21 Density-Matrix Renormalization Group Algorithms

Eric Jeckelmann

Institut für Theoretische Physik, Leibniz Universität Hannover, 30167 Hannover, Germany

In this chapter I will introduce the basic Density Matrix Renormalization Group (DMRG) algorithms for calculating ground states in quantum lattice many-body systems using the one-dimensional spin-$\frac{1}{2}$ Heisenberg model as illustration. I will attempt to present these methods in a manner which combines the advantages of both the traditional formulation in terms of renormalized blocks and superblocks and the new description based on matrix-product states. The latter description is physically more intuitive but the former description is more appropriate for writing an actual DMRG program. Pedagogical introductions to DMRG which closely follow the original formulation are available in [2, 1]. The conceptual background of DMRG and matrix-product states is discussed in the previous chapter and should be read before. Extensions of the basic DMRG algorithms are presented in the chapters that follow this one.

21.1 Introduction

The DMRG was developed by White [3, 4] in 1992 to overcome the problems arising in the application of real-space renormalization groups to quantum lattice many-body systems in solid-state physics. Since then the approach has been extended to a great variety of problems in all fields of physics and even in quantum chemistry. The numerous applications of DMRG are summarized in two recent review articles [5, 6]. Additional information about DMRG can be found at `http://www.dmrg.info`.

Originally, DMRG has been considered as an extension of real-space renormalization group methods. The key idea of DMRG is to renormalize a system using the information provided by a reduced density matrix rather than an effective Hamiltonian (as done in most renormalization groups), hence the name density-matrix renormalization. Recently, the connection between DMRG and matrix-product states has been emphasized (for a recent review, see [7]) and has lead to significant extensions of the DMRG approach. From this point of view, DMRG is an algorithm for optimizing a variational wavefunction with the structure of a matrix-product state.

The outline of this chapter is as follows: First I briefly introduce the DMRG matrix-product state and examine its relation to the traditional DMRG blocks and

E. Jeckelmann: *Density-Matrix Renormalization Group Algorithms*, Lect. Notes Phys. **739**, 597–619 (2008)
DOI 10.1007/978-3-540-74686-7_21

superblocks in Sect. 1. In the next three Sect. I present a numerical renormalization group method, then the infinite-system DMRG algorithm, and finally the finite-system DMRG algorithm. In Sect. 5 the use of additive quantum numbers is explained. In the next two sections the estimation of numerical errors and code optimization are discussed. In the last section some extensions of DMRG are presented.

21.2 Matrix-Product States and (Super-)Blocks

We consider a quantum lattice system with N sites $n = 1, \ldots, N$. Let $\mathcal{B}(n) = \{|s_n\rangle; s_n = 1, \ldots, d_n\}$ denote a complete basis of the Hilbert space for site n (all bases used here are orthonormal). The tensor product of these bases yields a complete basis of the system Hilbert space $\mathcal{H}$

$$\{|\boldsymbol{s} = (s_1, \ldots, s_N)\rangle = |s_1\rangle \otimes \cdots \otimes |s_N\rangle; s_n = 1, \ldots, d_n; n = 1, \ldots, N\} \ . \quad (21.1)$$

For instance, for the spin-$\frac{1}{2}$ Heisenberg model $d_n = 2$ and $\mathcal{B}(n) = \{|\uparrow\rangle, |\downarrow\rangle\}$.

Any state $|\psi\rangle$ of $\mathcal{H}$ can be expanded in this basis: $|\psi\rangle = \sum_{\boldsymbol{s}} c(\boldsymbol{s})|\boldsymbol{s}\rangle$. As explained in Chap. 20, the coefficients $c(\boldsymbol{s})$ can take the form of a matrix product. Here we consider a particular matrix-product state

$$c(\boldsymbol{s}) = \mathsf{A}_1(s_1) \ldots \mathsf{A}_j(s_j) \mathsf{C}_j \mathsf{B}_{j+1}(s_{j+1}) \ldots \mathsf{B}_N(s_N) \ , \quad (21.2)$$

where C_j is a $(a_j \times b_{j+1})$-matrix (i.e., with a_j rows and b_{j+1} columns). The $(a_{n-1} \times a_n)$-matrices $\mathsf{A}_n(s_n)$ (for $s_n = 1, \ldots, d_n; n = 1, \ldots, j$) and the $(b_n \times b_{n+1})$-matrices $\mathsf{B}_n(s_n)$ (for $s_n = 1, \ldots, d_n; n = j+1, \ldots, N$) fulfill the orthonormalization conditions

$$\sum_{s_n=1}^{d_n} (\mathsf{A}_n(s_n))^\dagger \, \mathsf{A}_n(s_n) = \mathsf{I} \quad \text{and} \quad \sum_{s_n=1}^{d_n} \mathsf{B}_n(s_n) \, (\mathsf{B}_n(s_n))^\dagger = \mathsf{I} \ , \quad (21.3)$$

(I is the identity matrix), and the boundary condition $a_0 = b_{N+1} = 1$. These conditions imply that $a_n \leq d_n a_{n-1} \leq \prod_{k=1}^{n} d_k$ and $b_n \leq d_n b_{n+1} \leq \prod_{k=n}^{N} d_k$.

Obviously, this matrix-product state splits the lattice sites in two groups. The sites $n = 1, \ldots, j$ make up a left block $L(j)$ and the sites $n = j+1, \ldots, N$ constitute a right block $R(j+1)$. From the matrix elements of $\mathsf{A}_n(s_n)$ and $\mathsf{B}_n(s_n)$ we can define third-rank tensors

$$\phi_\alpha^{L(n)}(\alpha', s_n) = [\mathsf{A}_n(s_n)](\alpha', \alpha) \ , \quad (21.4)$$

for $s_n = 1, \ldots, d_n$; $\alpha = 1, \ldots, a_n$, and $\alpha' = 1, \ldots, a_{n-1}$, and

$$\phi_\beta^{R(n)}(s_n, \beta') = [\mathsf{B}_n(s_n)](\beta, \beta') \ , \quad (21.5)$$

for $s_n = 1, \ldots, d_n$; $\beta = 1, \ldots, b_n$, and $\beta' = 1, \ldots, b_{n+1}$. Using these tensors one can iteratively define a set of orthonormal states in the Hilbert space associated with each left block

$$\left|\phi_\alpha^{L(1)}\right\rangle = |s_1\rangle \; ; \; \alpha = s_1 = 1, \ldots, d_1;$$
$$\left|\phi_\alpha^{L(n)}\right\rangle = \sum_{\alpha'=1}^{a_{n-1}} \sum_{s_n=1}^{d_n} \phi_\alpha^{L(n)}(\alpha', s_n) \left|\phi_{\alpha'}^{L(n-1)}\right\rangle \otimes |s_n\rangle \; ; \; \alpha = 1, \ldots, a_n \; , \tag{21.6}$$

and each right block

$$\left|\phi_\beta^{R(N)}\right\rangle = |s_N\rangle \; ; \; \beta = s_N = 1, \ldots, d_N;$$
$$\left|\phi_\beta^{R(n)}\right\rangle = \sum_{\beta'=1}^{b_{n+1}} \sum_{s_n=1}^{d_n} \phi_\beta^{R(n)}(s_n, \beta') \, |s_n\rangle \otimes \left|\phi_{\beta'}^{R(n+1)}\right\rangle \; ; \; \beta = 1, \ldots, b_n \; . \tag{21.7}$$

The orthonormality of each set of block states (i.e., the states belonging to the same block Hilbert space) follows directly from the orthonormalization conditions for the matrices $\mathsf{A}_n(s_n)$ and $\mathsf{B}_n(s_n)$.

Every set of block states spans a subspace of the Hilbert space associated with the block. Using these states one can build an effective or renormalized (i.e., approximate) representation of dimension a_n or b_n for every block. By definition, an effective representation of dimension a_n for the block $L(n)$ is made of vector and matrix representations in a subspace basis $\mathcal{B}(L, n)$ for every state and operator (acting on sites in $L(n)$) which our calculation requires. Note that if $a_n = \prod_{k=1}^{n} d_k$, the block state set is a complete basis of the block Hilbert space and the "effective" representation is actually exact. An effective representation of dimension b_n for a right block $R(n)$ is defined similarly using a subspace basis $\mathcal{B}(R, n)$.

If we combine the left block $L(j)$ with the right block $R(j+1)$, we obtain a so-called superblock $\{L(j) + R(j+1)\}$ which contains the sites 1 to N. The tensor-product basis $\mathcal{B}(SB, j) = \mathcal{B}(L, j) \otimes \mathcal{B}(R, j+1)$ of the block bases is called a superblock basis and spans a $(a_j b_{j+1})$-dimensional subspace of the system Hilbert space $\mathcal{H}$. The matrix-product state given by (21.2) can be expanded in this basis

$$|\psi\rangle = \sum_{\alpha=1}^{a_j} \sum_{\beta=1}^{b_{j+1}} [\mathsf{C}_j](\alpha, \beta) \, |\phi_\alpha^{L(j)} \phi_\beta^{R(j+1)}\rangle \; , \tag{21.8}$$

where $[\mathsf{C}_j](\alpha, \beta)$ denotes the matrix elements of C_j and

$$|\phi_\alpha^{L(j)} \phi_\beta^{R(j+1)}\rangle = |\phi_\alpha^{L(j)}\rangle \otimes |\phi_\beta^{R(j+1)}\rangle \in \mathcal{B}(SB, j) \; . \tag{21.9}$$

We note that the square norm of $|\psi\rangle$ is given by $\langle\psi|\psi\rangle = \mathrm{Tr} \; \mathsf{C}_j^\dagger \mathsf{C}_j$.

If $a_j = \prod_{n=1}^{j} d_n$ and $b_{j+1} = \prod_{n=j+1}^{N} d_n$, the superblock basis $\mathcal{B}(SB, j)$ is a complete basis of $\mathcal{H}$ and any state $|\psi\rangle \in \mathcal{H}$ can be written in the form (21.8). For a large lattice these conditions mean that some matrix dimensions are very large (at least $2^{N/2}$ for a spin-$\frac{1}{2}$ model). However, a matrix-product state is numerically

tractable only if all matrix dimensions are kept small, for instance $a_n, b_k \leq m$ with m up to a few thousands. A matrix-product state with restricted matrix sizes can be considered as an approximation for states in $\mathcal{H}$. In particular, it can be used as a variational ansatz for the ground state of the system Hamiltonian H. Thus the system energy $E = \langle\psi|H|\psi\rangle/\langle\psi|\psi\rangle$ is a function of the matrices $\mathsf{A}_n(s_n)$, $\mathsf{B}_n(s_n)$, and C_j. It has to be minimized with respect to these variational parameters subject to the constraints (21.3) to determine the ground state. In the following sections I will present three algorithms (a numerical renormalization group, the infinite-system DMRG method, and the finite-system DMRG method) for carrying out this minimization.

21.3 Numerical Renormalization Group

The Numerical Renormalization Group (NRG) method was developed by Wilson a few decades ago to solve the Kondo impurity problem [8]. The key idea is a decomposition of the lattice into subsystems (blocks) of increasing size. To calculate the ground state properties of a large lattice one starts from an exact representation of a small subsystem and builds effective representations of larger subsystems iteratively, adding one site at every iteration as illustrated in Fig. 21.1. Here I formulate this procedure for a quantum lattice system in the framework of a matrix-product state (21.2). To find a fixed point in an infinite chain, we consider that $j \equiv N$ in (21.2) while for a finite lattice size N we set $b_n = 1$ for all sites n. In both cases the right blocks do not play any role and j is increased by one in every iteration using the following procedure.

We want to calculate an effective representation of dimension a_{j+1} for the left block $L(j+1)$ assuming that we know an effective representation of dimension a_j for the left block $L(j)$. First, from the known bases $\mathcal{B}(L, j)$ of $L(j)$ and $\mathcal{B}(j+1)$ for the site $j+1$ we can define a tensor-product basis of dimension $a_j d_{j+1}$ for $L(j+1)$

$$\left|\phi_\alpha^{L(j)}\ s_{j+1}\right\rangle = \left|\phi_\alpha^{L(j)}\right\rangle \otimes |s_{j+1}\rangle\ , \tag{21.10}$$

with $\alpha = 1, \ldots, a_j$ and $s_{j+1} = 1, \ldots, d_{j+1}$. Second, every operator acting on sites in $L(j+1)$ can be decomposed into a sum of operator pairs

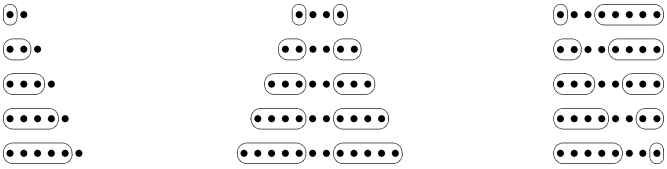

Fig. 21.1. Schematic representations of the NRG method (**left**), the infinite-system DMRG (**center**), and the finite-system DMRG (**right**). *Solid circles* are lattice sites and ovals are blocks. Going from top to bottom corresponds to the iterations $L(1) \to L(2) \to \cdots \to L(5)$ for the three methods. In the right picture, going from bottom to top corresponds to the iterations $R(N = 8) \to R(7) \to \cdots \to R(4)$ in a sweep from right to left of the finite-system DMRG

$$\mathcal{O} = \sum_k \mathcal{O}^{L,k}\, \mathcal{O}^{S,k} \;, \tag{21.11}$$

where the operators $\mathcal{O}^{L,k}$ act only on sites in $L(j)$ and the operators $\mathcal{O}^{S,k}$ act only on the site $j+1$. For instance, the (one-dimensional) Heisenberg Hamiltonian on the block $L(j+1)$

$$H = \sum_{n=1}^{j} \boldsymbol{S}_n \boldsymbol{S}_{n+1} \;, \tag{21.12}$$

can be decomposed as

$$H = \sum_{n=1}^{j-1} \boldsymbol{S}_n \boldsymbol{S}_{n+1} \otimes I + S_j^z \otimes S_{j+1}^z + \frac{1}{2}\left(S_j^+ \otimes S_{j+1}^- + S_j^- \otimes S_{j+1}^+\right) \;, \tag{21.13}$$

where I is the identity operator and $\boldsymbol{S}_n, S_n^z, S_n^+, S_n^-$ are the usual spin operators for the site n. As a result the matrix representation of $\mathcal{O}$ in the basis (21.10)

$$\mathcal{O}(\alpha, s_{j+1}, \alpha', s'_{j+1}) = \left\langle \phi_\alpha^{L(j)}\; s_{j+1} \right| \mathcal{O} \left| \phi_{\alpha'}^{L(j)}\; s'_{j+1} \right\rangle \;, \tag{21.14}$$

is given by

$$\mathcal{O}(\alpha, s_{j+1}, \alpha', s'_{j+1}) = \sum_k \mathcal{O}^{L,k}(\alpha, \alpha')\, \mathcal{O}^{S,k}(s_{j+1}, s'_{j+1}) \;, \tag{21.15}$$

where

$$\mathcal{O}^{L,k}(\alpha, \alpha') = \left\langle \phi_\alpha^{L(j)} \right| \mathcal{O}^{L,k} \left| \phi_{\alpha'}^{L(j)} \right\rangle \tag{21.16}$$

denotes the known matrix representations of $\mathcal{O}^{L,k}$ in the basis $\mathcal{B}(L,j)$ of the block $L(j)$. The matrix representations of the site operators

$$\mathcal{O}^{S,k}(s_{j+1}, s'_{j+1}) = \langle s_{j+1} |\, \mathcal{O}^{S,k} \left| s'_{j+1} \right\rangle \;, \tag{21.17}$$

can be calculated exactly. For instance, they correspond to the Pauli matrices for the spin operators S_n^x, S_n^y, S_n^z in the spin-$\frac{1}{2}$ basis $\mathcal{B}(n) = \{|\uparrow\rangle, |\downarrow\rangle\}$.

Using this procedure we can construct the matrix representation (21.14) of the Hamiltonian (restricted to the block $L(j+1)$) in the basis (21.10). This matrix can be fully diagonalized numerically. In practice, this sets an upper limit of a few thousands on $a_j d_{j+1}$. The eigenvectors are denoted $\phi_\mu^{L(j+1)}(\alpha, s_{j+1})$ for $\mu = 1, \ldots, a_j d_{j+1}$ and are ordered by increasing eigenenergies $\epsilon_\mu^{L(j+1)}$. The a_{j+1} eigenvectors with the lowest eigenenergies are used to define a new basis $\mathcal{B}(L, j+1)$ of $L(j+1)$ through (21.6) and the other eigenvectors are discarded. The matrix representation in $\mathcal{B}(L, j+1)$ for any operator acting in $L(j+1)$

$$\mathcal{O}(\mu, \mu') = \left\langle \phi_\mu^{L(j+1)} \right| \mathcal{O} \left| \phi_{\mu'}^{L(j+1)} \right\rangle \;;\; \mu, \mu' = 1, \ldots, a_{j+1} \;, \tag{21.18}$$

can be calculated using the orthogonal transformation and projection defined by the reduced set of eigenvectors. Explicitly, we have to perform two successive matrix products

$$M(\alpha, s_{j+1}, \mu') = \sum_{\alpha'=1}^{a_j} \sum_{s'_{j+1}=1}^{d_{j+1}} \mathcal{O}(\alpha, s_{j+1}, \alpha', s'_{j+1})\, \phi_{\mu'}^{L(j+1)}(\alpha', s'_{j+1}) \,,$$

$$\mathcal{O}(\mu, \mu') = \sum_{\alpha=1}^{a_j} \sum_{s_{j+1}=1}^{d_{j+1}} \left(\phi_{\mu}^{L(j+1)}(\alpha, s_{j+1})\right)^* M(\alpha, s_{j+1}, \mu') \,. \quad (21.19)$$

Vector representations of states in $L(j+1)$ can be obtained using the same principles. Therefore, we have obtained an effective representation of dimension a_{j+1} for the block $L(j+1)$. We note that the block states (21.6) are not explicitly calculated. Only matrix and vector representations for operators and states in that basis and the transformation from a basis to the next one need to be calculated explicitly.

Once the effective representation of $L(j+1)$ has be determined, the procedure can be repeated to obtain the effective representation of the next larger block. This procedure has to be iterated until $j+1 = N$ for a finite system or until a fixed point is reached if one investigates an infinite system. After the last iteration physical quantities for the (approximate) ground state and low-energy excitations can be calculated using the effective representation of $L(N)$. For instance, expectation values are given by

$$\langle\psi|\mathcal{O}|\psi\rangle = \sum_{\mu,\mu'=1}^{a_N} [\mathsf{C}_N^\dagger](\mu)\, \mathcal{O}(\mu, \mu')\, [\mathsf{C}_N](\mu') \,, \quad (21.20)$$

where $\mathcal{O}(\mu, \mu')$ is the matrix representation of $\mathcal{O}$ in the basis $\mathcal{B}(L, N)$ and C_N is the $(a_N \times 1)$-matrix corresponding to the state $|\psi\rangle$ in (21.2) and (21.8). For the ground state we obviously have $[\mathsf{C}_N](\mu) = \delta_{\mu,1}$.

The NRG method is efficient and accurate for quantum impurity problems such as the Kondo model but fails utterly for quantum lattice problems such as the Heisenberg model. One reason is that in many quantum systems the exact ground state can not be represented accurately by a matrix-product state (21.2) with restricted matrix sizes. However, another reason is that in most cases the NRG algorithm does not generate the optimal block representation for the ground state of a quantum lattice system and thus does not even find the matrix-product state (21.2) with the minimal energy for given matrix sizes.

21.4 Infinite-System DMRG Algorithm

The failure of the NRG method for quantum lattice problems can be understood qualitatively. The subsystem represented by a block $L(n)$ always has an artificial boundary at which the low-energy eigenstates of a quantum lattice Hamiltonian

tend to vanish. Thus at later iterations the low-energy eigenstates of the effective Hamiltonian in larger subsystems have unwanted features like nodes where the artificial boundaries of the previous subsystems were located. White and Noack [9] have shown that this difficulty can be solved in single-particle problems if the effects of the subsystem environment are taken into account self-consistently. DMRG is the application of this idea to many-particle problems. In his initial papers [3, 4], White described two DMRG algorithms: The infinite-system method presented in this section and the finite-system method discussed in the next section.

The infinite-system method is certainly the simplest DMRG algorithm and is the starting point of many other DMRG methods. In this approach the system size increases by two sites in every iteration, $N \to N+2$, as illustrated in Fig. 21.1. The right block $R(j+1)$ is always an image (reflection) of the left block $L(j)$, which implies that $j \equiv N/2$ in (21.2). Therefore, the superblock structure is $\{L(N/2) + R(N/2+1)\}$ and an effective representation for the N-site system is known if we have determined one for $L(N/2)$.

As in the NRG method an iteration consists in the calculation of an effective representation of dimension a_{j+1} for the block $L(j+1)$ assuming that we already know an effective representation of dimension a_j for the block $L(j)$. First, we proceed as with the NRG method and determine an effective representation of dimension $a_j d_{j+1}$ for $L(j+1)$ using the tensor product basis (21.10). Next, the effective representation of $R(j+2)$ is chosen to be an image of $L(j+1)$. The quantum system is assumed to be homogeneous and symmetric (invariant under a reflection $n \to n' = N - n + 3$ through the middle of the (N+2)-site lattice) to allow for this operation. Therefore, one can define a one-to-one mapping between the site and block bases on the left- and right-hand sides of the superblock. We consider a mapping between the tensor product bases for $L(j+1)$ and $R(j+2)$

$$\left|\phi_\alpha^{L(j)} s_{j+1}\right\rangle \leftrightarrow \left|s_{j+2}\, \phi_\beta^{R(j+3)}\right\rangle \ . \tag{21.21}$$

Thus, the matrix representation of any operator acting in $R(j+2)$

$$\mathcal{O}(s_{j+2}, \beta, s'_{j+2}, \beta') = \left\langle s_{j+2}\, \phi_\beta^{R(j+3)}\right| \mathcal{O} \left|s'_{j+2}\, \phi_{\beta'}^{R(j+3)}\right\rangle \ , \tag{21.22}$$

is given by the matrix representation (21.14) of the corresponding (reflected) operator in $L(j+1)$ through the basis mapping.

A superblock basis $\mathcal{B}(SB, j+1)$ of dimension $\mathcal{D}_{j+1} = a_j d_{j+1} d_{j+2} b_{j+3}$ can be defined using the tensor product of the block bases

$$\left|\phi_\alpha^{L(j)} s_{j+1}\, s_{j+2}\, \phi_\beta^{R(j+3)}\right\rangle = \left|\phi_\alpha^{L(j)} s_{j+1}\right\rangle \otimes \left|s_{j+2}\, \phi_\beta^{R(j+3)}\right\rangle \ , \tag{21.23}$$

for $\alpha = 1, \ldots, a_j$; $s_{j+1} = 1, \ldots, d_{j+1}$; $s_{j+2} = 1, \ldots, d_{j+2}$; and $\beta = 1, \ldots, b_{j+3}$. Every operator acting on the superblock (i.e., the $(N+2)$-site lattice) can be decomposed in a sum of operator pairs

$$\mathcal{O} = \sum_{k=1}^{n_k} \mathcal{O}^{L,k}\, \mathcal{O}^{R,k} \;, \tag{21.24}$$

where the operator parts $\mathcal{O}^{L,k}$ and $\mathcal{O}^{R,k}$ act on sites in $L(j+1)$ and $R(j+2)$, respectively. As an example, the Heisenberg Hamiltonian on a $(N+2)$-site chain can be written

$$H = \sum_{n=1}^{j} \boldsymbol{S}_n \boldsymbol{S}_{n+1} \otimes I + I \otimes \sum_{n=j+2}^{N+1} \boldsymbol{S}_n \boldsymbol{S}_{n+1} + S^z_{j+1} \otimes S^z_{j+2} + \frac{1}{2}\left(S^+_{j+1} \otimes S^-_{j+2} + S^-_{j+1} \otimes S^+_{j+2}\right) \;, \tag{21.25}$$

where I is the identity operator. Therefore, the matrix representation of any operator in the superblock basis

$$\mathcal{O}(\alpha, s_{j+1}, s_{j+2}, \beta, \alpha', s'_{j+1}, s'_{j+2}, \beta') = \left\langle \phi^{L(j)}_{\alpha} s_{j+1}\; s_{j+2}\; \phi^{R(j+3)}_{\beta} \right| \mathcal{O} \left| \phi^{L(j)}_{\alpha'} s'_{j+1}\; s'_{j+2}\; \phi^{R(j+3)}_{\beta'} \right\rangle \;, \tag{21.26}$$

(for $\alpha, \alpha' = 1, \ldots, a_j$; $s_{j+1}, s'_{j+1} = 1, \ldots, d_{j+1}$; $s_{j+2}, s'_{j+2} = 1, \ldots, d_{j+2}$; and $\beta, \beta' = 1, \ldots, b_{j+3}$) is given by the sum of the tensor products of the matrix representations (21.14) and (21.22) for the block operators

$$\mathcal{O}(\alpha, s_{j+1}, s_{j+2}, \beta, \alpha', s'_{j+1}, s'_{j+2}, \beta') = \sum_{k=1}^{n_k} \mathcal{O}^{L,k}(\alpha, s_{j+1}, \alpha', s'_{j+1})\, \mathcal{O}^{R,k}(s_{j+2}, \beta, s'_{j+2}, \beta') \;. \tag{21.27}$$

Storing the matrix representations (21.14) and (21.22) for the block operators requires a memory amount $\propto n_k[(a_j d_{j+1})^2 + (d_{j+2} b_{j+3})^2]$, but calculating and storing the superblock matrix (21.26) require $n_k (\mathcal{D}_{j+1})^2$ additional operations and a memory amount $\propto (\mathcal{D}_{j+1})^2$. As the number of operator pairs n_k is typically much smaller than the matrix dimensions a_j, b_{j+3} ($n_k = 5$ in the Heisenberg model on a open chain), one should not calculate the superblock matrix representation (21.26) explicitly but work directly with the right-hand side of (21.27). For instance, the application of the operator $\mathcal{O}$ to a state $|\psi\rangle \in \mathcal{H}$ yields a new state $|\psi'\rangle = \mathcal{O}|\psi\rangle$, which can be calculated without computing the superblock matrix (21.26) explicitly. If

$$[\mathsf{C}_{j+1}](\alpha, s_{j+1}, s_{j+2}, \beta) = \left\langle \phi^{L(j)}_{\alpha} s_{j+1}\; s_{j+2}\; \phi^{R(j+3)}_{\beta} \right| \psi \Big\rangle \;, \tag{21.28}$$

is the vector representation of $|\psi\rangle$ in the superblock basis (21.23), the vector representation C'_{j+1} of $|\psi'\rangle$ in this basis is obtained through double matrix products with the block operator matrices in (21.27)

$$
\begin{aligned}
&V_k(\alpha', s'_{j+1}, s_{j+2}, \beta) \\
&\qquad = \sum_{\beta'=1}^{b_{j+3}} \sum_{s'_{j+2}=1}^{d_{j+2}} [\mathsf{C}_{j+1}](\alpha', s'_{j+1}, s'_{j+2}, \beta')\, \mathcal{O}^{R,k}(s_{j+2}, \beta, s'_{j+2}, \beta') \;, \\
&[\mathsf{C}'_{j+1}](\alpha, s_{j+1}, s_{j+2}, \beta) \\
&\qquad = \sum_{k=1}^{n_k} \sum_{\alpha'=1}^{a_j} \sum_{s'_{j+1}=1}^{d_{j+1}} \mathcal{O}^{L,k}(\alpha, s_{j+1}, \alpha', s'_{j+1}) V_k(\alpha', s'_{j+1}, s_{j+2}, \beta) \;.
\end{aligned}
\tag{21.29}
$$

Performing these operations once requires only $n_k \mathcal{D}_{j+1}(a_j d_{j+1} + d_{j+2} b_{j+3})$ operations, while computing a matrix-vector product using the superblock matrix (21.26) would require $(\mathcal{D}_{j+1})^2$ operations. In practice, this sets an upper limit of the order of a few thousands for the matrix dimensions a_n, b_n.

As we want to calculate the ground state of the system Hamiltonian H, the next task is to set up the superblock representation (21.27) of H and then to determine the vector representation (21.28) of its ground state in the superblock basis. To determine the ground state without using the superblock matrix (21.26) of H we use iterative methods such as the Lanczos algorithm or the Davidson algorithm, see Chap. 18. These algorithms do not require an explicit matrix for H but only the operation $|\psi'\rangle = H|\psi\rangle$, which can be performed very efficiently with (21.29) as discussed above.

Once the superblock ground state C_{j+1} has been determined, the next step is finding an effective representation of dimension $a_{j+1} < a_j d_{j+1}$ for $L(j+1)$ which described this ground state as closely as possible. Thus we look for the best approximation $\widetilde{\mathsf{C}}_{j+1}$ of the superblock ground state C_{j+1} with respect to a new basis $\mathcal{B}(L, j+1)$ of dimension a_{j+1} for $L(j+1)$. As discussed in Chap. 20 this can be done using the Schmidt decomposition or more generally reduced density matrices. Choosing the density-matrix eigenvectors with the highest eigenvalues is an optimal choice for constructing a smaller block basis (see Sect. 21.7). Therefore, if the DMRG calculation targets a state with a vector representation $[\mathsf{C}_{j+1}](\alpha, s_{j+1}, s_{j+2}, \beta)$ in the superblock basis (21.23), we calculate the reduced density matrix for the left block $L(j+1)$

$$
\begin{aligned}
&\rho(\alpha, s_{j+1}, \alpha', s'_{j+1}) \\
&\qquad = \sum_{s_{j+2}=1}^{d_{j+2}} \sum_{\beta=1}^{b_{j+3}} \left([\mathsf{C}_{j+1}](\alpha, s_{j+1}, s_{j+2}, \beta)\right)^* [\mathsf{C}_{j+1}](\alpha', s'_{j+1}, s_{j+2}, \beta)
\end{aligned}
\tag{21.30}
$$

for $\alpha, \alpha' = 1, \ldots, a_j$ and $s_{j+1}, s'_{j+1} = 1, \ldots, d_{j+1}$. This density matrix has $a_j d_{j+1}$ eigenvalues $w_\mu \geq 0$ with

$$
\sum_{\mu=1}^{a_j d_{j+1}} w_\mu = 1 \;. \tag{21.31}
$$

We note $\phi_\mu^{L(j+1)}(\alpha, s_{j+1})$ the corresponding eigenvectors. The a_{j+1} eigenvectors with the largest eigenvalues are used to define a new basis $\mathcal{B}(L, j+1)$ of $L(j+1)$ through (21.6) and the other eigenvectors are discarded. As done in the NRG method, the matrix representation of any operator in $L(j+1)$ can be calculated using the orthogonal transformation and projection (21.19) defined by the reduced set of eigenvectors. If necessary, vector representations of states in $L(j+1)$ can be obtained using the same principles.

Thus, we have obtained an effective representation of dimension a_{j+1} for the block $L(j+1)$. We note that as with the NRG method the block states (21.6) are not explicitly calculated. Only matrix and vector representations of operators and states in that basis and the transformation from a basis to the next one need to be calculated explicitly. The procedure can be repeated to obtain an effective representation of the next larger blocks (i.e., for the next larger lattice size). Iterations are continued until a fixed point has been reached.

As an illustration Fig. 21.2 shows the convergence of the ground state energy per site as a function of the superblock size N in the one-dimensional spin-$\frac{1}{2}$ Heisenberg model. The energy per site $E_{\mathrm{DMRG}}(N)$ is calculated from the total energy E_0 for two consecutive superblocks $E_{\mathrm{DMRG}}(N) = [E_0(N) - E_0(N-2)]/2$. The exact result for an infinite chain is $E_{\mathrm{exact}} = \frac{1}{4} - \ln(2)$ according to the Bethe ansatz solution [10]. The matrix dimensions a_n, b_n are chosen to be not greater than a number m which is the maximal number of density-matrix eigenstates kept at each iteration. As N increases, $E_{\mathrm{DMRG}}(N)$ converges to a limiting value $E_{\mathrm{DMRG}}(m)$ which is the minimal energy for a matrix-product state (21.2) with matrix dimensions up to m. This energy minimum $E_{\mathrm{DMRG}}(m)$ is always higher than the exact ground state energy E_{exact} as expected for a variational method. The error in $E_{\mathrm{DMRG}}(m)$ is dominated by truncation errors, which decrease rapidly as the number m increases (see the discussion of truncation errors in Sect. 21.7).

Once a fixed point has been reached, ground state properties can be calculated. For instance, a ground state expectation value $\bar{\mathcal{O}} = \langle\psi|\mathcal{O}|\psi\rangle$ is obtained in two

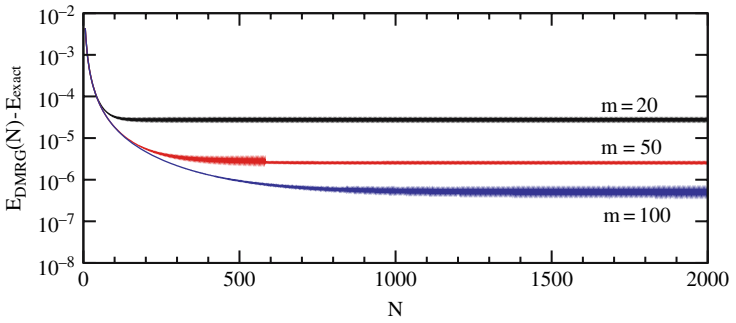

Fig. 21.2. Convergence of the ground state energy per site calculated with the infinite-system DMRG algorithm in a spin-$\frac{1}{2}$ Heisenberg chain as a function of the superblock size N for three different numbers m of density-matrix eigenstates kept

steps: First, one calculates $|\psi'\rangle = \mathcal{O}|\psi\rangle$ using (21.29), then the expectation value is computed as a scalar product $\bar{\mathcal{O}} = \langle\psi|\psi'\rangle$. Explicitly,

$$\langle\psi|\psi'\rangle = \sum_{\alpha=1}^{a_j} \sum_{s_{j+1}=1}^{d_{j+1}} \sum_{s_{j+2}=1}^{d_{j+2}} \sum_{\beta=1}^{b_{j+3}} ([\mathsf{C}_{j+1}](\alpha, s_{j+1}, s_{j+2}, \beta))^* \, [\mathsf{C}'_{j+1}](\alpha, s_{j+1}, s_{j+2}, \beta) \; . \tag{21.32}$$

Experience shows that the infinite-system DMRG algorithm yields accurate results for local physical quantities such as spin density and short-range spin correlations in quantum lattice systems with "good" features (infinite homogeneous one-dimensional systems with short-range interactions such as the Heisenberg model on an open chain). These local quantities are calculated using operators $\mathcal{O}$ acting only on sites around the middle of the last (i.e., longest) lattice (more precisely, in an interval of length $N - N'$ around the center of a N-site lattice if the fixed point has been reached after $N'/2 < N/2$ iterations). For other types of systems and other physical quantities the infinite-system algorithm fails in most cases. The reasons for these failures are the same as for NRG. First, the exact ground state may not be represented accurately by a matrix-product state (21.2) with restricted matrix sizes. For instance, the matrix-product state cannot reproduce long-range power-law correlations, see Chap. 20. Second, very often the infinite-system DMRG algorithm does not generate the optimal block representation for the ground state (i.e., does not find the best possible matrix-product state (21.2) for preset matrix sizes) when the system does not have the good features mentioned above.

21.5 Finite-System DMRG Algorithm

The finite-system method is a more versatile DMRG algorithm than the infinite-system method as it can be applied to almost any quantum lattice problem. It is also more reliable as it always finds the best possible matrix-product representation (21.2) for a given quantum state. In the finite-system DMRG method the lattice size N is kept constant. The superblock structure is $\{L(j) + R(j+1)\}$, where j is varied iteratively by one site from $N-2$ to 2 in a sweep from right to left and from 2 to $N-2$ in a sweep from left to right, see Fig. 21.1. If the system has the reflection symmetry used in the infinite-system algorithm, j need to be varied from $N/2$ to 2 and back only. At the start of the finite-system algorithm, one calculates effective representations for the left blocks $L(1)$ to $L(N-3)$ using the NRG method, the infinite-system DMRG algorithm, or other methods, even using random transformations in (21.6), as they can be poor approximations of the optimal representations. This initial calculation is called the warmup sweep.

We first proceed with a sweep through the lattice from right to left, reducing j by one at every iteration starting from $j = N-2$. For this purpose, we have to compute an effective representation of dimension b_{j+1} for $R(j+1)$ using the

effective representation of dimension b_{j+2} for the right block $R(j+2)$ calculated in the previous iteration. For the first iteration $j = N-2$, the exact representation of $R(N)$ is used. As done for left blocks in the NRG and infinite-system DMRG algorithm, we first define a tensor-product basis of dimension $d_{j+1}b_{j+2}$ for the new right block using the site basis $\mathcal{B}(j+1)$ and the subspace basis $\mathcal{B}(R, j+2)$ of $R(j+2)$

$$\left| s_{j+1}\, \phi_\beta^{R(j+2)} \right\rangle = |s_{j+1}\rangle \otimes \left| \phi_\beta^{R(j+2)} \right\rangle \ , \tag{21.33}$$

for $s_{j+1} = 1, \ldots, d_{j+1}$ and $\beta = 1, \ldots, b_{j+2}$. The matrix representation (21.22) of any operator $\mathcal{O}$ acting in $R(j+1)$ can be calculated similarly to (21.15)

$$\mathcal{O}(s_{j+1}, \beta, s'_{j+1}, \beta') = \sum_k \mathcal{O}^{S,k}(s_{j+1}, s'_{j+1})\, \mathcal{O}^{R,k}(\beta, \beta') \ , \tag{21.34}$$

where the $\mathcal{O}^{S,k}(s_{j+1}, s'_{j+1})$ are site-operator matrices (21.17) and $\mathcal{O}^{R,k}(\beta, \beta')$ denotes the known matrix representations of operators acting on sites of $R(j+2)$ in the basis $\mathcal{B}(R, j+2)$. Thus we obtain an effective representation of dimension $d_{j+1}b_{j+2}$ for $R(j+1)$. Next, we use the available effective representation of dimension a_{j-1} for the left block $L(j-1)$, which has been obtained during the previous sweep from left to right (or the result of the warmup sweep if this is the first sweep from right to left). With this block $L(j-1)$ we build an effective representation of dimension $a_{j-1}d_j$ for $L(j)$ using a tensor-product basis (21.10) as done in the NRG and infinite-system DMRG methods.

Now we consider the superblock $\{L(j) + R(j+1)\}$ and its tensor-product basis analogue to (21.23) and set up the representation of operators in this basis, especially the Hamiltonian, similarly to (21.27). As for the infinite-system algorithm we determine the ground state C_j of the superblock Hamiltonian in the superblock basis using the Lanczos or Davidson algorithm and the efficient implementation of the matrix-vector product (21.29). Typically, we have already obtained a representation of the ground state C_{j+1} for the superblock configuration $\{L(j+1) + R(j+2)\}$ in the previous iteration. This state can be transformed exactly in the superblock basis for $\{L(j) + R(j+1)\}$ using

$$[\mathsf{C}_j^{\mathrm{G}}](\alpha, s_j, s_{j+1}, \beta) = \sum_{\alpha'=1}^{a_j} \phi_{\alpha'}^{L(j)}(\alpha, s_j) \sum_{s_{j+2}=1}^{d_{j+2}} \sum_{\beta'=1}^{b_{j+3}} [\mathsf{C}_{j+1}](\alpha', s_{j+1}, s_{j+2}, \beta') \left(\phi_\beta^{R(j+2)}(s_{j+2}, \beta') \right)^* \ , \tag{21.35}$$

for $\alpha = 1, \ldots, a_{j-1}$; $s_j = 1, \ldots, d_j$; $s_{j+1} = 1, \ldots, d_{j+1}$; and $\beta = 1, \ldots, b_{j+2}$. The functions $\phi_\beta^{R(j+2)}(s_{j+2}, \beta')$ are the density-matrix eigenvectors of $R(j+2)$ calculated in the previous iteration while the functions $\phi_{\alpha'}^{L(j)}(\alpha, s_j)$ are the density-matrix eigenvectors of $L(j)$ calculated during the previous sweep from left to right (or during the warmup sweep if this is the first sweep from right to left). The state

$\mathsf{C}_j^{\mathrm{G}}$ can be used as the initial vector for the iterative diagonalization routine. When the finite-system DMRG algorithm has already partially converged, this initial state $\mathsf{C}_j^{\mathrm{G}}$ is a good guess for the exact ground state C_j of the superblock Hamiltonian in the configuration $\{L(j) + R(j+1)\}$ and thus the iterative diagonalization method converges in a few steps. This can result in a speed up of one or two orders of magnitude compared to a diagonalization using a random initial vector $\mathsf{C}_j^{\mathrm{G}}$.

Once the superblock representation C_j of the targeted ground state has been obtained, we calculate the reduced density matrix for the right block $R(j+1)$

$$\rho(s_{j+1}, \beta, s'_{j+1}, \beta') = \sum_{s_j=1}^{d_j} \sum_{\alpha=1}^{a_{j-1}} \left([\mathsf{C}_j](\alpha, s_j, s_{j+1}, \beta)\right)^* [\mathsf{C}_j](\alpha, s_j, s'_{j+1}, \beta') \ , \quad (21.36)$$

for $\beta, \beta' = 1, \ldots, b_{j+2}$ and $s_{j+1}, s'_{j+1} = 1, \ldots, d_{j+1}$. We denote the eigenvectors of this density matrix $\phi_\mu^{R(j+1)}(s_{j+1}, \beta)$ with $\mu = 1, \ldots, d_{j+1} b_{j+2}$. The b_{j+1} eigenvectors with the largest eigenvalues are chosen to define a new basis $\mathcal{B}(R, j+1)$ of $R(j+1)$ through (21.7) and the other eigenvectors are discarded. As already mentioned in the previous section, this is the optimal choice for preserving C_j while reducing the basis dimension from $d_{j+1} b_{j+2}$ to b_{j+1} (see Chap. 20 and Sect. 21.7 for more detail). The matrix representation of any operator acting only on sites in $R(j+1)$ can be calculated in the new basis with two successive matrix products

$$M(s_{j+1}, \beta, \mu') = \sum_{\beta'=1}^{b_{j+2}} \sum_{s'_{j+1}=1}^{d_{j+1}} \mathcal{O}(s_{j+1}, \beta, s'_{j+1}, \beta') \, \phi_{\mu'}^{R(j+1)}(s'_{j+1}, \beta') \ ,$$

$$\mathcal{O}(\mu, \mu') = \sum_{\beta=1}^{b_{j+2}} \sum_{s_{j+1}=1}^{d_{j+1}} \left(\phi_\mu^{R(j+1)}(s_{j+1}, \beta)\right)^* M(s_{j+1}, \beta, \mu') \ , \quad (21.37)$$

for $\mu, \mu' = 1, \ldots, b_{j+1}$ as done for a right block in (21.19).

Thus we have obtained an effective representation of dimension b_{j+1} for the right block $R(j+1)$. We note that the block states (21.7) are not explicitly calculated. Only matrix and vector representations of operators and states in that basis and the transformation from a basis to the next one need to be calculated explicitly. The procedure is repeated in the next iteration to obtain an effective representation for the next larger right block. Iterations are continued until the sweep from right to left is completed ($j + 1 = 3$). This right-to-left sweep is illustrated in the right picture of Fig. 21.1 going from bottom to top.

Then we exchange the roles of the left and right blocks and perform a sweep from left to right. Effective representations for the left blocks $L(j), j = 2, \ldots, N-3$, are built iteratively. The effective representation for $R(j+3)$ which has been calculated during the last right-to-left sweep is used to make an effective tensor-product-basis representation of $R(j+2)$ and thus to complete the superblock $\{L(j+1) + R(j+2)\}$. This left-to-right sweep corresponds to the right picture of Fig. 21.1 going from top to bottom.

When this left-to-right sweep is done, one can start a new couple of sweeps back and forth. The ground state energy calculated with the superblock Hamiltonian decreases progressively as the sweeps are performed. This results from the progressive optimization of the matrix-product state (21.2) for the ground state. Figure 21.3 illustrates this procedure for the total energy of a 400-site Heisenberg chain. The matrix dimensions a_n, b_n are chosen to be not greater than $m = 20$ (maximal number of density-matrix eigenstates kept at each iteration). The sweeps are repeated until the procedure converges (i.e., the ground state energy converges). In Fig. 21.3 the DMRG energy converges to a value $E_{\mathrm{DMRG}}(m = 20)$ which lies about 0.008 above the exact result for the 400-site Heisenberg chain. As it corresponds to a variational wavefunction (21.2) the DMRG energy $E_{\mathrm{DMRG}}(m)$ always lies above the exact ground state energy and decreases as m increases.

Once convergence is achieved, ground state properties can be calculated with (21.29) and (21.32) as explained in the previous section. Contrary to the infinite-system algorithm, however, the finite-system algorithm yields consistent results for the expectation values of operators acting on any lattice site. For example, we show in Fig. 21.4 the staggered spin bond order $(-1)^n(\langle \boldsymbol{S}_n \boldsymbol{S}_{n+1}\rangle + \ln(2) - 1/4)$ and the staggered spin-spin correlation function $C(r) = (-1)^r \langle \boldsymbol{S}_n \boldsymbol{S}_{n+r}\rangle$ obtained in the 400-site Heisenberg chain using up to $m = 200$ density-matrix eigenstates. A strong staggered spin bond order is observed close to the chain edges (Friedel oscillations) while a smaller one is still visible in the middle of the chain because of its finite size. For a distance up to $r \approx 100$ the staggered spin-spin correlation function $C(r)$ decreases approximately as a power-law $1/r$ as expected but a deviation from this behavior occurs for larger r because of the chain edges. Finite-size

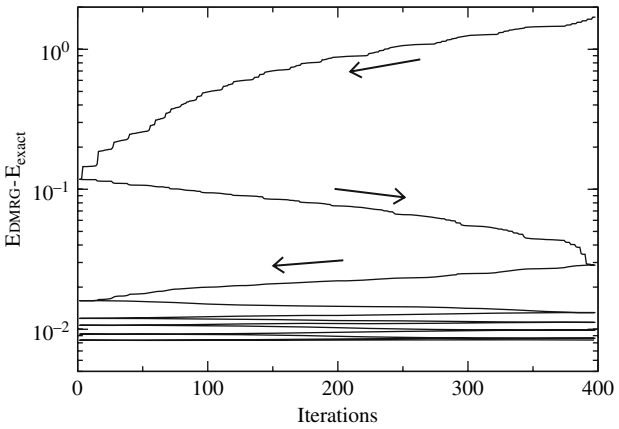

Fig. 21.3. Convergence of the ground state energy calculated with the finite-system DMRG algorithm using $m = 20$ density-matrix eigenstates as a function of the iterations in a 400-site spin-$\frac{1}{2}$ Heisenberg chain. Arrows show the sweep direction for the first three sweeps starting from the top

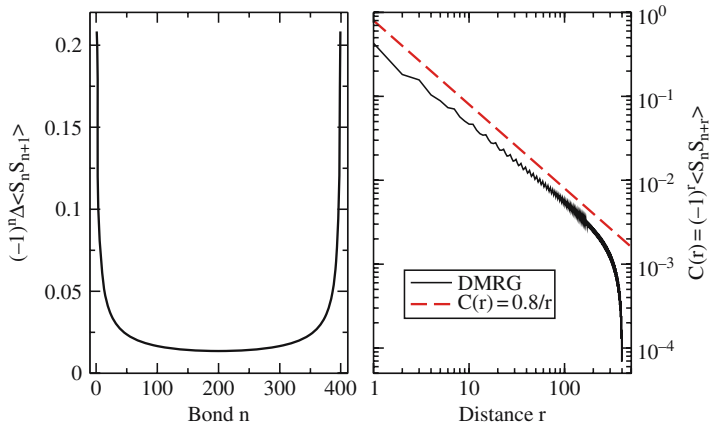

Fig. 21.4. Staggered spin bond order (**left**) $(-1)^n(\langle \boldsymbol{S}_n \boldsymbol{S}_{n+1}\rangle - \frac{1}{4} + \ln(2))$ and staggered spin-spin correlation function (**right**) $C(r) = (-1)^r \langle \boldsymbol{S}_n \boldsymbol{S}_{n+r}\rangle$. Both quantities have been calculated using the finite-system DMRG algorithm with $m = 200$ in a 400-site spin-$\frac{1}{2}$ Heisenberg chain. The *dashed line* is a guide for the eye

and chain-end effects are unavoidable and sometimes troublesome features of the finite-size DMRG method.

Contrary to the infinite-system algorithm the finite-system algorithm always finds the optimal matrix-product state (21.2) with restricted matrix sizes. Nevertheless, experience shows that the accuracy of DMRG calculations depends significantly on the system investigated because the matrix-product state (21.2) with restricted matrix sizes can be a good or a poor approximation of the true ground state. In practice, this implies that physical quantities calculated with DMRG can approach the exact results rapidly or slowly for an increasing number m of density-matrix eigenstates kept. This so-called truncation error is discussed in Sect. 21.7. For instance, the finite-system DMRG method yields excellent results for gapped one-dimensional systems but is less accurate for critical systems or in higher dimensions for the reason discussed in Chap. 20.

21.6 Additive Quantum Numbers

Symmetries and quantum numbers play an important role in the solution and analysis of quantum many-body systems. Here I discuss additive quantum numbers, which constitute the simplest implementation of quantum numbers within the basic DMRG methods. A quantum number is additive when the quantum number of the tensor product of two states is given by the sum of the quantum numbers of both states. The use of other symmetries and quantum numbers is described in [7, 11].

We consider an operator Q acting in $\mathcal{H}$ which is the sum of Hermitian site operators, $Q = \sum_{n=1}^{N} Q_n$, where the operator Q_n acts only on the site n. A typical

example is the z-component of the total spin $S^z = \sum_{n=1}^{N} S_n^z$ in a spin system such as the Heisenberg model. If Q commutes with the system Hamiltonian H, eigenstates of H can be chosen so that they are also eigenstates of Q. If the target of the DMRG calculation is an eigenstate of Q (for instance, the ground state of H), one can show that the reduced density operators for left and right blocks commute with the operators

$$Q^{L(j)} = \sum_{n=1}^{j} Q_n \text{ and } Q^{R(j+1)} = \sum_{n=j+1}^{N} Q_n \,, \tag{21.38}$$

respectively. As a consequence, the density-operator eigenstates (21.6) and (21.7) can be chosen to be eigenstates of $Q^{L(j)}$ or $Q^{R(j+1)}$ and the block basis states can be labeled with an index identifying their quantum number (the corresponding eigenvalue of $Q^{L(j)}$ or $Q^{R(j+1)}$). For instance, the left block basis becomes

$$\mathcal{B}(L,j) = \left\{ \left| \phi_{r,\alpha}^{L(j)} \right\rangle ;\ r = 1,2,\ldots;\alpha = 1,\ldots,a_{r,j} \right\} \,, \tag{21.39}$$

where the index r numbers the possible quantum numbers $q_r^{L(j)}$ of $Q^{L(j)}$, α numbers $a_{r,j}$ basis states with the same quantum number, and $\sum_r a_{r,j} = a_j$.

We note that $Q^{L(j+1)} = Q^{L(j)} + Q_{j+1}$. Thus if we choose the site basis states in $\mathcal{B}(j+1)$ to be eigenstates of the site operator Q_{j+1} and denote $|t, s_{j+1}\rangle$ a basis state with quantum number $q_t^{S(j+1)}$, the tensor product state (21.10) becomes

$$\left| \phi_{r,\alpha}^{L(j)}; t, s_{j+1} \right\rangle = \left| \phi_{r,\alpha}^{L(j)} \right\rangle \otimes |t, s_{j+1}\rangle \,, \tag{21.40}$$

and its quantum number (eigenvalue of $Q^{L(j+1)}$) is given by $q_p^{L(j+1)} = q_r^{L(j)} + q_t^{S(j+1)}$. Therefore, the corresponding density-matrix eigenstates take the form $\phi_{p,\alpha}^{L(j+1)}(r,\alpha',t,s_{j+1})$ and vanish if $q_p^{L(j+1)} \neq q_r^{L(j)} + q_t^{S(j+1)}$, see (21.6). Similarly, the density-matrix eigenstates for a right block are noted $\phi_{p,\beta}^{R(j+1)}(t,s_{j+1},r,\beta')$ and vanish if $q_p^{R(j+1)} \neq q_r^{R(j+2)} + q_t^{S(j+1)}$. We can save computer time and memory if we use this rule to compute and store only the terms which do not identically vanish.

Furthermore, as $Q = Q^{L(j)} + Q_{j+1} + Q_{j+2} + Q^{R(j+3)}$, a superblock basis state (21.23) can be written

$$\left| \phi_{p,\alpha}^{L(j)}; r, s_{j+1}; t, s_{j+2}; \phi_{v,\beta}^{R(j+3)} \right\rangle = \left| \phi_{p,\alpha}^{L(j)}; r, s_{j+1} \right\rangle \otimes \left| t, s_{j+2}; \phi_{v,\beta}^{R(j+3)} \right\rangle \,, \tag{21.41}$$

and its quantum number (eigenvalue of Q) is given by $q = q_p^{L(j)} + q_r^{S(j+1)} + q_t^{S(j+2)} + q_v^{R(j+3)}$. Therefore, the superblock representation (21.28) of a state $|\psi\rangle$ with a quantum number q can be written $[\mathsf{C}_{j+1}](p,\alpha,r,s_{j+1},t,s_{j+2},v,\beta)$ and vanishes if $q \neq q_p^{L(j)} + q_r^{S(j+1)} + q_t^{S(j+2)} + q_v^{R(j+2)}$. Here again we can save computer time and memory if we use this rule to compute and store only the components of C_{j+1} which do not identically vanish.

If an operator $\mathcal{O}$ has a simple commutation relation with Q of the form $[Q, \mathcal{O}] = \Delta q\ \mathcal{O}$, where Δq is a number, the matrix elements $\langle\mu|\mathcal{O}|\nu\rangle$ of $\mathcal{O}$ in the eigenbasis of Q vanish but for special combinations of the eigenstates $|\mu\rangle$ and $|\nu\rangle$. Similar rules apply for the related operators $Q^{L(n)}$, $Q^{R(n)}$, and Q_n. Explicitly, for the matrices (21.17) of site operators one finds that $\langle p, s_n|\mathcal{O}|p', s'_n\rangle = 0$ for $q_p^{S(n)} \neq q_{p'}^{S(n)} + \Delta q$ if $[Q_n, \mathcal{O}] = \Delta q\ \mathcal{O}$. For instance, for the spin operator S_n^+ with $[S^z, S_n^+] = \hbar S_n^+$ only $\langle\uparrow\ |S_n^+|\ \downarrow\rangle$ does not vanish. For the matrix representation of left block operators (21.14) one finds that

$$\left\langle \phi_{p\alpha}^{L(j)}; r, s_{j+1} \middle| \mathcal{O} \middle| \phi_{p'\alpha'}^{L(j)}; r', s'_{j+1} \right\rangle = 0\ , \tag{21.42}$$

for $q_p^{L(j)} + q_r^{S(j+1)} \neq q_{p'}^{L(j)} + q_{r'}^{S(j+1)} + \Delta q$ if $[Q^{L(j+1)}, \mathcal{O}] = \Delta q\ \mathcal{O}$. A similar rule, applies to matrix representations (21.22) in a right block

$$\left\langle t, s_{j+2}; \phi_{v\beta}^{R(j+3)} \middle| \mathcal{O} \middle| t', s'_{j+2}; \phi_{v'\beta'}^{R(j+3)} \right\rangle = 0\ , \tag{21.43}$$

for $q_v^{R(j+3)} + q_t^{S(j+2)} \neq q_{v'}^{R(j+3)} + q_{t'}^{S(j+2)} + \Delta q$ if $[Q^{R(j+2)}, \mathcal{O}] = \Delta q\ \mathcal{O}$. Therefore, we can reduce the computer time and memory used if we compute and save only the matrix elements which are not identically zero because of the conservation of additive quantum numbers. Moreover, if we implement these rules, the computational cost of the operations (21.15), (21.19), (21.29), (21.30), (21.32), and (21.34) to (21.37) is also substantially reduced.

In summary, using additive quantum numbers increases the complexity of a DMRG program but can reduce the computational effort significantly. In Chap. 22 it is shown that quantum numbers and symmetries can also be used with DMRG to investigate additional properties such as excited states.

21.7 Truncation Errors

There are three main sources of numerical errors in the finite-system DMRG method:

- The iterative diagonalization algorithm used to find the ground state of the superblock Hamiltonian (diagonalization error),
- the iterative optimization of matrices in the matrix-product state (21.2) (convergence error), and
- the restrictions put on the matrix dimensions a_n and b_n (truncation error).

Diagonalization errors originate from errors in the calculation of the matrix C_j in (21.2) but they propagate to the other matrices through the density-matrix based selection of the block basis states. These errors can always be made negligible compared to the other two error sources in ground state calculations. However, as the superblock diagonalization is the most time-consuming task and the other two error sources limit the overall accuracy anyway, one should not determine the superblock

ground state with too much precision but strike a balance between accuracy and computational cost. In DMRG algorithms that target other states than the ground state (for instance, dynamical correlation functions, see Chap. 22), the diagonalization error may become relevant.

Convergence errors corresponds to non-optimal matrices $\mathsf{A}_n(s_n)$ and $\mathsf{B}_n(s_n)$ in the matrix-product state (21.2). They are negligible in DMRG calculations for ground state properties in non-critical one-dimensional open systems with nearest-neighbor interactions. For such cases DMRG converges after very few sweeps through the lattice. Convergence problems occur frequently in critical or inhomogeneous systems and in systems with long-range interactions (this effectively includes all systems in dimension larger than one, see the last section). However, if one performs enough sweeps through the lattice (up to several tens in hard cases), these errors can always be made smaller than truncation errors (i.e., the finite-system DMRG algorithm always finds the optimal matrices for a matrix-product state (21.2) with restricted matrix sizes).

Truncation errors are usually the dominant source of inaccuracy in the finite-system DMRG method. They can be systematically reduced by increasing the matrix dimensions a_n, b_n used in (21.2). In actual computations, however, they can be significant and it is important to estimate them reliably. In the finite-system DMRG algorithm a truncation error is introduced at every iteration when a tensor-product basis of dimension $a_j d_{j+1}$ for the left block $L(j+1)$ is reduced to a basis of dimension a_{j+1} during a sweep from left to right and, similarly, when a tensor-product basis of dimension $b_{j+2}d_{j+1}$ for the right block $R(j+1)$ is reduced to a basis of dimension b_{j+1} during a sweep from right to left. Each state $|\psi\rangle$ which is defined using the original tensor-product basis (usually, the superblock ground state) is replaced by an approximate state $|\widetilde{\psi}\rangle$ which is defined using the truncated basis. It has been shown [1] that the optimal choice for constructing a smaller block basis for a given target state $|\psi\rangle$ consists in choosing the eigenvectors with the highest eigenvalues w_μ from the reduced density-matrix (21.30) or (21.36) of $|\psi\rangle$ for this block. More precisely, this choice minimizes the differences $= \left||\psi\rangle - |\widetilde{\psi}\rangle\right|^2$ between the target state $|\psi\rangle$ and its approximation $|\widetilde{\psi}\rangle$.

The minimum of S is given by the weight P of the discarded density-matrix eigenstates. With $w_1 \geq w_2 \geq \cdots \geq w_{a_j d_{j+1}}$ we can write

$$S_{\min} = P(a_{j+1}) = \sum_{\mu=1+a_{j+1}}^{a_j d_{j+1}} w_\mu = 1 - \sum_{\mu=1}^{a_{j+1}} w_\mu \tag{21.44}$$

for the left block $L(j+1)$ and similarly $S_{\min} = P(b_{j+1}) = 1 - \sum_{\mu=1}^{b_{j+1}} w_\mu$ for the right block $R(j+1)$. It can be shown that errors in physical quantities depend directly on the discarded weight. For the ground-state energy the truncation introduces an error

$$\frac{\langle\widetilde{\psi}|H|\widetilde{\psi}\rangle}{\langle\widetilde{\psi}|\widetilde{\psi}\rangle} - \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \propto P(a_{j+1}) \text{ or } P(b_{j+1}) , \tag{21.45}$$

while for other expectation values the truncation error is

$$\frac{\langle\widetilde{\psi}|\mathcal{O}|\widetilde{\psi}\rangle}{\langle\widetilde{\psi}|\widetilde{\psi}\rangle} - \frac{\langle\psi|\mathcal{O}|\psi\rangle}{\langle\psi|\psi\rangle} \propto \sqrt{P(a_{j+1})} \text{ or } \sqrt{P(b_{j+1})} \tag{21.46}$$

for $P(a_{j+1}), P(b_{j+1}) \ll 1$, respectively. Therefore, truncation errors for physical quantities are small when the discarded weight is small. Clearly, the discarded weight is small when the eigenvalues w_μ of the reduced density-matrices (21.30) and (21.36) decrease rapidly with increasing index μ. As discussed in Chap. 20, there are various quantum systems for which the spectrum of reduced density-matrices for subsystems has this favorable property. For such quantum systems the matrix-product state (21.2) is a good approximation and DMRG truncation errors decrease rapidly with increasing matrix sizes a_{j+1} and b_{j+1}.

In practice, there are two established methods for choosing the matrix dimensions in a systematic way in order to cope with truncation errors. First, we can perform DMRG sweeps with matrix dimensions not greater than a fixed number m of density-matrix eigenstates kept, $a_n, b_n \lesssim m$. In that approach, physical quantities [ground state energy $E_{\mathrm{DMRG}}(m)$ and other expectation values $\bar{\mathcal{O}}_{\mathrm{DMRG}}(m)$] are calculated for several values of m and their scaling with increasing m is analyzed. Usually, one finds a convergence to a fixed value with corrections that decreases monotonically with m. This decrease is exponential in favorable cases (gapped one-dimensional systems with short-range interactions) but can be as slow as m^{-2} for systems with non-local Hamiltonians. As an example, we show in Fig. 21.5 the truncation error in the ground state energy for a 100-site Heisenberg chain. For open boundary conditions (a favorable case for a matrix-product state (21.2) and thus for DMRG) the error decreases very rapidly with m until it reaches the order of

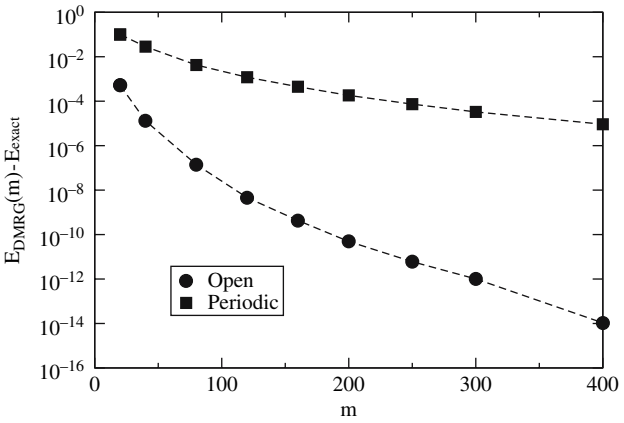

Fig. 21.5. Error in the ground state energy calculated with the finite-system DMRG algorithm as a function of the number m of density-matrix eigenstates kept for the spin-$\frac{1}{2}$ Heisenberg Hamiltonian on a one-dimensional 100-site lattice with open (*circles*) and periodic (*squares*) boundary conditions

magnitude of round-off errors in the computer system used. For periodic boundary conditions (a less favorable case) the error decreases slowly with m and is still significant for the largest number of density-matrix eigenstates considered $m = 400$.

In the second approach the density-matrix eigenbasis is truncated so that the discarded weight is approximately constant, $P(a_{j+1}), P(b_{j+1}) \lesssim P$, and thus a variable number of density-matrix eigenstates is kept at every iteration. The physical quantities obtained with this procedure depends on the chosen discarded weight P. Empirically, one finds that the relations (21.45) and (21.46) hold for DMRG results calculated with various P. For the energy one has $E_{\mathrm{DMRG}}(P) \approx E(P = 0) + cP$ and for other expectation values $\bar{\mathcal{O}}_{\mathrm{DMRG}}(P) \approx \bar{\mathcal{O}}(P = 0) + c'\sqrt{P}$ if P is small enough. Therefore, we can carry out DMRG calculations for several values of the discarded weight P and obtain results $E(P = 0)$ and $\bar{\mathcal{O}}(P = 0)$ in the limit of vanishing discarded weight $P \to 0$ using an extrapolation. In practice, this procedure yields reliable estimations of the truncation errors and often the extrapolated results are more accurate than those obtained directly with DMRG for the smallest value of P used in the extrapolation.

It should be noted that if one works with a fixed number m of density-matrix eigenstates kept, it is possible to calculate an average discarded weight $P(m) = \sum_j P(b_{j+1})$ over a sweep. In many cases, the physical quantities $E_{\mathrm{DMRG}}(m)$ and $\bar{\mathcal{O}}_{\mathrm{DMRG}}(m)$ scale with $P(m)$ as in (21.45) and (21.46), respectively. Therefore, an extrapolation to the limit of vanishing discarded weight $P(m) \to 0$ is also possible (see [12] for some examples).

21.8 Computational Cost and Optimization

Theoretically, the computational cost for one calculation with the infinite-system algorithm or for one sweep of the finite-system algorithm is proportional to $Nn_k m^3 d^3$ for the number of operations and to $Nn_k m^2 d^2$ for the memory if one assumes that about m density-matrix eigenstates are kept at every iteration and the site Hilbert space dimension is d. In practice, various optimization techniques such as the additive quantum numbers of Sect. 21.6 lead to a more favorable scaling with m. The actual computational effort varies greatly with the model investigated and the physical quantities which are computed. Using a highly optimized code the infinite-system DMRG simulations shown in Fig. 21.2 take from 30 seconds for $m = 20$ to 7 minutes for $m = 100$ on 3 GHz Pentium 4 processor while the finite-system DMRG calculations shown in Figs. 21.3 and 21.4 take about 20 minutes. These calculations use less than to 300 MBytes of memory. For more difficult problems with $m \lesssim 10^4$, the computational cost can reach thousands of CPU hours and hundreds of GBytes of memory.

Even for less challenging problems, it is useful to consider some basic optimization issues for a DMRG code. First of all, an efficient dynamical memory management should be used because many vectors, matrices, and tensors of higher ranks with variable sizes have too be stored temporarily. Even for the simplest applications this amounts to the allocation and release of GBytes of memory during a DMRG

simulation. Second, processor-optimized linear algebra routines should be used because most of the CPU time is spent for linear algebra operations such as products of matrices. Generic BLAS routines [13] can be as much as two orders of magnitude slower than processor-optimized ones. Third, the most expensive part of a DMRG iteration is usually the calculation of the superblock representation (21.28) of the target state, typically the ground state of the superblock Hamiltonian. Thus one should use the procedures (21.29) and (21.35) and the efficient iterative algorithms described in Chap. 18 to perform this task. Finally, one should also consider using parallelization and optimization techniques for high-performance computers (see Chap. 27). Unfortunately, the basic DMRG algorithms presented in Sects. 21.4 and 21.5 are inherently sequential and a parallelization is possible only at a low level. For instance, the superblock product (21.29) or, at an even lower level, matrix products (i.e., the BLAS routines) can be parallelized. The parallelization of a DMRG code is discussed in more detail in [14].

21.9 Basic Extensions

The finite-system DMRG algorithm described in Sect. 21.5 can readily be applied to one-dimensional quantum spin chains. It can also be used to study fermions, bosons, and systems in higher dimensions without much difficulty.

To apply the DMRG algorithm to fermion systems we just have to take into account the fermion commutation sign in the operator decomposition (21.11) and (21.24) and in the tensor product of their matrix representations (21.15), (21.27), and (21.34). Using the total number of fermion as an additive quantum number is very helpful for that purpose. For instance, if $|\alpha\rangle, |\alpha'\rangle, |\beta\rangle, |\beta'\rangle$ denote states with a fixed number of fermions and $\mathcal{O}_1$,$\mathcal{O}_2$ are operators, the matrix element of the tensor-product operator $\mathcal{O}_1 \otimes \mathcal{O}_2$ for the tensor-product states $|\alpha\beta\rangle = |\alpha\rangle \otimes |\beta\rangle$ and $|\alpha'\beta'\rangle = |\alpha'\rangle \otimes |\beta'\rangle$ is

$$\langle\alpha\beta\,|\mathcal{O}_1 \otimes \mathcal{O}_2|\,\alpha'\beta'\rangle = (-1)^{q|\Delta q|}\,\langle\alpha\,|\mathcal{O}_1|\,\alpha'\rangle\,\langle\beta\,|\mathcal{O}_2|\,\beta'\rangle\ , \tag{21.47}$$

where q is the number of fermions in the state $|\alpha'\rangle$ and Δq is the difference between the number of fermions in the states $|\beta\rangle$ and $|\beta'\rangle$.

To apply the DMRG method to boson systems such as electron-phonon models, we must first choose an appropriate finite basis for each boson site to represent the infinite Hilbert space of a boson as best as possible, which is done also in exact diagonalization methods [12]. Then the finite-system DMRG algorithm can be used without modification. However, the computational cost scales as d^3 for the CPU time and as d^2 for the memory if d states are used to represent each boson site. Typically, $d = 10-100$ is required for accurate computations in electron-phonon models. Therefore, simulating boson systems with the standard DMRG algorithms is significantly more demanding than spin systems. More sophisticated DMRG algorithms have been developed to reduce the computational effort involved in solving boson systems. The best algorithms scale as d or $d\ln(d)$ and are presented in [12].

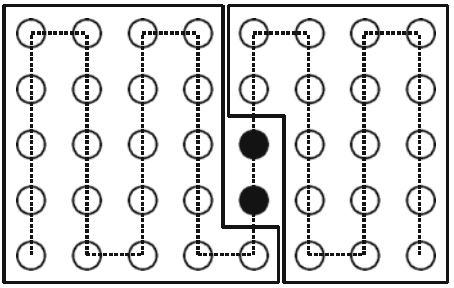

Fig. 21.6. Schematic representations of the site sequence (*dashed line*) in a two-dimensional lattice. The site in the bottom left corner is site 1. The superblock structure $\{L(21)+$ site 22 + site 23 $+R(24)\}$ is shown with *solid lines* delimiting the left and right blocks and full circles indicating both sites

The finite-system DMRG method can be applied to quantum systems with various degrees of freedom, on lattices in dimension larger than one, and to the non-local Hamiltonians considered in quantum chemistry and momentum space, see Chap. 24. We just have to order the lattice sites from 1 to N in some way to be able to carry out the algorithm described in Sect. 21.5. For instance, Fig. 21.6 shows one possible site sequence for a two-dimensional cluster. It should be noted that sites which are close in the two-dimensional lattice are relatively far apart in the sequence. This corresponds to an effective long-range interaction between the sites even if the two-dimensional system includes short-range interactions only, and results in a slower convergence and larger truncation errors than in truly one-dimensional systems with short-range interactions. As a consequence, reordering of the lattice sites can significantly modify the accuracy of a DMRG calculation and various site sequences should be considered for those systems which do not have a natural order. The difficulty with DMRG simulations and more generally with matrix-product states in dimensions larger than one is discussed fully in Chap. 20.

References

1. I. Peschel, X. Wang, M. Kaulke, K. Hallberg (eds.), *Density-Matrix Renormalization, A New Numerical Method in Physics* (Springer, Berlin, 1999) 597, 614
2. R.M. Noack, S.R. Manmana, in *Lectures on the Physics of Highly Correlated Electron Systems IX: Ninth Training Course in the Physics of Correlated Electron Systems and High-Tc Superconductors*, *AIP Conf. Proc.*, Vol. 789, ed. by A. Avella, F. Mancini (AIP, 2005), pp. 93–163 597
3. S.R. White, Phys. Rev. Lett. **69**(19), 2863 (1992) 597, 603
4. S.R. White, Phys. Rev. B **48**(14), 10345 (1993) 597, 603
5. U. Schollwock, Rev. Mod. Phys. **77**(1), 259 (2005) 597
6. K. Hallberg, Adv. Phys. **55**, 477 (2006) 597
7. I.P. McCulloch (2007). URL `http://arxiv.org/abs/cond-mat/0701428`. Preprint 597, 611

8. K.G. Wilson, Rev. Mod. Phys. **47**(4), 773 (1975) 600
9. S.R. White, R.M. Noack, Phys. Rev. Lett. **68**(24), 3487 (1992) 603
10. L. Hulthén, Arkiv Mat. Astr. Fysik **26A**(11), 1 (1938) 606
11. I.P. McCulloch, M. Gulàcsi, Europhys. Lett. **57**(6), 852 (2002) 611
12. E. Jeckelmann, H. Fehske, in *Proceedings of the International School of Physics "Enrico Fermi" - Course CLXI Polarons in Bulk Materials and Systems with Reduced Dimensionality* (IOS Press, Amsterdam, 2006), pp. 247–284 616, 617
13. Basic Linear Algebra Subprograms (BLAS). URL `http://www.netlib.org/blas/` 617
14. G. Hager, E. Jeckelmann, H. Fehske, G. Wellein, J. Comput. Phys. **194**, 795 (2004) 617