + 64v + 1168v^2 + 16576v^3 + 201232v^4 + 2204608v^5 + 22411504v^6 \\
+ 215447872v^7 + 1981980688v^8 + 17602809920v^9 + 151866668752v^{10} \\
+ 1278888344256v^{11} + 10550227820400v^{12} + 85510907958720v^{13} \\
+ 682500568307184v^{14} + 5374496030148928v^{15} + 41821018545214608v^{16} \\
+ 321992795063663936v^{17} + 2455641803116052752v^{18} + 18567879503614668736v^{19} \\
+ 13931065514229882000v^{20} + 103785402668655887552v^{21} + O(v^{22}); (36) \]

\[ \chi_6 = 16 + 1088v + 36416v^2 + 853952v^3 + 15974528v^4 + 255491264v^5 + 3638767040v^6 \\
+ 47395195712v^7 + 574950589568v^8 + 6581949043264v^9 + 71803170318144v^{10} \\
+ 752047497945024v^{11} + 7606707093034368v^{12} + 74649010982738112v^{13} \\
+ 713458387977120192v^{14} + 6661638582474716480v^{15} + 6092351962198124752v^{16} \\
+ 54692327751320201536v^{17} + 482463182433394315584v^{18} \\
+ 41987611565592990702272v^{19} + O(v^{20}); (37) \]

\[- \chi_8 = 272 + 31744v + 1673728v^2 + 58110976v^3 + 1538207872v^4 + 33584739328v^5 \\
+ 634387677184v^6 + 10699575811072v^7 + 164723097021568v^8 \\
+ 2352360935459840v^9 + 31540880634427392v^{10} + 400802365468148736v^{11} \\
+ 4862781935250449280v^{12} + 56665753776838026240v^{13} \\
+ 637305912177206767104v^{14} + 6945658883867865975808v^{15} \\
+ 73600395257678784586368v^{16} + 76047682319542275111936v^{17} + O(v^{18}). (38) \]

On the sc lattice, we obtained
\[
\chi = 1 + 6v + 30v^2 + 150v^3 + 726v^4 + 3510v^5 + 16710v^6 + 79494v^7 + 375174v^8 \\
+ 1769686v^9 + 8306862v^{10} + 38975286v^{11} + 182265822v^{12} + 852063558v^{13} \\
+ 3973784886v^{14} + 18527532310v^{15} + 86228667894v^{16} + 401225368086v^{17} \\
+ 1864308847838v^{18} + 8660961643254v^{19} + 40190947325670v^{20} \\
+ 186475398518726v^{21} + 86404776466406v^{22} + 4006394107568934v^{23} \\
+ O(v^{24}); \quad (39)
\]

\[
M_2 = 6v + 72v^2 + 582v^3 + 4032v^4 + 25542v^5 + 153000v^6 + 880422v^7 + 4920576v^8 \\
+ 26879670v^9 + 144230088v^{10} + 762587910v^{11} + 3983525952v^{12} + 20595680694v^{13} \\
+ 105558845736v^{14} + 536926539990v^{15} + 2713148048256v^{16} + 13630071574614v^{17} \\
+ 68121779384520v^{18} + 3389583104998v^{19} + 167899803744448v^{20} \\
+ 8287136476787862v^{21} + 40764741656730408v^{22} + 19990133482355526v^{23} \\
+ O(v^{24}). \quad (40)
\]

Acknowledgments

We would like to thank Michele Caselle and Ettore Vicari for many useful discussions. Special thanks are due to Paolo Rossi, who worked out exact results for the 1D spin-1 Ising model, and to Andrea Pelissetto, for his critical reading of the manuscript.

References

[1] A. J. Guttmann, in *Phase Transitions and Critical Phenomena*, vol. 13, C. Domb and J. Lebowitz eds. (Academic Press, New York, 1989), p. 1.

[2] M. Wortis, in *Phase Transitions and Critical Phenomena*, vol. 3, C. Domb and M. S. Green eds. (Academic Press, London, 1974), p. 113.

[3] B. G. Nickel, in *Phase Transitions, Cargèse 1980*, M. Levy, J. C. Le Guillou, and J. Zinn-Justin eds. (Plenum, NY, 1982), p. 291.

[4] B. G. Nickel and J. J. Rehr, J. Stat. Phys. **61**, 1 (1990).

[5] M. Lüscher and P. Weisz, Nucl. Phys. **B300**, 325 (1988).

[6] M. Campostrini, A. Pelissetto, P. Rossi, and E. Vicari, Phys. Rev. **E60**, 3526 (1999).

[7] J. W. Essam and M. E. Fisher, Rev. Mod. Phys. **42**, 272 (1970).

[8] D. E. Knuth, *The art of computer programming*, vol. 3 (Addison-Wesley publishing co., Reading, 1973).

[9] M. Campostrini, A. Pelissetto, P. Rossi, and E. Vicari, Nucl. Phys. **B459**, 207 (1996).

[10] P. Rossi, private communication.
[11] P. Butera and M. Comi, Phys. Rev. B\textbf{56}, 8212 (1997).

[12] B. R. Heap, J. Math. Phys. \textbf{7}, 1582 (1966).