



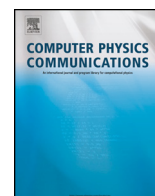
OIST

OKINAWA INSTITUTE OF SCIENCE AND TECHNOLOGY GRADUATE UNIVERSITY
沖縄科学技術大学院大学

Quantum spin solver near saturation: QS3

Author	Hiroshi Ueda, Seiji Yunoki, Tokuro Shimokawa
journal or publication title	Computer Physics Communications
volume	277
page range	108369
year	2022-04-27
Publisher	Elsevier B.V.
Rights	(C) 2022 The Authors.
Author's flag	publisher
URL	http://id.nii.ac.jp/1394/00002506/

doi: [info:doi/10.1016/j.cpc.2022.108369](https://doi.org/10.1016/j.cpc.2022.108369)



Quantum spin solver near saturation: QS³ ^{☆,☆☆}

Hiroshi Ueda ^{a,b,c,*}, Seiji Yunoki ^{c,d,e,f}, Tokuro Shimokawa ^g



^a Center for Quantum Information and Quantum Biology, Osaka University, Toyonaka, 560-0043, Japan

^b JST, PRESTO, Kawaguchi, 332-0012, Japan

^c Computational Materials Science Research Team, RIKEN Center for Computational Science (R-CCS), Kobe, 650-0047, Japan

^d Computational Condensed Matter Physics Laboratory, RIKEN Cluster for Pioneering Research (CPR), Wako, 351-0198, Japan

^e Computational Quantum Matter Research Team, RIKEN Center for Emergent Matter Science (CEMS), Wako, 351-0198, Japan

^f Quantum Computational Science Research Team, RIKEN Center for Quantum Computing (RQC), Wako, 351-0198, Japan

^g Theory of Quantum Matter Unit, Okinawa Institute of Science and Technology Graduate University, Onna, 904-0495, Japan

ARTICLE INFO

Article history:

Received 13 July 2021

Received in revised form 1 April 2022

Accepted 2 April 2022

Available online 20 April 2022

Keywords:

Exact diagonalization

Lanczos method

Quantum spin system

Dilute hardcore Bose system

Magnetism

Dynamical structure factor

ABSTRACT

We develop a program package named QS³ [kǰú:-és-kǰú:b] based on the (thick-restart) Lanczos method for analyzing spin-1/2 XXZ-type quantum spin models on spatially uniform/non-uniform lattices near fully polarized states, which can be mapped to dilute hardcore Bose systems. All calculations in QS³, including eigenvalue problems, expectation values for one/two-point spin operators, and static/dynamical spin structure factors, are performed in the symmetry-adapted bases specified by the number N_{\downarrow} of down spins and the wave number \mathbf{k} associated with the translational symmetry without using the bit representation for specifying spin configurations. Because of these treatments, QS³ can support large-scale quantum systems containing more than 1000 sites with dilute N_{\downarrow} . We show the benchmark results of QS³ for the low-energy excitation dispersion of the isotropic Heisenberg model on the $10 \times 10 \times 10$ cubic lattice, the static and dynamical spin structure factors of the isotropic Heisenberg model on the 10×10 square lattice, and the open-MP parallelization efficiency on the supercomputer (Ohtaka) based on AMD Epyc 7702 installed at the Institute for the Solid State Physics (ISSP). Theoretical backgrounds and the user interface of QS³ are also described.

Program summary

Program Title: QS³

CPC Library link to program files: <https://doi.org/10.17632/yc6jytfv8v.1>

Developer's repository link: <https://github.com/QS-Cube/ED>

Licensing provisions: MIT

Programming language: Fortran90

External routines/libraries: BLAS, LAPACK

Nature of problem: Physical properties (such as total energy, magnetic moment, two-point spin correlation function, and dynamical structure factor)

Solution method: Application software based on the full diagonalization method and the exact diagonalization method using the Lanczos and thick-restart Lanczos techniques for quantum spin $S = 1/2$ models such as the XXZ model.

Restrictions: Spin $S = 1/2$ systems with U(1) symmetry.

Unusual features: Massively large quantum spin systems with U(1) symmetry near the saturation field can be solved with or without considering translational symmetry, which is difficult to treat using standard exact diagonalization libraries with the bit representation for specifying spin configurations.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

[☆] The review of this paper was arranged by Prof. Weigel Martin.

^{☆☆} This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author at: Center for Quantum Information and Quantum Biology, Osaka University, Toyonaka, 560-0043, Japan.

E-mail address: ueda.hiroshi.qiqb@osaka-u.ac.jp (H. Ueda).

1. Introduction

The exact diagonalization (ED) method [1] is a traditional approach and indeed one of the most powerful numerical methods to correctly understand the nature of quantum many-body systems from finite-size cluster calculations. This method enables us

to obtain directly our target eigenvectors and the corresponding eigenvalues of Hamiltonian \hat{H} and to evaluate all of the static, dynamical, and thermal properties without any bias by storing in physical memory of computers several vectors of the size as large as the dimension of the Hamiltonian matrix. Moreover, in contrast to a quantum Monte Carlo method, which in general suffers from the so-called negative sign problem for fermion and quantum frustrated spin systems, the ED method does not have such kind of difficulty. Therefore, the ED method has been used at the forefront in the research field of quantum many-body systems.

The only disadvantage of the ED method is that the accessible system size is severely limited to small clusters. A direct way to alleviate this disadvantage is to introduce the message-passing-interface (MPI) parallelization [2] and adapt the Hamiltonian symmetry [3]. For examples, sophisticated ED program packages such as ALPS [4–6], SpinPack [7], Rokko [8], and $\mathcal{H}\Phi$ [9] are designed with MPI techniques for high performance on large-scale supercomputers and use U(1) and/or lattice symmetry to blockdiagonalize the Hamiltonian matrix. Indeed, these large-scale computations with ingenious ways have led to new findings and many discoveries in the long history of studies in the quantum many-body systems, some of which can be found, e.g., in Refs. [10–19].

Nonetheless, the accessible system sizes are still small because the required computational resources increase exponentially with the system size. For example, for a spin $S=1/2$ Heisenberg model without magnetic field, the conventional Lanczos algorithm [20] can treat up to 50 spins by fully using the power of a large modern supercomputer [21]. In some particular cases, the system size limitation to small clusters may not be an issue, e.g., in strong random systems [22–27] and in the high-temperature limit, where the typical correlation lengths are usually small. However, the accessible system size becomes crucial when, for example, gapless or incommensurate phases are discussed in various intriguing quantum many-body systems.

Let us now remind a distinct advantage of the U(1) symmetry adaption for a quantum many-body system that can be mapped to a dilute particle system with a conserved number of particles, e.g., an $S = 1/2$ XXZ model near the saturation field. In such a system, the ED method can treat much larger system sizes with 10^3 sites and more, in principle, because the required matrix dimension is scaled as $O(N^{N_\downarrow})$, instead of $O(\exp(N))$, where N is the system size and N_\downarrow is the number of particles. A protocol in previous ED program packages [7,9,28–30] is to use a bitwise operation and thus the power of the U(1) symmetry is not fully exploited. While the bitwise operation is essential to speedup higher-level arithmetic operations, it requires that all basis states for expressing the Hamiltonian matrix are represented as binary numbers. Therefore, we cannot express the basis for more than 64 sites [31] in a single binary number, e.g., when an $S = 1/2$ spin model is treated on 64-bit processors.

Considering the fact that no ED program packages are currently available (with one exception), each researcher has to individually implement their own computational program to perform numerical diagonalization for analyzing quantum spin systems near saturated magnetization with more than 64 sites [32–34]. We should however note that although it is not well known, the ALPS numerical diagonalization library can treat quantum spin systems with more than 64 sites, but the details of the implementation are not documented in the ALPS manual [4] and its original paper [5,6]. In order to facilitate the further development of the methodology in this research field, it is highly desirable to establish a numerical diagonalization library for large-scale spin systems near the saturation by providing a systematically explained algorithm with the open-source library that anyone can easily use.

In this paper, we develop an ED program package named QS³ (Quantum Spin Solver near Saturation). This package can treat

$S=1/2$ XXZ spin models near the saturation field where the U(1) symmetry is preserved with small $N_\downarrow = O(1)$. This is a symmetry adapted open source ED package developed without using the bit representation. QS³ also adapts the lattice symmetry of the Hamiltonian and thus it is able to treat $O(10^3)$ sites near the saturation field. A prototype of QS³ has been already used for analyzing the ground state phase diagram of $S=1/2$ XXZ spin models on the triangular lattice near the saturation field with system sizes up to 1296 sites [34].

The QS³ code is based on the Lanczos and thick-restart Lanczos methods [35,36] to calculate the low-energy eigenvalues and the corresponding eigenvectors of the Hamiltonian matrix, and is supported by the external libraries BLAS and LAPACK [37]. Available physical quantities are the local magnetization, two-point correlation functions, static spin structure factors, and dynamical spin structure factors. For the calculation of the dynamical spin structure factors, the continued fraction expansion is employed with the Lanczos method [1,38,39].

The QS³ code is specialized to the analysis for quantum magnets under a high magnetic field, dilute hard-core Bose gases, and low-energy properties of ferromagnets. Thus, it is particularly beneficial, for example, to the study of a field-induced spin nematic state [40–51], which often emerges in a high magnetic field. The QS³ code can also be used to estimate magnetic couplings of effective spin Hamiltonians for some materials by directly comparing the spin excitations calculated numerically and measured experimentally by inelastic neutron scattering in a sufficiently high magnetic field. Therefore, the QS³ package is useful for both theoretical and experimental researchers. The QS³ package is designed to be executed on generally available computing resources such as laptops and small workstations, and therefore only OpenMP is used for parallelizations. The QS³ package provides several samples for demonstration calculating physical quantities of $S=1/2$ XXZ models on three different lattices, the square, triangular, and cubic lattices, which preserve the translational symmetry. The QS³ package also supports the analysis for systems without the translational symmetry.

The rest of this paper is organized as follows. The basic usage of the QS³ package is first described in Sec. 2. The algorithms implemented in QS³ are then explained in Sec. 3. In Sec. 4, benchmark results on the square and cubic lattices are provided, and the bottlenecks and characteristics of the QS³ calculations are discussed. Finally, the paper is summarized with brief discussion of future extension of the QS³ package in Sec. 5.

2. Basic usage of QS³

2.1. How to download and build QS³

The QS³ package, containing the Fortran source codes, samples, and manual, is available on GitHub (<https://github.com/QS-Cube/ED>). For building QS³, Fortran compiler with BLAS/LAPACK library [37] is prerequisite.

For those who have their own Git accounts, simply clone the repository on their local computers:

```
$ git clone https://github.com/QS-Cube/ED.git
```

Otherwise, go to the web page and click the “Code” button and “Download ZIP” to get “ED-main.zip”. The zip file is unpacked as

```
$ unzip ED-main.zip
$ cd ED-main
```

A simple Makefile is provided to build the executable files “QS3.exe” for systems preserving the translational symmetry and

“QS3_only_u1.exe” for systems without the translational symmetry. The following procedures after the cloning or downloading will generate the executable file and execute sample programs

```
$ cd script
$ ./make.sh
```

Before executing make.sh, open this file and select the compiler (ARG1=gfortran/fort) and linear algebra library (ARG2=lapack/mkl). After executing samples, each result is stored in the separate directory, “output_ex1”, “output_ex2”, ..., and “output_ex5”.

2.2. Model

The QS³ package can treat the following $S=1/2$ XXZ-type spin Hamiltonian:

$$\hat{\mathcal{H}} = \sum_{r < r'} \{ J_{r,r'}^{xy} (\hat{S}_r^x \hat{S}_{r'}^x + \hat{S}_r^y \hat{S}_{r'}^y) + J_{r,r'}^z \hat{S}_r^z \hat{S}_{r'}^z \} - h^z \sum_{r=1}^N \hat{S}_r^z, \quad (1)$$

where $\hat{\mathbf{S}}_r = (\hat{S}_r^x, \hat{S}_r^y, \hat{S}_r^z)$ is a spin $S = 1/2$ operator at site r on an N -site cluster, $J_{r,r'}^{z(xy)}$ is the $z(xy)$ component of the two-body exchange interaction between the r th and r' th spins, and h^z is the uniform magnetic field applied along the z direction. Since the Hamiltonian $\hat{\mathcal{H}}$ commutes with the z component of the total spin, i.e., $[\hat{\mathcal{H}}, \sum_r \hat{S}_r^z] = 0$, the U(1) symmetry is preserved. We also assume that the Hamiltonian $\hat{\mathcal{H}}$ is translationally invariant under periodic boundary conditions. In the QS³ package, the lattice structures and the range of the exchange interactions can be varied as long as the U(1) and translational symmetries are preserved. The details will be described below.

2.3. How to use QS³

Here we explain in detail how to use QS³ by providing a concrete and simple example of the $S = 1/2$ isotropic Heisenberg model, i.e., $J_{r,r'}^{xy} = J_{r,r'}^z = J_1$ when sites r and r' are nearest neighbors and $J_{r,r'}^{xy} = J_{r,r'}^z = 0$ otherwise, on the 6×6 square lattice shown in Fig. 1(a). We calculate the 10 lowest eigenvalues and the corresponding eigenvectors in the subspace of $\langle \sum_r \hat{S}_r^z \rangle = 15$ and momentum $\mathbf{k} = (k_x, k_y) = (0, 0)$ by using the thick-restart Lanczos algorithm. Here, $\langle \sum_r \hat{S}_r^z \rangle$ is the expectation value of the z component of the total spin with respect to an eigenvector. We also calculate the local magnetization and the two-point correlation function from the obtained eigenvectors, and the dynamical spin structure factor $S^+(\mathbf{q}, \omega)$ at wave vector $\mathbf{q} = (q_x, q_y) = (0, 0)$ by means of the continued fraction method.

2.3.1. Set a main input file

One should first create several input files in the “input_ex1” directory. A concrete example of the main input file, input.dat, for the $S=1/2$ isotropic Heisenberg model on the 6×6 square lattice is shown below.

```
&input_parameters
NOS = 36,
NOD = 3,
LX = 6,
LY = 6,
LZ = 1,
KX = 0,
KY = 0,
KZ = 0,
NOxxz = 72,
ALG = 2,
```

```
cal_lm = 1,
cal_cf = 1,
cal_dsf = 1,
wr_wf = 1,
re_wf = 0,
FILExxz = "input_ex1/list_xxz_term_36.dat",
FILEwf = "work/",
OUTDIR = "output_ex1/",
&end
&input_static
NOV = 2,
NOLM = 36,
NOCF = 1296,
FILElm = "input_ex1/list_local_mag.dat",
FILECF = "input_ex1/list_cf_ss.dat",
&end
&input_dynamic
spsmsz = 1,
itr_dsf = 200,
QX = 0.0d0,
QY = 0.0d0,
QZ = 0.0d0,
rfield = 0.495d0,
FILEpos = "input_ex1/list_site_position_36_type1.dat",
&end
&input_lancz
lnc_ene_conv0 = 1.0d-14,
minitr = 20,
maxitr = 10000,
itrint = 5,
&end
&input_TRLan
NOE = 10,
NOK = 15,
NOM = 30,
maxitr = 10000,
lnc_ene_conv = 1.0d-14,
i_vec_min = 1,
i_vec_max = 1,
&end
```

The main input file given above consists of five parts, input_parameters, input_static, input_dynamic, input_lancz, and input_TRLan. The meaning of each part and the variables used there are explained below.

input_parameters

This part requires the users to set fundamental conditions, number of spins, number of down spins, linear dimensions of the cluster, momentum sector, location of the input file specifying the lattice structure and the exchange interactions, locations of outputs for the results, and an algorithm for the calculation of eigenvalues/eigenvectors. The details of the variables are explained below.

NOS (INTEGER): Number N of spins.

NOD (INTEGER): Number N_\downarrow of down spins. The users can select the z component of the total spin, $M = (N/2 - N_\downarrow)$, by adjusting this valuable.

LX, LY, LZ (INTEGER): Linear dimensions L_x , L_y , and L_z of the cluster in the x , y , and z directions.

KX, KY, KZ (INTEGER): Momentum sector K_x , K_y , and K_z . The users should set a allowed momentum value that is compatible with the cluster size and shape, $\mathbf{k} \cdot \mathbf{a}_\alpha = 2\pi K_\alpha / L_\alpha$ with $\alpha \in \{x, y, z\}$, where \mathbf{k} is the momentum and \mathbf{a}_α is the α component of the primitive translation vectors.

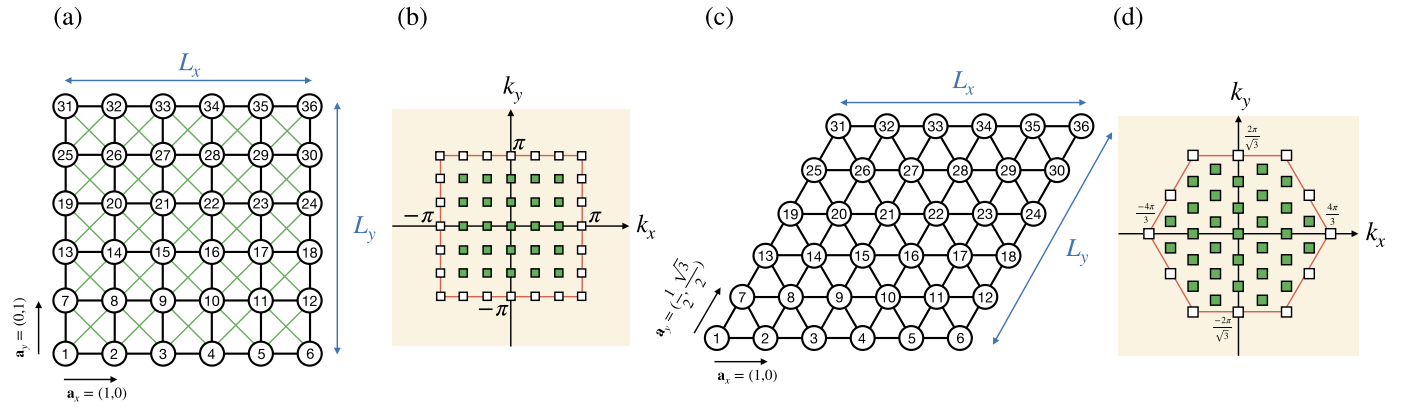


Fig. 1. Examples of lattice structures and the corresponding momentum spaces. (a, b) The 6×6 square lattice cluster and (c, d) the 6×6 triangular lattice cluster. The numbers in each circle in (a) and (c) denote site numbers used in the input file. \mathbf{a}_x and \mathbf{a}_y represent the primitive translational vectors, where the lattice constant is set to be one. L_x (L_y) is the linear dimension of the lattice along the \mathbf{a}_x (\mathbf{a}_y) direction (in these examples, $L_x = L_y = 6$). In (b) and (d), the red lines indicate the first Brillouin Zone, and the green and white squares represent the allowed momentum points, $\mathbf{k} = (k_x, k_y)$, for the clusters in (a) and (c), respectively, under periodic boundary conditions; the white squares are the momentum points on the Brillouin zone boundary and pairs of momentum points located on the opposite corners and sides are equivalent. The QS³ package can compute eigenvalues and eigenvectors of the Hamiltonian matrix at each momentum sector, separately. These momentum points correspond to wave vector points, $\mathbf{q} = (q_x, q_y)$, which can be chosen for computing the dynamical spin structure factor $\mathbf{S}(\mathbf{q}, \omega)$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

NOxxz (INTEGER): Number N_{xxz} of the two-body exchange interactions.

ALG (INTEGER): The users can choose algorithms by setting 1: Conventional Lanczos, 2: Thick-restart Lanczos, 3: Full diagonalization.

cal_lm, **cal_cf**, **cal_dsf** (INTEGER): The local magnetization (lm), spin correlation function (cf), and dynamical structure factor (dsf) are calculated by setting the corresponding variables to 1. Otherwise, these variables should be 0.

wr_wf (INTEGER): If $wr_wf = 1$, the computed eigenvectors are output in the directory specified by FILEwf.

re_wf (INTEGER): If $re_wf = 0$, the whole diagonalization calculation starts from a random initial vector. If $re_wf = 1$, only the expectation values of physical quantities are computed after reading the eigenvectors already prepared in the directory specified by FILEwf.

FILExxz (CHARACTER): The location of the file that defines the lattice structure and the two-body exchange interactions $J_{r,r'}^z$ and $J_{r,r'}^{xy}$. The lattice structure is specified in terms of pairs of sites connected by the nonzero interactions. The details are described below.

FILEwf (CHARACTER): The location of output for the computed eigenvectors. The number of eigenvectors is specified by NOE (the number of the lowest eigenvalues/eigenvectors calculated by an eigensolver specified by ALG).

OUTDIR (CHARACTER): The location of output for the computed expectation values such as the local magnetization, spin correlation function, and dynamical structure factor.

input_static

This part requires the users to set the conditions for the calculation of the static physical quantities, i.e., the local magnetization and two-point correlation function. The users should prepare separately the two files that specify the sites and the pairs of sites for the calculation of the local magnetization and the two-point correlation function, respectively.

NOV (INTEGER): Number of the lowest eigenvectors used for computing static physical quantities.

NOLM (INTEGER): Number of sites for which the local magnetization is computed. If $NOLM=0$, the local magnetization is not computed.

NOCF (INTEGER): Number of pairs of sites for which the two-point correlation function is computed. If $NOCF=0$, the correlation function is not computed.

FILElm (CHARACTER): The location of the file specifying the site definition of the cluster for the calculation of the local magnetization. The details of the file are explained below.

FILECF (CHARACTER): The location of the file specifying the pairs of sites for the calculation of the two-point correlation function. The details of the file are explained below.

input_dynamic

This part requires the users to set the conditions for the calculation of the dynamical spin structure factor. The user should prepare the file specifying the sites of the cluster, separately.

spsmsz (INTEGER): $S^+(\mathbf{q}, \omega)$, $S^-(\mathbf{q}, \omega)$, and $S^Z(\mathbf{q}, \omega)$ are computed by setting the value $spsmsz = 1, 2$, and 3 , respectively. If $spsmsz = 0$, the dynamical spin structure factor is not computed.

itr_dsf (INTEGER): Number of iterations for the continued fraction method. See Sec. 3.9 for the details.

QX,QY,QZ (REAL8): Wave vector point \mathbf{q} at which the dynamical spin structure factor is computed, $\mathbf{q} \cdot \mathbf{a}_\alpha = Q_\alpha$.

rfield (REAL8): Magnetic field value h_z .

FILEpos (CHARACTER): The location of the file specifying the site positions. The details of the file are explained below.

input_lancz

This part requires the users to set the conditions for the conventional Lanczos algorithm when the users set $ALG = 1$. The users must set $NOE = 1$ below because only the lowest eigenvalue with the corresponding eigenvector is computed here.

lnc_ene_conv0 (REAL8): Convergence condition for the Lanczos iteration.

min(max)itr (INTEGER): The minimum/maximum number of iterations for the Lanczos method.

itrint (INTEGER): Every itrint iterations, the convergence of the Lanczos iteration is checked.

input_TRLan

This part requires the users to set the conditions for the thick-restart Lanczos algorithm when the users set $ALG = 2$.

NOE (INTEGER): Number of the lowest eigenvalues/eigenvectors computed by the thick-restart Lanczos method.

NOK (INTEGER): N_K value. See Algorithm 10.

NOM (INTEGER): N_M value. See Algorithm 10.

maxitr (INTEGER): I_M value. See Algorithm 10. The maximum number of iterations for the thick-restart Lanczos method.

lnc_ene_conv (REAL8): Convergence condition for the thick-restart Lanczos iteration.

i_vec_min/max (INTEGER): Store the i_vec_min-th to i_vec_max-th lowest eigenvectors computed by the thick-restart Lanczos method in the location specified by FILEwf.

2.3.2. Set a file specifying the model

The users are required to specify the lattice structure and the exchange interactions in an input file. We provide in the input directory an example, list_xxz_term_36.dat, for the $S = 1/2$ isotropic ferromagnetic Heisenberg model on the 6×6 square lattice, which corresponds to the input of example 1) in QS³. The file location should be specified in the main input file using the FILExxz variable. The first part of list_xxz_term_36.dat are shown below.

```
1 2 -1.0E+00 -1.0E+00
2 3 -1.0E+00 -1.0E+00
3 4 -1.0E+00 -1.0E+00
4 5 -1.0E+00 -1.0E+00
....
```

In this data statement, the first and second columns denote the pair of sites (r and r'), and the third and fourth columns represent the corresponding exchange interactions, $J_{r,r'}^{xy}$ and $J_{r,r'}^z$, respectively [see Fig. 1].

2.3.3. Set files for computing physical quantities

The users are required to set several input files for the calculation of physical quantities. For the local magnetization (\hat{s}_r^z), the users have to specify the site r for which the local magnetization is computed. For the two-point correlation function ($\hat{s}_r^+ \hat{s}_{r'}^-$) and ($\hat{s}_r^x \hat{s}_{r'}^x$), the users have to specify the pair of sites (r, r') for which the correlation function is computed. As concrete examples, we provide two files in the input directory, list_local_mag.dat and list_cf_ss.dat, for the local magnetization and the two-point correlation function, respectively.

For the dynamical spin structure factor, the users are required to specify the site positions in the cluster. We also provide an example input file, list_site_position_36_type1.dat. Note that the lattice constant is set to be one [see Fig. 1(a)]. The location of these files is specified using FILExxz, FILElm, and FILECF variables in the main input file.

2.3.4. Run and results

After preparing all these files described above, the users can perform the calculation as follows:

```
$. /QS3.exe < input_ex1/input.dat >> output_ex1/output.dat
2>&1
```

Here, output.dat is the result file for the calculation, from which the users can check the status of the calculation. The computed results of the physical quantities are output in the directory specified by the OUTDIR variable in the main input file. The eigenvectors are output in the directory specified by the FILEwf variable in the main input file.

3. Implemented algorithms

3.1. Representation of states with the $U(1)$ symmetry

The QS³ package diagonalizes the Hamiltonian matrix for $\hat{\mathcal{H}}$ given in Eq. (1), consisting of N spins that can be as large as $\mathcal{O}(10^3)$, with a small number $N_\downarrow = \mathcal{O}(1) \ll N$ of down spins or equivalently with a large total magnetization value $M \equiv \langle \phi | \sum_r \hat{s}_r^z | \phi \rangle = N/2 - N_\downarrow$, where $|\phi\rangle$ is an eigenstate of $\hat{\mathcal{H}}$.

A standard way to construct the spin basis states, and accordingly represent the Hamiltonian matrix, is to use the bit representation, where up (\uparrow) and down (\downarrow) spins are expressed as 0- and 1-bit values, respectively. However, this is not a practical way for our purpose because a single four-byte (eight-byte) integer can only represent spin basis states up to $N=32$ (64) [31] in a standard 32 (64)-bit operating system. Furthermore, the definition and operation of arbitrary-byte integers are not supported in standard numerical programming languages.

Let us now explain how to construct the spin basis states in the QS³ package. We first introduce the following fully polarized state as a vacuum state:

$$|v\rangle \equiv \overbrace{|\uparrow\uparrow\cdots\uparrow\rangle}^{N \text{ spins}}. \quad (2)$$

Each spin basis state $|a\rangle$ is then constructed by acting the $S = 1/2$ descending operator \hat{s}_r^- on the vacuum state, i.e.,

$$|a\rangle = \prod_{m=1}^{N_\downarrow} \hat{s}_{r_m}^- |v\rangle, \quad (3)$$

where r_m is the position of the m th down spin in real space. In the QS³ package, a set of $\{r_m\}_{1 \leq m \leq N_\downarrow}$ is stored in N_\downarrow -dimensional integer vector (array) $\mathbf{n} \equiv (n_1, n_2, \dots, n_{N_\downarrow}) = (r_1, r_2, \dots, r_{N_\downarrow})$ in ascending order, $1 \leq r_1 < r_2 < \dots < r_{N_\downarrow} \leq N$, implying that $r_m \geq m$.

We now focus on a subspace of the entire Hilbert space of $\hat{\mathcal{H}}$ by setting the number N_\downarrow of down spins. The dimension of the subspace is equal to the binomial coefficient ${}_N C_{N_\downarrow} := \binom{N}{N_\downarrow}$ and the spin basis states $|a\rangle$ in this subspace are numerated as $a = 1, 2, \dots, {}_N C_{N_\downarrow}$. For a given set of $\{r_m\}_{1 \leq m \leq N_\downarrow}$, we can define an integer index a through the following bijective function F :

$$a = F(\mathbf{n}); F(\mathbf{n}) = 1 + \sum_{m=1}^{N_\downarrow} r_{m-1} C_{m-1}. \quad (4)$$

This one-to-one correspondence between a and $\{r_m\}_{1 \leq m \leq N_\downarrow}$ can be understood as follows: When the N_\downarrow th down spin is located at r_{N_\downarrow} th site, the target spin configuration $|a\rangle$ should be listed after $r_{N_\downarrow} - 1 C_{N_\downarrow}$ patterns for arranging N_\downarrow spins stored within sites from the first site to the $(r_{N_\downarrow} - 1)$ -th site. We can apply the same procedure for $m = N_\downarrow - 1$ down to $m = 1$, recursively, assuming that $k C_m = 0$ if $k < m$.

The inverse bijective function $\mathbf{n} = \bar{F}(a)$ is given in Algorithm 1 with the binary search algorithm (Algorithm 2). It is highly instructive to first consider a concrete example. For example, let us consider the case of $N = 8$, $N_\downarrow = 4$, and $\mathbf{n} = (2, 4, 6, 8)$, which corresponds to $a = 7C_4 + 5C_3 + 3C_2 + 1C_1 + 1 = 50$, according to Eq. (4). Now, giving $s = a = 50$ as the input, we explain how Algorithm 1 outputs $\mathbf{n} = (2, 4, 6, 8)$. The algorithm first searches for $n_4 = r_4 = 8$ that satisfies $7C_4 = 35 < s = 50 \leq 8C_4 = 70$ and updates $s := s - 35 = 15$. Second, the algorithm searches for $n_3 = r_3 = 6$ that satisfies $5C_3 = 10 < s = 15 \leq 6C_3 = 20$ and updates $s := s - 10 = 5$. Third, the algorithm searches for $n_2 = r_2 = 4$ that satisfies $3C_2 = 3 < s = 5 \leq 4C_2 = 6$ and updates $s := s - 3 = 2$. Finally, the algorithm assigns $n_1 = r_1 = s = 2$.

More generally, Algorithm 1 first searches for $n_{N_\downarrow} = r_{N_\downarrow}$ that satisfies $r_{N_\downarrow} - 1 C_{N_\downarrow} < a \leq r_{N_\downarrow} C_{N_\downarrow}$, followed by a research for $n_{N_\downarrow-1} = r_{N_\downarrow-1}$ that satisfies $r_{N_\downarrow-1} - 1 C_{N_\downarrow-1} < a - r_{N_\downarrow-1} C_{N_\downarrow} \leq r_{N_\downarrow-1} C_{N_\downarrow-1}$, until a search for $n_2 = r_2$ that satisfies $r_2 - 1 C_2 < a - \sum_{m=3}^{N_\downarrow} r_{m-1} C_{m-1} \leq r_2 C_2$. Finally, it uses Eq. (4) to determine $n_1 = r_1 = a - \sum_{m=2}^{N_\downarrow} r_{m-1} C_{m-1}$ and returns \mathbf{n} . Note that the memory cost with $\mathcal{O}(N_\downarrow N C_{N_\downarrow})$ bytes for keeping the basis sets $\{\mathbf{n}\}$

in a computer, which may become a memory bottleneck for the Lanczos method, can be reduced to $O(1)$ by representing the spin basis states with $|a\rangle$ at the expense of additional numerical cost of $O(N_\downarrow \ln(N - N_\downarrow))$ for the use of the function \bar{F} . This additional computation makes the difference from the standard technique using bit representation in computational time.

Algorithm 1 Generate \mathbf{n} for given a .

Input: integers N_\downarrow , N , and a with $1 \leq a \leq N C_{N_\downarrow}$.
Output: N_\downarrow -dimensional integer vector \mathbf{n} with $1 \leq n_1 < n_2 < \dots < n_{N_\downarrow} \leq N$.
1: **function** F_BAR(a, N_\downarrow, N)
2: $s := a$
3: $j := N$
4: **for** $m = N_\downarrow$ to 2 with $m := m - 1$ **do**
5: $(j_0, f) = \text{BINARY_SEARCH}(s - 1, \{k C_m\}_{1 \leq k \leq j}, m, j)$
 \triangleright A logical parameter f is not used in this function.
 \triangleright Note that $k C_m = 0$ if $k < m$.
6: $j := j_0$
7: $n_m := j_0 + 1$
8: $s := s - j C_m$
9: **end for**
10: $n_1 := s$
11: **return** (\mathbf{n})
12: **end function**

Algorithm 2 Check whether s_0 is in \mathbf{s} .

Input: integer s_0 , l_s , and l_e ($\geq l_s$), and l_e -dimensional integer vector \mathbf{s} .
Output: integer p and logical f .
1: **function** BINARY_SEARCH($s_0, \mathbf{s}, l_s, l_e$)
2: Search integer p in $\{s_k\}_{l_s \leq k \leq l_e}$ satisfying $s_p \leq s_0 < s_{p+1}$ with $l_s \leq p \leq l_e$ by the binary search where $s_{l_e+1} = \infty$.
3: $f := \begin{cases} \text{True} & s_p = s_0 \\ \text{False} & \text{otherwise} \end{cases}$
4: **return** (p, f)
5: **end function**

3.2. Generation of the Hamiltonian matrix

The Hamiltonian matrix is block diagonal with respect to the number N_\downarrow of down spins and the dimension of the block-diagonal matrix specified with (N, N_\downarrow) is $N C_{N_\downarrow}$. There is a nonzero diagonal contribution of $\langle a | \hat{s}_r^z \hat{s}_{r'}^z | a \rangle = \pm 1/4$ to the Hamiltonian matrix, where the sign of the value is minus when either r or r' is in \mathbf{n} and otherwise it is plus. For this check, the QS³ package uses Algorithm 2 of the binary search. On the other hand, the off-diagonal operator $(\hat{s}_r^+ \hat{s}_{r'}^- + \hat{s}_r^- \hat{s}_{r'}^+)$ acting on a state $|a\rangle$ can generate a different basis state $|a'\rangle$ and the corresponding array $\mathbf{n}' = \bar{F}(a')$. We can also use the binary search to determine whether a new state is generated. Namely, the new state $|a'\rangle$ is generated when two logical variables f_r and $f_{r'}$, which are given by $(p_t, f_t) := \text{BINARY_SEARCH}(t, \mathbf{n}, 1, N_\downarrow)$ with $t \in \{r, r'\}$, are different. If this is the case, the QS³ package uses Algorithm 3 to make the new array \mathbf{n}' and we can obtain the off-diagonal matrix element $\langle a' | \hat{s}_r^+ \hat{s}_{r'}^- + \hat{s}_r^- \hat{s}_{r'}^+ | a \rangle = 1$. To construct the full matrix elements, we have to consider all sets of $\{r, r'\}$ compatible with the nonzero exchange interactions in $\hat{\mathcal{H}}$ given in Eq. (1). For this purpose, the QS³ package uses Algorithm 4, where the N_{xxz} variable is the number of the exchange interactions, i.e., the number of pairs $\{r, r'\}$ connected via the nonzero exchange interactions, and should be equal to NOxxz in the input file. Note that the contribution of the Zeeman term in Eq. (1) is excluded in Algorithm 4 because it is simply constant within the subspace of a fixed N_\downarrow .

3.3. Representative states and Hamiltonian matrix elements in symmetry-adapted basis sets

Not only the U(1) symmetry in spin space, but also lattice symmetry such as translational symmetry and point group symmetry

Algorithm 3 Spin exchange interaction between sites r and r' .

Input: integer r, p, r' ($> r$), and p' ($\geq p$), N_\downarrow -dimensional integer vector \mathbf{n} , and logical f , where p, p' , and f are given by $(p, f) := \text{BINARY_SEARCH}(r, \mathbf{n}, 1, N_\downarrow)$ and $(p', f') := \text{BINARY_SEARCH}(r', \mathbf{n}, 1, N_\downarrow)$, assuming that $f' \neq f$.
Output: N_\downarrow -dimensional integer vector \mathbf{n}' with $1 \leq n'_1 < n'_2 < \dots < n'_{N_\downarrow} \leq N$.
1: **function** SPIN_EXCHANGE($r, p, f, r', p', \mathbf{n}$)
2: **if** $f = \text{False}$ **then**
3: $\mathbf{n}' := (n_1, \dots, n_p, r, n_{p+1}, \dots, n_{p'-1}, n_{p'+1}, \dots, n_{N_\downarrow})$
4: **else**
5: $\mathbf{n}' := (n_1, \dots, n_{p-1}, n_{p+1}, \dots, n_{p'}, r, n_{p'+1}, \dots, n_{N_\downarrow})$
6: **end if**
7: **return** (\mathbf{n}')
8: **end function**

Algorithm 4 Generation of full Hamiltonian matrix.

Input: N_{xxz} -dimensional integer vectors \mathbf{r} and \mathbf{r}' , and N_{xxz} -dimensional real vectors \mathbf{J}^{xy} and \mathbf{J}^z .
Output: $N C_{N_\downarrow}$ -dimensional real matrix $\mathbf{H} = \{h_{a,a'}\}$.
1: **function** GEN_FULL_HAM($\mathbf{r}, \mathbf{r}', \mathbf{J}^{xy}, \mathbf{J}^z$)
2: $\mathbf{H} := 0$
3: **for** $a = 1$ to $N C_{N_\downarrow}$ **do**
4: $\mathbf{n} := \text{F_BAR}(a, N_\downarrow, N)$
5: **for** $n = 1$ to N_{xxz} **do**
6: $(p, f) := \text{BINARY_SEARCH}(r_n, \mathbf{n}, 1, N_\downarrow)$
7: $(p', f') := \text{BINARY_SEARCH}(r'_n, \mathbf{n}, 1, N_\downarrow)$
8: **if** $f = f'$ **then**
9: $h_{a,a} := h_{a,a} + J_n^z/4$
10: **else**
11: $h_{a,a} := h_{a,a} - J_n^z/4$
12: $\mathbf{n}' := \text{SPIN_EXCHANGE}(r, p, f, r', p', \mathbf{n})$
13: $a' := F(\mathbf{n}')$
14: $h_{a,a'} := J_n^{xy}/2$
15: **end if**
16: **end for**
17: **end for**
18: **return** (\mathbf{H})
19: **end function**

can be used to reduce the dimension of the Hamiltonian matrix to be diagonalized and thus the computational cost. Here, we describe how to block-diagonalize the Hamiltonian $\hat{\mathcal{H}}$ based on the symmetry-adapted basis sets.

First, we briefly explain how to construct the symmetry-adapted basis sets that are the eigenstates of the lattice translational operator \hat{T} . A pedagogical introduction for the construction of the symmetry-adapted basis sets can be found in Ref. [3]. For simplicity, we consider a periodic chain with N spins, namely $(L_x, L_y, L_z) = (N, 1, 1)$, in which the translational operator \hat{T} is defined by shifting the position of the spin one site right, $\hat{T}|a\rangle = \prod_{m=1}^{N_\downarrow} \hat{s}_{m+1}^- |v\rangle$, with $\hat{s}_{L+1} = \hat{s}_1$ under periodic boundary conditions. Note that the translational operator \hat{T} commutes with the Hamiltonian, i.e., $[\hat{T}, \hat{H}] = 0$, and the accessible eigenvalues of \hat{T} are given as $\{e^{ik} \mid k = 2\pi K/N, 0 \leq K < N\}$ with momentum k or momentum sector K .

The symmetry-adapted basis states with a given momentum k are given as

$$|a, k\rangle = \frac{1}{\sqrt{N_{a,k}}} \sum_{j=1}^N e^{-ikj} \hat{T}^j |a\rangle, \quad (5)$$

where $|a\rangle$ is a single reference state with a fixed number N_\downarrow of down spins and it is defined in Eq. (3). One can easily confirm that $|a, k\rangle$ in Eq. (5) is an eigenstate of the translational operator, i.e., $\hat{T}|a, k\rangle = e^{ik}|a, k\rangle$. If the reference state $|a\rangle$ is not compatible with the momentum k , the state $|a, k\rangle$ generated in Eq. (5) vanishes. The compatibility of the chosen reference state $|a\rangle$ and the normalization factor $N_{a,k}$ can be determined as

$$N_{a,k} = \frac{N \left| \langle a | \sum_{j=1}^N e^{-ikj} \hat{T}^j | a \rangle \right|^2}{\langle a | \sum_{j=1}^N \hat{T}^j | a \rangle}. \quad (6)$$

If this quantity is zero, it implies that the chosen reference state $|a\rangle$ is not compatible with the momentum k . Otherwise, this quantity gives the normalization factor of the state $|a, k\rangle$.

Considering the cyclicity of the translated states, $\{|a_j\rangle \equiv \hat{T}^j | a \rangle\}_{1 \leq j \leq N}$, we can simply choose only one state as a representative among $\{|a_j\rangle\}_{1 \leq j \leq N}$. The QS³ package chooses one state with the smallest integer a , i.e., $|a\rangle \equiv |\min_j a_j\rangle$, which is used to generate $|a, k\rangle$ in Eq. (5). We have to check all possible ${}_N C_{N_\downarrow}$ -states, $\{|a\rangle\}$, in this way, and determine which states and how many states are representatives in the target subspace specified with (k, N_\downarrow) . The QS³ package uses Algorithm 5 to check whether or not a state is representative and evaluate the corresponding normalization factor, and uses Algorithm 6 to make a list of the representative states $\{|a\rangle\}$ and a list of the corresponding normalization factors $\{N_{a,k}\}$, representing the symmetry-adapted basis sets $\{|a, k\rangle\}$.

Algorithm 5 Check whether a state $|a\rangle$ is representative, and evaluate the corresponding normalization factor.

Input: integer a with $1 \leq a \leq {}_N C_{N_\downarrow}$, and real $k \in \{2\pi K/N\}_{0 \leq K < N}$
Output: real $N_{a,k} \geq 0$.

```

1: function CHECK_STATE( $a, k$ )
2:    $\mathbf{n} := \text{F\_BAR}(a, N_\downarrow, N)$ 
3:    $N_{a,k} := 0$ ;  $c^* := 0$ ;  $n_c := 0$ 
4:   for  $j = 1$  to  $N$  do
5:      $\mathbf{n} := \text{SHIFT\_FUNC}(\mathbf{n})$ 
6:      $\triangleright$  The function SHIFT_FUNC( $\mathbf{n}$ ) gives a  $N_\downarrow$ -dimensional vector corresponding to a translated state,  $\hat{T}^j | a \rangle$ .
7:      $\mathbf{n} := \text{INSERTION\_SORT}(\mathbf{n})$ 
8:      $\triangleright$  The function INSERTION_SORT( $\mathbf{n}$ ) sorts the vector elements in ascending order by using the insertion sort algorithm. When  $N_\downarrow$  is  $\mathcal{O}(1)$ , we confirm that the insertion sort algorithm is generally faster than the quick sort algorithm.
9:      $a' := F(\mathbf{n})$ 
10:    if  $a' < a$  then
11:      return ( $N_{a,k} = 0$ )
12:    else
13:      if  $a' = a$  then
14:         $c^* := c^* + e^{ikj}$ ;  $n_c := n_c + 1$ 
15:      end if
16:    end if
17:  end for
18:  return ( $N_{a,k} = |c^*|^2 N / n_c$ )

```

Algorithm 6 Making lists of representative states and the corresponding normalization factors.

Input: real k
Output: integer d with $0 \leq d \leq {}_N C_{N_\downarrow}$, d -dimensional integer vector σ , and d -dimensional non-negative real vector \mathbf{R}

```

1: function MK_LIST( $k$ )
2:    $d := 0$ 
3:   for  $a = 1$  to  ${}_N C_{N_\downarrow}$  do
4:      $N_{a,k} := \text{CHECK\_STATE}(a, k)$ 
5:     if  $N_{a,k} > 0$  then
6:        $d := d + 1$ 
7:        $\sigma_d := a$ 
8:        $R_d := \sqrt{N_{a,k}}$ 
9:     end if
10:  end for
11:  return ( $d, \sigma, \mathbf{R}$ )
12: end function

```

We are now ready to construct the Hamiltonian matrix based on the symmetry-adapted basis sets $\{|a, k\rangle\}$. A state obtained after operating the Hamiltonian $\hat{\mathcal{H}}$ to each basis state $|a, k\rangle$ is given by

$$\hat{\mathcal{H}}|a, k\rangle = \frac{1}{\sqrt{N_{a,k}}} \sum_n \sum_{j=1}^N e^{-ikj} \hat{T}^j \hat{h}_n |a\rangle, \quad (7)$$

where

$$\hat{h}_n |a\rangle = \frac{J_n^{xy}}{2} (\hat{s}_{r_n}^+ \hat{s}_{r_n}^- + \hat{s}_{r_n}^- \hat{s}_{r_n}^+) |a\rangle + J_n^z \hat{s}_{r_n}^z \hat{s}_{r_n}^z |a\rangle. \quad (8)$$

Note that the Zeeman term in Eq. (1) can be treated separately because the U(1) symmetry is adapted in the basis sets. We should also note that the off-diagonal term in Eq. (8) flips a spin in the representative state $|a\rangle$ and the generated state, $|a^{(n)}\rangle = (\hat{s}_{r_n}^+ \hat{s}_{r_n}^- + \hat{s}_{r_n}^- \hat{s}_{r_n}^+) |a\rangle$, is not necessarily a representative state. Therefore, we have to check if the flipped state $|a^{(n)}\rangle$ is compatible with the momentum k . If it is the case, we have to seek the representative state $|a^{(n)}\rangle \equiv |\min_j a_j^{(n)}\rangle$ by applying translational operations onto $|a^{(n)}\rangle$, i.e., $|a_j^{(n)}\rangle \equiv \hat{T}^j |a^{(n)}\rangle$.

Consequently, we can write Eq. (7) as

$$\hat{H}|a, k\rangle = \sum_n \frac{J_n^{xy}}{2} e^{-ik\ell_n} \sqrt{\frac{N_{a^{(n)},k}}{N_{a,k}}} \left(1 - \delta_{f_{n,a}, f'_{n,a}}\right) |a^{(n)}, k\rangle + \sum_n \langle a | J_n^z \hat{s}_{r_n}^z \hat{s}_{r_n}^z | a \rangle |a, k\rangle, \quad (9)$$

where ℓ_n is obtained from the relationship $|a^{(n)}\rangle = T^{\ell_n} |a^{(n)}\rangle$ with $1 \leq \ell_n \leq N$. The two variables $f_{n,a}$ and $f'_{n,a}$ are logical ones given by $(p_{n,a}, f_{n,a}) := \text{BINARY_SEARCH}(r_{n,\text{F_BAR}}(a, N_\downarrow, N), 1, N_\downarrow)$ and $(p'_{n,a}, f'_{n,a}) := \text{BINARY_SEARCH}(r'_{n,\text{F_BAR}}(a, N_\downarrow, N), 1, N_\downarrow)$, respectively. These variables are used to judge if each off-diagonal term in the Hamiltonian contributes. In the QS³ package, Algorithm 7 is used to search the representative state $|a^{(n)}\rangle$ for the off-diagonal matrix elements and to obtain the corresponding ℓ_n value, and Algorithm 8 is to construct the Hamiltonian matrix. Note the order of two for-loops associated with a and n in Algorithm 8 that is chosen to enhance the performance of open MP parallelization applying to a .

Algorithm 7 Seeking the representative state $a^{(n)}$ and the ℓ_n value for operations in Eq. (9).

Input: integer $a^{(n)}$ and N_\downarrow
Output: integer $a^{(n)}$ and ℓ_n .

```

1: function REPRESENTATIVE( $a^{(n)}, N_\downarrow$ )
2:    $\alpha^{(n)} := a^{(n)}$ 
3:    $\mathbf{n} := \text{F\_BAR}(a^{(n)}, N_\downarrow, N)$ 
4:   for  $j = 1$  to  $N$  do
5:      $\mathbf{n} := \text{SHIFT\_FUNC}(\mathbf{n})$ 
6:      $\mathbf{n} := \text{INSERTION\_SORT}(\mathbf{n})$ 
7:      $a := F(\mathbf{n})$ 
8:     if  $a \leq \alpha^{(n)}$  then
9:        $\alpha^{(n)} := a$ ;  $\ell_n := j$ 
10:    end if
11:  end for
12:  return ( $\alpha^{(n)}, \ell_n$ )
13: end function

```

3.4. Full diagonalization

One can full diagonalize the whole Hamiltonian to obtain all eigenvalues $\{E_\nu\}$ and the corresponding eigenvectors $\{|\nu\rangle\}$ by separately diagonalizing block diagonalized Hamiltonian matrices constructed via Algorithm 4 or Algorithm 8 with different symmetry sectors. Accordingly, one can for example compute the temperature dependence of any physical quantity \hat{A} based on the thermal average,

Algorithm 8 Generation of Hamiltonian matrix with the symmetry-adapted basis sets.

Input: N_{xxz} -dimensional integer vectors \mathbf{r} and \mathbf{r}' , N_{xxz} -dimensional real vectors \mathbf{J}^{xy} and \mathbf{J}^z , d -dimensional integer vector σ and real vector \mathbf{R} for the lists of the representative states and their normalization factors, respectively, and real k

Output: d -dimensional complex matrix $\mathbf{H} = \{h_{a,a'}\}$.

```

1: function GEN_FULL_HAM_SYM_ADAPT( $\mathbf{r}, \mathbf{r}', \mathbf{J}^{xy}, \mathbf{J}^z, \sigma, \mathbf{R}, k$ )
2:    $\mathbf{H} := 0$ 
3:   for  $a = 1$  to  $d$  do
4:      $\mathbf{n} = \text{F\_BAR}(\sigma_a, N_\downarrow, N)$ 
5:     for  $n = 1$  to  $N_{\text{xxz}}$  do
6:        $(p, f) := \text{BINARY\_SEARCH}(r_n, \mathbf{n}, 1, N_\downarrow)$ 
7:        $(p', f') := \text{BINARY\_SEARCH}(r'_n, \mathbf{n}, 1, N_\downarrow)$ 
8:       if  $f = f'$  then
9:          $h_{a,a} := h_{a,a} + J_n^z/4$ 
10:      else
11:         $h_{a,a} := h_{a,a} - J_n^z/4$ 
12:         $a' := \text{F}(\text{SPIN\_EXCHANGE}(r, p, f, r', p', \mathbf{n}))$ 
13:         $(a', \ell) := \text{REPRESENTATIVE}(a', N_\downarrow)$ 
14:         $(a', f) := \text{BINARY\_SEARCH}(a', \sigma, 1, d)$ 
15:        if  $f = \text{True}$  then
16:           $h_{a,a'} := h_{a,a'} + \frac{J_n^{xy}}{2} e^{ik\ell} \sqrt{R_{a'}/R_a}$ 
17:        end if
18:      end if
19:    end for
20:  end for
21:  return ( $\mathbf{H}$ )
22: end function

```

$$\langle \hat{A} \rangle_{\beta, N}^{\text{ens}} = \sum_{\nu} \frac{e^{-\beta E_{\nu}}}{Z(\beta)} \langle \nu | \hat{A} | \nu \rangle, \quad (10)$$

where β is the inverse temperature, $Z(\beta) = \sum_{\nu} e^{-\beta E_{\nu}}$ is the partition function, and the summation of ν runs over all symmetry sectors with different values of N_{\downarrow} and/or k . However, note that the accessible matrix dimension is very limited in the full diagonalization calculation, typically up to $O(10^4)$ on a currently available standard computer. Therefore, one may not be able to treat all subspaces of the Hamiltonian even when the Hamiltonian matrix is block diagonalized with different symmetry sectors.

The QS³ package is specialized for the system under a high magnet field, in which one can treat much larger system sizes near the saturation field. This implies that one may access the finite temperature physics of large systems but at sufficiently low temperature where the low-energy eigenvalues are reasonably separated from those for the symmetry sectors with larger N_{\downarrow} and thus the latter contribution to the thermal average $\langle \hat{A} \rangle_{\beta, N}^{\text{ens}}$ can be simply discarded.

The QS³ package uses DHEEVR/ZHEEVR routine in LAPACK [37] for the full diagonalization to obtain all eigenvalues and eigenvectors of the Hamiltonian matrix constructed with the symmetry-adapted basis sets.

3.5. Multiplying Hamiltonian to state vectors (matrix-vector product)

In order to calculate the lowest eigenvalue (and also the several lowest eigenvalues) and the corresponding eigenvector(s) of the Hamiltonian matrix, one can also employ the conventional Lanczos method, instead of the full diagonalization, which allows us to treat larger system sizes. The main and most time-consuming part in the Lanczos method is a matrix-vector product, and the QS³ package does this operation based on the symmetry-adapted basis sets. Assuming that both the U(1) and translational symmetries are adapted, the resulting vector after the matrix-vector product operation $\hat{H}|\phi\rangle$ is expressed with the basis sets $\{|a, k\rangle\}$ and each element $\psi_{a,k}$ can be obtained as

$$\psi_{a,k} = \langle a, k | \psi \rangle = \langle a, k | \hat{H} | \phi \rangle$$

$$= \sum_n \frac{J_n^{xy}}{2} e^{ik\ell_n} \sqrt{\frac{N_{a^{(n)},k}}{N_{a,k}}} \left(1 - \delta_{f_n, a, f'_n, a}\right) \phi_{a^{(n)},k} + \sum_n \langle a | J_n^z \hat{S}_{r_n}^z \hat{S}_{r'_n}^z | a \rangle \phi_{a,k}, \quad (11)$$

where a state vector $|\phi\rangle = \sum_a \phi_{a,k} |a, k\rangle$ is an input vector. The QS³ package uses Algorithm 9 to do this procedure. Note that the calculation of each element is done on the fly and hence the accessible vector dimension can be enlarged up to $O(10^8)$.

Algorithm 9 Perform $\psi := \mathbf{H}\phi$.

Input: integer vectors σ , \mathbf{r} , and \mathbf{r}' , real k , real vectors \mathbf{R} , \mathbf{J}^{xy} , and \mathbf{J}^z as in Algorithm 8, and d -dimensional complex vector ϕ .

Output: d -dimensional complex vector ψ .

```

1: function HAM_TO_VEC( $\mathbf{r}, \mathbf{r}', \mathbf{J}^{xy}, \mathbf{J}^z, \sigma, k, \mathbf{R}, \phi$ )
2:    $\psi := 0$ 
3:   for  $a = 1$  to  $d$  do
4:      $\mathbf{n} = \text{F\_BAR}(\sigma_a, N_\downarrow, N)$ 
5:     for  $n = 1$  to  $N_{\text{xxz}}$  do
6:        $(p, f) := \text{BINARY\_SEARCH}(r_n, \mathbf{n}, 1, N_\downarrow)$ 
7:        $(p', f') := \text{BINARY\_SEARCH}(r'_n, \mathbf{n}, 1, N_\downarrow)$ 
8:       if  $f = f'$  then
9:          $\psi_a := \psi_a + \frac{J_n^z}{4} \phi_a$ 
10:      else
11:         $\psi_a := \psi_a - \frac{J_n^z}{4} \phi_a$ 
12:         $a' := \text{F}(\text{SPIN\_EXCHANGE}(r, p, f, r', p', \mathbf{n}))$ 
13:         $(a', \ell) := \text{REPRESENTATIVE}(a', N_\downarrow)$ 
14:         $(a', f) := \text{BINARY\_SEARCH}(a', \sigma, 1, d)$ 
15:        if  $f = \text{True}$  then
16:           $\psi_a := \psi_a + \frac{J_n^{xy}}{2} e^{ik\ell} \sqrt{R_{a'}/R_a} \phi_{a'}$ 
17:        end if
18:      end if
19:    end for
20:  end for
21:  return ( $\psi$ )
22: end function

```

3.6. Calculating expectation values

The QS³ package can evaluate the local magnetization $\langle \phi | \hat{S}_r^z | \phi \rangle$ and the two-point spin correlation function $\langle \phi | \hat{S}_r^\alpha \hat{S}_{r'}^\beta | \phi \rangle$ where $(\alpha, \beta) \in \{(z, z), (\pm, \mp)\}$ after computing eigenvectors $|\phi\rangle$ of the Hamiltonian matrix. When the eigenvector $|\phi\rangle = \sum_a \phi_{a,k} |a, k\rangle$ respects the translational symmetry with the momentum k , the expectation value of an operator preserving the translational symmetry can be evaluated simply by reusing Algorithm 8, where the matrix elements of the Hamiltonian matrix are evaluated in the symmetry-adapted basis sets. Therefore, in the QS³ package, the translationally-symmetrized operators $\frac{1}{N} \sum_{j=1}^N \hat{T}^j \hat{O} \hat{T}^{-j}$ with $\hat{O} = \hat{S}_r^\alpha$ and $\hat{S}_r^\alpha \hat{S}_{r'}^\beta$ are used, instead of directly treating the local operators \hat{O} , for the expectation values: $\langle \phi | \hat{O} | \phi \rangle = \frac{1}{N} \langle \phi | (\sum_{j=1}^N \hat{T}^j \hat{O} \hat{T}^{-j}) | \phi \rangle$.

3.7. Thick-restart Lanczos method

The calculation of the lowest N_K eigenvalues of the Hamiltonian represented by the $d (\gg N_K)$ -dimensional matrix \mathbf{H} is often required when the low-energy physics of the quantum system is analyzed. When the lowest energy eigenvalue, namely $N_K = 1$, is focused, the Lanczos method is often used because it is simple to implement and also very powerful. In the Lanczos method, \mathbf{H} is mapped to a $N_M (> N_K)$ -dimensional tridiagonal matrix \mathbf{T} with $d \times N_M$ -dimensional projection matrix $\Psi = \{\psi_x\}_{1 \leq x \leq N_M}$ through procedures called the Lanczos process described in lines 1–8 and 10–13 in Algorithm 10. It is known that the lowest N_K eigenvalues of \mathbf{T} obtained through a sufficient number of Lanczos processes

(i.e. $d \gg N_M \gg N_K$) in principle reproduce those of the original Hamiltonian matrix \mathbf{H} . However, generally the Lanczos vectors $\{\psi_x\}_{1 \leq x \leq N_M}$ becomes non-orthogonal to each other as the Lanczos process progresses due to the round-off error, resulting in problems of pseudo-eigenvalues. Therefore, the low-energy eigenvalues cannot be obtained correctly in the standard Lanczos method.

To mitigate this round-off error, we need to introduce a restart procedure into the method, in addition to a reorthogonalization procedure for Lanczos vectors. There are many variants to introduce these procedures in the Lanczos method, and the thick-restart Lanczos method [35,36] employed in $Q S^3$ is one of them. An algorithm for the thick-restart Lanczos method is provided in Algorithm 10. In the first part (the lines 1–13) of this algorithm, the procedure of the conventional Lanczos method with a reorthogonalization process in the line 9 is employed to generate $N_M + 1$ Lanczos vectors, Ψ and ψ_{N_M+1} , and construct the tridiagonal matrix

$$\mathbf{T} = \Psi^\dagger \mathbf{H} \Psi := \begin{pmatrix} \alpha_1 & \beta_1 & & & & \\ \beta_1^* & \alpha_2 & \beta_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \beta_{N_M-2}^* & \alpha_{N_M-1} & \beta_{N_M-1} & \\ & & & \beta_{N_M-1}^* & \alpha_{N_M} & \end{pmatrix}, \quad (12)$$

where $\alpha_x = \psi_x^\dagger \mathbf{H} \psi_x$, $\beta_x = \psi_x^\dagger \mathbf{H} \psi_{x+1}$, and other elements are zero. Then, as shown in the line 13, this tridiagonal matrix \mathbf{T} is diagonalized to obtain the eigenvalues $\mathbf{e} = \{e_x\}$ in ascending order and the unitary matrix $\mathbf{C} = \{c_{x,x'}\}$ such that $\mathbf{T} = \mathbf{C}(\text{diag}[\mathbf{e}])\mathbf{C}^\dagger$.

The thick-restart Lanczos method focuses on the lowest $N_K (< N_M)$ eigenvalues and the corresponding eigenvectors by the keeping $N_K + 1$ vectors as shown in the lines 15 and 26, $\{\{\psi_y\}_{1 \leq y \leq N_K}, \psi_{N_K+1}\} := \{\Psi\{c_{x,y}\}_{1 \leq y \leq N_K}, \psi_{N_M+1}\}$. Then, the method generates the $N_M - N_K$ Lanczos vectors $\{\psi_{N_K+2}, \dots, \psi_{N_M+1}\}$, as shown in the lines 29–33, according to the procedures in the conventional Lanczos method with the initial vector ψ_{N_M+1} , and also the matrix \mathbf{T} through the procedures in the lines 34–37, namely

$$\mathbf{T} := \begin{pmatrix} e_1 & & & \beta_1 & & & & & & \\ & \ddots & & \vdots & & & & & & \\ & & e_{N_K} & \beta_{N_K} & & & & & & \\ \beta_1^* & \dots & \beta_{N_K}^* & \alpha_{N_K+1} & \beta_{N_K+1} & & & & & \\ & & & \ddots & \ddots & \ddots & & & & \\ & & & & \beta_{N_M-2}^* & \alpha_{N_M-1} & \beta_{N_M-1} & & & \\ & & & & & \beta_{N_M-1}^* & \alpha_{N_M} & & & \end{pmatrix}, \quad (13)$$

where $\{\beta_y\}_{1 \leq y \leq N_K} := \{\beta_{N_M} c_{N_M,y}\}$ and only elements not generally zero are shown. This matrix \mathbf{T} is then diagonalized to obtain the eigenvalues $\mathbf{e} = \{e_x\}$ in ascending order and the corresponding unitary matrix $\mathbf{C} = \{c_{x,x'}\}$ as shown in the line 38. This procedure is repeated until the lowest N_K eigenvalues are converged within the specified convergence ratio ϵ or the total number of iterations exceeds a given integer I_M . This is the second part (the lines 14–40) of the algorithm described in Algorithm 10. Therefore, the thick-restart Lanczos method requires maximally the $N_M (> N_K)$ dimensional Krylov subspace. The integer numbers N_M and I_M and the real number ϵ are input parameters, which determine the quality of the calculation.

3.8. Multiplying an operator to state vectors represented with the symmetry-adapted basis sets

The QS^3 package computes the static and dynamical spin structure factors after obtaining a target eigenvector $|\phi\rangle$ of the Hamil-

Algorithm 10 Thick-restart Lanczos method.

Input: integer $N_K (> 0)$, $N_M (> N_K)$, d , and $I_M (> 0)$, and real $\epsilon \ll 1$

Output: real $\mathbf{e} = \{e_x\}_{1 \leq x \leq N_M}$ and complex $\Psi = \{\psi_{ax}\}_{1 \leq a \leq d, 1 \leq x \leq N_M}$.

```

1: function THICK_RESTART_LANCZOS( $N_K, N_M, I_M, \epsilon$ )
2:    $\Psi_x = \{\psi_{ax}\}_{1 \leq a \leq d}$ ;  $\psi_1 := \text{RANDOM\_VEC}(d)$ 
   ▶ The function RANDOM_VEC( $d$ ) returns  $d$ -dimensional complex random vector.
3:    $\beta_0 := \sqrt{|\psi_1|^2}$ 
4:   for  $x = 1$  to  $N_M$  do
5:      $\psi_x := \psi_x / \beta_{x-1}$ 
6:      $\mathbf{v} := \text{HAM\_TO\_VEC}(\mathbf{r}, \mathbf{r}', \mathbf{J}^{xy}, \mathbf{J}^z, \sigma, k, \psi_x)$ 
7:      $\alpha_x := \psi_x^\dagger \cdot \mathbf{v}$ 
8:      $\psi_{x+1} := \begin{cases} \mathbf{v} - \alpha_x \psi_x & (x=1) \\ \mathbf{v} - \alpha_x \psi_x - \beta_{x-1} \psi_{x-1} & (x>1) \end{cases}$ 
9:      $\psi_{x+1} := \text{REORTHOGONALIZATION}(\Psi, x+1)$ 
   ▶ The function REORTHOGONALIZATION( $\Psi, x$ ) performs reorthogonalization, for
   example, with the modified Gram-Schmidt procedure, to numerically keep the
   orthogonality  $\psi_{x' < x}^\dagger \cdot \psi_x = 0$ .
10:     $\beta_x := \sqrt{|\psi_{x+1}|^2}$ 
11:  end for
12:   $\psi_{N_M+1} := \psi_{N_M+1} / \beta_{N_M}$ 
13:   $(\mathbf{e}, \mathbf{C} = \{c_{x,x'}\}) := \text{DIAG\_TRI}(\{\alpha_x\}, \{\beta_x\})$ 

```

The function $\text{DIAG_TRI}(\{\alpha_x\}, \{\beta_x\})$ returns eigenvalues \mathbf{e} (ascending order) and the corresponding eigenvectors \mathbf{C} of a real symmetric tridiagonal matrix with diagonal elements $\{\alpha_x\}$ and sub-diagonal elements $\{\beta_x\}$.

```

14:  for  $l = 1$  to  $I_M$  do
15:     $\{\psi_y\}_{1 \leq y \leq N_K} := \Psi\{c_{x,y}\}$ 
16:    if  $l = 1$  then
17:       $\mathbf{e}' = \{e_y\}_{1 \leq y \leq N_K}$ 
18:    else
19:      if  $\max_{1 \leq y \leq N_K} |e'_y / e_y - 1| < \epsilon$  then
20:        Exit
21:      else
22:         $\mathbf{e}' = \{e_y\}_{1 \leq y \leq N_K}$ 
23:      end if
24:    end if
25:     $\{\alpha_y\}_{1 \leq y \leq N_K} := \{e_y\}$ 
26:     $\psi_{N_K+1} := \psi_{N_M+1}$ 
27:     $\{\beta_y\}_{1 \leq y \leq N_K} := \{\beta_{N_M} c_{N_M,y}\}$ 
28:    lines 6 and 7 with  $x = N_K + 1$ .
29:     $\mathbf{v} := \mathbf{v} - \sum_y \beta_y \psi_y$ 
30:     $\psi_{N_K+2} := \mathbf{v} - \alpha_{N_K+1} \psi_{N_K+1}$ 
31:     $\psi_{N_K+2} := \text{REORTHOGONALIZATION}(\Psi, N_K + 2)$ 
32:     $\beta_{N_K+1} := \sqrt{|\psi_{N_K+2}|^2}$ 
33:    lines 4-12 with the starting value of  $x = N_K + 2$ .
34:     $\mathbf{T} := 0$ 
35:     $\{t_{xx}\}_{1 \leq x \leq N_M} = \{\alpha_x\}$ 
36:     $\{t_{y,N_K+1}\}_{1 \leq y \leq N_K} = \{t_{N_K+1,y}^*\} = \{\beta_y\}$ 
37:     $\{t_{z,z+1}\}_{N_K+1 \leq z \leq N_M-1} = \{t_{z+1,z}^*\} = \{\beta_z\}$ 
38:     $(\mathbf{e}, \mathbf{C}) := \text{DIAG}(\mathbf{T})$ 
   ▶ The function  $\text{DIAG}(\mathbf{T})$  returns eigenvalues (ascending order) and the
   corresponding eigenvectors of Hermitian matrix  $\mathbf{T}$ .
39:  end for
40:  return  $(\mathbf{e}, \Psi)$ 
41: end function

```

tonian matrix. Considering a periodic chain with N spins, as an example, the Fourier transform of the spin operator at wave number q with $\alpha = \pm, z$ is given as

$$\hat{S}_q^\alpha = \frac{1}{\sqrt{N}} \sum_{j=1}^N e^{-iqj} \hat{T}_j^\dagger \hat{S}_1^\alpha (\hat{T}_j)^\dagger. \quad (14)$$

One can easily show that the operator \hat{S}_q^α satisfies the following relation:

$$\hat{S}_q^\alpha \hat{T}^j = e^{-iqj} \hat{T}^j \hat{S}_q^\alpha. \quad (15)$$

Using this relation, one of the basic operations, $\hat{S}_q^\alpha |\phi\rangle$, necessary for computing the static and dynamical spin structure factors can be rewritten as

$$\hat{S}_q^\alpha |\phi\rangle = \sum_a \phi_{a,k} \hat{S}_q^\alpha |a, k\rangle \quad (16)$$

with

$$\begin{aligned}\hat{S}_q^\alpha |a, k\rangle &= \frac{1}{\sqrt{N_{a,k}}} \sum_j e^{-ikj} \hat{S}_q^\alpha \hat{T}^j |a\rangle \\ &= \frac{1}{\sqrt{N_{a,k}}} \sum_j e^{-i(k+q)j} \hat{T}^j \hat{S}_q^\alpha |a\rangle \\ &= \frac{1}{\sqrt{N_{a,k}N}} \sum_{j,j'} e^{-i(k+q)j} \hat{T}^j e^{-iqj'} \hat{T}^{j'} \hat{S}_1^\alpha \left(\hat{T}^{j'}\right)^\dagger |a\rangle,\end{aligned}\quad (17)$$

where $|\phi\rangle = \sum_a \phi_{a,k} |a, k\rangle$ is an eigenvector of the Hamiltonian matrix and it is in the subspace specified with momentum k and the number N_\downarrow of down spins. Note that in computing the transverse components of the spin structure factors with $\alpha = \pm$ in Eq. (17), we have to consider a transition between states with different $U(1)$ symmetry sectors, i.e., from a state $|a\rangle$ with N_\downarrow down spins to a state $\hat{S}_q^{\alpha=\pm} |a\rangle$ with $N_\downarrow \mp 1$ down spins.

Let us now introduce the spin state $|b\rangle$ defined as

$$\hat{T}^{j'} \hat{S}_1^\alpha \left(\hat{T}^{j'}\right)^\dagger |a\rangle = c |b\rangle \quad (18)$$

with $c = \langle b | \hat{T}^{j'} \hat{S}_1^\alpha \left(\hat{T}^{j'}\right)^\dagger |a\rangle$. Note that $|b\rangle$ as well as c depends on α , j' , and a . In general, the state $|b\rangle$ is not the representative state $|b\rangle \equiv |\min_j b_j\rangle$ for states $\{|b_j\rangle \equiv \hat{T}^j |b\rangle\}_{1 \leq j \leq N}$, and $|b\rangle$ can be translated to $|b\rangle$ by repeatedly applying the translational operator \hat{T} , i.e., $|b\rangle = \hat{T}^\ell |b\rangle$ with $1 \leq \ell \leq N$, where ℓ depends on $|b\rangle$. Therefore, we can rewrite Eq. (17) using the representative state $|b\rangle$ as

$$\begin{aligned}\hat{S}_q^\alpha |a, k\rangle &= \frac{1}{\sqrt{N_{a,k}N}} \sum_{j,j'} e^{-i(k+q)j} \hat{T}^j e^{-iqj'} c \hat{T}^{-\ell} |b\rangle \\ &= \frac{1}{\sqrt{N_{a,k}N}} \sum_{j,j'} e^{-i(qj'+(k+q)\ell)} c e^{-i(k+q)(j-\ell)} \hat{T}^{j-\ell} |b\rangle \\ &= \frac{1}{\sqrt{N_{a,k}N}} \sum_{j'} \sqrt{N_{b,k+q}} e^{-i(qj'+(k+q)\ell)} c |b, k+q\rangle.\end{aligned}\quad (19)$$

A concrete procedure for performing $\hat{S}_q^\alpha |\phi\rangle$ is shown in Algorithm 11. In the same manner, we can perform $\hat{S}_q^+ |\phi\rangle$ and $\hat{S}_q^- |\phi\rangle$.

3.9. Continued fraction expansion based on the Lanczos algorithm

Using the continued fraction expansion based on the Lanczos algorithm [1,38,39], the QS³ package computes the dynamical spin structure factor

$$S_q^\alpha(\omega) = -\frac{1}{\pi} \text{Im} \langle \phi | \hat{S}_q^{\alpha\dagger} \frac{1}{\omega - \hat{H} + E_0 + i\eta} \hat{S}_q^\alpha | \phi \rangle, \quad (20)$$

where E_0 is the ground state energy (i.e., lowest eigenvalue) with the corresponding ground state $|\phi\rangle$ of the Hamiltonian \hat{H} and positive real number η is the broadening factor. We can rewrite the above equation as

$$S_q^\alpha(\omega) = -\frac{1}{\pi} \text{Im} \frac{\langle \phi | \hat{S}_q^{\alpha\dagger} \hat{S}_q^\alpha | \phi \rangle}{z - \alpha_1 - \frac{\beta_1^2}{z - \alpha_2 - \frac{\beta_2^2}{z - \alpha_3 - \dots}}} \quad (21)$$

with $z = \omega - E_0 + i\eta$. α and β in Eq. (21) are obtained by the tridiagonalization procedure of the Hamiltonian matrix in the Lanczos iteration shown in lines 3-11 of Algorithm 10 with the initial state $|\psi\rangle_1 = \hat{S}_q^\alpha |\phi\rangle$ that can be prepared by the procedure shown in Algorithm 11.

Algorithm 11 Perform $|\psi'\rangle := \hat{S}_q^- |\phi\rangle$.

Input: real k and $q \in \{2\pi K/N\}_{0 \leq K < N}$, d -dimensional integer vector σ and real vector \mathbf{R} for the lists of representative states specified with N_\downarrow down spins and their normalization factors, respectively, d -dimensional complex vector ϕ , and d' -dimensional integer vector σ' and real vector \mathbf{R}' for the lists of representative states specified with $N_\downarrow + 1$ down spins and their normalization factors, respectively.

Output: d' -dimensional complex vector ψ' .

```

1: function SMO_TO_VEC( $\sigma, \mathbf{R}, k, \phi, \sigma', \mathbf{R}', q$ )
2:    $\psi' := 0$ 
3:   for  $r = 1$  to  $N$  do
4:     for  $a = 1$  to  $d$  do
5:        $\mathbf{n} = \text{F\_BAR}(\sigma_a, N_\downarrow, N)$ 
6:        $(p, f) := \text{BINARY\_SEARCH}(r, \mathbf{n}, 1, N_\downarrow)$ 
7:       if  $f = \text{False}$  then
8:          $\mathbf{n}' := (n_1, \dots, n_p, r, n_{p+1}, \dots, n_{N_\downarrow})$ 
9:          $b := F(\mathbf{n}')$ 
10:         $(b, \ell) := \text{REPRESENTATIVE}(b, N'_\downarrow)$ 
11:         $(b, f) := \text{BINARY\_SEARCH}(b, \sigma', 1, d')$ 
12:        if  $f = \text{True}$  then
13:           $\psi'_b := \psi'_b + \sqrt{\frac{R_b}{R_a N}} e^{-i(qr+(k+q)\ell)} \phi_a$ 
14:        end if
15:      end if
16:    end for
17:  end for
18:  return  $(\psi')$ 
19: end function
```

4. Benchmark results

4.1. Parallelization efficiency with openMP

Here we show a benchmark result of the QS³ package for the numerical diagonalization. The most time consuming part in the Lanczos algorithm is the Hamiltonian-vector multiplication in Algorithm 9 with computational complexity $O(N C_{N_\downarrow} N_{\text{xxz}} N_\downarrow \ln N_\downarrow)$. The QS³ package adopts OpenMP to parallelize this procedure. For a typical benchmark, we consider an $S = 1/2$ isotropic antiferromagnetic Heisenberg model on a simple cubic lattice of 216 sites ($L_x = L_y = L_z = 6$) and calculate the ground state in the subspace with momentum $\mathbf{k} = (0, 0, 0)$ and $N_\downarrow = 5$, by setting the parameters $(K_x, K_y, K_z) = (0, 0, 0)$, using the conventional Lanczos algorithm. The dimension of the Hilbert space (i.e., the Hamiltonian matrix) is ${}_{216}C_5 = (3,739,729,608)$ with only adapting the $U(1)$ symmetry and can be reduced $1/N$ times smaller down to 17,313,563 when the translational symmetry is also adapted. Fig. 2 shows the efficiency of the parallelization of the QS³ package executed using the supercomputer (Ohtaka) in ISSP with AMD Epyc 7702 2.0 GHz. We confirm almost linear acceleration with increasing the number of threads up to 128, although the slope becomes somewhat smaller when the number of threads exceeds around 20.

4.2. Energy-dispersion relation

One of the essential physical quantities to understand the low-energy physics of a quantum spin model is the energy-dispersion relation $E_0(\mathbf{k})$, the ground state energy at each momentum \mathbf{k} . Most of the currently available exact diagonalization libraries compute this quantity but are sufficient for practical use only in one-dimensional systems because of the severe limitations of the accessible system sizes. The QS³ package can evaluate the energy-dispersion relation around the saturation field even in three-dimensional systems.

For demonstration, we show the energy-dispersion relation for an $S = 1/2$ isotropic antiferromagnetic Heisenberg model on a simple cubic lattice of 1000 sites ($L_x = L_y = L_z = 10$) in Fig. 3. When only the $U(1)$ symmetry is used, the dimension of the Hilbert space with $N_\downarrow = 3$ is ${}_{1000}C_3 = 166,167,000$ and approx-

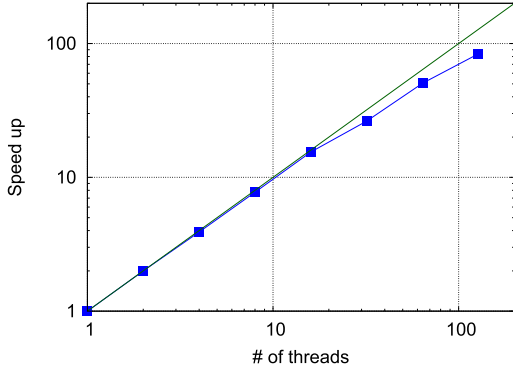


Fig. 2. Parallelization efficiency of the QS³ package. The conventional Lanczos algorithm is used to calculate the ground state of an $S = 1/2$ isotropic antiferromagnetic Heisenberg model on a simple cubic lattice of 216 sites with $(K_x, K_y, K_z) = (0, 0, 0)$ and $N_\downarrow = 5$. The vertical axis means multi-thread efficiency in the Lanczos iteration compared to the single-thread. The dark-green line indicates an ideal speed-up.

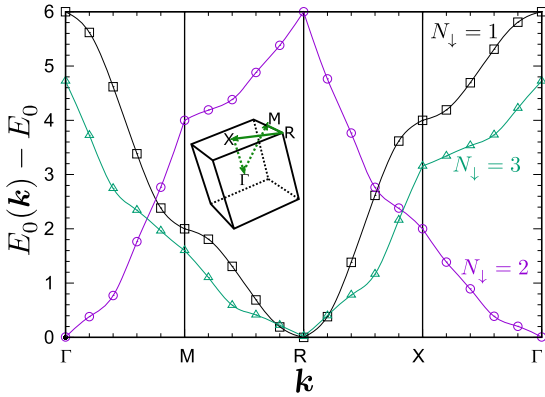


Fig. 3. Energy-dispersion relation $E_0(\mathbf{k})$ of an $S = 1/2$ isotropic antiferromagnetic Heisenberg model on a simple cubic lattice of 1000 sites ($L_x = L_y = L_z = 10$) with N_\downarrow down spins along the high symmetric momentum \mathbf{k} line indicated in the inset. The lowest eigenenergies $E_0(\mathbf{k})$ with $N_\downarrow = 1, 2$, and 3 near the saturation field are plotted relative to the ground state energy E_0 of the fully polarized state with $N_\downarrow = 0$. The high symmetric momentum points are indicated by Γ : $(0, 0, 0)$, M: $(0, \pi, \pi)$, R: (π, π, π) , and X: $(0, 0, \pi)$. Solid lines are the cubic-spline interpolation in each path, i.e., $\Gamma \rightarrow M$, $M \rightarrow R$, $R \rightarrow X$, and $X \rightarrow \Gamma$. Note that the ground-state momentum alternates between Γ and R for N_\downarrow even and odd, respectively, leading to a corresponding alternation of the minimum of the dispersion relation.

imately 2.5 GByte of physical memory is required to store a state vector with the double-complex precision. This implies that the total physical memory up to about 8 GByte is required to obtain the ground state by means of the conventional Lanczos method. This is rather expensive to perform on a standard laptop computer. However, by incorporating the translational symmetry, the required storage per a state vector is reduced down to around 2.5 MByte, and thus the computation can be executed easily with a laptop computer.

4.3. Static and dynamical structure factors

In Fig. 4, we also demonstrate the calculation of the static and dynamical spin structure factors for an $S = 1/2$ isotropic antiferromagnetic Heisenberg model on a square lattice of 100 sites ($L_x = L_y = 10$) with $N_\downarrow = 2$. The static spin structure factor $S_{\mathbf{q}}^\alpha$ is defined as

$$S_{\mathbf{q}}^\alpha = \langle \phi | \hat{S}_{\mathbf{q}}^{\alpha\dagger} \hat{S}_{\mathbf{q}}^\alpha | \phi \rangle \quad (22)$$

with $|\phi\rangle$ being the ground state and it is related to the dynamical spin structure factor $S_{\mathbf{q}}^\alpha(\omega)$ via

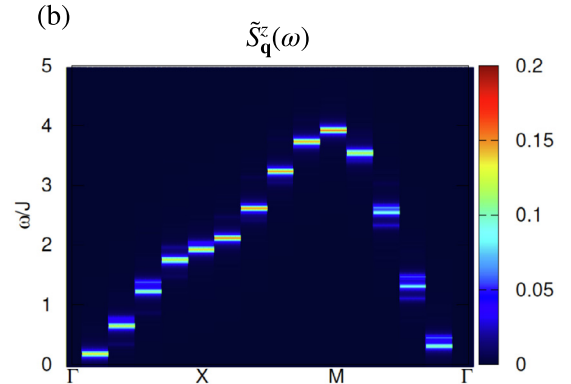
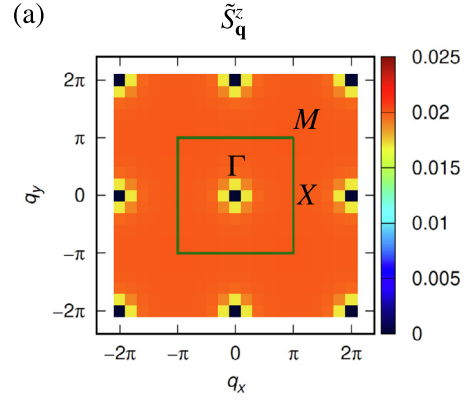


Fig. 4. (a) The z-component of the static spin structure factor $\tilde{S}_{\mathbf{q}}^z$ and (b) the z-component of the dynamical spin structure factor $\tilde{S}_{\mathbf{q}}^z(\omega)$ along the high symmetric momentum line for an $S = 1/2$ isotropic antiferromagnetic Heisenberg model on a square lattice of 100 sites ($L_x = L_y = 10$) with $N_\downarrow = 2$ near the saturation field. Note that the trivial component in the structure factors is subtracted for visibility (see the text). The high symmetric momenta are indicated by Γ : $(0, 0)$, X: $(\pi, 0)$, and M: (π, π) .

$$S_{\mathbf{q}}^\alpha = \int S_{\mathbf{q}}^\alpha(\omega) d\omega. \quad (23)$$

A nearly fully polarized state always displays a trivial but dominant sharp peak in the longitudinal structure factor at the Γ point and the symmetrically equivalent momenta. For ease of visibility, this trivial component is subtracted from the calculated static and dynamical longitudinal spin structure factors, denoted respectively as $\tilde{S}_{\mathbf{q}}^z$ and $\tilde{S}_{\mathbf{q}}^z(\omega)$ in Fig. 4. Here, the dynamical spin structure factor $\tilde{S}_{\mathbf{q}}^z(\omega)$ at $\mathbf{q} = \mathbf{0}$ is given as

$$\tilde{S}_{\mathbf{q}=\mathbf{0}}^z(\omega) \equiv S_{\mathbf{q}=\mathbf{0}}^z(\omega) - \frac{\eta}{\pi(\omega^2 + \eta^2)} \frac{M^2}{N} \quad (24)$$

and $\tilde{S}_{\mathbf{q}}^z(\omega)$ at $\mathbf{q} \neq \mathbf{0}$ is exactly the same as $S_{\mathbf{q}}^z(\omega)$.

5. Summary

We have developed the exact diagonalization package QS³ for analyzing spin-1/2 quantum lattice models with XXZ interactions near the saturation field. The QS³ package employs the symmetry-adapted basis sets with respect to the translational symmetry as well as the U(1) symmetry. In order to access large system sizes up to $O(1000)$, the QS³ package does not use the traditional bit representation for spin configurations. Introducing OpenMP parallelization, the bottleneck of the calculation, i.e., large dimension matrix-vector multiplication, is efficiently accelerated by the parallelization. The QS³ package computes fundamental physical quantities such as the local magnetization, two-point spin correlation

function, and the dynamical spin structure factor. These quantities are essential and observable in experiments. As demonstrated in the benchmark, the QS³ package can treat three-dimensional systems to understand the ground state as well as the low-energy excitations with potentially interesting properties.

For the future development, the QS³ package will be extended to treat the point-group symmetry in addition to the translational symmetry. We will also introduce the multiple degrees of freedom per unit cell, i.e., multiple spins per unit cell, to treat more general lattice geometries such as the kagome and pyrochlore lattices. In addition, we will extend the application of the QS³ packages to dilute fermionics, soft-core bosons, and higher-spin systems. These extensions are straightforward in terms of the coding employed in the QS³ package and most likely increase a value of the QS³ package as a research tool not only in condensed matter physics but also in quantum chemistry. For example, the QS³ package will be able to handle the full configuration interaction (full CI) calculation for molecules with a small number of electrons occupying many orbitals, which are difficult to treat by an available open-source package, e.g., given in Ref. [52].

Furthermore, we can implement a function to simulate quantum circuits with symmetry constraints. This direction of development is important to provide reference data for benchmark results of future large-scale universal quantum computers and to investigate quantum accelerated algorithms for quantum many-body systems. These extensions are in part under progress and will be reported in the near future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

H.U. thanks W. Mizukami for helpful comments. T.S. is supported by the Theory of Quantum Matter Unit of the Okinawa Institute of Science and Technology Graduate University (OIST). The work was partially supported by KAKENHI Nos. JP17K14359, JP18H01183, JP19K14665, JP21K03477, JP21H04446, JP21H05182 and JP21H05191, and by JST Nos. JPMJPR1911 and JPMJPF2014. This work was also supported by MEXT Q-LEAP Grant Number JPMXS0120319794 and the COE research grant in computational science from Hyogo Prefecture and Kobe City through Foundation for Computational Science. We are grateful for allocating computational resources of the HOKUSAI BigWaterfall supercomputing system at RIKEN. The QS³ package was also developed and performed using the facilities of the Supercomputer Center, ISSP, the University of Tokyo, the facilities of the research center for computational science of national institutes of natural sciences, and the facilities of computing section, OIST.

References

- [1] E. Dagotto, *Rev. Mod. Phys.* 66 (1994) 763.
- [2] MPI Forum, <https://www.mpi-forum.org/>.
- [3] A.W. Sandvik, *AIP Conf. Proc.* 1297 (2010) 135.
- [4] ALPS, <http://alps.comp-phys.org>.
- [5] A. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gürtler, A. Honecker, R. Igarashi, M. Körner, A. Kozhevnikov, A. Läuchli, S. Manmana, M. Matsumoto, I. McCulloch, F. Michel, R. Noack, G. Pawłowski, L. Pollet, T. Pruschke, U. Schollwöck, S. Todo, S. Trebst, M. Troyer, P. Werner, S.

- Wessel, in: *Proceedings of the 17th International Conference on Magnetism*, J. Magn. Magn. Mater. 310 (2007) 1187.
- [6] B. Bauer, L.D. Carr, H.G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S.V. Isakov, D. Koop, P.N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J.D. Picon, L. Pollet, E. Santos, V.W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M.L. Wall, P. Werner, S. Wessel, *J. Stat. Mech. Theory Exp.* 2011 (2011) P05001.
- [7] Spin Pack, <https://www-e.ovgu.de/jschulen/spin/>.
- [8] Rokko, <https://github.com/t-sakashita/rokkko>.
- [9] M. Kawamura, K. Yoshimi, T. Misawa, Y. Yamaji, S. Todo, N. Kawashima, *Comput. Phys. Commun.* 217 (2017) 180.
- [10] J. Oitmaa, D.D. Betts, *Can. J. Phys.* 56 (1978) 897, <https://doi.org/10.1139/p78-120>.
- [11] P.W. Leung, V. Elser, *Phys. Rev. B* 47 (1993) 5459.
- [12] C. Waldtmann, H.U. Everts, B. Bernu, C. Lhuillier, P. Sindzingre, P. Lecheminant, L. Pierre, *Eur. Phys. J. B, Condens. Matter Complex Syst.* 2 (1998) 501.
- [13] J. Richter, J. Schulenburg, *Eur. Phys. J. B* 73 (2010) 117.
- [14] H. Nakano, T. Sakai, *J. Phys. Soc. Jpn.* 79 (2010) 053707, <https://doi.org/10.1143/JPSJ.79.053707>.
- [15] H. Nakano, T. Sakai, *J. Phys. Soc. Jpn.* 80 (2011) 053704, <https://doi.org/10.1143/JPSJ.80.053704>.
- [16] A.M. Läuchli, J. Sudan, E.S. Sørensen, *Phys. Rev. B* 83 (2011) 212401.
- [17] H. Nakano, T. Sakai, *J. Phys. Soc. Jpn.* 84 (2015) 063705, <https://doi.org/10.7566/JPSJ.84.063705>.
- [18] H. Nakano, N. Todoroki, T. Sakai, *J. Phys. Soc. Jpn.* 88 (2019) 114702, <https://doi.org/10.7566/JPSJ.88.114702>.
- [19] A.M. Läuchli, J. Sudan, R. Moessner, *Phys. Rev. B* 100 (2019) 155142.
- [20] C. Lanczos, *J. Res. Natl. Bur. Stand. B* 45 (1950) 255.
- [21] A. Wietek, A.M. Läuchli, *Phys. Rev. E* 98 (2018) 033309.
- [22] K. Watanabe, H. Kawamura, H. Nakano, T. Sakai, *J. Phys. Soc. Jpn.* 83 (2014) 034714, <https://doi.org/10.7566/JPSJ.83.034714>.
- [23] H. Kawamura, K. Watanabe, T. Shimokawa, *J. Phys. Soc. Jpn.* 83 (2014) 103704, <https://doi.org/10.7566/JPSJ.83.103704>.
- [24] T. Shimokawa, K. Watanabe, H. Kawamura, *Phys. Rev. B* 92 (2015) 134407.
- [25] K. Uematsu, H. Kawamura, *J. Phys. Soc. Jpn.* 86 (2017) 044704, <https://doi.org/10.7566/JPSJ.86.044704>.
- [26] K. Uematsu, H. Kawamura, *Phys. Rev. B* 98 (2018) 134427.
- [27] K. Uematsu, H. Kawamura, *Phys. Rev. Lett.* 123 (2019) 087201.
- [28] TITPACK Ver. 2, http://www.qa.iir.titech.ac.jp/~nishimori/titpack2_new/index-e.html.
- [29] P. Weinberg, M. Bukov, *SciPost Phys.* 2 (2017) 003.
- [30] P. Weinberg, M. Bukov, *SciPost Phys.* 7 (2019) 20.
- [31] When we use standard 32 (64)-bit unsigned integer, the range is restricted from 0 to 2³²⁽⁶⁴⁾-1.
- [32] A. Honecker, *J. Phys. Condens. Matter* 11 (1999) 4697.
- [33] D. Sellmann, X.-F. Zhang, S. Eggert, *Phys. Rev. B* 91 (2015) 081104.
- [34] D. Yamamoto, H. Ueda, I. Danshita, G. Marmorini, T. Momoi, T. Shimokawa, *Phys. Rev. B* 96 (2017) 014431.
- [35] K. Wu, A. Canning, H. Simon, L.-W. Wang, *J. Comput. Phys.* 154 (1999) 156.
- [36] K. Wu, H. Simon, *SIAM J. Matrix Anal. Appl.* 22 (2000) 602, <https://doi.org/10.1137/S0895479898334605>.
- [37] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, et al., *LAPACK Users' Guide: Software, Environments, and Tools*, third edition, Society for Industrial and Applied Mathematics, 1999.
- [38] R. Haydock, V. Heine, M.J. Kelly, *J. Phys. C* 5 (1972) 2845.
- [39] E.R. Gagliano, C.A. Balseiro, *Phys. Rev. Lett.* 59 (1987) 2999.
- [40] M. Blume, Y.Y. Hsieh, *J. Appl. Phys.* 40 (1969) 1249, <https://doi.org/10.1063/1.1657616>.
- [41] A. Andreev, I.A. Grishchuk, *Zh. Eksp. Teor. Fiz.* 87 (1984) 267.
- [42] N. Shannon, T. Momoi, P. Sindzingre, *Phys. Rev. Lett.* 96 (2006) 027213.
- [43] B. Schmidt, P. Thalmeier, N. Shannon, *Phys. Rev. B* 76 (2007) 125113.
- [44] T. Vekua, A. Honecker, H.-J. Mikeska, F. Heidrich-Meisner, *Phys. Rev. B* 76 (2007) 174420.
- [45] L. Kecke, T. Momoi, A. Furusaki, *Phys. Rev. B* 76 (2007) 060407.
- [46] T. Hikihara, L. Kecke, T. Momoi, A. Furusaki, *Phys. Rev. B* 78 (2008) 144404.
- [47] J. Sudan, A. Läuchli, A.M. Läuchli, *Phys. Rev. B* 80 (2009) 140402.
- [48] H.T. Ueda, K. Totsuka, *Phys. Rev. B* 80 (2009) 014417.
- [49] M. Sato, T. Hikihara, T. Momoi, *Phys. Rev. Lett.* 110 (2013) 077206.
- [50] H.T. Ueda, T. Momoi, *Phys. Rev. B* 87 (2013) 144417.
- [51] H.T. Ueda, *J. Phys. Soc. Jpn.* 84 (2015) 023601, <https://doi.org/10.7566/JPSJ.84.023601>.
- [52] K.D. Vogiatzis, D. Ma, J. Olsen, L. Gagliardi, W.A. de Jong, *J. Chem. Phys.* 147 (2017) 184111.