

CZECH TECHNICAL UNIVERSITY IN PRAGUE  
FACULTY OF NUCLEAR SCIENCES AND PHYSICAL ENGINEERING  
DEPARTMENT OF PHYSICS

## RESEARCH WORK

# Bose Gases in One-Dimensional Optical Lattices

Author: Miroslav Urbanek  
Advisor: Ing. Pavel Soldán, Dr.  
Year: 2005/2006

# Contents

<b>Introduction</b>	<b>3</b>
<b>1 Bose gases in optical lattices</b>	<b>4</b>
1.1 Optical traps . . . . .	4
1.2 Optical lattices . . . . .	6
1.3 The Bose-Hubbard model . . . . .	7
1.4 Phase transitions in the Bose-Hubbard model . . . . .	8
<b>2 Density matrix renormalization group algorithm</b>	<b>10</b>
2.1 Blocks renormalization group algorithm . . . . .	11
2.2 Density matrix renormalization group algorithm . . . . .	11
<b>3 The DMRG algorithm applied to the single-particle quantum mechanics</b>	<b>15</b>
3.1 Relation to the many-body problem . . . . .	15
3.2 System decomposition . . . . .	17
3.3 Superblock Hamiltonian . . . . .	18
3.4 Warmup and sweeps . . . . .	19
3.5 Wavefunction computation . . . . .	20
3.6 Results . . . . .	20
3.6.1 The harmonic oscillator . . . . .	21
3.6.2 The Double-Well potential . . . . .	21
3.6.3 The Morse potential . . . . .	24
<b>4 The DMRG algorithm applied to the 1-D Bose-Hubbard model</b>	<b>28</b>
4.1 Superblock Hamiltonian . . . . .	29
4.2 Warmup and sweeps . . . . .	31
4.3 Phase diagram . . . . .	31
4.4 Results . . . . .	32
<b>Conclusions</b>	<b>36</b>
<b>References</b>	<b>37</b>
<b>Appendix</b>	<b>38</b>

# Introduction

The state of matter known as Bose-Einstein condensate was first predicted in 1920s. It took seventy years to prepare it experimentally. Following the first experimental preparation in 1995, the interest in the field has greatly increased. There have been many advances recently. These include progress in laboratory technology - improved cooling, trapping and detection of particles - as well as advances in theoretical description.

The most common ways to keep the Bose-Einstein condensate stable is to use magneto-optical traps and optical lattices. Using these devices it is possible to control the condensate in a very fine manner. They are important for practical purposes and it is desirable to understand them well theoretically.

This work could be divided into four parts. The first one deals with a particular model known as *Bose-Hubbard model*. It describes bosons trapped in a periodic optical potential. Although the model is quite simple, it predicts many properties of a real system. For example, the phase transition between “superfluid” and “insulator” phase predicted by the model has been observed experimentally as well.

The most interesting results that could be obtained from the Bose-Hubbard model are solutions to problems involving a great number of particles. Such problems are essentially many-body problems. The complexity of many-body problems grows exponentially with increasing particle number. Therefore it is very hard to get any quantitative results even for small systems.

However, there exist methods and algorithms that make it possible to get results in reasonable time. The second part of this work deals with one such algorithm called *Density Matrix Renormalization Group* algorithm. The algorithm is introduced and briefly outlined in the second part.

The technical details of the algorithm are explained in the third section on one of the most familiar quantum mechanical problems - a particle in a potential. Using a simple model it is easy to understand various concepts of the algorithm.

The final part of the work builds on the previous parts - density matrix renormalization group algorithm is applied to the Bose-Hubbard model. The results for different model parameters are calculated and compared.

The core of the work is computer modeling of this interesting physical system. All the program code was written in the MATLAB environment by myself.

Finally, I would like to thank my advisor, Dr. Ing. Pavel Soldán, for great support and patience.

# 1 Bose gases in optical lattices

The confinement of ultracold bosons using optical devices is based on quantum mechanical phenomena. This section contains an introduction to the optical trap and the optical lattice mechanism. Subsequently, the model describing atoms in optical lattices is analyzed.

## 1.1 Optical traps

Optical traps are devices that exploit laser light to trap and confine atoms. They make use of the *AC-Stark effect* - an effect of alternating electric field on energy levels of a neutral atom. In this section I will show a simple derivation of formulas describing this phenomenon.

The Hamiltonian of a neutral atom in electric field could be expressed in terms of multipole expansion. It is assumed that the field is spatially uniform on the atomic scale and thus quadrupole and higher terms could be neglected. The Hamiltonian is then

$$H = H_0 + H', \quad (1)$$

where  $H_0$  is the Hamiltonian of the atom without the influence of the electric field and  $H'$  represents the dipole term of multipole expansion.

The atom-field interaction  $H'$  is given by the correspondence principle as

$$H' = -\mathbf{d} \cdot \boldsymbol{\varepsilon}, \quad (2)$$

where  $\boldsymbol{\varepsilon}$  is the electric field vector, and

$$\mathbf{d} = -e \sum_j \mathbf{r}_j \quad (3)$$

is the electric dipole moment operator for the atom electrons.

It is assumed that the time-dependent electric field is in the  $z$  direction and varies in time as  $\varepsilon(t) = \varepsilon_0 \cos \omega t$ . Therefore the interaction Hamiltonian is

$$H' = -d_z \cos \omega t = -\frac{d_z \varepsilon_0}{2} (e^{i\omega t} + e^{-i\omega t}) \quad (4)$$

The Hamiltonian is time-dependent, thus the dynamics of the system could be examined by means of time-dependent perturbation theory. Suppose that the wave function  $\psi$  can be expressed as

$$|\psi(t)\rangle = \sum_n a_n(t) e^{-iE_n t/\hbar} |n\rangle, \quad (5)$$

where  $|n\rangle$  are eigenkets of  $H_0$ ,  $E_n$  are the corresponding eigenvalues of  $H_0$ , and  $a_n$  are complex functions of time  $t$ . By inserting  $|\psi(t)\rangle$  in the Schrödinger equation

$$H|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle, \quad (6)$$

the set of coupled ordinary differential equations for the coefficients  $a_n$  is obtained,

$$i\hbar\dot{a}_n(t) = \sum_k \langle n|H'|k\rangle a_k(t) e^{i\omega_{nk}t}, \quad (7)$$

where

$$\omega_{nk} = \frac{E_n - E_k}{\hbar}. \quad (8)$$

Let us consider an atom initially in an eigenstate  $|m\rangle$  of the unperturbed Hamiltonian  $H$  ( $a_m = 1$ , and  $a_n = 0$  for  $n \neq m$ ), and assume that the electric field is turned on at the time  $t = 0$ . The equations for coefficients  $a_n$  to the first approximation are then

$$a_n^{(1)}(t) = -\frac{1}{2i\hbar} \int_0^t dt' \langle n|d_i\varepsilon_0|m\rangle [e^{i(\omega_{nm}+\omega)t'} + e^{i(\omega_{nm}-\omega)t'}]. \quad (9)$$

Suppose that the frequency of the electric field is not equal to any of the transition frequencies so the integration can be simply carried out and the expression

$$a_n^{(1)}(t) = -\frac{1}{2\hbar} \langle n|d_i\varepsilon_0|m\rangle \left[ \frac{e^{i(\omega_{nm}+\omega)t} - 1}{\omega_{nm} + \omega} + \frac{e^{i(\omega_{nm}-\omega)t} - 1}{\omega_{nm} - \omega} \right]. \quad (10)$$

is obtained.

The expression to the second approximation is found by inserting (10) into (7). The interesting part is the change of the original state  $|m\rangle$ . The differential equation for the coefficient  $a_m(t)$  is

$$\begin{aligned} i\hbar\dot{a}_m^{(2)}(t) &= -\frac{\varepsilon_0^2}{2\hbar} \sum_n \langle m|d_i|n\rangle \langle n|d_i|m\rangle e^{i\omega_{mn}t} \cos \omega t \left[ \frac{e^{i(\omega_{nm}+\omega)t} - 1}{\omega_{nm} + \omega} + \frac{e^{i(\omega_{nm}-\omega)t} - 1}{\omega_{nm} - \omega} \right] = \\ &= -\frac{\varepsilon_0^2}{2\hbar} \sum_n |\langle m|d_i|n\rangle|^2 e^{i\omega_{mn}t} \cos \omega t \left[ \frac{e^{i(\omega_{nm}+\omega)t} - 1}{\omega_{nm} + \omega} + \frac{e^{i(\omega_{nm}-\omega)t} - 1}{\omega_{nm} - \omega} \right]. \end{aligned} \quad (11)$$

The coefficient  $a_m(t)$  can be also expressed in terms of a complex phase  $\phi_m$

$$a_m(t) = e^{-\frac{i}{\hbar}\phi_m(t)}. \quad (12)$$

Its derivative with respect to time is approximately

$$\dot{a}_m(t) = -\frac{i}{\hbar}\dot{\phi}_m(t) e^{-\frac{i}{\hbar}\phi_m(t)} \approx -\frac{i}{\hbar}\dot{\phi}_m(t). \quad (13)$$

Hence the derivative of the complex phase in our approximation is

$$\dot{\phi}_m(t) = i\hbar\dot{a}_m^{(2)}(t) = -\frac{\varepsilon_0^2}{2\hbar} \sum_n |\langle m|d_i|n\rangle|^2 e^{i\omega_{mn}t} \cos \omega t \left[ \frac{e^{i(\omega_{nm}+\omega)t} - 1}{\omega_{nm} + \omega} + \frac{e^{i(\omega_{nm}-\omega)t} - 1}{\omega_{nm} - \omega} \right]. \quad (14)$$

The shift in the energy is given by the average rate at which the phase of the state decreases in time. The time average over one oscillation period of the electric field is carried out and the expression

$$\Delta E = \langle \dot{\phi}_m \rangle_t = -\frac{\varepsilon_0^2}{4\hbar} \sum_n |\langle m|d_i|n\rangle|^2 \left[ \frac{1}{\omega_{nm} + \omega} + \frac{1}{\omega_{nm} - \omega} \right]. \quad (15)$$

is obtained.

If the frequency of the alternating electric field is close to the transition frequency from the state  $|m\rangle$  to an arbitrary state  $|e\rangle$  ( $\omega_{em}$ ), then the corresponding term in (15) becomes dominant and the other terms can be neglected. It is convenient to introduce the *detuning*  $\delta = \omega - \omega_{em}$ . The positive detuning is called *blue detuning* and the negative one is *red detuning*. The expression for the energy shift is then reduced to

$$\Delta E = \frac{\varepsilon_0^2}{4\hbar(\omega - \omega_{em})} |\langle m|d_i|e\rangle|^2 = \frac{\varepsilon_0^2}{4\hbar\delta} |\langle m|d_i|e\rangle|^2. \quad (16)$$

The energy shift of atoms in state  $|m\rangle$  depends on the detuning of the frequency of the electric field and on the intensity of the field  $\varepsilon_0$ . The source of alternating electric field in optical traps is usually a laser light. It is very simple to focus a laser light on a point in space. Although the frequency of the light is the same everywhere, the intensity is maximal at the focal point. Therefore the maximal energy shift is also in this point. The blue detuning results in a positive shift, the red detuning in a negative shift. Using a red detuned laser it is therefore possible to trap atoms in the focal point, as atoms tend to be in points of minimal energy.

Experiments involving optical traps are usually performed with atoms in the ground state, which are precooled to  $\mu\text{K}$  range.

## 1.2 Optical lattices

The optical trap effect can be used to create a periodic trapping structure - optical lattice. Instead of focusing the laser light on just one spatial point, where the light intensity has its maximum, the periodic maxima and minima are created. This can be achieved by creating a standing light wave, using a laser and a mirror (Fig. 1). The amplitude of electric intensity differs along the  $x$  axis and thus an additional potential for atoms is created. In the case of red detuned laser, ground state energy of atoms is minimal at points of maximal amplitude. The spatial dependency of the electric intensity is

$$\varepsilon_0(x) \approx \varepsilon_0 \cos(kx). \quad (17)$$

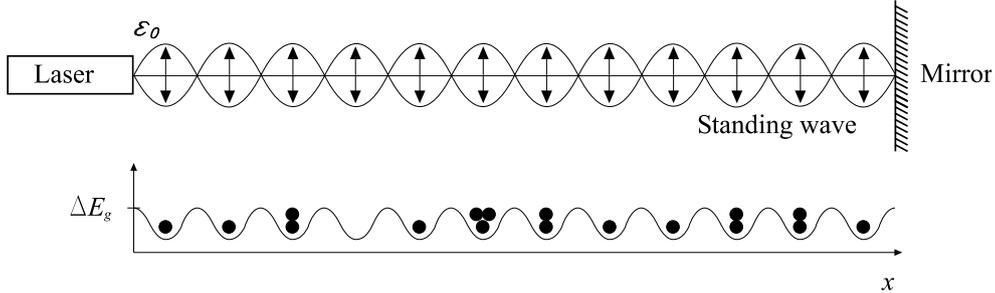
According to (16), this gives lattice potential

$$V_{\text{latt}}(x) \approx V_0 \cos^2(kx). \quad (18)$$

If the same wavelength is used in all dimensions, one gets 3D optical lattice. However, if the wavelength is significantly smaller and the intensity greater in one dimension than

in the other two dimensions, the positions of atoms is effectively restricted to this one dimension. Thus an 1-D, 2-D or 3-D optical lattices can be created by setting proper laser parameters. It is also possible to create different lattice geometries (rectangular, triangular) by a different laser geometry setup.

Figure 1: Principle of an optical lattice: The standing wave of a laser light creates a periodic trapping potential. The energy shift of the ground state  $\Delta E_g$  forces bosonic particles to occupy potential wells. The potential depth depends on the laser light intensity.



Cold atoms interacting with a spatially modulated optical potential resemble in many aspects electrons in an ion lattice potential of a solid crystal. Therefore similar ideas and theory are used in both fields. The main differences are:

- Optical lattices have bigger spacing.
- Optical lattices are mostly defect free.
- They are very easily controlled - e.g. changing potential depth by changing the intensity of the laser.

### 1.3 The Bose-Hubbard model

The simplest non-trivial model that describes interacting ultracold bosons in a periodic optical potential is the Bose-Hubbard model. It includes the main physics describing strongly interacting bosons, which is governed by the competition between the kinetic and the interaction energy. It originates from the Hubbard model, which is very popular in the field of solid state physics, as it describes the transition between conducting and insulating phase in electron systems.

The Bose-Hubbard model predicts a quantum phase transition from a *superfluid* to a *Mott insulator* phase. This transition has been observed experimentally [5].

The Bose-Hubbard Hamiltonian is

$$\hat{H} = -J \sum_{\langle i,j \rangle} \hat{b}_i^\dagger \hat{b}_j + \frac{1}{2} U \sum_i \hat{n}_i (\hat{n}_i - 1), \quad (19)$$

where  $\hat{b}_i$  is the annihilation operator for the site  $i$ ,  $\hat{b}_i^\dagger$  is the creation operator for the site  $i$ , and  $\hat{n}_i = \hat{b}_i^\dagger \hat{b}_i$  is the particle number operator. The parameters are:  $J$  is the hopping matrix

element between adjacent sites  $i, j$  (the notation  $\langle i, j \rangle$  restricts the sum to the nearest-neighbors sites), and  $U$  corresponds to the strength of the onsite repulsion of two atoms on the lattice site  $i$ .

The ratio  $U/J$  determines which term in the (19) is dominant and thus the properties of the system.

The realization of the Bose-Hubbard model using optical lattices has the advantage that the interaction matrix element  $U$  and the tunneling matrix element  $J$  can be controlled by adjusting the intensity of the laser beams. With increase of the optical potential the atomic wave function becomes more and more localized and the onsite interaction increases, while at the same time the tunneling matrix element is reduced. The numerical value of parameters  $U$  and  $J$  can be calculated from lattice parameters.

## 1.4 Phase transitions in the Bose-Hubbard model

At the zero temperature the physics described by the Bose-Hubbard Hamiltonian can be divided into two different regimes. One is the interaction dominated regime when  $J$  is much smaller than  $U$ , and the system is in the Mott insulator phase. The other is the kinetic energy dominated regime, where tunneling overwhelms the repulsion and the system exhibits superfluid properties. The onset of superfluidity is a consequence of the competition between the kinetic energy, which tries to delocalize the particles, and the interaction energy, which tries to localize them and make the fluctuations small.

In the superfluid regime the kinetic energy term dominates the Hamiltonian. Quantum correlations can be neglected and the system can be described by a macroscopic wave function since the many body state is almost a product of identical single particle wave functions. There is a macroscopic well defined phase between adjacent sites. Because the atoms are delocalized, they exhibit an interference pattern when the laser is turned off. In the limit where the tunneling term dominates, ground state of the system for a lattice of  $N$  sites filled with  $M$  atoms is given by

$$|\psi_{SF}\rangle_{U=0} \propto \left( \sum_{i=1}^N \hat{b}_i^\dagger \right)^M |0\rangle. \quad (20)$$

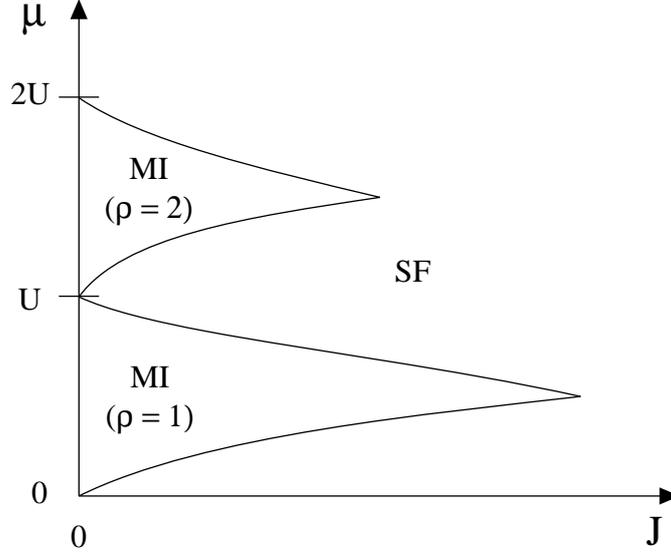
In the Mott insulator phase, the interactions dominate the Hamiltonian and the ground state of the system consists of localized atomic wave functions with a fixed number of atoms per site instead. In the limit, where interaction term dominates, the ground state of the system for a lattice with  $N$  sites and an integer particle density of  $\rho$  atoms per lattice site is described by the wave function

$$|\psi_{MI}\rangle_{J=0} \propto \prod_{i=1}^N \left( \hat{b}_i^\dagger \right)^\rho |0\rangle. \quad (21)$$

The phase diagram of Bose-Hubbard model in a  $J - \mu$  plane, where  $\mu$  is the chemical potential, includes lobe-like structures (Fig. 2). Each Mott lobe is characterized by having a

fixed integer particle density  $\rho$ . Inside the Mott insulator phases the compressibility  $\partial\rho/\partial\mu$  vanishes.

Figure 2: Schematic phase diagram of the Bose-Hubbard model. It shows two Mott insulator (MI) phases with particle densities  $\rho = 1$  and  $\rho = 2$ , surrounded by the superfluid phase (SF).



The physics behind this diagram can be understood as follows: If we start at some point in the Mott insulating phase and increase  $\mu$  keeping  $J$  fixed, there is going to be a point at which the kinetic energy of an extra particle will balance the on-site interaction energy cost. With an extra particle free to move around the lattice phase coherence is recovered, and the system enters the superfluid regime. Similarly, by decreasing  $\mu$  from a point in the Mott phase, at some point it will be eventually energetically favorable to remove one particle from the system. The extra mobile hole will also induce phase coherence, and the system will condense in a superfluid state. Since the kinetic energy of the system increases with  $J$ , the width of the lobes decreases with  $J$ . The distance in the  $\mu$  direction at fixed  $J$  between the upper and lower part of the lobe is the energy gap. At  $J = 0$  the gap is just equal to  $U$ .

Mott insulator phases only occur at integer densities; non-integer density states lie entirely in the superfluid phase because there is always an extra particle that can hop without any energy cost.

## 2 Density matrix renormalization group algorithm

The problem of finding bound states of a quantum system is one of the most fundamental problems in the field of quantum physics. The basic approach is to construct the Hamiltonian and to diagonalize it. In special cases this can be done analytically. However only a small number of interesting Hamiltonians can be solved this way. A perturbative approach is also useful in certain cases, but only if the perturbations are very small. The only method, which is applicable almost in all cases is a numerical approach - to construct the matrix Hamiltonian and to find the eigenstates of this matrix. There are two issues:

1. As many systems have infinite number of allowed states, the number of base vectors has to be truncated to a finite number. Only the most essential states have to be retained. The question is what are the most important states. The answer depends on the bound states one is interested in. If the problem is to find the ground state, that is the state with minimal energy, the most important base vectors to retain are the vectors of minimal energy.

For example, in the case of the Bose-Hubbard model (19), the number of particles on one site can be infinite. Nevertheless, it is clear that the states with more particles on the site have higher energy due to the interaction term  $U$ . Thus if one wants to find the ground state of this model, it is reasonable to limit the number of allowed particles on the site.

2. If the quantum system is composed of subsystems, the total Hilbert space is the tensor product of Hilbert spaces of the subsystems. The total space dimension is a product of the corresponding subsystem dimensions. As the number of subsystems increases linearly, the dimension of total space increases exponentially. Thus, for a big system, it is very complicated to diagonalize the matrix Hamiltonian, because of the matrix size. This is called the *many-body problem*.

For example, consider the Bose-Hubbard model and limit the number of particles on the site to  $n = 5$ . If the system has  $N$  sites, the dimension of the total space is  $d = n^N$ . For the system of  $N = 10$  sites this gives dimension  $d = 9765625$ . Obviously, it is very complicated to create, store and diagonalize such a big matrix. Even such a small system of  $N = 10$  sites and  $n = 5$  maximum particles per site is in fact too complex for numerical computation on most of today's computers.

In the case of ultracold physics, one is usually interested in the ground state properties. Thus the solution to the first issue is clear - limit the number of particles onsite.

There are many methods and approximate methods how to overcome the second issue, the many-body problem, for example *mean field theory* model or *quantum Monte Carlo*. However, there exists an approach which is very powerful especially for one dimensional problems - the *Renormalization Group* approach. I will focus primarily on this approach and on a particular method called the *Density Matrix Renormalization Group (DMRG)* algorithm.

The basic idea of renormalization group methods is quite simple - to divide the system into blocks and to treat every block separately, considering interactions between blocks. Only a fixed number of states is retained in every block. The way of creating the blocks and choosing the retained states characterizes the particular renormalization group method.

## 2.1 Blocks renormalization group algorithm

The basic method is a *Blocks Renormalization Group (BRG)* algorithm in one dimension. Consider a lattice system and let  $m$  be number of states kept in a block. Then the algorithm is:

1. Describe interactions on an initial sublattice ("block")  $A$  of length  $l$  by a block Hamiltonian  $H_A$  acting on an  $m$ -dimensional Hilbert space.
2. Form a compound block  $AA$  of length  $2l$  and the Hamiltonian  $H_{AA}$ , consisting of two block Hamiltonians and interblock interactions.  $H_{AA}$  has dimension  $m^2$ .
3. Diagonalize  $H_{AA}$  to find the  $m$  lowest-lying eigenstates.
4. Project  $H_{AA}$  onto the truncated space spanned by the  $m$  lowest-lying eigenstates,  $H_{AA} \rightarrow H'_{AA}$ .
5. Restart from step (2), with doubled block size:  $2l \rightarrow l$ ,  $AA \rightarrow A$ , and  $H'_{AA} \rightarrow H_A$ , until the lattice size is reached.

The most time consuming part of this algorithm is the diagonalization of  $H_{AA}$ . In every block only  $m$  lowest-lying states are retained, and thus the size of matrix Hamiltonians  $H_{AA}$  is always the same. The complexity of the algorithm grows linearly with doubling the block length. Without the truncation of high energy states the system would grow exponentially, thus the improvement is significant.

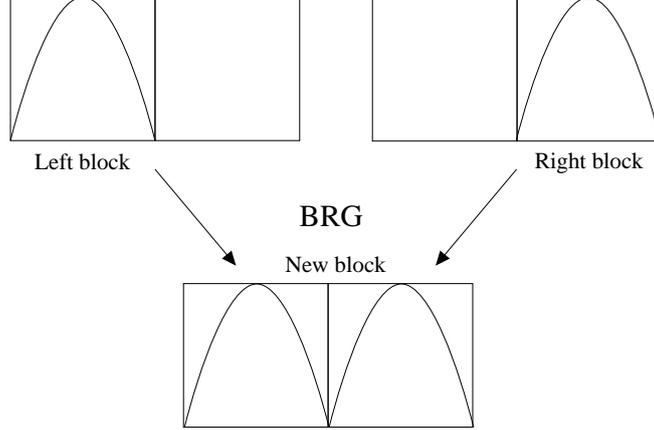
However, the outlined procedure gives very poor results for certain problems. The breakdown can best be understood visually (Fig. 3): assuming an already rather large block size, where discretization can be neglected, the lowest-lying states of the left block all vanish at the compound block center. The true ground state of a new block has its maximum amplitude right there, thus it cannot be properly approximated by a combining the states from left and right block only.

The problem lies in an inappropriate boundary conditions imposed on the states in both blocks. The blocks cannot be treated as independent systems. They have to be embedded in some environment.

## 2.2 Density matrix renormalization group algorithm

The boundary condition problem could be solved by introducing an interaction between the block and the environment. Instead of finding the ground state of the block alone, the compound system of the block and the environment is constructed - the *superblock*. The

Figure 3: Poor results of the BRG: All the states lying in left and right block have boundary conditions such that the wavefunction vanishes at the block boundary. These states cannot accurately represent low lying states of the compound block.



ground state of the superblock is found, projected onto the block and then the block space is truncated. The question is how to perform the projection and the truncation.

The compound system space is a tensor product of the block space and the environment space. A pure state in compound space is expressed as

$$|\psi\rangle = \sum_{i,j} \psi_{i,j} |\alpha_i\rangle \otimes |\beta_j\rangle, \quad (22)$$

where  $|\alpha_i\rangle$  for  $i = 1, \dots, m$  and  $|\beta_j\rangle$  for  $j = 1, \dots, m$  are base vectors of block space and environment space, respectively. The state is generally an entangled state for the block and the environment, therefore it could not be simply described as a single block state and a single environment state. Nevertheless, the block states, which best describe the ground state of superblock, are sought.

The key is to trace environment out and to construct density matrix  $\rho_B$  for the block

$$\langle \alpha_{i'} | \rho_B | \alpha_i \rangle = \sum_j \psi_{i,j}^* \psi_{i',j}. \quad (23)$$

$\rho_B$  essentially describes the states on the block which belong to the ground state on the superblock. The density matrix is positive-semidefinite, self-adjoint and has a unit trace.  $\rho_B$  has  $m$  eigenvalues  $w_i \geq 0$  and respective eigenstates  $|\gamma_i\rangle$ , with  $\sum_i w_i = 1$ .

The most relevant states are the states with the highest eigenvalues. Therefore in the truncation of the block space it is most reasonable to retain these states as they best describe the ground state of the superblock in the block space.

## 1D system

The DMRG for 1D system is briefly outlined here. The BRG algorithm above uses exponential growth of the system. For the DMRG algorithm the linear growth is more appropriate, that means instead of doubling the block every step it is better to add one site at the time. 1D system is a chain with interactions between sites:

$$\bullet - \bullet - \bullet - \bullet.$$

Adding one site at the time, the block becomes:

$$\bullet - \bullet - \bullet - \bullet - \circ.$$

There are many ways how to select and describe environment. A good approach in 1D case is to use the same type of system, inverted:

$$\bullet - \bullet - \bullet - \bullet - \circ - \circ - \bullet - \bullet - \bullet - \bullet.$$

The right and left part could be represented as blocks:

$$L - \circ - \circ - R.$$

Let  $m$  be the number of states kept in a block and  $n$  be the number of allowed states on a site. One DMRG step works as follows:

1. Describe interactions on the left block  $L$  and on the right block  $R$  by a block Hamiltonians  $H_L$  and  $H_R$  acting on an  $m$ -dimensional Hilbert spaces.
2. Describe interactions on left and right central sites by Hamiltonians  $H_{CL}$  and  $H_{CR}$ .
3. Form a compound superblock and respective Hamiltonian  $H_{SB}$ , consisting of left block, left central site, right central site and right block Hamiltonians and interactions between them.  $H_{SB}$  has dimension  $m^2n^2$ .
4. Diagonalize  $H_{SB}$  to find its ground state.
5. Create density matrix for the left part (left block and central left site combined).
6. Find  $m$  density matrix eigenstates with highest eigenvalues.
7. Project left part Hamiltonian ( $H_L$  and  $H_{CL}$  combined) onto the  $m$ -dimensional truncated space spanned by these eigenstates.
8. Do the steps 5-7 for the right part as well.

If the size of the original block (number of included sites) was  $l$ , the new block size is  $l + 1$ .

There are several methods for calculating parameters of physical systems using this algorithm. One approach, *infinite-system DMRG*, is used for a very long chain. The size of the block is increased in every step. The other approach, *finite-system DMRG*, is used for finite length systems. We will deal with the latter case only.

## Warmup

The first step in finite-system DMRG is a warmup phase - preparing block data (Hamiltonians and operators describing interactions with adjacent site) for different system lengths. Repeating the above process from an initial small block, the size of the system grows at the rate of two sites in a step (left and right block grows by one site each). The block data for different system sizes are stored. The process is repeated until the size of the left and the right blocks and two sites between them reaches the size of the system  $N$ :

$$L(N/2 - 1) - \circ - \circ - R(N/2 - 1),$$

where  $L(n)$ ,  $R(n)$  are left and right block containing  $n$  sites.

## Sweep

During a sweep, the division point between left and right parts moves in both directions. Firstly to the right; new left blocks are created:

$$L(n) - \circ - \circ - R(N - n - 2) \rightarrow L(n + 1) - \circ - \circ - R(N - n - 1),$$

When the smallest possible right block is reached, the sense of motion is inversed and the right blocks for various partitioning of the system are created or updated:

$$L(n) - \circ - \circ - R(N - n - 2) \rightarrow L(n - 1) - \circ - \circ - R(N - n - 3),$$

When the smallest possible left block is reached, the sense is inverted one more time. The left blocks are updated until the division point reaches the center of system.

In one sweep, the block data for all possible divisions of the chain are updated. The sweeping process is repeated a few times. The block data and values of various observables usually converge after one or two sweeps.

DMRG for specific models is explained in a more detail in next sections.

### 3 The DMRG algorithm applied to the single-particle quantum mechanics

A simplified version of the DMRG algorithm can be successfully applied to a 1D quantum mechanical problem - to find the ground state and few excited states of a particle confined on the real line ([6], [7]).

Particle dynamics in appropriate units is governed by the Hamiltonian

$$H = p^2 + V(x), \quad (24)$$

where  $p$  is the momentum operator and  $V(x)$  is the potential operator. The first step is to discretize the real line - to constrain the particle position to the discrete values  $x_n = h(n - \frac{N+1}{2})$  for  $n = 1, 2, \dots, N$ .  $N$  is the number of allowed sites and  $h$  is the lattice spacing. The second step is to create a discretized Hamiltonian. The well-known approximate formula for the second derivative,

$$f''(x_n) = \frac{f(x_{n-1}) - 2f(x_n) + f(x_{n+1}))}{h^2}, \quad (25)$$

is used for this purpose. The Hamiltonian then becomes an  $N \times N$  matrix

$$H_{ij} = \begin{cases} 2/h^2 + V_i & i = j \\ -1/h^2 & |i - j| = 1 \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

In the limit  $h \rightarrow 0^+$  and  $N \rightarrow \infty$  this Hamiltonian coincides with Hamiltonian (24).

#### 3.1 Relation to the many-body problem

The DMRG algorithm outlined in section 2.2 deals with many-body problems. From that point of view every lattice site is a two dimensional Hilbert space  $\mathcal{H}$ . The site could be in two states - there is no particle on the site ( $|0\rangle$ ) or there is a particle on the site ( $|1\rangle$ ). The Hilbert space for the system of all lattice sites,

$$\mathcal{H} = \mathcal{H} \otimes \dots \otimes \mathcal{H}, \quad (27)$$

is a product space of respective site Hilbert spaces. Such a system has dimension  $2^N$ , where  $N$  is the number of lattice sites. It is clearly not equivalent to the system described by Hamiltonian (26), which has dimension  $N$ .

The relation between these systems is the fact that there is just one particle in the system. The states with one particle form a Hilbert subspace  $\mathcal{S} \subset \mathcal{H}$ . The basis of this

subspace is:

$$\begin{aligned}\psi_1 &= |1\rangle \otimes |0\rangle \otimes |0\rangle \dots, \\ \psi_2 &= |0\rangle \otimes |1\rangle \otimes |0\rangle \dots, \\ \psi_3 &= |0\rangle \otimes |0\rangle \otimes |1\rangle \dots, \\ &\vdots\end{aligned}$$

The Hamiltonian (26) is a projection of a many-body Hamiltonian to the subspace  $\mathcal{S}$ .

The Hamiltonian for a single-particle problem is therefore much simpler. In this case it is not necessary to construct tensor product space at all. Moreover, the calculation of density matrix is very simple too.

The state of the compound system has the form (22):

$$|\psi\rangle = \sum_{i,j} \psi_{i,j} |\alpha_i\rangle \otimes |\beta_j\rangle, \quad (28)$$

where  $|\alpha_i\rangle$  are states on the system space and  $|\beta_j\rangle$  are states on the environment space.

Because there is one particle in the whole system, states  $|\alpha_i\rangle$  and  $|\beta_j\rangle$  can contain only zero or one particle each. Let us suppose that the states containing no particles are states  $|\alpha_0\rangle$  and  $|\beta_0\rangle$ . The compound-system state has to contain exactly one particle. The base vectors containing one particle are only the following ones:

$$|\alpha_i\rangle \otimes |\beta_0\rangle, \quad \text{for } i > 1, \quad (29)$$

$$|\alpha_0\rangle \otimes |\beta_j\rangle, \quad \text{for } j > 1. \quad (30)$$

$$(31)$$

Therefore all possible state vectors could be expressed as:

$$|\psi\rangle = \sum_i \psi_i |\alpha_i\rangle \otimes |\beta_0\rangle + \sum_j \varphi_j |\alpha_0\rangle \otimes |\beta_j\rangle. \quad (32)$$

The density matrix for compound-system state  $|\psi\rangle$ , constructed according to formula (23), has block form

$$\rho_B = \begin{pmatrix} \sum_j \varphi_j^2 & 0 \\ 0 & B \end{pmatrix}, \quad (33)$$

where block  $B$  is

$$B_{ij} = \psi_i \psi_j^*. \quad (34)$$

To find the most relevant states describing a system without environment it is necessary to find eigenvectors of the density matrix. Let  $X$  be the vector

$$X = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \end{pmatrix}. \quad (35)$$

The block matrix  $B$  is then equal to

$$B = XX^\dagger. \quad (36)$$

Matrix  $B$  is Hermitian and thus can be diagonalized to a matrix  $D$  by a unitary matrix  $U$ :

$$D = UBU^\dagger = UXX^\dagger U^\dagger = YY^\dagger, \quad (37)$$

where  $Y = UX$ .  $Y$  is a column vector and  $Y^\dagger$  is a row vector. The diagonal form  $D$  of the matrix  $B$  could be therefore expressed as a product of a column vector  $Y$  and a row vector  $Y^\dagger$ . This is possible only if the vector  $Y$  has one non-zero element at most. This non-zero element could be found by multiplying vectors  $Y$  and  $Y^\dagger$  in different order:

$$Y^\dagger Y = X^\dagger U^\dagger U X = X^\dagger X = \sum_i \psi_i^2. \quad (38)$$

Thus the matrix  $B$  has only one non-zero eigenvalue  $p = \sum_i \psi_i^2$ . The eigenvector corresponding to  $p$  is the vector  $X$ .

Now it is easy to find eigenvalues and eigenvectors of the density matrix  $\rho_B$  - it has only two non-zero eigenvalues  $p_i$ :

$$p_1 = \sum_i \psi_i^2, \quad \text{eigenvector } |\psi_1^B\rangle = \sum_i \psi_i |\alpha_i\rangle, \quad (39)$$

$$p_2 = \sum_j \varphi_j^2, \quad \text{eigenvector } |\psi_2^B\rangle = |\alpha_0\rangle. \quad (40)$$

$$(41)$$

It is therefore trivial to find the most relevant states of the density matrix.  $|\psi_1^B\rangle$  is a projection of (32) to the respective subsystem.  $|\psi_2^B\rangle$  is a state with no particle in the subsystem.

For a single-particle problem, it is not necessary to construct density matrix at all. The relevant states are a simple projection and an empty state.

### 3.2 System decomposition

Following section 2.2, the basic idea is to divide system into a left block  $L$ , a right block  $R$  and two lattice sites  $\circ$  in between,

$$L - \circ - \circ - R. \quad (42)$$

The number of states in the right and the left block is truncated to  $m$ . For every block it is necessary to store an  $m \times m$  block matrix Hamiltonian and a hook vector of length  $m$ . A hook vector describes interaction between the block and the adjacent site. Therefore the blocks are sets  $L = \{H_L, T_L\}$ ,  $R = \{H_R, T_R\}$ .

All such decompositions of the system are considered. The possible left and right blocks are

$$\{L(m), L(m+1) \dots L(N-m-2)\} \quad \{R(m), R(m+1) \dots R(N-m-2)\}, \quad (43)$$

since a block with less than  $m$  sites cannot have  $m$  independent “states”.

If  $j$  for  $m < j < N - m - 2$  is a position in the lattice, the decomposition for  $j$  could be expressed as  $L(j) - \circ - \circ - R(N - j - 2)$ .

### 3.3 Superblock Hamiltonian

Superblock (SB) Hamiltonian is a  $(2m+2) \times (2m+2)$  matrix Hamiltonian constructed for a particular decomposition of the system from the left block  $H_L$ , left hook vector  $T_L$ , the two central sites  $H_{CL}$ ,  $H_{CR}$ , the right block  $H_R$ , and the right hook vector  $T_R$ ,

$$H_{SB} = \begin{pmatrix} H_L & T_L & 0 & 0 \\ T_L^\dagger & H_{CL} & -1/h^2 & 0 \\ 0 & -1/h^2 & H_{CR} & T_R^\dagger \\ 0 & 0 & T_R & H_R \end{pmatrix}. \quad (44)$$

$H_{CL}$  and  $H_{CR}$  are diagonal terms from (26).

The SB Hamiltonian can be constructed for every decomposition of the system. It is used to find the next block and hook vector - in the process of either

$$L(p) \rightarrow L(p+1) \quad \text{or} \quad R(p) \rightarrow R(p+1).$$

Let us describe this process for the left block. The first step is to diagonalize the superblock and to obtain its eigenstates  $\psi_i$ , ( $i = 1, \dots, 2m+2$ ). Every state has its respective energy and the states are sorted according to their energies. The interesting states are the  $m$  lowest energy states in the left part of the superblock. The left part of SB Hamiltonian

$$H_{SB} \rightarrow H_{SBL} = \begin{pmatrix} H_L & T_L \\ T_L^\dagger & h_{CL} \end{pmatrix}, \quad (45)$$

and left part of the eigenstates

$$\psi_i = \begin{pmatrix} a_1 \\ \dots \\ a_{m+1} \\ a_{m+2} \\ \dots \\ a_{2m+2} \end{pmatrix} \rightarrow \psi_i^L = \begin{pmatrix} a_1 \\ \dots \\ a_{m+1} \end{pmatrix}. \quad (46)$$

are created.

The goal is to find a new  $m \times m$  matrix Hamiltonian that describes  $m$  lowest lying states in  $H_{SBL}$ . At the beginning the first  $m$  states  $\psi_i^L$  are chosen. In case that the space

spanned by these vectors has lower dimension than  $m$ , the first  $m + 1, m + 2, \dots$  states  $\psi_i^L$  are chosen until the dimension of this space is  $m$ . Although the eigenstates of  $H_{SB}$  are orthogonal, the projected states  $\psi_i^L$  are not necessarily orthogonal. Therefore the Gram-Schmidt orthonormalization process is carried out for chosen  $\psi_i^L$ . As a result  $m$  orthonormal vectors  $\psi_i^O$ , which describe the  $m$  lowest energy states in  $H_{SBL}$ , are obtained.

The new matrix Hamiltonian that describes interaction between these states is constructed as

$$(H_{LN})_{ij} = \langle \psi_i^O | H | \psi_j^O \rangle. \quad (47)$$

After the introduction of a matrix

$$O = (\psi_1^O, \dots, \psi_m^O), \quad (48)$$

it can be expressed as transformation

$$H_{LN} = O^\dagger H_{SBL} O. \quad (49)$$

The transformation of the hook vector is even simpler. The hook vector describes interaction between states  $\psi_i^O$  and right central site. The only terms in  $H_{SB}$  matrix, which are significant here, are the terms  $(H_{SB})_{m+1, m+2}$  and it's conjugate  $(H_{SB})_{m+2, m+1}$ . The transformation is

$$(T_{LN})_i = (\psi_i^O)_{m+1} (H_{SB})_{m+1, m+2}. \quad (50)$$

The algorithm for the left part of the SB Hamiltonian is analogous.

As explained in section 3.1, it is not necessary to construct a density matrix for our one-particle case. The results obtained are the same as results obtained by a construction of full density matrix.

### 3.4 Warmup and sweeps

The DMRG algorithm consists of a warmup phase, in which the blocks are initialized, and several sweep cycles. The initial left block is the upper left  $m \times m$  part of (26) and the initial right block is the bottom left  $m \times m$  part of (26)

$$\begin{aligned} (H_L)_{ij} &= (H)_{ij}, & i, j &= 1, \dots, m, \\ (H_R)_{ij} &= (H)_{ij}, & i, j &= N - m + 1, \dots, N. \end{aligned} \quad (51)$$

The initial hook vectors are constructed analogously

$$\begin{aligned} (T_L)_i &= (H)_{i, m+1}, & i, j &= 1, \dots, m, & \Rightarrow T_L &= (0, \dots, 0, -1/h^2) \\ (T_R)_i &= (H)_{i, N-m}, & i, j &= N - m + 1, \dots, N, & \Rightarrow T_R &= (-1/h^2, 0, \dots, 0). \end{aligned} \quad (52)$$

These are the blocks  $L(m)$  and  $R(m)$ . The initial superblock is created from  $L(m)$ ,  $R(m)$  and from the sites at positions  $m + 1$  and  $N - m$  in the lattice. The virtual connection between these sites is used only in the warmup phase. According the scheme above it is possible to construct blocks  $L(m + 1)$  and  $R(m + 1)$  from this superblock. The process is

repeated until the blocks  $L(j)$ ,  $R(j)$  for  $j = m, \dots, N/2 - 1$  are obtained. The warmup phase is finished here, as it is possible to create a proper decomposition of the system (with the real interaction between central sites)  $L(N/2 - 1) - \circ - \circ - R(N/2 - 1)$ .

The sweep cycle is a cycle in which every block is updated once. It consists of three parts in which the system is decomposed as  $L(j) - \circ - \circ - R(N - j - 2)$ :

- $j = N/2 - 1, \dots, N - m - 3$ . Left blocks  $L(j + 1)$  are updated. The right blocks  $R(j)$  for  $j = m, \dots, N/2 - 1$  were created during the warmup phase, so such process is possible.
- $j = N - m - 2, \dots, m + 1$ . Right blocks  $R(j + 1)$  are updated.
- $j = m, \dots, N/2 - 1$ . Left blocks  $L(j + 1)$  are updated.

The full DMRG algorithm consists of a warmup phase and several sweep cycles until convergence is attained. Five cycles are usually sufficient, as the improvement after more cycles is minimal. However, the accuracy of the convergence depends mostly on  $m$  and  $N$ .

### 3.5 Wavefunction computation

In the single-particle DMRG algorithm it is also possible to compute the wavefunction of a state. For every decomposition of the system the eigenstates  $\psi_i$  of the superblock are computed. The elements  $a_{m+1}$ ,  $a_{m+2}$  in (46) are in fact the values of the wavefunction  $i$  at positions  $m + 1$  and  $m + 2$ . By computing  $a_{m+1}$ ,  $a_{m+2}$  for every decomposition of the system we obtain the whole wavefunction (on the left and right lattice edge is used the whole left and right part of eigenstate  $\psi_i$  as wavefunction values).

### 3.6 Results

In this section the certain potentials are examined using DMRG single-particle algorithm.

It is important to realize the differences between continuous and discretized single-particle problem. *Discretized potential has hard walls on the lattice ends and thus the bound states resemble bound states of a particle in a box for sufficiently high energies.* In every problem it should be carefully examined which states are bound states of the particle in the given potential and which states are bound states of the “hard wall on lattice ends” potential. Naturally, the algorithm could not find any states with eigenvalues from continuous part of the spectrum.

The accuracy generally grows with increasing the number of lattice sites and lowering the lattice spacing.

The expectation values are most accurate, if measured in the middle of the lattice. If measured for other system decompositions, the values slightly differ. The convention in the data below is that the last printed digit is the only digit that was altered in the last sweep. If all the printed digits were altered in the last sweep, the number is in parentheses.

### 3.6.1 The harmonic oscillator

The potential of a harmonic oscillator is

$$V(x) = kx^2, \quad (53)$$

and its energy levels are

$$E_n = \left(n + \frac{1}{2}\right) \hbar\omega. \quad (54)$$

The constant was chosen as  $k = 1$ . Then  $\hbar\omega = 2$  and the energy levels are  $E_n = 2n + 1$ .

Firstly, energies of first two energy states were calculated for various lattice parameters. That means that results of [6] were recalculated. The results are in Tab. 1.

Table 1: Accuracy of DMRG for the harmonic oscillator ( $N$  - number of lattice sites,  $h$  - lattice spacing,  $R = N \times h$ ,  $m$  - number of calculated states,  $s$  - number of sweeps,  $\Delta E_0 = 1 - E_0$ ,  $\Delta E_1 = 3 - E_1$  - difference between exact and calculated energy levels).

$N$	$h$	$R$	$m$	$s$	$\Delta E_0$	$\Delta E_1$
100	0.1	10	2	5	$6.253912540 \times 10^{-4}$	$3.12852166338 \times 10^{-3}$
1000	0.01	10	2	5	$6.24989 \times 10^{-6}$	$3.124334 \times 10^{-5}$
5000	0.002	10	2	5	$2.498 \times 10^{-7}$	$1.2427 \times 10^{-6}$
10000	0.001	10	2	5	$6.2 \times 10^{-8}$	$3.05 \times 10^{-7}$
20000	0.0005	10	2	5	$1.6 \times 10^{-8}$	$7.1 \times 10^{-8}$
20000	0.001	20	2	5	$6.2 \times 10^{-8}$	$3.12 \times 10^{-7}$
50000	0.0002	10	2	5	$(4.94) \times 10^{-9}$	$(7.79) \times 10^{-9}$
50000	0.0004	20	2	5	$(9.47) \times 10^{-9}$	$5 \times 10^{-8}$
100000	0.0001	10	2	5	$(2.08) \times 10^{-9}$	$(-2.39) \times 10^{-9}$
100000	0.0002	20	2	5	$(2.82) \times 10^{-9}$	$1 \times 10^{-8}$

Next the first nine energy levels and wavefunctions for selected lattice parameters were calculated. The results are in Tab. 2, Fig. 4 and Fig. 5.

As we can see, the results are very accurate even for a small number of sites and a big lattice spacing.

### 3.6.2 The Double-Well potential

The expression for this potential is

$$V(x) = -x^2 + Cx^4. \quad (55)$$

The potential has two minima at the positions

$$x_0^{(\pm)} = \pm \frac{1}{\sqrt{2C}}, \quad (56)$$

Table 2: Calculated energy levels for the harmonic oscillator ( $N = 10000$ ,  $h = 0.001$ ,  $m = 9$ ).

Level	Energy
$E_0$	0.9999999376
$E_1$	2.9999996948
$E_2$	4.9999993548
$E_3$	7.0000008701
$E_4$	9.0000226114
$E_5$	11.0001928891
$E_6$	13.0011952591
$E_7$	15.0058306483
$E_8$	17.022869670

Figure 4: Energy levels for the harmonic oscillator ( $N = 10000$ ,  $h = 0.001$ ,  $m = 9$ ).

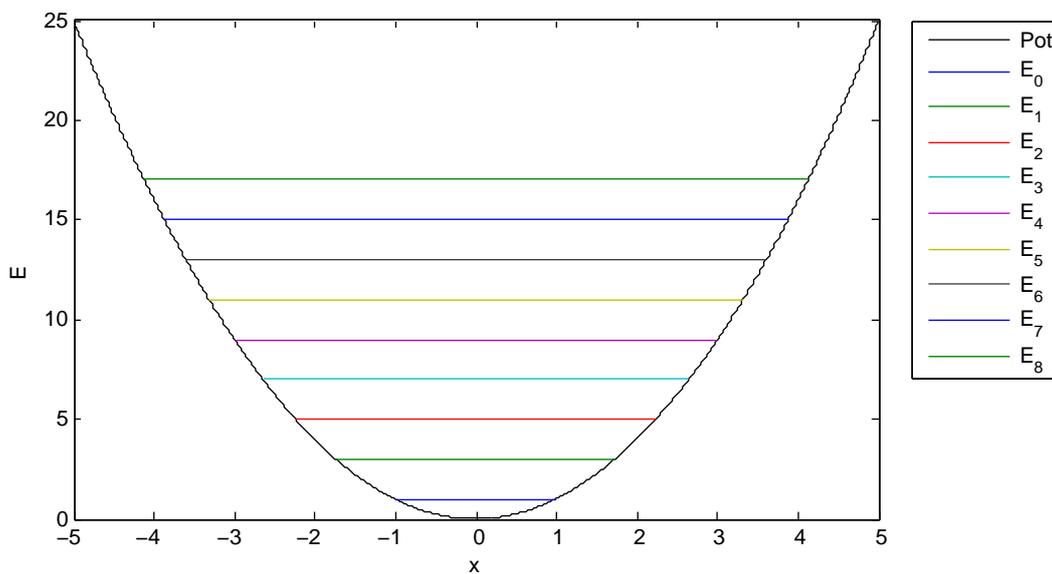
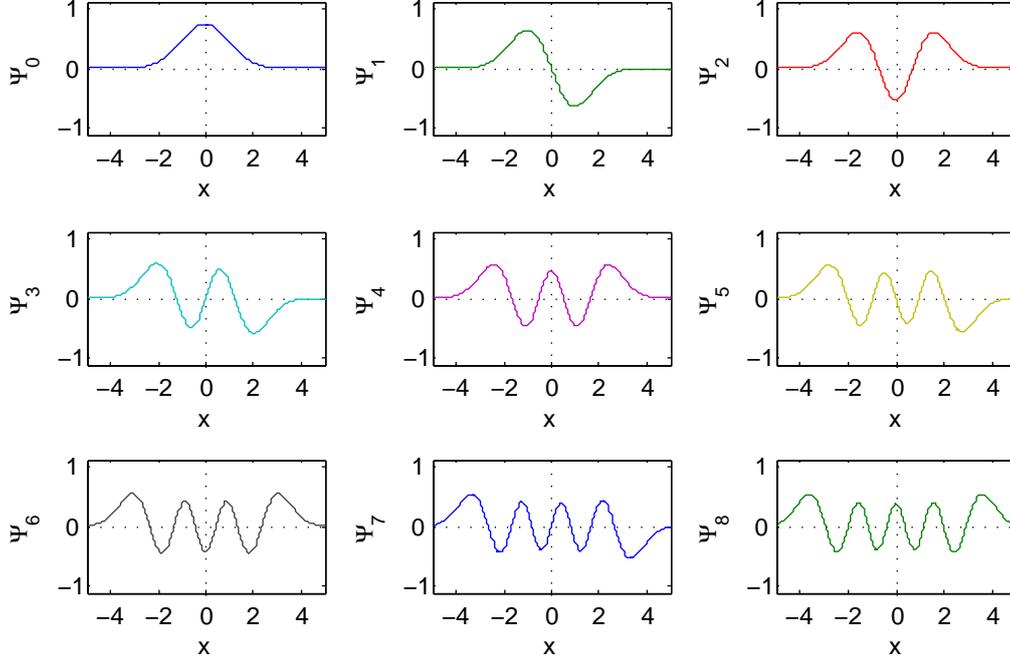


Figure 5: Wavefunctions for the harmonic oscillator ( $N = 10000$ ,  $h = 0.001$ ,  $m = 9$ ).



and a barrier between them. The zero value of potential is on the top of the barrier. Thus states localized in the potential wells have negative energy.

This potential is interesting because there is a small energy gap between the ground state and the first excited state, which depends on the value of  $C$ . For small values of  $C$  this non-trivial behavior cannot be treated perturbatively. The lower is the value of  $C$ , the bigger is the distance between potential minima. Therefore for small values of  $C$  the lattice length has to be big enough.

Again, the results of [6] were recalculated (Tab. 3).

Table 3: Double-Well potential ( $C$  - coupling constant,  $E_0$  - energy of the ground state,  $E_1$  - energy of the first excited state).

$N$	$h$	$R$	$m$	$s$	$C$	$E_0$	$E_1 - E_0$
1000	0.01	10	2	5	1	0.65764425361	2.17682571030
20000	0.0005	10	2	5	1	0.657652983	2.176883053
20000	0.0005	10	2	5	0.6	0.391952619	1.633284793
40000	0.001	40	2	5	0.1	-1.265492922	$1.12433687 \times 10^{-1}$
60000	0.0005	30	2	5	0.06	-2.823639492	$7.299766 \times 10^{-3}$
90000	0.0004	36	2	5	0.02	-11.10647243	$(2.13) \times 10^{-9}$

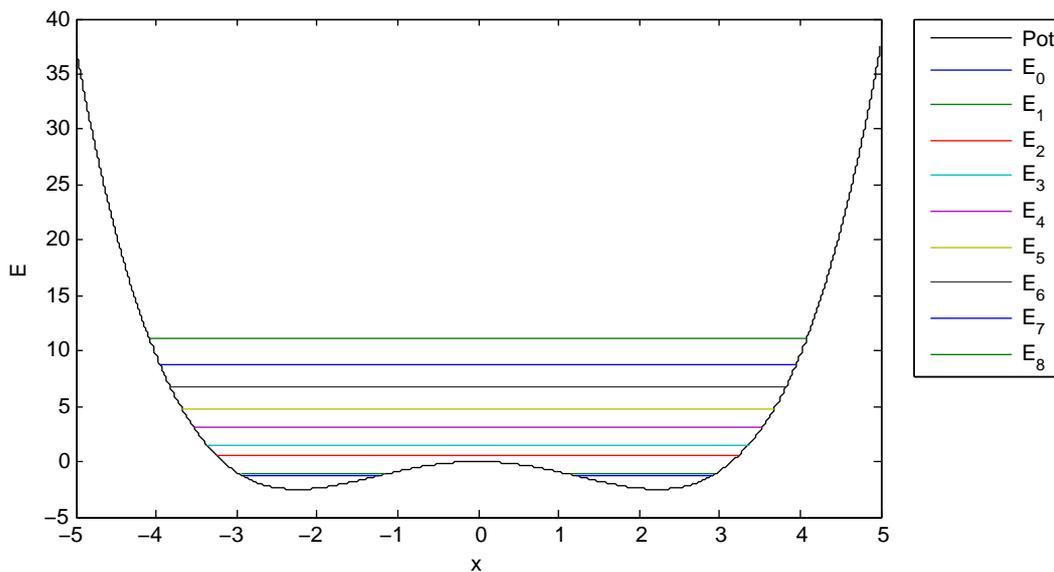
The nine energy levels and wavefunction for particular parameters are in Tab. 4, Fig.

6 and Fig. 7.

Table 4: Calculated energy levels for the Double-Well potential ( $N = 10000$ ,  $h = 0.001$ ,  $m = 9$ ,  $C = 0.1$ ).

Level	Energy
$E_0$	-1.2654923985
$E_1$	-1.1530585362
$E_2$	0.5094905070
$E_3$	1.5435531220
$E_4$	3.1051577298
$E_5$	4.8361932723
$E_6$	6.7570799236
$E_7$	8.8354601140
$E_8$	11.055370858

Figure 6: Energy levels for the Double-Well potential ( $N = 10000$ ,  $h = 0.001$ ,  $m = 9$ ,  $C = 0.1$ ).

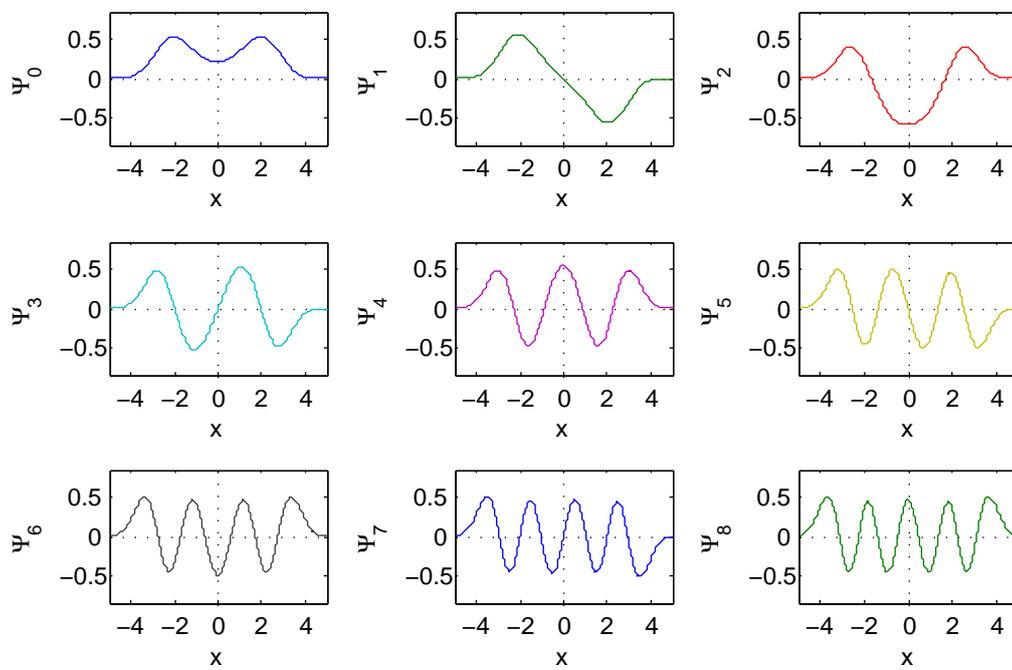


### 3.6.3 The Morse potential

The Morse potential is an empirical potential that approximately describes the chemical bond between two atoms. It is asymmetric indicating that it is harder to compress a bond than to pull it apart. The expression for this potential is

$$V(x) = D_e (1 - e^{-\alpha(x-x_e)})^2, \quad (57)$$

Figure 7: Wavefunctions for the Double-Well potential ( $N = 10000$ ,  $h = 0.001$ ,  $m = 9$ ,  $C = 0.1$ ).



where  $D_e$  is the depth of potential minimum,  $\alpha = \sqrt{\frac{k}{2D_e}}$  is the bond parameter and  $x_e$  is the equilibrium distance.

The allowed energy levels can be calculated analytically ([8]). These energies are given by the formula

$$E_n = \left(n + \frac{1}{2}\right) \hbar\omega - \left(n + \frac{1}{2}\right)^2 \frac{\hbar^2\omega^2}{4D_e}, \quad (58)$$

where  $\hbar\omega = 2\alpha\sqrt{D_e}$ .

With the Morse potential, the energy between adjacent levels decreases with increasing  $n$  as is seen in nature. It fails at the value of  $n$  where  $E_{n+1} - E_n$  is calculated to be zero or negative. The Morse potential is a good approximation for vibrational energy levels of diatomic molecules below this limit.

The constants were chosen as  $D_e = 50$ ,  $\alpha = 1$ ,  $x_e = -49$ . Difference between the values calculated by DMRG and the formula (58) is in Tab. 5. DMRG energy levels and wavefunctions are in Fig. 8 and Fig. 9.

The agreement between DMRG and the formula (58) values is very good, the first seven bound states were found. However, the last bound state was missing, apparently because the very small difference between  $E_6$  and  $E_7$ .

Table 5: The difference between energy levels of Morse potential obtained by DMRG and by the formula ( $N = 100000$ ,  $h = 0.001$ ,  $m = 9$ ).

Level	Formula	DMRG	$\Delta[10^{-3}]$
$E_0$	6.821067812	6.821526186	0.4
$E_1$	18.963203436	18.965486526	2.2
$E_2$	29.105339059	29.111083114	5.7
$E_3$	37.247474683	37.257029846	9.5
$E_4$	43.389610307	43.401014550	11.4
$E_5$	47.531745931	47.541282695	9.5
$E_6$	49.673881554	49.677919001	4.0
$E_7$	49.816017178	-	-

Figure 8: Energy levels for the Morse potential ( $N = 100000$ ,  $h = 0.001$ ,  $m = 9$ ).

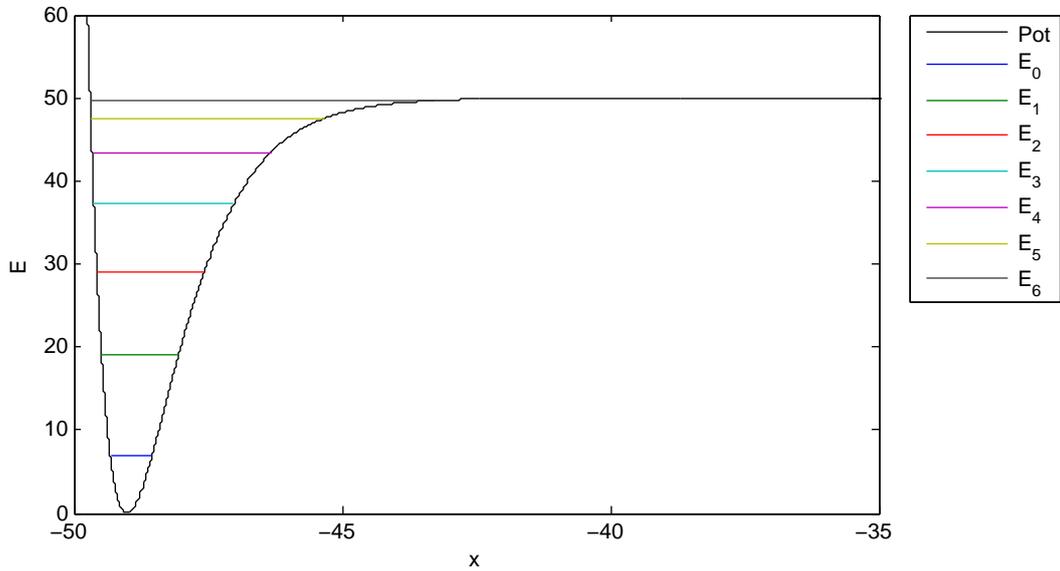
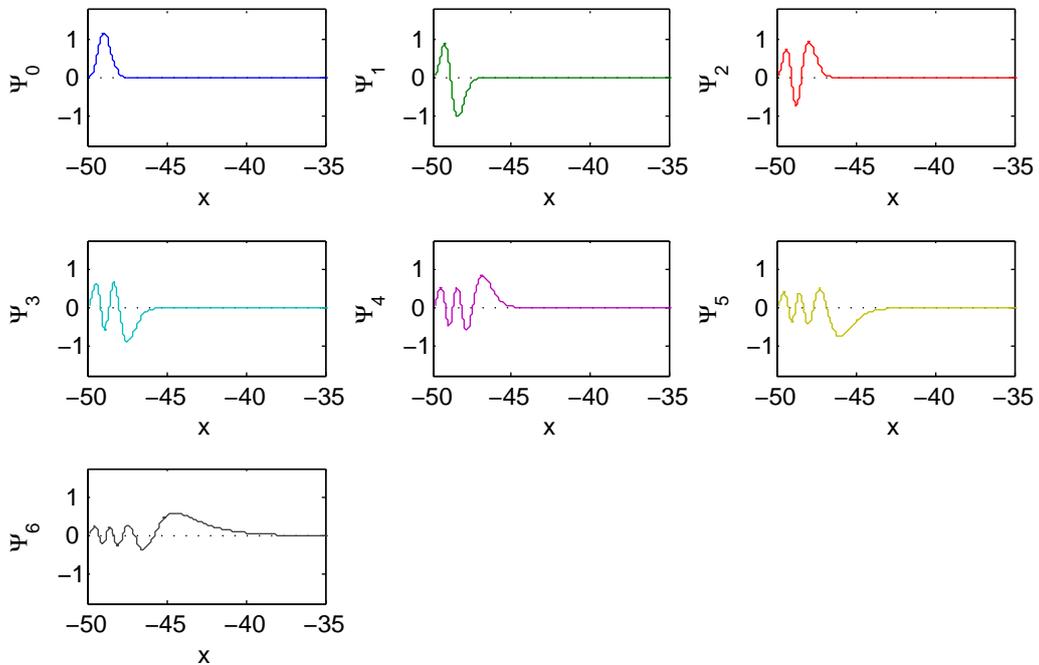


Figure 9: Wavefunctions for the Morse potential ( $N = 100000$ ,  $h = 0.001$ ,  $m = 9$ ).



## 4 The DMRG algorithm applied to the 1-D Bose-Hubbard model

The Bose-Hubbard model describes a chain of lattice sites. The particles in the model are bosons, thus the number of particles located on any site is unlimited. However, for practical purposes it is necessary to limit this number - let  $M$  be the maximum number of particles located on a single lattice site.

A single lattice site could be now in  $M + 1$  different states - there are  $0, 1, \dots, M - 1$  or  $M$  particles on the site. The Hilbert space of a single site is thus  $M + 1$  dimensional space  $\mathcal{H}_0$ . Let  $\hat{n}$  be the particle number operator of a site. The eigenvectors of  $\hat{n}$  form a basis  $\Phi$  of  $\mathcal{H}_0$ :

$$\hat{n}\phi_i = n_i\phi_i, \quad \phi_i \in \Phi, \quad (59)$$

where  $n_i$  is a number of particles in state  $\phi_i$ . All the following matrix operators are written in basis  $\Phi$ .

For the construction of Hamiltonian (19) it is necessary to construct annihilation operator  $\hat{b}$ , creation operator  $\hat{b}^\dagger$ , and particle number operator  $\hat{n}$  on the lattice site. The matrix forms of these operators -  $B$ ,  $B^\dagger$  and  $N$ , respectively - are

$$B = \begin{pmatrix} 0 & \sqrt{1} & 0 & \dots & 0 \\ 0 & 0 & \sqrt{2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \sqrt{M} \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \quad (60)$$

$$B^\dagger = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ \sqrt{1} & 0 & \dots & 0 & 0 \\ 0 & \sqrt{2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \sqrt{M} & 0 \end{pmatrix}, \quad (61)$$

$$N = B^\dagger B = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & M \end{pmatrix}. \quad (62)$$

Let us first analyze a simple example of just two lattice sites to understand how the matrix form of Hamiltonian (19) is constructed. The system Hilbert space is  $\mathcal{H}_0 \otimes \mathcal{H}_0$ . The Hamiltonians on the first and second sites without interaction between them are:

$$H_1 = \frac{1}{2}UN_1(N_1 - I_1) \otimes I_2, \quad (63)$$

$$H_2 = I_1 \otimes \frac{1}{2}UN_2(N_2 - I_2), \quad (64)$$

where indices 1 and 2 denote operators acting on the first and second site, respectively. Obviously, in this simple case the corresponding matrices are the same for both indices.  $I_1$  and  $I_2$  are unit matrix of dimension  $M + 1$  and  $\otimes$  is the *Kronecker product*. The interaction Hamiltonian is:

$$H_{INT} = -J(B_1^\dagger \otimes B_2 + B_1 \otimes B_2^\dagger). \quad (65)$$

The complete Hamiltonian for the Bose-Hubbard model with two lattice sites is therefore

$$H = H_1 + H_2 + H_{INT}. \quad (66)$$

The matrix Hamiltonian for more lattice sites is constructed analogously.

## 4.1 Superblock Hamiltonian

The DMRG algorithm is based on a division of the system to a left block, a right block and two central lattice sites. In every block the number of states is limited. Only the most relevant states are preserved. For the preserved states it is necessary to store information describing them. The information stored for every block is:

1.  $H_B$  - Hamiltonian acting on the block,
2.  $N_B$  - particle number operator on the block,
3.  $C_B$  - creation operator of right-most site contained in the block (in case of left blocks), or of the left-most site contained in the block (in case of right blocks); this is necessary for construction of the interaction Hamiltonian between a block and an adjacent site.

A single site is also a block. The operators for a single site are known:

$$H_B = \frac{1}{2}UN(N - I), \quad (67)$$

$$N_B = N, \quad (68)$$

$$C_B = B^\dagger. \quad (69)$$

The superblock is constructed in the same way as in example (66). The Hamiltonian without interaction between sites is

$$\begin{aligned} H_{SITE} = & H_L \otimes I_{CL} \otimes I_{CR} \otimes I_R \\ & + I_L \otimes H_{CL} \otimes I_{CR} \otimes I_R \\ & + I_L \otimes I_{CL} \otimes H_{CR} \otimes I_R \\ & + I_L \otimes I_{CL} \otimes I_{CR} \otimes H_R, \end{aligned} \quad (70)$$

where  $H_L, H_R, H_{CL}$  and  $H_{CR}$  are stored Hamiltonians for the left block, right block, central left site and central right site;  $I_L, I_R, I_{CL}$  and  $I_{CR}$  are unit matrices on respective sites. The

interaction Hamiltonian is

$$\begin{aligned}
H_{INT} = & \left[ -J \left( C_L \otimes C_{CL}^\dagger + C_L^\dagger \otimes C_{CL} \right) \right] \otimes I_{CR} \otimes I_R \\
& + I_L \otimes \left[ -J \left( C_{CL} \otimes C_{CR}^\dagger + C_{CL}^\dagger \otimes C_{CR} \right) \right] \otimes I_R \\
& + I_L \otimes I_{CL} \otimes \left[ -J \left( C_{CR} \otimes C_R^\dagger + C_{CR}^\dagger \otimes C_R \right) \right]
\end{aligned} \tag{71}$$

The complete Hamiltonian on the whole lattice is therefore

$$H = H_{SITE} + H_{INT} \tag{72}$$

The number of particles  $n$  located on the whole lattice is usually specified beforehand. Only some states defined on the product space contain the desired number of particles. Thus it is possible to reduce the Hilbert space and Hamiltonian  $H$ . The particle number operator on the whole lattice is

$$\begin{aligned}
N = & N_L \otimes I_{CL} \otimes I_{CR} \otimes I_R \\
& + I_L \otimes N_{CL} \otimes I_{CR} \otimes I_R \\
& + I_L \otimes I_{CL} \otimes N_{CR} \otimes I_R \\
& + I_L \otimes I_{CL} \otimes I_{CR} \otimes N_R.
\end{aligned} \tag{73}$$

$N$  is a diagonal matrix in the selected basis  $\Phi$ . Every number  $m_i$  on the diagonal specifies the number of particles in the corresponding product state. Only the states with indices  $j$ , for which  $m_j = n$ , are relevant as only these states contain  $n$  particles. The rows and columns of the Hamiltonian matrix  $H$  with indices  $j$  are selected and the projected Hamiltonian  $H_{PROJ}$  is created. The diagonalization of  $H_{PROJ}$  is much faster than the diagonalization of  $H$ .

In the diagonalization of  $H_{PROJ}$  only the lowest energy state is sought. Obtained state is a ground state on the whole system. The density matrix (23) is constructed, where ‘block’ is the left block and the central left site, and ‘environment’ is the right block and the central right site, or vice versa (depends on the direction of the sweep). The density matrix is diagonalized and eigenstates with the highest eigenvalues (weights) are the most important states to retain in the block.

There are two ways how to select the most important states  $\psi_i^O$ . The first one is to select a fixed number of states  $s_{max}$  every time. That means the number of states that are stored in every block is also  $s_{max}$ . The second way is to store only states with weights greater than some truncation limit  $w_{lim}$ . Both ways have advantages and disadvantages.

The current implementation of the DMRG algorithm combines both strategies. Only the states with weight greater than  $w_{lim}$  are selected. Afterwards, if the number of these states is greater than  $s_{max}$ , only the  $s_{max}$  states with highest weights are stored. This method ensures:

1. The number of states stored in blocks is small as there are usually only a few states with significant weights. Consequently, it makes the algorithm run much faster.

2. The number of states stored in blocks is fixed. The algorithm will not get stuck due to matrix  $H_{PROJ}$  being too large.

Finally, the operators have to be transformed to a new basis consisting of the selected states. Let us describe the situation in the case of a new left block being created from the left block and the central left site (the situation for new right block is analogous).

Let  $m$  be the number of relevant states  $\psi_i^O$ , extracted from density matrix, and  $O$  be the matrix constructed from these states.

$$O = (\psi_1^O, \dots, \psi_m^O), \quad (74)$$

The states  $\psi_i^O$  are states on the product space of the left block space and the central left site space. The block operators on this space are:

$$H_B = H_L \otimes I_{CL} + I_L \otimes H_{CL} + \left[ -J \left( C_L \otimes C_{CL}^\dagger + C_L^\dagger \otimes C_{CL} \right) \right], \quad (75)$$

$$N_B = N_L \otimes I_{CL} + I_L \otimes N_{CL}, \quad (76)$$

$$C_B = I_L \otimes C_{CL}. \quad (77)$$

$$(78)$$

These operators are transformed to a new basis using matrix  $O$ :

$$H_{NB} = O^\dagger H_B O, \quad (79)$$

$$N_{NB} = O^\dagger N_B O, \quad (80)$$

$$C_{NB} = O^\dagger C_B O. \quad (81)$$

$$(82)$$

Operator  $H_{NB}$ ,  $N_{NB}$  and  $C_{NB}$  describe the composite block of the left block and the central left site. Thus the new left block is created.

## 4.2 Warmup and sweeps

The warmup and sweeps are analogous to the single particle algorithm described in section (3.4). A small difference is that during the warmup phase the total number of particles  $n$  has to grow according to the actual system size.

The calculation of a “wavefunction” (expected number of particles occupying a particular lattice site) is possible for the Bose-Hubbard model too. It is analogous to the algorithm in section (3.5).

## 4.3 Phase diagram

There are many quantitative properties of the Bose-Hubbard model that can be calculated using the DMRG algorithm. One of them is an exact shape of the phase diagram (Fig. 2).

Let  $N$  be the number of lattice sites. The Mott insulator phase has integer particle density. For example, there are exactly  $N$  particles on the lattice in the Mott insulator case with particle density  $\rho = 1$ . The superfluid phase has a non integer particle density. The border between them could be therefore characterized as a state, where there are  $N - 1$  (*hole*) or  $N + 1$  (*particle*) particles.

For a particular hopping element  $J$  the ground state energy  $E_0$  for a system with  $N$  particles is calculated. The chemical potential

$$\mu = \left( \frac{\partial U}{\partial N} \right)_{S,V} \quad (83)$$

is in fact an increase in energy caused by adding or removing one particle ( $S$  and  $V$  are constant entropy and volume, respectively). The ground state energies  $E_H$  and  $E_P$  for systems with  $N - 1$  and  $N + 1$  particles are calculated. The upper border  $\mu_P$  and the lower border  $\mu_H$  is then obtained as:

$$\mu_P = E_P - E_0, \quad (84)$$

$$\mu_H = E_0 - E_H. \quad (85)$$

The chemical potentials  $\mu_P$  and  $\mu_H$  differ for a system in the insulator phase.  $\mu_P = \mu_H$  for a system in the superfluid phase.

Using this algorithm it is possible to obtain complete phase diagram.

## 4.4 Results

The results of DMRG applied to the Bose-Hubbard model are summarized in the following figures.

Fig. 10 shows the particle density in the Mott insulator phase for different number of particles. If the number of lattice sites and the number of particles is equal, exactly one particle fills exactly one site. The result corresponds to formula (21). Particle wavefunctions are localized. Adding and removing particles creates ‘particles’ and ‘holes’, respectively.

Fig. 11 shows the particle density in the superfluid phase. The difference is apparent: particle wavefunctions are delocalized, every particle wavefunction is spread out over almost entire lattice. The situation corresponds to (21).

The phase diagram in  $J - \mu$  plane is shown in Fig. 12. The diagram has an expected shape as in Fig. 2. If the tunneling term  $J$  is large enough, the system is always in the superfluid phase. For lower values of  $J$  it is possible for the system to be in the superfluid phase.

Figure 10: Particle density of the ground state in the Mott insulator phase ( $J = 0.1, U = 1$ ) for various particle numbers.  $\langle n_i \rangle$  denotes the expectation value for the number of particles at the lattice site  $i$ . Calculation parameters: lattice size  $N = 100$ , number of sweeps  $s = 10$ , maximal number of particles on a site  $n = 4$ , truncation weight limit  $w = 10^{-7}$ , truncation states limit  $m = 60$ .

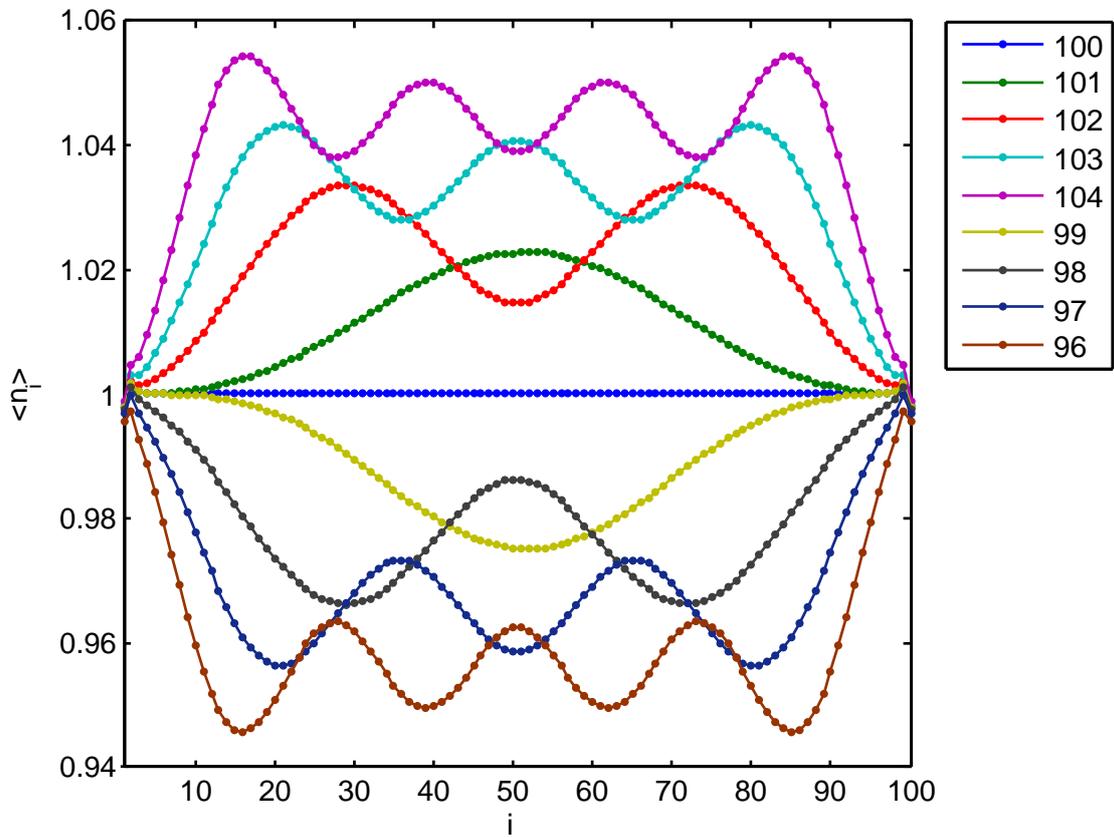


Figure 11: Particle density of the ground state in the superfluid phase ( $J = 10, U = 1$ ) for various particle numbers. Calculation parameters: lattice size  $N = 100$ , number of sweeps  $s = 10$ , maximal number of particles on a site  $n = 4$ , truncation weight limit  $w = 10^{-7}$ , truncation states limit  $m = 60$ .

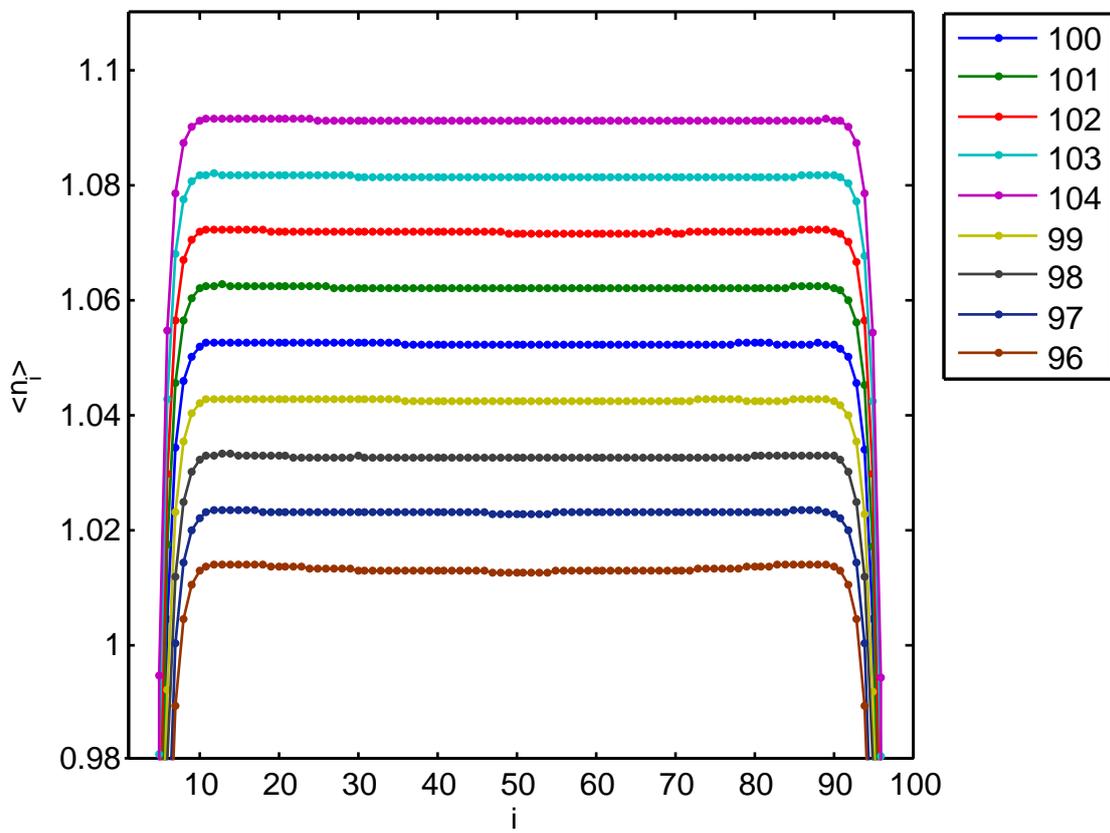
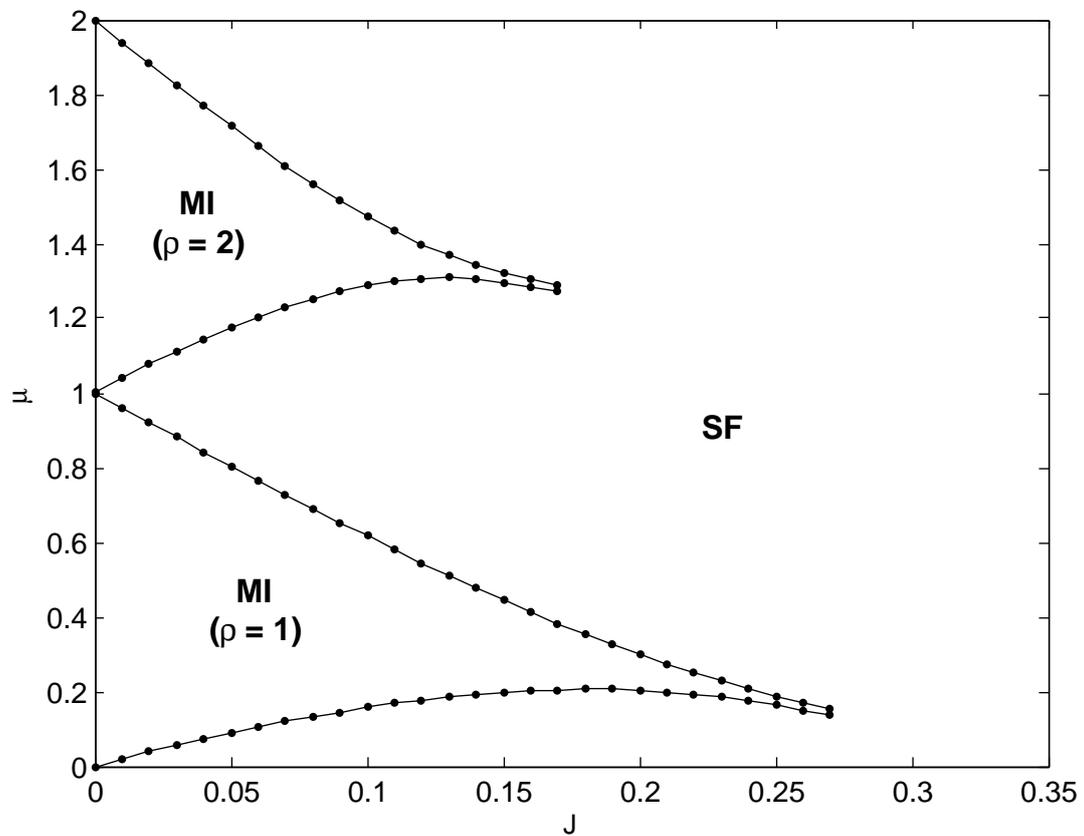


Figure 12: The phase diagram as calculated by DMRG (MI: Mott insulator with density  $\rho$ , SF: Superfluid phase). Calculation parameters: lattice size  $N = 100$ , number of sweeps  $s = 2$ , particle number limit  $n = 4$ , truncation weight limit  $w = 10^{-7}$ , truncation states limit  $m = 60$ .



## Conclusions

The DMRG algorithm was used to calculate the ground state of the Bose-Hubbard model with various parameters. The results are in good correspondence with theoretical predictions. However, the theoretical predictions are possible only for special cases, while DMRG could be applied universally. The potential of the DMRG algorithm is thus quite large.

There are many ways how to extend current work, for example:

1. Design a DMRG-like algorithm for 2D and 3D problems.
2. Parallelization. The current implementation is unparallelized. There could be interesting performance gains by making the algorithm distributed.
3. More complex model. Bose-Hubbard model is a simple model with one particle species. The model could be extended to include more particles species and more complicated interactions.

## References

- [1] C. J. Pethic and H. Smith: *Bose-Einstein Condensate in Dilute Gases* (Cambridge University Press, Cambridge, England, 2001).
- [2] R. Fitzpatrick: *Quantum Mechanics: A graduate level course* (Lecture notes, <http://farside.ph.utexas.edu/teaching/qm/qm.html>).
- [3] A. M. Rey *Ultracold bosonic atoms in optical lattices*, 2004, Dissertation at the University of Maryland.
- [4] D. Jaksch, C. Bruder, C. W. Gardiner, J. I. Cirac and P. Zoller, *Cold bosonic atoms in optical lattices*, 1998, Phys. Rev. Lett. **81**, 3108.
- [5] M. Greiner, O. Mandel, T. Esslinger, T. W. Hänsch, and I. Bloch, 2002, Nature **415**, 39.
- [6] M. A. Martín-Delgado, G. Sierra and R. M. Noack: *The density matrix renormalization group applied to single-particle quantum mechanics*, 1999, J. Phys. A **32**, 6079.
- [7] J. R. Laguna: *Real Space Renormalization Group Techniques and Applications*, 2002, cond-mat/0207340.
- [8] P. M. Morse: *Diatomic molecules according to the wave mechanics. II. Vibrational levels*, 1929, Phys. Rev. **34**, 57.
- [9] S. R. White: *Density matrix algorithms for quantum renormalization groups*, 1993, Phys. Rev. B **48**, 14.
- [10] T. D. Kühner, S. R. White and H. Monien: *One-dimensional Bose-Hubbard model with nearest-neighbor interaction*, 2000, Phys. Rev. B **61**, 18.

## Appendix

This section contains the source code of the DMRG algorithm for the Bose-Hubbard model implementation in MATLAB. The function `dmrgbh` does the actual calculation and the scripts `dens` and `phase` calculate the density and phase diagram, respectively.

### File `createsite.m`:

```
function S = createsite (maxpart)

%
% Creates one site
%

global U;

X = 1:maxpart;
I = eye(maxpart+1);

C = diag(sqrt(X),-1);
N = C*C';
H = 1/2*U*N*(N-I);

S.m = maxpart+1;
S.NTOT = diag(N,0);
S.C = C;
S.H = H;
S.N = N;
```

### File `superblock.m`:

```
function [G,E] = superblock (BL,SL,SR,BR,ntotal)

%
% Creates the superblock and finds its eigenvalues and eigenvectors.
%

global J;

IBL = speye(BL.m);
ISL = speye(SL.m);
ISR = speye(SR.m);
IBR = speye(BR.m);

H1 = sparse (-J*(kron(BL.C,SL.C') + kron(BL.C',SL.C)));
H2 = sparse (-J*(kron(SL.C,SR.C') + kron(SL.C',SR.C)));
H3 = sparse (-J*(kron(SR.C,BR.C') + kron(SR.C',BR.C)));

HSITE = kron(BL.H,kron(ISL,kron(ISR,IBR))) + kron(IBL,kron(SL.H,kron(ISR,IBR))) + kron(IBL,kron(ISL,kron(SR.H,IBR))) +
kron(IBL,kron(ISL,kron(ISR,IBR,H)));
HINT = kron(H1,kron(ISR,IBR)) + kron(IBL,kron(H2,IBR)) + kron(IBL,kron(ISL,H3));
HSB = HSITE + HINT;

if ntotal == -1
    H = HSB;
else
    JBL = ones(BL.m,1);
    JSL = ones(SL.m,1);
    JSR = ones(SR.m,1);
    JBR = ones(BR.m,1);
    NTOT = round(kron(BL.NTOT,kron(JSL,kron(JSR,JBR))) + kron(JBL,kron(SL.NTOT,kron(JSR,JBR))) + kron(JBL,kron(JSL,kron(
(SR.NTOT,JBR))) + kron(JBL,kron(JSL,kron(JSR,BR.NTOT)))));
    IND = find(NTOT == ntotal);
    H = HSB(IND,IND);
end;

H = (H + H')/2;    % ensures that the superblock is fully symmetric (round-off error)

opt.issym=1;
opt.isreal=1;
opt.disp=0;
%[V,D] = eig(full(H));          % various matrix diagonalization routines
%[V,D] = eigs(H,1,'sa',opt);
%[V,D] = jdcg(H,1,opt);
[V,D] = jdqr(H,1,'sr',opt);

if ntotal == -1
    G = V(:,1);
else
    G = zeros(BL.m*SL.m*SR.m*BR.m,1);
```

```

    G(IND) = V(:,1);
end;

E = D(1,1);

```

## File getleft.m:

```

function BLO = getleft (BL,SL,SR,BR,G)

global J truncmax trunclimit;

m1 = BL.m*SL.m;
mr = BR.m*SR.m;
DM = zeros(m1,m1);
for i = 1:m1
    for j = 1:m1
        V1 = G((i-1)*mr+1:i*mr);
        V2 = G((j-1)*mr+1:j*mr);
        DM(i,j) = V1'*V2;
    end
end
DM = (DM + DM')/2;

JBL = ones(BL.m,1);
JSL = ones(SL.m,1);
NTOT = round(kron(BL.NTOT,JSL) + kron(JBL,SL.NTOT));

V = [];
D = [];
S = [];

for I = unique (NTOT')
    IND = find (NTOT == I);
    n = size(IND,1);
    DMX = DM(IND,IND);
    [VO,DO] = eig(DMX);
    VX = zeros(m1,n);
    VX(IND,:) = VO;
    DX = diag(DO,0);
    SX = I*ones(n,1);
    V = [V,VX];
    D = [D;DX];
    S = [S;SX];
end;

[D,IX] = sort (D,'descend');
V = V(:,IX);
S = S(IX);

IND = find (D > trunclimit);
n = size(IND,1);

if (n <= truncmax)
    BLO.m = n;
    BLO.NTOT = S(IND);
    O = V(:,IND);
else
    BLO.m = truncmax;
    BLO.NTOT = S(1:truncmax);
    O = V(:,1:truncmax);
end;

IBL = eye(BL.m);
ISL = eye(SL.m);

C = kron(IBL,SL.C);
BLO.C = O'*C*O;

H = kron(BL.H,ISL) + kron(IBL,SL.H) - J*(kron(BL.C,SL.C') + kron(BL.C',SL.C));
BLO.H = O'*H*O;

N = kron(BL.N,ISL) + kron(IBL,SL.N);
BLO.N = O'*N*O;

```

## File getright.m:

```

function BRO = getright (BL,SL,SR,BR,G)

global J truncmax trunclimit;

m1 = BL.m*SL.m;
mr = BR.m*SR.m;

```

```

DM = zeros(mr);
for i = 1:mr
    for j = 1:mr
        V1 = G(i:mr:end);
        V2 = G(j:mr:end);
        DM(i,j) = V1'*V2;
    end
end
end

DM = (DM + DM')/2;

JSR = ones(SR.m,1);
JBR = ones(BR.m,1);
NTOT = round(kron(SR.NTOT,JBR) + kron(JSR,BR.NTOT));

V = [];
D = [];
S = [];

for I = unique (NTOT')
    IND = find (NTOT == I);
    n = size(IND,1);
    DMX = DM(IND,IND);
    [VO,DO] = eig(DMX);
    VX = zeros(mr,n);
    VX(IND,:) = VO;
    DX = diag(DO,0);
    SX = I*ones(n,1);
    V = [V,VX];
    D = [D;DX];
    S = [S;SX];
end;

[D,IX] = sort (D,'descend');
V = V(:,IX);
S = S(IX);

IND = find (D > truncclimit);
n = size(IND,1);

if (n <= truncmax)
    BRO.m = n;
    BRO.NTOT = S(IND);
    O = V(:,IND);
else
    BRO.m = truncmax;
    BRO.NTOT = S(1:truncmax);
    O = V(:,1:truncmax);
end;

IBR = eye(BR.m);
ISR = eye(SR.m);

C = kron(SR.C,IBR);
BRO.C = O'*C*O;

H = kron(ISR,BR.H) + kron(SR.H,IBR) - J*(kron(SR.C,BR.C') + kron(SR.C',BR.C));
BRO.H = O'*H*O;

N = kron(SR.N,IBR) + kron(ISR,BR.N);
BRO.N = O'*N*O;

```

## File warmup.m:

```

function [L,R] = warmup (N,L,R,S,rho,extra)

for j = 2:N/2-1

    if rho == -1
        ntotal = -1;
    else
        ntotal = round(2*j*rho+extra);
    end;

    G = superblock (L(j-1),S,S,R(j-1),ntotal);
    L(j) = getleft (L(j-1),S,S,R(j-1),G);
    R(j) = getright (L(j-1),S,S,R(j-1),G);
end;

```

## File sweep.m:

```

function [L,R] = sweep (N,L,R,S,ntotal)

```

```

for j = N/2:N-3
    G = superblock (L(j-1),S,S,R(N-j-1),ntotal);
    L(j) = getleft (L(j-1),S,S,R(N-j-1),G);
end;

for j = N-2:-1:3
    G = superblock (L(j-1),S,S,R(N-j-1),ntotal);
    R(N-j) = getright (L(j-1),S,S,R(N-j-1),G);
end;

for j = 2:N/2-1
    G = superblock (L(j-1),S,S,R(N-j-1),ntotal);
    L(j) = getleft (L(j-1),S,S,R(N-j-1),G);
end;

```

## File meas.m:

```

function [E,NPART] = meas (N,L,R,S,ntotal)

BL = L(N/2-1);
SL = S;
SR = S;
BR = R(N/2-1);

IBL = speye(BL.m);
IBR = speye(BR.m);
ISL = speye(SL.m);
ISR = speye(SR.m);

[G,E] = superblock (BL,SL,SR,BR,ntotal);

NO = kron(BL.N,kron(ISL,kron(ISR,IBR))) + kron(IBL,kron(SL.N,kron(ISR,IBR))) + kron(IBL,kron(ISL,kron(SR.N,IBR))) + kron
    (IBL,kron(ISL,kron(ISR,BR.N)));
NPART = G'*NO*G;

```

## File measdens.m:

```

function DENS = measdens (N,L,R,S,ntotal)

DENS = zeros(N,1);

for j = 2:N-2

    BL = L(j-1);
    SL = S;
    SR = S;
    BR = R(N-j-1);

    IBL = speye(BL.m);
    IBR = speye(BR.m);
    ISL = speye(SL.m);
    ISR = speye(SR.m);

    G = superblock (BL,SL,SR,BR,ntotal);

    if j == 2
        NO = kron(BL.N,kron(ISL,kron(ISR,IBR)));
        NP = G'*NO*G;
        DENS(1) = NP;
    end;

    if j == N-2
        NO = kron(IBL,kron(ISL,kron(SR.N,IBR)));
        NP = G'*NO*G;
        DENS(N-1) = NP;

        NO = kron(IBL,kron(ISL,kron(ISR,BR.N)));
        NP = G'*NO*G;
        DENS(N) = NP;
    end;

    NO = kron(IBL,kron(SL.N,kron(ISR,IBR)));
    NP = G'*NO*G;
    DENS(j) = NP;
end;

```

## File dmrgbh:

```

function [E,NPART,DENS] = dmrgbh (J0,U0,N,rho,extra,truncmax0,trunclimit0,sweeps,mdens)

```

```

%
% DMRG for the Bose-Hubbard model
%
% J0 - kinetic energy parameter
% U0 - interaction energy parameter
% N - number of lattice sites
% rho - lattice filling
% extra - extra particles to filling (particle = +1, hole = -1, etc..)
% truncmax0 - maximum number of states retained in the DMRG
% trunclimit0 - truncation weight limit in the DMRG
% sweeps - number of sweeps
% mdens - whether calculate particle density (true/false)
%

global J U truncmax trunclimit;

J = J0;
U = U0;
truncmax = truncmax0;
trunclimit = trunclimit0;

if rho == -1
    ntotal = -1;
else
    ntotal = round(N*rho+extra);
end;

S = createsite(4);

L(1) = S;
R(1) = S;

disp('Warmup ...');
[L,R] = warmup (N,L,R,S,rho,extra);
for i = 1:sweeps
    disp(['Sweep ' num2str(i) ' ...']);
    [L,R] = sweep (N,L,R,S,ntotal);
end;

[E,NPART] = meas (N,L,R,S,ntotal);

if mdens
    DENS = measdens (N,L,R,S,ntotal);
else
    DENS = 0;
end;

```

## File dens.m:

```

%
% Particle density
%

clear all hidden;

J = 0.1;
U = 1;
N = 100;
truncmax = 60;
trunclimit = 10e-7;
sweeps = 5;

rho = [1;1.01;1.02;1.03;1.04;0.99;0.98;0.97;0.96];
I = 1:size(rho,1);

for i = I
    [E,NPART,DENS] = dmrgbh(J,U,N,rho(i),0,truncmax,trunclimit,sweeps,true);
    DATA(:,i) = DENS;
end;

```

## File phase.m:

```

%
% Phase diagram
%

clear all hidden;

rho = 1;

N = 100;
truncmax = 60;

```

```
trunclimit = 10e-7;
sweeps = 2;

start = 0.00;
step = 0.01;
stop = 0.30;

X = start:step:stop;

for i = X
    E1 = dmrgbh(i,1,N,rho,0,truncmax,trunclimit,sweeps,false);
    E2 = dmrgbh(i,1,N,rho,-1,truncmax,trunclimit,sweeps,false);
    E3 = dmrgbh(i,1,N,rho,+1,truncmax,trunclimit,sweeps,false);

    DATA(round((i-start)/step+1),:) = [E1-E2, E3-E1];
end;
```