Divide and conquer the Hilbert space of translation-symmetric spin systems

Alexander Weiße
*Max-Planck-Institut für Mathematik, Vivatsgasse 7, 53111 Bonn, Germany**
(Dated: October 5, 2012, revised: January 14, 2013)

Iterative methods that operate with the full Hamiltonian matrix in the untrimmed Hilbert space of a finite system continue to be important tools for the study of one- and two-dimensional quantum spin models, in particular in the presence of frustration. To reach sensible system sizes such numerical calculations heavily depend on the use of symmetries. We describe a divide-and-conquer strategy for implementing translation symmetries of finite spin clusters, which efficiently uses and extends the "sublattice coding" of H. Q. Lin. With our method, the Hamiltonian matrix can be generated on-the-fly in each matrix vector multiplication, and problem dimensions beyond 10^{11} become accessible.

PACS numbers: 02.70.-c, 02.20.-a, 75.10.Jm, 05.30.-d

I. INTRODUCTION

Lattice spin models have attracted continuous research activity, from the early days of quantum mechanics until the present. Unfortunately, only very few spin models can be solved analytically and in the thermodynamic limit, with geometries usually restricted to one dimension [1, 2]. Numeric methods, therefore, are indispensable for an understanding of quantum spin models. Density matrix renormalisation [3] and related variational approaches have revolutionised the study of one-dimensional systems and are able to deliver very precise results for eigenstates and correlation functions. However, two-dimensional spin systems, systems with frustrated interactions [4], and dynamic correlations in such systems are still a domain for "exact" iterative methods that operate with the full Hamiltonian matrix of a finite cluster. Taking advantage of symmetries increases the accessible cluster size, which is crucial for significant results.

In this work we consider spin models on finite lattices with periodic boundary conditions and describe an efficient approach for the construction of a translation symmetric basis of the corresponding Hilbert space. The underlying ideas are related to divide-and-conquer strategies and fast Fourier transform. The core decomposition trick we use was invented more than two decades ago by H. Q. Lin [5], but for unknown reasons did not really catch on. For many years the symmetrised Hilbert space dimensions reached in studies of quantum spin models therefore lagged behind, compared to simulations of Hubbard-type models or electron-phonon models.

II. STANDARD APPROACH

Let us start with a short review of the common method for the construction of translation symmetric spin states and the performance issues connected to it. Consider a quantum spin model with translation symmetry on a one-dimensional lattice with n sites and periodic boundary conditions, $\vec{s}_n \equiv \vec{s}_0$,

$$H = \sum_{i,j} J_{j-i}\ \vec{s}_i \cdot \vec{s}_j\,. \tag{1}$$

The Hamiltonian H commutes with the total spin and its components $S^\alpha = \sum_{i=0}^{n-1} s_i^\alpha$, $\alpha \in \{x, y, z\}$, and with the translation operator

$$T : \vec{s}_i \to \vec{s}_{i+1}\,. \tag{2}$$

Assuming local spins with amplitude $|\vec{s}_i| = 1/2$, the Hilbert space of the n-site system is the product of n two-dimensional spaces and has dimension 2^n. Using the conservation of S^z, this space can be decomposed into $n+1$ subspaces corresponding to the eigenvalues of S^z, $-n/2, -n/2+1, \ldots, n/2$. Fixing $u = S^z + n/2$, a subspace is spanned by all products of u up-spins and $n-u$ down-spins (the two eigenstates of s_i^z) and its dimension is $\binom{n}{u}$, where obviously $\sum_{u=0}^{n}\binom{n}{u} = 2^n$. On a computer, these states are usually represented as bit patterns of length n, and the above decomposition is equivalent to grouping patterns according to their digit sum. Understanding bit patterns as integers defines an order and allows for the construction of ordered lists, which can be efficiently searched for specific patterns.

The above Hamiltonian also conserves the amplitude of the total spin, $\vec{S}^2$, but the construction of the corresponding eigenstates is more involved and rarely adopted in numeric computations on finite clusters. Instead, lattice symmetries are used to further decompose spaces of given S^z into smaller subspaces. As the title implies, in this work we focus on the translation symmetry.

Since the Hamiltonian H commutes with the translation T, the matrix elements of H between different eigenspaces of T vanish. Thus, if H is expressed in an orthonormal basis of eigenstates of T, the original problem splits into n independent pieces, each having a dimension roughly a factor of n smaller. The projection operator

$$P_k = \frac{1}{n}\sum_{j=0}^{n-1} e^{2\pi i j k/n} T^j \tag{3}$$

* weisse@mpim-bonn.mpg.de

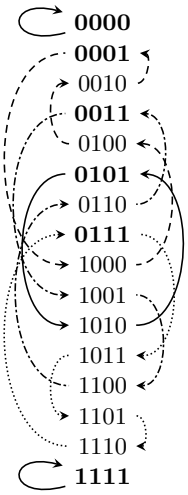

Figure 1. Decomposition of the set of S^z eigenstates with $n = 4$ into orbits. Arrows indicate the action of the translation T. The six representatives are shown in bold face.

maps an arbitrary state onto an eigenstate of T with eigenvalue $\exp(-2\pi ik/n)$, namely

$$\begin{aligned} TP_k|\psi\rangle &= \frac{1}{n}\sum_{j=0}^{n-1} e^{2\pi ijk/n} T^{j+1}|\psi\rangle \\ &= \exp(-2\pi ik/n)\, P_k|\psi\rangle\,. \end{aligned} \tag{4}$$

Here we used $T^n = 1$, which also implies $\exp(-2\pi ik) = 1$ and $k \in \mathbb{Z}$. Due to the periodicity there are only n distinct projectors and it suffices to consider the momenta $k = 0$, $1, \ldots, (n-1)$.

The projector P_k maps states, which are related to each other by an arbitrary translation, $|\phi\rangle = T^j|\psi\rangle$, onto the same eigenstate of T (up to a phase factor). To avoid this ambiguity and to obtain a symmetrised basis we need to partition the set of all S^z eigenstates, $\mathcal{S}$, into orbits, i.e., disjoint subsets $\mathcal{S}_r$ that are closed under the translation group,

$$\forall|\psi\rangle \in \mathcal{S}_r :\ T^j|\psi\rangle \in \mathcal{S}_r\,. \tag{5}$$

Each orbit can be represented by one of its elements. For instance, if we use bit patterns to represent S^z eigenstates, we can sort all elements of an orbit by the corresponding integer values and choose the smallest as the representative of the orbit. All other members of this orbit are obtained from the representative by applying all translations T^j, $j = 1, \ldots, (n-1)$. Figure 1 illustrates this decomposition for the case $n = 4$, where $\mathcal{S}$ contains $2^4 = 16$ elements which belong to 6 disjoint orbits. The sizes of the orbits differ, since some of the bit patterns are invariant under non-trivial subgroups of the full translation group of the n-site lattice. The orbit size is then given by n divided by the order of the subgroup.

Let us denote the set of all representatives with $\mathcal{R}$ and the size of the orbit of $|r\rangle \in \mathcal{R}$ with ω_r. Then, for given momentum k a translation symmetric basis of the Hilbert space is formed by the states

$$|k, r\rangle = \sqrt{\nu_{k,r}}\ P_k|r\rangle,\ \ |r\rangle \in \mathcal{R}\,. \tag{6}$$

The prefactor $\sqrt{\nu_{k,r}}$ ensures that $|k, r\rangle$ is normalised, $\langle k, r|k, r\rangle = 1$, or it is zero, which means that for momentum k the representative $|r\rangle$ does not contribute to the basis. More precisely,

$$\nu_{k,r} = \begin{cases} 0 & \text{if } n/\omega_r \text{ divides } k, \\ \omega_r & \text{otherwise.} \end{cases} \tag{7}$$

Of course, we can restrict the set $\mathcal{R}$ to states with a given eigenvalue of S^z (i.e., patterns with given digit sum), which yields a basis that makes use of both symmetries of H, S^z, and T.

III. THE PROBLEM

The recipe for the construction of the symmetrised basis $\{|k, r\rangle\}$ does not look particularly complicated. However, following it becomes very time-consuming for large n. To find all representatives $\mathcal{R}$ we can loop over the 2^n eigenstates of S^z in $\mathcal{S}$ (or at least the $\binom{n}{S^z+n/2}$ eigenstates of fixed S^z), apply all n translations, and check if the considered bit pattern has the minimal integer value within its orbit. If this is the case, it qualifies for the set $\mathcal{R}$. The process can be improved by memorising bit patterns that were already encountered in previous orbits and apply the translations only to new ones. Still, we need to perform of the order of 2^n translations and store all the patterns.

The performance issues become more serious once we start to calculate matrix elements of the Hamiltonian H with respect to the basis $\{|k, r\rangle\}$,

$$\begin{aligned} \langle k, r'|H|k, r\rangle &= \sqrt{\nu_{k,r'}\nu_{k,r}}\ \langle r'|P_k H P_k|r\rangle \\ &= \sqrt{\nu_{k,r'}\nu_{k,r}}\ \langle r'|P_k H|r\rangle\,. \end{aligned} \tag{8}$$

The application of H on a representative $|r\rangle$ yields many different bit patterns—usually their number is some multiple of the lattice size n. In general, these bit patterns are not representatives, and we need to find the orbit they belong to as well as the translation, which maps the pattern to the representative of its orbit. The latter tells us which part of the projector P_k contributes to the matrix element, in particular, which phase factor. If we apply all translations to all bit patterns generated by H and then look up the observed representatives in a list, the construction of the (sparse) matrix representation of H requires huge amounts of bit operations and processing time.

In contrast, for lattice models such as the Hubbard model or electron-phonon models, the Hilbert space is the product of subspaces which can be symmetrised individually. The subspaces are small enough such that orbit representatives can be identified through simple

n	Dimension
36	252 088 496
38	930 138 522
40	3 446 167 860
42	12 815 663 844
44	47 820 447 028
46	178 987 624 514

Table I. Problem dimensions for $S^z = 0$, $k = 0$ as a function of the chain length n.

table look-ups, and it is common practise to construct the Hamiltonian matrix on-the-fly in each step of an iterative calculation. Methods such as the Lanczos eigenvalue solver [6] then need memory only for a few vectors with the dimension of the Hilbert space, and huge problems can be studied.

With the standard approach for quantum spin models this is impractical. Instead, the matrix has to be kept in memory or stored on disk. The former limits the accessible system sizes, and the latter is not efficient either, since disk access is slow and the matrix dimensions are huge (cf. Table I).

The program SPINPACK [7], which employs many symmetries and is frequently used to calculate the lowest eigenstates of spin models, follows the above strategies, and the problem of identifying the orbit and representative for a given bit pattern seriously limits its performance. The authors of the code even considered implementing the required bit operations with specialised hardware based on field-programmable gate arrays (FPGA) [8]. The authors of Ref. [9], on the other hand, argue that the overhead for using symmetries outweighs the benefits of the reduced problem dimensions, and in their large-scale exact diagonalisation study make no use of translation symmetries.

Below we resolve all these issues and present an answer to the following problem: *Find a fast and memory-efficient algorithm, which for a given arbitrary bit pattern identifies the orbit the pattern belongs to and the translation that maps it to the representative of this orbit.*

IV. DIVIDE-AND-CONQUER APPROACH

A. The basic idea

Let us assume that the number of lattice sites n is even. We can then divide the set of all sites into two subsets of equal size, such that the neighbours of a given site all belong to the other sublattice. The translation T of the entire lattice is then decomposed into two operations: the exchange of the two sublattices and a translation within one of the two. In Figure 2 we illustrate this concept for a lattice of $n = 8$ sites.

Such a decomposition—termed "sublattice coding"—was used by H. Q. Lin [5] to construct the orbit representatives in exact diagonalisation studies of translation

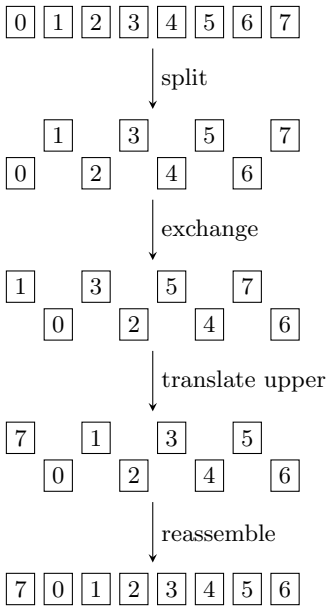

Figure 2. The action of a translation on a lattice decomposed into two intertwined sublattices.

symmetric spin clusters with up to 32 sites. Unfortunately, the description of his algorithm is rather brief and details of the implementation remain vague. Therefore, the potential of this trick seems to have been missed (see, e.g., the discussion of Ref. [5] in Refs. [10, 11]).

An improved implementation of the Lin decomposition was proposed by Schulz, Ziman, and Poilblanc in Ref. [12]. Here an arbitrary spin state is decomposed into its sublattice states and the representatives of the sublattice orbits are determined through look-ups in moderately sized tables. The operation that maps one sublattice state to its representative is then applied to the other sublattice, which in most cases (80 % according to Ref. [12]) yields the correct orbit representative of the full lattice. However, an ambiguity remains and additional symmetry operations can be necessary to identify the correct representative.

In what follows we explain our interpretation and extension of the Lin approach, which has been used for a couple of years and efficiently handles very large spin systems. We properly decompose all symmetry operations and for each given spin state arrive at a unique orbit representative purely through look-ups in moderately sized tables. There is no need to apply symmetry operations to spin states on the full lattice, i.e., the above ambiguity is resolved.

As a starting point, we formalise the above decomposition of lattice translations by introducing the "zipper product" of two bit patterns $|a\rangle = (a_0, \ldots, a_{n/2-1})$ and $|b\rangle = (b_0, \ldots, b_{n/2-1})$,

$$(a_0, \ldots, a_{n/2-1}) \circledcirc (b_0, \ldots, b_{n/2-1}) := (a_0, b_0, \ldots, a_{n/2-1}, b_{n/2-1}) \,. \quad (9)$$

For clarity, we indicate the size of the translated lattice as an index to the translation operator, T_n, and assume that it translates patterns to the right,

$$T_n(a_0, \ldots, a_{n-1}) = (a_{n-1}, a_0, \ldots, a_{n-2})\,. \tag{10}$$

Then, the above procedure can be summarised as

$$T_n(|a\rangle \circledcirc |b\rangle) = (T_{n/2}\,|b\rangle) \circledcirc |a\rangle\,. \tag{11}$$

This equation relates the translations on the n-site lattice to the translations on the $n/2$-site sublattice, and we can use it to derive orbits and representatives of the full lattice from the orbits and representatives of the sublattice. Multiple application of T_n on a state $|a\rangle \circledcirc |b\rangle$ yields

$$\begin{aligned} T_n^{2j}(|a\rangle \circledcirc |b\rangle) &= (T_{n/2}^j|a\rangle) \circledcirc (T_{n/2}^j|b\rangle)\,, \\ T_n^{2j+1}(|a\rangle \circledcirc |b\rangle) &= (T_{n/2}^{j+1}|b\rangle) \circledcirc (T_{n/2}^j|a\rangle)\,, \end{aligned} \tag{12}$$

with $j = 0, \ldots, (n/2-1)$, which illustrates how the orbit of $|a\rangle \circledcirc |b\rangle$ is built from the sublattice orbits of $|a\rangle$ and $|b\rangle$.

Let us now consider two representatives of the sublattice $|r\rangle, |r'\rangle \in \mathcal{R}_{n/2}$ subject to two conditions: First, $r < r'$, where the order is defined in terms of the integer value of the bit patterns. Second, the orbits of both representatives have maximal size $n/2$, i.e., $T_{n/2}^j|\psi\rangle \neq |\psi\rangle$ $\forall j = 1, \ldots, (n/2-1)$ and $|\psi\rangle \in \{|r\rangle, |r'\rangle\}$. Then, the $n/2$ states

$$|r, r', j\rangle = |r\rangle \circledcirc (T_{n/2}^j|r'\rangle) \text{ with } j = 0, \ldots, (n/2-1) \tag{13}$$

are representatives for orbits of the full translation group on the n-site lattice. Since the translation T_n involves the exchange of the two sublattices (see Eq. 11), the orbit of $|r, r', j\rangle$, which is given by $T_n^i|r, r', j\rangle$ with $i = 0, \ldots, (n-1)$, contains all $n^2/2$ states that can be obtained by combining the sublattice orbits of $|r\rangle$ and $|r'\rangle$, namely $(T_{n/2}^j|r\rangle) \circledcirc (T_{n/2}^l|r'\rangle)$ *and* $(T_{n/2}^l|r'\rangle) \circledcirc (T_{n/2}^j|r\rangle)$, with $j, l = 0, \ldots, (n/2-1)$. This explains the above condition $r < r'$.

For $r = r'$, the range of j needs to be reduced,

$$|r, r, j\rangle = |r\rangle \circledcirc (T_{n/2}^j|r\rangle) \text{ with } j = 0, \ldots, \lfloor (n-1)/4 \rfloor\,, \tag{14}$$

as otherwise we would count states twice. Note, that for $n/2$ odd one of the generated representatives is invariant under $T_n^{n/2}$, i.e., the orbit has size $n/2$ only.

The general case, where $r \leq r'$ and $|r\rangle$ or $|r'\rangle$ are invariant under non-trivial subgroups of the $n/2$-site translation group, leads to further restrictions on the values of j. In addition, the generated representatives $|r, r', j\rangle$ will be invariant under subgroups of the n-site translation group. It is then convenient to identify all subgroups of the $n/2$-site translations (which correspond to the divisors of $n/2$) and to tabulate all possible combinations and the resulting restrictions on j. These tables are small and can also hold other basic details, like the orbit size $\omega_{r,r',j}$ of $|r, r', j\rangle$, which does not depend on r and r' directly but only on the maximal subgroups that $|r\rangle$ and $|r'\rangle$ are invariant under.

What are the advantages of the decomposition into two sublattices? First, the number of representatives of the sublattice is approximately equal to the square root of the number of representatives of the full lattice, $|\mathcal{R}_{n/2}| \approx \sqrt{|\mathcal{R}_n|}$. Hence, the construction of $\mathcal{R}_n$ from $\mathcal{R}_{n/2}$ is much faster than the traditional approach we described earlier. Second, $|\mathcal{R}_{n/2}|$ is small enough, that we can store in memory the map $|\psi\rangle \to T_{n/2}^j|r\rangle$ from an arbitrary state $|\psi\rangle \in \mathcal{S}_{n/2}$ to its representative $|r\rangle \in \mathcal{R}_{n/2}$ and the corresponding exponent j. Moreover, we can use these tables to directly identify the representative $|r\rangle \in \mathcal{R}_n$ and the exponent j for an arbitrary state $|\psi\rangle \in \mathcal{S}_n$ on the full lattice. Hence, we can solve the problem of Section III with a few look-ups in moderately sized tables (typically a few megabytes).

B. Implementation

We start from a lattice with $n/2$ sites and determine all subgroups of the translation group generated by $T_{n/2}$. They are given by the divisors of $n/2$, namely, if $d \mid n/2$ then $T_{n/2}^d$ generates a subgroup. For example, setting $n/2 = 4$ we find three subgroups indexed with g,

g	$d = \omega_g$	Example
0	1	0000
1	2	0101
2	4	0001

(15)

which match the three different orbit types shown in Figure 1 and the corresponding orbit sizes ω_g.

Next, we construct representatives for all orbits in $\mathcal{S}_{n/2}$. Since we are dealing with only half of the target lattice size n, we can use the approach sketched in the first paragraph of Section III. For each representative $|r\rangle \in \mathcal{R}_{n/2}$, we also determine the maximal subgroup g it is invariant under, i.e., we find the minimal non-zero $d \mid n/2$ such that $T_{n/2}^d|r\rangle = |r\rangle$. For the example $n/2 = 4$, we obtain

r	$\lvert r\rangle$	d	g
0	0000	1	0
1	0001	4	2
2	0011	4	2
3	0101	2	1
4	0111	4	2
5	1111	1	0

(16)

Having selected a set of representatives $\mathcal{R}_{n/2}$, we can tabulate the map $|\psi\rangle \to T_{n/2}^j|r\rangle$, i.e., we can construct an array which takes the integer value of a bit pattern as the index and returns both, the index r of the corresponding representative $|r\rangle$ and the exponent j. For the example

$n/2 = 4$ this looks as follows:

$\psi \equiv \lvert\psi\rangle$	r	j
0 0000	0	0
1 0001	1	0
2 0010	1	3
3 0011	2	0
4 0100	1	2
5 0101	3	0
6 0110	2	3
7 0111	4	0
8 1000	1	1
9 1001	2	1
10 1010	3	1
11 1011	4	1
12 1100	2	2
13 1101	4	2
14 1110	4	3
15 1111	5	0

(17)

Knowing all details about the sublattice with $n/2$ sites, we can construct a set of tables, which characterise the symmetrised states of the full n-site lattice. Consider an arbitrary state $|a\rangle \circledS |b\rangle$ on the full lattice: For both sublattice states, $|a\rangle$ and $|b\rangle$, we can immediately look up the indices r_a and r_b of corresponding representatives $|r_a\rangle$ and $|r_b\rangle$, as well as the exponents j_a and j_b. In addition, given r_a and r_b we know the subgroups g_a and g_b of the representatives $|r_a\rangle$ and $|r_b\rangle$. The only information missing for locating $|a\rangle \circledS |b\rangle$ within its orbit,

$$|a\rangle \circledS |b\rangle = T_n^i |r, r', j\rangle = T_n^i \left[|r\rangle \circledS (T_{n/2}^j |r'\rangle) \right] , \quad (18)$$

are the exponents i and j. These exponents depend only on the values of j_a, j_b, g_a, and g_b, and on the order of r_a and r_b, namely, whether $r_a < r_b$, $r_a = r_b$ or $r_a > r_b$. When we defined the representatives $|r, r', j\rangle$ of the full lattice in Eq. (13), we demanded $r < r'$. Thus, if we encounter $r_a > r_b$, then $|a\rangle \circledS |b\rangle$ is created from the representative $|r_b, r_a, j\rangle$ by a translation T_n^i with odd exponent i. For $r_a < r_b$, the representative is $|r_a, r_b, j\rangle$ and the exponent i is even. For $r_a = r_b$ other restrictions apply, as discussed in the paragraph of Eq. (14). We can tabulate all possible cases in three arrays,

$$\begin{aligned} e^< &: j_a, j_b, g_a, g_b \to i, j \,, \\ e^= &: j_a, j_b, g \to i, j \,, \\ e^> &: j_a, j_b, g_a, g_b \to i, j \,. \end{aligned} \quad (19)$$

At first glance these four-dimensional arrays appear large, but the j indices take only $n/2$ different values, and the g indices even fewer. In the Appendices A 1 to A 3 we show the maps $e^<$, $e^=$, and $e^>$ for the lattice with $n = 8$ sites. To build the arrays we perform a double-loop over the subgroups in Eq. (15), which fixes g_a and g_b. Then, for each combination of subgroups we pick two matching representatives, r_a and r_b, and loop in reverse order over i and j in Eq. (18). Looking up the exponents j_a and j_b of the resulting state $|a\rangle \circledS |b\rangle$ in Eq. (17) completes the data required for the arrays. In particular, for each input j_a, j_b, g_a, and g_b, the stored values of i, j are the minimal ones.

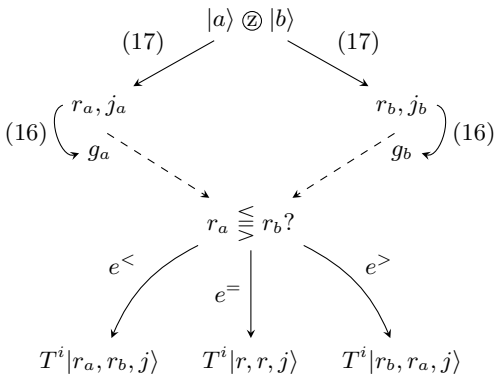

Figure 3. Schematic view of the table look-ups (solid lines) required to locate an arbitrary state $|a\rangle \circledS |b\rangle$ within its orbit.

In Figure 3 we summarise the algorithm to identify both the orbit of an arbitrary state $|a\rangle \circledS |b\rangle$ and its translation relative to the orbit's representative. Let us remark that, in general, the representative $|r, r', j\rangle$ is not the state with minimal integer value within its orbit. Using direct table look-ups, this property is no longer needed, and in our programs the components of zipped states are usually stored in separate variables.

The tables $e^<$ and $e^=$ can also be used to identify the values of j, for which $|r, r', j\rangle$ is a valid representative. In this case the set $(j_a = 0, j_b = j, g_a, g_b)$ is mapped to $(i = 0, j)$, i.e., $|r, r', j\rangle$ is not part of the orbit of some other representative $|r, r', j'\rangle$ with $j' < j$. We can store this information together with the size of the orbit of $|r, r', j\rangle$, which depends only on the corresponding subgroups g and g'. Similar to the previous arrays, we need to distinguish two cases, $\omega^<_{g,g',j}$ for $r < r'$ and $\omega^=_{g,j}$ for $r = r'$. If j is invalid, we set ω to zero, otherwise it will have some integer value $d \mid n$. In Appendix A 4 we list the latter quantities for $n = 8$.

In analogy to Eqs. (6) and (7), we now know the normalised, translation-symmetric basis states of the n-site lattice,

$$|k, r, r', j\rangle = \sqrt{\nu_{k,r,r',j}}\, P_k |r, r', j\rangle \,, \quad (20)$$

where

$$\nu_{k,r,r',j} = \begin{cases} 0 & \text{if } n/\omega \text{ divides } k \\ \omega & \text{otherwise} \end{cases} \quad (21)$$

and ω is $\omega^<_{g,g',j}$ or $\omega^=_{g,j}$, respectively. Note that for large n the number of representatives with $\omega < n$ is negligible compared to those with $\omega = n$ introduced in Eq. (13). Therefore, in a practical calculation a loop over the whole basis can include all $r \le r'$ and $j = 0, \ldots, (n/2 - 1)$, and

the few inactive states with $\omega = 0$ will waste hardly any resources.

In the preceding paragraphs we did not take into account the S^z symmetry of the original spin model (1). However, its inclusion is easy: When constructing the representatives of the sublattice, $\mathcal{R}_{n/2}$, we also calculate the S^z eigenvalue of each $|r\rangle$. Then, for the representatives of the full lattice, $|r, r', j\rangle$, we combine only those r and r', whose spin values add to the desired S^z of the full lattice. This requires a little extra book keeping, but does not affect the overall performance.

V. GENERALISATIONS

A. Two-dimensional lattices

Up to now we considered only one-dimensional lattices, but the generalisation to two dimensions is straightforward. Again we demand $n = n_x \times n_y$ to be even. Hence, one or both of n_x and n_y are even, and we can apply the decomposition into sublattices along one of the two space directions. Another option is the decomposition into a chequerboard pattern, which can also be used for quadratic clusters with rotated unit cell [13] and an even number of sites fulfilling $n = n_1^2 + n_2^2$ with $n_1, n_2 \in \mathbb{Z}$. The main condition for the decomposition is that the sublattices each have the same translation group. In Figure 4 we show the lattice with $20 = 5 \times 4$ sites decomposed along the y-direction and the lattice with $10 = 3^2 + 1^2$ sites decomposed in chequerboard fashion.

The construction of representatives for the orbits of the full-lattice translation group then follows the route described in Section IV. Merely the number and structure of the subgroups of the translation group differs slightly, since now the group is generated by two commuting elementary translations T_x and T_y. Also, the condition for vanishing norm $\nu_{k,r,r',j}$ is more complicated and will usually be tabulated.

B. Reflection symmetry

Apart from being translation symmetric, most of the considered quantum spin models are also invariant under reflections, i.e., the full lattice symmetry is described by the dihedral group or, in higher dimensions, by products thereof. In one dimension the reflection operator reads

$$R : \vec{s}_i \to \vec{s}_{n-1-i} \,. \tag{22}$$

It is fully compatible with the lattice decomposition introduced in Section IV, since R can be written as reflections of both sublattices and exchange of the two,

$$R(|a\rangle \circledcirc |b\rangle) = (R|b\rangle) \circledcirc (R|a\rangle) \,. \tag{23}$$

Hence, the reflections can be incorporated into the divide-and-conquer approach and used for a further reduction

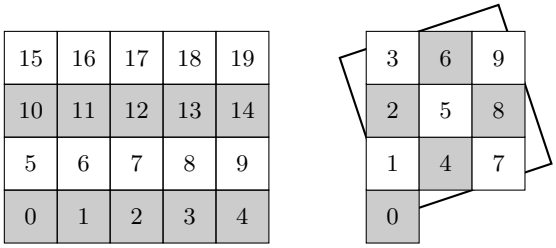

Figure 4. The decompositions of lattices with $20 = 5 \times 4$ and with $10 = 3^2 + 1^2$ sites into two sublattices.

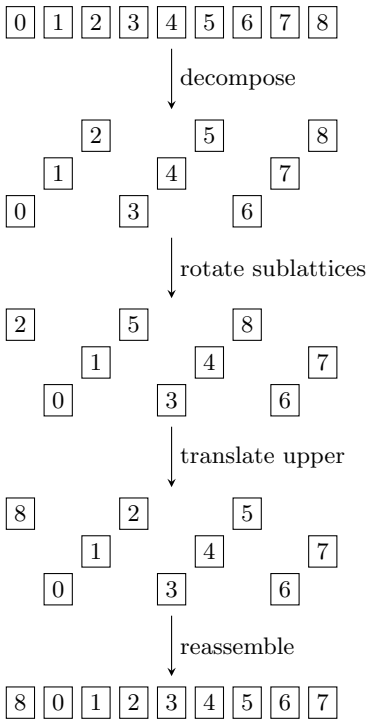

Figure 5. The action of a translation on a lattice decomposed into three sublattices of equal size.

of the Hilbert space dimension, or to make the matrix representation real [10].

C. Odd lattice size

The key prerequisite for the decompositions presented in the preceding sections, is the even number of lattice sites. In practise, this condition is not particularly restrictive, since many of the quantum spin models studied are anti-ferromagnetic. Fitting long-range order or correlations of this type into a finite cluster usually requires even n. However, lattices with an odd number of sites could be of interest for certain interaction types, geometries, or spin amplitudes other than one-half.

As long as n is not prime, we can take a small factor

n	S_z	$\frac{2\pi k}{n}$	Dimension	$E_{\min}$	Cores	Memory	Time / MVM	SPINPACK build matrix
1×32	0	0	18 784 170	-14.2065274389	1	290 MB	10.76 s	$\approx$ 280 s
1×34	0	π	68 635 478	-15.0912964656	1	1 GB	50.03 s	$\approx$ 1100 s
1×36	0	0	252 088 496	-15.9762358220	1	3.8 GB	225.15 s	$\approx$ 4600 s
6×6	0	$(0,0)$	252 091 362	-24.4393973993	1	3.8 GB	342.43 s	
1×36	0	0	252 088 496	-15.9762358220	16	7.18 GB	65.32 s	
6×6	0	$(0,0)$	252 091 362	-24.4393973993	16	7.20 GB	112.64 s	
1×38	0	π	930 138 522	-16.8613184638	16	26.2 GB	270.63 s	
1×40	0	0	3 446 167 860	-17.7465227882	24	97.1 GB	1299.15 s	
1×40	0	0	3 446 167 860	-17.7465227883	64	76.1 GB	174.53 s	
1×42	0	π	12 815 663 844	-18.6318313306	256	249 GB	207.19 s	
1×44	0	0	47 820 447 028	-19.5172298175	1024	1209 GB	322.60 s	
1×46	0	π	178 987 624 514	-20.4027064699	1024	3522 GB	1172.84 s	

Table II. Lanczos calculations of the ground state of the Heisenberg model using one core of a desktop computer with Intel Xeon 5150 processors at 2.66 GHz, many cores of a compute server with 8 quad-core Opteron 8384 processors at 2.7 GHz, and several nodes of a high-performance cluster with Power6 processors at 4.7 GHz. The eighth column shows the time required for one matrix vector multiplication (MVM), which includes on-the-fly generation of the Hamiltonian matrix. For comparison, in the last column we list the time SPINPACK 2.43 needs to generate the matrix (stored in RAM for these system sizes).

$m \mid n$ and split the lattice into m equal sublattices, such that the left and the right neighbor of a site belongs to the previous and the next sublattice, respectively. An arbitrary translation of the full lattice then corresponds to a cyclic permutation of the sublattices and internal translations within the sublattices.

Consider, for instance, a lattice where the number of sites is a multiple of 3. We can decompose this lattice into 3 sublattices, as illustrated in Figure 5. Now, the translation of the full lattice by a single site is equivalent to a cyclic permutation of the three sublattices and an internal translation within one of the three, or

$$T_n(|a\rangle \circledcirc |b\rangle \circledcirc |c\rangle) = (T_{n/3}|c\rangle) \circledcirc |a\rangle \circledcirc |b\rangle\,. \qquad (24)$$

Knowing the representatives $|r\rangle$ for the sublattice with $n/3$ sites, we can build representatives for the full lattice,

$$|r, r', r'', j', j''\rangle = |r\rangle \circledcirc (T_{n/3}^{j'}|r'\rangle) \circledcirc (T_{n/3}^{j''}|r''\rangle)\,. \qquad (25)$$

The restrictions needed to avoid double counting are more intricate, compared to the bi-partition. There are 6 permutations of 3 objects, and the cycle (123) connects the even and the odd permutations among each other. Thus, starting from $r \le r' \le r''$ or $r > r' > r''$ we can reach all possible combinations of sublattice states. The exponents, in general, can take all values $j', j'' = 0, \ldots, (n/3-1)$, but some will be switched off with appropriate norm factors, if two or all representatives are equal or have a higher symmetry. When we construct the corresponding tables of the orbit sizes $\omega_{g,g',g'',j',j''}$, we need to differentiate between a number of different cases. The three states r, r', and r'' can all be different and arranged in ascending or descending order, there can be equal pairs, or all three can be the same. Similarly, the three tables $e^<$, $e^=$, and $e^>$, which for the bi-partite lattice were sufficient to identify the orbit and the phase factor of an arbitrary state on the full lattice, now generalise to a whole set of tables covering all possible orderings of r, r', and r''.

As yet we did not have a good incentive to study lattices with an odd number of sites and, therefore, cannot comment on the performance of this setup. An implementation of the decomposition into three sublattices appears feasible, but the benefits of decompositions into five or more sublattices seem to be rather limited.

D. Higher spin

The translation symmetry of the lattice and the structure of the local Hilbert space at each site are more or less independent. Therefore, the construction of the translation symmetric basis can easily be extended to systems with spins of amplitude larger than 1/2. The efficient storage of the map from sublattice states to sublattice representatives, $|\psi\rangle \to T_{n/2}^j|r\rangle$, may require some care. Otherwise, all steps of the algorithm work as described above.

VI. PERFORMANCE

We implemented the divide-and-conquer approach for spin-1/2 chains and rectangular two-dimensional lattices ($n = n_x \times n_y$) already a few years ago, and used it mainly for the study of correlation functions. The latter can be efficiently calculated using Chebyshev expansion methods [14, 15], which at their core require fast matrix vector multiplications. For example, we calculated a set of static correlation functions [16] and the dynamic ESR-response [17, 18] of the one-dimensional XXZ model at finite temperature and finite magnetic field.

Of course, the described basis construction can also be used in Lanczos calculations of low-energy eigenstates. To give an impression of the performance of the algorithm, in Table II we show the time and memory consumption of

several ground-state calculations for the Heisenberg model on one- and two-dimensional lattices. Taking into account the translation and the S_z symmetries, the Hamilton matrix is computed on-the-fly in each iteration. For the momenta considered the matrix is real. Systems with up to 36 sites can be simulated on desktop computers or powerful laptops, as illustrated by the single-core data for an older Xeon CPU. For systems with up to 40 sites, we use a compute server with eight quad-core CPUs, and on a decent high-performance cluster [19] we are able to handle systems with 46 sites, corresponding to a matrix dimension of 1.8×10^{11}. The main limiting parameter for these calculations is the memory required for two double vectors with the dimension of the Hilbert space. On the largest clusters currently available one could certainly study systems with 50 sites, which requires approximately 40 TB of memory and is well below present limits.

A direct comparison of our timings with SPINPACK is difficult, since this code usually precomputes the entire Hamilton matrix and stores it in memory or on disk for later use in the Lanczos recursion. In the last column of Table II we show matrix generation times for not too large systems, where the matrix fits into available memory. These calculations take much longer than the matrix vector multiplication (MVM) in our approach, which includes on-the-fly matrix generation.

VII. SUMMARY

We present an efficient algorithm to construct translation symmetric basis states for quantum spin models on finite lattices with periodic boundary conditions. The approach extends an old trick by H. Q. Lin [5] and employs a divide-and-conquer strategy, such that direct table lookups can be used to map an arbitrary spin state to its orbit with respect to the translation group. The Hamiltonian matrix, which in iterative calculations like Lanczos or Chebyshev expansion needs to be applied repeatedly to a few quantum states, can then be constructed on-the-fly. This saves large amounts of memory or disk space and considerably increases the system size accessible to these types of simulations.

We thank Rechenzentrum Garching of the Max Planck Society for providing computing time on their high-performance clusters.

While this article was under review we learned of unpublished exact diagonalisation results [20] for systems with 48 spins (dimension 2.5×10^{11}), which were obtained with an extended version of the method in Ref. [12].

[1] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*, updated ed. (Dover, Mineola, NY, 2008) ISBN 978-0486462714

[2] B. M. McCoy, *Advanced Statistical Mechanics*, International Series of Monographs on Physics, Vol. 146 (Oxford Univ. Press, Oxford, 2010) ISBN 978-0-19-955663-2

[3] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992)

[4] *Introduction to Frustrated Magnetism: Materials, Experiments, Theory*, edited by C. Lacroix, P. Mendels, and F. Mila, Springer Series in Solid-State Sciences, Vol. 164 (Springer, Heidelberg, 2011)

[5] H. Q. Lin, Phys. Rev. B **42**, 6561 (1990)

[6] C. Lanczos, J. Res. Nat. Bur. Stand. **45**, 255 (1950)

[7] J. Schulenburg, "SPINPACK," Software package, `http://www-e.uni-magdeburg.de/jschulen/spin/index.html`

[8] J. Schulenburg, "SPINPACK using FPGA," Notes on FPGA implementation of permutations, `http://www-e.uni-magdeburg.de/jschulen/spin/fpga.html`

[9] A. M. Läuchli, J. Sudan, and E. S. Sørensen, Phys. Rev. B **83**, 212401 (2011)

[10] A. W. Sandvik, in *Lectures on the Physics of Strongly Correlated Systems XIV*, AIP Conference Proceedings, Vol. 1297, edited by A. Avella and F. Mancini (Amer. Inst. Phys., 2010) pp. 135–338, ISBN 978-0-7354-0851-7, 14th Training Course in the Physics of Strongly Correlated Systems, Salerno, ITALY, Oct 05-16, 2009

[11] A. M. Läuchli, in Lacroix *et al.* [4], pp. 481–511

[12] H. J. Schulz, T. A. L. Ziman, and D. Poilblanc, J. Phys. I France **6**, 675 (1996)

[13] J. Oitmaa and D. D. Betts, Can. J. Phys. **56**, 897 (1978)

[14] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske, Rev. Mod. Phys. **78**, 275 (2006)

[15] A. Weiße and H. Fehske, in *Computational Many-Particle Physics*, Lecture Notes in Physics, Vol. 739, edited by H. Fehske, R. Schneider, and A. Weiße (Springer, Heidelberg, 2008) pp. 545–577

[16] H. E. Boos, J. Damerau, F. Göhmann, A. Klümper, J. Suzuki, and A. Weiße, J. Stat. Mech., P08010(August 2008), `http://www.iop.org/EJ/abstract/1742-5468/2008/08/P08010/`

[17] M. Brockmann, F. Göhmann, M. Karbach, A. Klümper, and A. Weiße, Phys. Rev. Lett. **107**, 017202 (2011)

[18] M. Brockmann, F. Göhmann, M. Karbach, A. Klümper, and A. Weiße, Phys. Rev. B **85**, 134438 (2012)

[19] On the November 2012 TOP500 list the Power6 cluster at RZ Garching was ranked number 307. It is now replaced by a more powerful system., `http://top500.org/system/176520`

[20] A. Läuchli, R. Johanni, and R. Moessner, "An exact diagonalization perspective on the S=1/2 Kagome Heisenberg antiferromagnet," Talk at KITP on October 31, 2012, `http://online.kitp.ucsb.edu/online/fragnets12/laeuchli/`

Appendix A: Tables for $n = 8$

1. The map $e^{<} : j_a, j_b, g_a, g_b \to (i, j)$

$e^{<}$	g_b	0				1				2			
g_a	$j_a \backslash j_b$	0	1	2	3	0	1	2	3	0	1	2	3
0	0	(0,0)	-	-	-	(0,0)	(2,0)	-	-	(0,0)	(2,0)	(4,0)	(6,0)
	1	-	-	-	-	-	-	-	-	-	-	-	-
	2	-	-	-	-	-	-	-	-	-	-	-	-
	3	-	-	-	-	-	-	-	-	-	-	-	-
1	0	(0,0)	-	-	-	(0,0)	(0,1)	-	-	(0,0)	(0,1)	(4,0)	(4,1)
	1	(2,0)	-	-	-	(2,1)	(2,0)	-	-	(6,1)	(2,0)	(2,1)	(6,0)
	2	-	-	-	-	-	-	-	-	-	-	-	-
	3	-	-	-	-	-	-	-	-	-	-	-	-
2	0	(0,0)	-	-	-	(0,0)	(0,1)	-	-	(0,0)	(0,1)	(0,2)	(0,3)
	1	(2,0)	-	-	-	(2,1)	(2,0)	-	-	(2,3)	(2,0)	(2,1)	(2,2)
	2	(4,0)	-	-	-	(4,0)	(4,1)	-	-	(4,2)	(4,3)	(4,0)	(4,1)
	3	(6,0)	-	-	-	(6,1)	(6,0)	-	-	(6,1)	(6,2)	(6,3)	(6,0)

2. The map $e^{=} : j_a, j_b, g \to (i, j)$

	$g = 0$				$g = 1$				$g = 2$			
$j_a \backslash j_b$	0	1	2	3	0	1	2	3	0	1	2	3
0	(0,0)	-	-	-	(0,0)	(3,0)	-	-	(0,0)	(0,1)	(5,1)	(7,0)
1	-	-	-	-	(1,0)	(2,0)	-	-	(1,0)	(2,0)	(2,1)	(7,1)
2	-	-	-	-	-	-	-	-	(1,1)	(3,0)	(4,0)	(4,1)
3	-	-	-	-	-	-	-	-	(6,1)	(3,1)	(5,0)	(6,0)

3. The map $e^{>} : j_a, j_b, g_a, g_b \to (i, j)$

$e^{>}$	g_a	0				1				2			
g_b	$j_b \backslash j_a$	0	1	2	3	0	1	2	3	0	1	2	3
0	0	(1,0)	-	-	-	(3,0)	(1,0)	-	-	(7,0)	(1,0)	(3,0)	(5,0)
	1	-	-	-	-	-	-	-	-	-	-	-	-
	2	-	-	-	-	-	-	-	-	-	-	-	-
	3	-	-	-	-	-	-	-	-	-	-	-	-
1	0	(1,0)	-	-	-	(1,1)	(1,0)	-	-	(5,1)	(1,0)	(1,1)	(5,0)
	1	(3,0)	-	-	-	(3,0)	(3,1)	-	-	(7,0)	(7,1)	(3,0)	(3,1)
	2	-	-	-	-	-	-	-	-	-	-	-	-
	3	-	-	-	-	-	-	-	-	-	-	-	-
2	0	(1,0)	-	-	-	(1,1)	(1,0)	-	-	(1,3)	(1,0)	(1,1)	(1,2)
	1	(3,0)	-	-	-	(3,0)	(3,1)	-	-	(3,2)	(3,3)	(3,0)	(3,1)
	2	(5,0)	-	-	-	(5,1)	(5,0)	-	-	(5,1)	(5,2)	(5,3)	(5,0)
	3	(7,0)	-	-	-	(7,0)	(7,1)	-	-	(7,0)	(7,1)	(7,2)	(7,3)

4. Sizes of the orbits represented by $|r, r', j\rangle$

		$\omega^{<}_{g,g',j}$									$\omega^{=}_{g,j}$		
	g	0			1			2			0	1	2
j	g'	0	1	2	0	1	2	0	1	2	0	1	2
0		2	4	8	4	4	8	8	8	8	1	4	8
1		0	0	0	0	4	8	0	8	8	0	0	8
2		0	0	0	0	0	0	0	0	8	0	0	0
3		0	0	0	0	0	0	0	0	8	0	0	0