Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator

(revised version: December 2011)

by

Boris N. Khoromskij, and Ivan V. Oseledets

Preprint no.: 69 2010

DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator [*]

B.N. Khoromskij [†] I.V. Oseledets [‡]

November 17, 2011

Abstract

In this paper we discuss how to combine two approaches: a Quantized Tensor Train (QTT) model and an advanced optimization method Density Matrix Renormalization Group (DMRG) to obtain efficient numerical algorithms for high-dimensional eigenvalue problems arising in quantum molecular computations. The QTT-format is used to approximate molecular Schrödinger operators. Theoretical estimates on the TT/QTT-ranks are provided for the case of multivariate polynomial potential energy surfaces (PES).

The numerical experiments are presented for the approximation of a 6-dimensional PES of a HONO molecule, as well as for the computation of the ground state of Henon-Heiles Hamiltonian with a large number of degrees of freedom up to 256.

AMS Subject Classification: 65F30, 65F50, 65N35, 65F10
Key words: QTT, DMRG, TT-format, quantum molecular computations

1 Introduction

In the recent years essential progress has been achieved in the numerical solution of high-dimensional problems in scientific computing by using different dimensionality and model reduction techniques. This paper focuses on the numerical solution of the so-called molecular Schrödinger equation, which describes the vibrational behaviour of the nuclei.

Several low-parametric formats were successfully used to represent multidimensional solutions to such equations, for example see the book [32]. These formats include a multi-configurational time-dependent Hartree (MCTDH) method proposed by Meyer et. al in 1990 [36, 35] and its hierarchical, cascadic or multilayer versions [33, 32], with which particular systems up to 500 degrees of freedom have been treated. However, the complicated hierarchical structure of such methods limits their wide applicability, and the original MCTDH method, which has exponential scaling in the dimension is still preferable for systems with moderate number of degrees of freedom [35].

[*] Supported by RFBR grants 08-01-00115, 09-01-00565, 09-01-12058 and RFBR/DFG grant 09-01-91332, by Rus. Gov. Contracts П940, П1178, П1112, by Rus. President grant MK-140.2011.1, by Priority Research Program OMN-3; Partially supported by DFG-SPP 1324, Germany. Part of this work was performed during the stay of the second author in Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig

[†] Max-Planck-Institut für Mathematik in den Naturwissenschaften, Inselstr. 22-26, D-04103 Leipzig, Germany. (bokh@mis.mpg.de)

[‡] Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8, Moscow (ivan.oseledets@gmail.com, http://spring.inm.ras.ru/osel)

If only a few lowest eigenvalues and eigenvectors of a molecular Hamiltonian are required, a vibrational self-consistent field iteration (*VSCF*) [6, 2, 3, 4, 5, 15, 16], which relies on rank-1 approximation to the wavefunction[1] , is usually used. VSCF and its enhancements are implemented in many state-of-art quantum chemistry software packages.

On the other hand, a great development was made in the modelling of quantum spin many-body systems. The so-called density matrix renormalization group (DMRG) [51, 52, 45] is a numerical variational technique devised to obtain the low energy physics of quantum many-body systems with high accuracy. It was invented in 1992 by Steven R. White and it is nowadays the most efficient method for 1-dimensional quantum systems, but its generalization to multidimensional case is still an open question. It was then noticed, that DMRG is a minimization method for the Rayleigh quotient in the Matrix Product States (MPS) [45], which also arise in the study of entanglement in quantum systems. The "meeting" of these two communities (the one that uses DMRG and the quantum information community) has led to deeper understanding of the role of the matrix product states in the modelling of quantum many-body systems. Also, several results were obtained by applying more complex tensor networks to two- and three- dimensional quantum many-body systems [50, 14].

In all above mentioned problems the main computational task is the efficient low-complexity representation of large multidimensional data arrays, commonly named tensors in the numerical multilinear algebra community, see a recent review [30]. The main problems considered in this community are related to the efficient computations of *the canonical decomposition* of a tensor or *the Tucker decomposition* of it. These two decompositions has known drawbacks: the canonical decomposition can not be computed via robust algorithms and complexity of the Tucker decomposition has exponential dependence from the dimension, which make them computationally infeasible for large-scale applications. However, in the case of moderate spacial dimension the proper combination of both models leads to efficient tensor-structured numerical methods [21, 25, 26, 19, 18].

In 2009 an active development of *new MPS-type tensor formats* began. The first attempt used the hierarchical splitting of dimensions (see the hierarchical-type dimension splitting in [20], the $\mathcal{H}$-Tucker format [12, 10] and the Tree-Tucker format [43, 38]), which are free from the curse of dimensionality, and what is important, basic MLA operations can be implemented in $\mathcal{O}(d)$ cost, so the general approach of tensor-structured iterations can be used. However, recursive structures are not always easy to implement, and it was found that the simplest choice of the tree, that leads to the so-called *Tensor Train (TT) format* [39, 42] is usually preferable: it retains the same complexity and leads to much simpler algorithms. For more detailed discussion on theoretical aspects of $\mathcal{H}$-Tucker and TT formats, see [11, 7].

One of the main advantages of the TT-format is the existence of a fast and robust *rounding procedure* with guaranteed accuracy, which is based on standard SVD and QR decompositions. The existence of the robust compression tool allowed, first experimentally, to find hidden tensor structures in unusual cases. For example, if a vector of length $n = 2^L$, $\{v_k\}$, comprised from values of regular function g on some grid x_k (i.e., $v_k = g(x_k)$) is considered as a $2 \times 2 \times \ldots \times 2$ L-dimensional tensor (this can be called the quantization of a vector) then in many cases it has surprisingly low TT-ranks [40, 23]. Generalization of this idea to the functions of f variables leads to the log-volume complexity to represent certain high-order $n \times n \times \ldots \times n$ f-dimensional function-related tensors. This format is called the *QTT-format.*

[1] In this paper the computation of vibrational states is considered thus no antisymmetry/symmetry conditions are imposed on the solution. The adaptation of the TT-format to the antisymmetric case can be in principle considered, but this is beyond the scope of this paper.

Recently, new theoretical estimates were obtained for QTT-ranks of certain classes of multivariate functions, but still the gap between theoretical and experimental results is large. Therefore, in the QTT-approach, the high-dimensional tensor, representing a function on $n \times n \times \ldots \times n$ tensor grid is transformed into a $L \log n$-dimensional tensor with mode sizes 2, and then represented (approximated) in the TT-format. The resulting approximation is very close in form to the MPS states used in the DMRG. The connection between TT and DMRG was discussed by R. Schneider et. al., see [13, 48].

It is worth to note, that the QTT-format was already successfully applied to solving high-dimensional elliptic equations [28], multiparametric SPDEs [27], and in electronic structure calculation [18] and the log-volume complexity was confirmed numerically and theoretically [9], so it is a natural idea to apply this format to other high-dimensional problems, in particular to the spectral problem that arises in the computation of the vibrational ground state. We refer to survey and lecture notes [24, 22] addressing the recent advances in tensor numerical methods.

We will consider the computation of the lowest eigenpair of the molecular Hamiltonian, which has the form

$$H\psi = (-\frac{1}{2}\Delta + V)\psi = E\psi. \tag{1.1}$$

The equation (1.1) will be solved in the QTT-format.

Note, that one of the bottlenecks in the quantum molecular computations is the representation of the potential V in high dimensions. The QTT-model is a very promising candidate, and moreover when no analytical representation is available, one can use a TT-cross approximation approach [44] (see also a related work [1]), which interpolates a high-dimensional function on the adaptively chosen points, and if a low TT-rank approximation exists, then such approximation is guaranteed to recover the full potential. In this paper, we confine ourselves to a class of potentials represented analytically, where in several cases (for example, for Henon-Heiles potentials considered in [37] , generic cubic and quadrics potentials, etc.) we will obtain estimates for the QTT-ranks.

Suppose now, that V is approximated in the QTT-format. How to compute the wavefunction? Now is the main message of the paper. Let us note that the MPS used in the DMRG are closely related to the QTT representations, and since the DMRG algorithm has shown its effectiveness for calculating the ground-state wavefunctions and energies of certain spin Hamiltonians, it turns out that after the transformation of the Hamiltonian in (1.1) to the QTT-format, the DMRG approach can be adapted to find the ground state wavefunction.

The combination of the effective variational approach, based on the DMRG in the QTT-format and well-known numerical algorithms creates a new tool for dealing with high-dimensional quantum molecular problems.

The rest of the paper is organized as follows. In Section 2 the formulation of the problem is given, while in Section 3 the description of TT/QTT tensor formats is presented. Section 4 contains theoretical analysis on the tensor structure for matrices arising in the discretization process. Section 5 describes the numerical scheme for the solution of eigenvalue problem by the DMRG iteration. Numerical experiments are presented in Section 6, and directions for the future work are outlined in Section 7.

2 Formulation of the problem

We are going to solve an eigenvalue problem

$$H\psi = E\psi, \quad \psi = \psi(q_1, \ldots, q_f), \quad q_k \in \mathbb{R}, k = 1, \ldots, f, \tag{2.1}$$

where H is a molecular Schrödinger operator

$$H = -\frac{1}{2}\Delta + V,$$

Δ is a f-dimensional Laplace operator, $V = V(q_1, \dots, q_f)$ is the so-called *potential energy surface* (PES). This equation is describing the vibrational motion of molecules. The variables $q_1, \dots, q_f$ are the degrees of freedom (for example, the coordinates of atoms in a molecule) and the values of V should be obtained from the solution of the *electronic Schrödinger equation* using any reliable electronic structure calculation method. An efficient representation of the PES is computationally difficult but important problem. A lot of successful approaches exist for the particular molecules (see, for example, [31, 34, 46]), which are based on the interpolation of multivariate functions.

In this paper we assume, that V is given in some low-parametric form that we can work with. A simple, but practically important approximation to V is based on normal coordinates. Suppose the molecule contains N atoms, and initially, the molecular Schrödinger equation is formulated in $3N$ natural coordinates. When the molecule has no additional symmetries, the number of independent degrees of freedom is equal to $3N - 6$, due to the translational and rotational symmetry of a molecule as a rigid body. The equilibrium state of the molecule corresponds to the minimum of the potential $V(r_1, \dots, r_{3N})$. A classical approach to study the vibrations of the molecule is to expand V around the equilibrium into Taylor series. The first term vanishes since it is an equilibrium:

$$V \approx V^* + \sum_{i,j=1}^{3N} \frac{\partial^2 V}{\partial r_i \partial r_j} \Delta r_i \Delta r_j + \dots. \tag{2.2}$$

If the Taylor series are truncated at the second-order terms then a *harmonic approximation* is obtained. Eigenvalues and eigenfunctions of the corresponding Hamiltonian can be computed analytically from the eigenvalues of the Hessian matrix $\frac{\partial^2 V}{\partial r_i \partial r_j}$. These eigenvalues are called *vibrational frequencies.* Despite being very simple, the harmonic approximation often gives a very good agreement with the true eigenvalues and experimental data. However there are molecules where the PES is not well-approximated by a polynomial of second order. One can improve the approximation by considering high-order terms in the Taylor series: 4-th order polynomials, 6-th order, etc. For example, the vibrational ground state of a protein BPTI [49] was computed using a 4-th order polynomial. Thus, case of a *polynomial potential* is very important.

The eigenvalues of the Hessian of V give the vibrational frequencies, whereas its eigenvectors provide a linear transformation of the independent variables into *normal coordinates.* A second-order polynomial in the normal coordinates becomes a sum of squares:

$$V = \sum_{k=1}^{f} w_k^2 q_k^2,$$

where w_k are the vibrational frequencies, and f is the number of degrees of freedom, which is defined as the number of non-zero eigenvalues of the Hessian of V. The transformation into the normal coordinates is often very useful and gives wavefunctions with better structure, and it is a good idea to work with the PES in the normal coordinates.

Equation (2.1) lacks boundary conditions, however in the normal coordinates it is usually sufficient to impose Dirichlet boundary conditions, i.e. we work in the coordinates q_k and suppose that $\psi(q_1, \dots, q_f) \to 0$ uniformly, as $|q_k| \to \infty$. Then, the computational domain can be chosen to be a f-dimensional cube. In this cube, a uniform tensor grid with n points in each direction is introduced, and the unknowns will be the values of ψ on this grid. The total number of unknowns is equal to n^f.

Now consider the discretization of the Hamiltonian. The multiplication by V reduces to the multiplication by a diagonal matrix. For the Laplace operator, the simplest finite-difference approximation is used. It yields a matrix $\Delta_{(f)}$ which can be written as

$$\Delta_{(f)} = \Delta_1 \otimes I \otimes \dots \otimes I + \dots + I \otimes \dots \otimes \Delta_f, \tag{2.3}$$

where Δ_i, $i = 1, \dots, f$, are the discretizations of a one-dimensional Laplace operator with Dirichlet boundary conditions, and $\otimes$ is a tensor (Kronecker) product of matrices. In the simplest case of uniform grid, we have $\Delta_i = \gamma \mathrm{tridiag}[-1, 2, -1]$.

3 TT and QTT solution ansatz

Usually, some iterative method is used to solve the obtained eigenvalue problem. However, if an eigenvector is stored as a full vector, it requires n^f memory cells, which becomes too large even for moderate f, and one has to approximate the solution somehow. The eigenvector ψ on a tensor grid in $\mathbb{R}^f$ can be naturally considered as a f-dimensional array Ψ with elements

$$\Psi(i_1, \dots, i_f) = \psi(x_1(i_1), \dots, x_f(i_f)), \quad 1 \le i_k \le n,$$

where $x_k(i_k)$ are the nodes on one-dimensional grids chosen for the discretization. This multidimensional array (tensor) has to be approximated by some low-parametric representation, and in order to do that, we will use the QTT nonlinear approximation scheme.

We will assume that the number of grid points in one dimension is a power of $2, n = 2^L$. In this case, the QTT-format consists in the *quantization* of the array Ψ. It proceeds as follows. An f-dimensional array with mode sizes 2^L can be naturally treated as a Lf-dimensional array Φ with mode sizes 2. Each "one-dimensional" mode index i_k is represented by its binary digits:

$$i_k = \sum_{s=0}^{L-1} 2^s j_{ks}, \quad k = 1, \dots, f,$$

where j_{kl} take values 0 and 1.

Then, the element of Φ in position $(j_{11}, j_{12}, \dots, j_{1L}, j_{2L}, \dots, j_{f1}, \dots j_{fL})$ is equal to $\Psi(i_1, \dots, i_f)$. Maybe the simplest way to describe this process is in terms of the MATLAB **reshape** function:

$$\Phi = \texttt{reshape}(\Psi, \texttt{2*ones(1,L*f)}).$$

The elements of Ψ can be indexed by a Lf-tuple $i_{ks}, k = 1, \dots, L,\ s = 1, \dots, f$, where the index s corresponds to the initial degree of freedom, and the index k — to the introduced "virtual dimensions". Such tensorization allows the application of high-dimensional tensor techniques even for one-dimensional objects (i.e. values of a univariate function on a grid with 2^L points), and that lead to the reduced complexity and storage. The main format we will use is the TT-approximation of the quantized tensor. For a general tensor $\mathbf{A}$ with elements $A(i_1, \dots, i_d)$ the TT-decomposition is defined as a representation of form

$$A(i_1, \dots, i_d) = G_1(i_1) G_2(i_2) \dots G_d(i_d), \tag{3.1}$$

where $G_k(i_k)$ is a $r_{k-1} \times r_k$ matrix depending on the integer parameter i_k, and $r_0 = r_d = 1$. Elements of the matrices $G_k(i_k)$ can be put into a three-dimensional array with elements $G_k(\alpha_{k-1}, i_k, \alpha_k)$ [2] where α_k vary from 1 to r_k, and (3.1) can be written in the index form

$$A(i_1, \ldots, i_d) = \sum_{\alpha_0,\ldots,\alpha_d} G_1(\alpha_0, i_1, \alpha_1) G_1(\alpha_1, i_2, \alpha_2) \ldots G_p(\alpha_{d-1}, i_d, \alpha_d). \tag{3.2}$$

The TT-format comes with all basic MLA operations that can be implemented in linear in d and polynomial in $r = \max r_k$ complexity, thus if the (approximate) TT-ranks are small, then it gives a low-parametric approximation to the solution. Moreover, the TT-format has a robust *rounding procedure*, which allows fast reduction of the TT-ranks, while maintaining the accuracy.

The solution of the f-dimensional problem can be naturally associated with a f-dimensional tensor (in the TT-case) and with a Lf-dimensional tensor in the QTT-case, and the resulting tensor can be approximated by a tensor in the TT-format. But what about matrices? By a "vector in TT-format" we implicitly assume, that such vector has length $N = n_1 \cdot \ldots \cdot n_d$ and can be associated with a d-dimensional array. A square matrix M, acting on such vector, must have the size $N \times N$. Then the elements of M can be naturally indexed by a $2d$-tuple $(i_1, \ldots, i_d), (j_1, \ldots, j_d)$, where the multiindex $(i_1, \ldots, i_d)$ enumerates the rows of M, and $(j_1, \ldots, j_d)$ — the columns of M. The matrix M is said to be in the TT-format, if

$$M(i_1, \ldots, i_d; j_1, \ldots, j_d) = M_1(i_1, j_1) \ldots M_d(i_d, j_d), \tag{3.3}$$

where $M_k(i_k, j_k)$, $k = 1, \ldots, d$ are $m_{k-1} \times m_k$ matrices, $m_0 = m_d = 1$. This is equivalent to the permutation of the dimensions of a $2d$-dimensional tensor M and treating (i_k, j_k) as a long index. Such permutation is standard in the compression of high-dimensional operators, and is motivated by the observation, that for $m_k = 1$ the TT-format turns into a tensor (Kronecker) product of matrices:

$$M = M_1 \otimes M_2 \otimes \ldots \otimes M_p,$$

which is a direct generalization of a rank-1 tensor to the matrix case. To be able to do computations with vectors in TT(QTT) formats, the matrix itself (in our case, the Hamiltonian H) has to be transformed into the TT(QTT) format. In this case the basic operation, the matrix-by-vector product, can be implemented in terms of cores of a matrix and a vector in complexity, that is linear in the dimension. So the first step is the representation of the specific matrix in the QTT format, which will be used later on for the solution of (2.1).

The potential V in (2.1) can be given in some analytical form, obtained using other methods, i.e. as a polynomial in $q_1, \ldots, q_f$. In this case, the estimation of the TT-ranks can be done on the functional level. Indeed (3.1) can be generalized to the continuous case by replacing the discrete indices i_k by continuous variables x_k. It gives *the functional TT-decomposition* (FTT) [41] which for a function $F = F(x_1, \ldots, x_p)$ has the form

$$F(x_1, \ldots, x_p) \approx G_1(x_1) G_2(x_2) \ldots G_d(x_p), \tag{3.4}$$

where $G_k(x_k)$ is an $r_{k-1} \times r_k$ matrix, that depends on x_k. If a function has low FTT-ranks, then its function-related tensors (i.e. tensors, constructed from the values of this function on a tensor grid) also have low TT-ranks. This will be helpful in deriving theoretical estimates for the TT-ranks in the cases where V is given analytically.

[2] An abuse of notation is present: G_k is used to denote three entities: the parameter-dependent matrix $G_k(i_k)$, the three-dimensional core G_k of the TT-decomposition and the elements of this array $G_k(\alpha_{k-1}, i_k, \alpha_k)$. The precise meaning should be clear from the context

4 Representation of the matrix

4.1 Theoretical rank estimates for the polynomial PES

The discrete Hamiltonian consists of two terms: the discretization of the Laplace operator and the discretization of the potential. First, let us consider the Laplace operator. It allows a TT-decomposition with TT-ranks equal to 2.

Proposition 4.1 *[39, 17] The following rank estimate holds*

$$\operatorname{rank}_{\mathrm{TT}}(\Delta_{(f)}) \leq 2.$$

As for the QTT-ranks, they depend on the one-dimensional discretization. Suppose, a uniform grid is chosen, and Δ_i in (2.3) are

$$\Delta_i = \gamma_i \mathtt{tridiag}[-1, 2, -1].$$

For such case, the QTT-ranks are bounded by 4.

Proposition 4.2 *(Lemma 2.1 in [17]) The following rank bound holds*

$$\operatorname{rank}_{\mathrm{QTT}}(\Delta_{(f)}) \leq 4.$$

The second term in (2.1), the potential V is discretized by the collocation at the grid points, becoming a diagonal matrix. Thus, only TT-ranks of the function V have to be estimated.

For the important case of the polynomial potential, one can obtain the following estimate for the TT-ranks of the corresponding tensors.

Theorem 4.3 *For a general homogeneous polynomial potential of form*

$$V(q_1, \ldots, q_f) = \sum_{i_1, \ldots, i_s = 1}^{f} a(i_1, \ldots, i_s) \prod_{k=1}^{s} q_{i_k},$$

there exists a TT-decomposition with

$$r_k = S(k) = \sum_{j=0}^{s} \min\Big(P(k, s-j), P(f-k, j)\Big), \quad k = 1, \ldots, f-1, \tag{4.1}$$

where

$$P(k, j) = \frac{1}{j!} k(k+1) \cdot \ldots \cdot (k+j-1), \quad P(k, 0) = 1. \tag{4.2}$$

Proof. First note, that without any loss of generality it can be assumed that the coefficient tensor $a(i_1, \ldots, i_s)$ is "triangular", i.e. its element $a(i_1, \ldots, i_s)$ is non-zero only if $i_1 \geq i_2 \geq i_3 \ldots \geq i_s$.

To prove the statement, we will separate q_k one-by-one. This is exactly how the numerical algorithm for the computation of the TT-decomposition (called *TT-SVD*) works [39].

At the first step, q_1 is separated:

$$V = p_1^{(0)} q_1^s + p_1^{(1)} q_1^{s-1} + \ldots + p_1^{(s)}. \tag{4.3}$$

Here $p_1^{(i)}$, $i = 0, \ldots, s$ are homogeneous polynomials of the variables $q_2, \ldots, q_f$ of degree not greater than i. Thus,

$$V = V_1(q_1) \begin{pmatrix} p_1^{(0)} \\ \vdots \\ p_1^{(s)} \end{pmatrix},$$

where $V_1(q_1)$ is a row of length $(s+1)$. Therefore, $r_1 \leq s+1$. At the second step, one can in the analogous way separate the variable q_2 from other variables for each $p_1^{(i)}$:

$$p_1^{(i)} = H_{i2}(q_2) \begin{pmatrix} p_{2i}^{(0)} \\ \vdots \\ p_{2i}^{(i)} \end{pmatrix},$$

where $H_{i2}(q_2)$ is a row of length $(i+1)$. Using this representation, we obtain an intermediate decomposition

$$V = V_1(q_1)\widehat{V}_2(q_2)W(q_3,\ldots,q_f),$$

where $\widehat{V}_2$ is a block matrix

$$\widehat{V}_2(q_2) = \begin{pmatrix} H_{12} \\ \vdots \\ H_{s2} \end{pmatrix}.$$

The column vector $W(q_3,\ldots,q_f)$ is comprised from all functions $p_{2i}^{(j)}, j = 0,\ldots,i,\ i = 0,\ldots,s$. Thus, the second rank is bounded by the dimension of the linear span of these functions. Let us look at the functions $p_{2i}^{(j)}$ more closely. They came from the functions $p_1^{(i)}$. The function $p_1^{(i)}$ is a homogeneous polynomial of degree not higher than i. After the second step, the $i-th$ function "produced" exactly i functions which are homogeneous polynomials in remaining variables of degrees $0,\ldots,i$. For example, after the second step there will be exactly 2 linear functions among $p_1^{(i)}$, three functions that are quadratic and so on. These functions will be splitted later on.

In Table 4.1 one can see how many different polynomials appear at the first steps of the algorithm for the case $s = 4$. It is not difficult to obtain a recursive formula for these numbers.

Step	Order 1	Order 2	Order 3	Order 4
1	1	1	1	1
3	4	3	2	1
4	10	6	3	1
5	20	10	4	1
⋮	⋮	⋮	⋮	⋮

Table 4.1: The number of polynomials of different degrees for the case $s = 4$

Denote by $C(k,j)$ the number of polynomials of degree j that appear after the k-th step of the algorithm. Evidently, they can come only from the previous step and from polynomials of degree not smaller than j, thus

$$C(k,j) = \sum_{i=j}^{s} C(k-1,i). \tag{4.4}$$

An obvious initial condition is $C(k,s) = 1$ — there is always only one function of degree s and $C(1,j) = 1, j = 1,\ldots,s$ — only one function of each degree at the first step. The solution of (4.4) is obviously unique, since the values at the step $(k-1)$ determine the values at the step k. It is easy to verify that

$$C(k,s-1) = k.$$

Using that one can obtain

$$C(k,s-2) = C(k-1,s-2)+C(k-1,s-1)+C(k,s) = C(k-1,s-2)+k = \ldots = \sum_{i=1}^{k} i = \frac{k(k+1)}{2}.$$

By analogous computations for $C(k,s-3)$ and so on, the following formula for $C(k,s-j), j = 1,\ldots,s-1$ is obtained:

$$C(k,s-j) = P(k,j) = \frac{1}{j!}k(k+1)\cdot\ldots\cdot(k+j-1). \tag{4.5}$$

Once the formula is found, it is easy to prove that it solves (4.4) by the induction argument.

It is important to note, that there is also another bound on the dimension of the linear span of functions that appear during the algorithm. At the k-step all the polynomials in question are homogeneous polynomials of $(f-k)$ variables. What is the dimension of the space of such polynomials? Suppose the variables are $q_{j_1},\ldots,q_{j_p}$ $p=(f-k)$ and the degree is bounded by l. Then, the dimension of such space is equal to the p-fold sum

$$D(p,l) = \sum_{j_1=1}^{l}\sum_{j_2=1}^{j_1}\cdots\sum_{j_p=1}^{j_{p-1}} 1,$$

which satisfies a simple recurrence relation

$$D(p,l) = \sum_{j=1}^{l} D(p-1,i). \tag{4.6}$$

Note that (4.6) and (4.4) are actually the same recurrence relations. Thus,

$$D(p,l) = P(p,l) = \frac{1}{l!}p(p+1)\cdot\ldots\cdot(p+l-1). \tag{4.7}$$

Finally, at the k-th step the dimension of the linear span of the polynomials of degree j is bounded by

$$\min(P(k,s-j),P(f-k,j)),$$

and the value of the k-th rank is bounded by (2 comes from polynomials of degree 0 and s).

$$r_k \le 2 + \sum_{j=1}^{s-1}\min(P(k,s-j),P(f-k,j)).$$

Together with the definition of $P(k,0)$ that completes the proof of the Theorem. ■

The formula (4.1) allows to compute upper bounds for the ranks for all values of f and s. The behaviour of these “rank curves” is different for odd and even values of s, see the Figure 4.1

It can be seen, that for the even values of s r_k, as a function of k has one maximum located at $\frac{f}{2}$. If f is odd, then it has two maxima $\lfloor\frac{f}{2}\rfloor$ and $\lceil\frac{f}{2}\rceil$. For the odd values of s the situation is more complicated: there are at least two maxima.

It is obvious that the function $S(k)$ in (4.1) is a piecewise-polynomial function of k, since $P(k,j)$ is a polynomial of degree j in the variable k. Moreover, this function is symmetric against $\frac{f}{2}$, thus it is sufficient to study its structure only on the interval $[0,\frac{f}{2}]$. The following Lemma allows to simplify in some sense the expression (4.1) for $S(k)$.

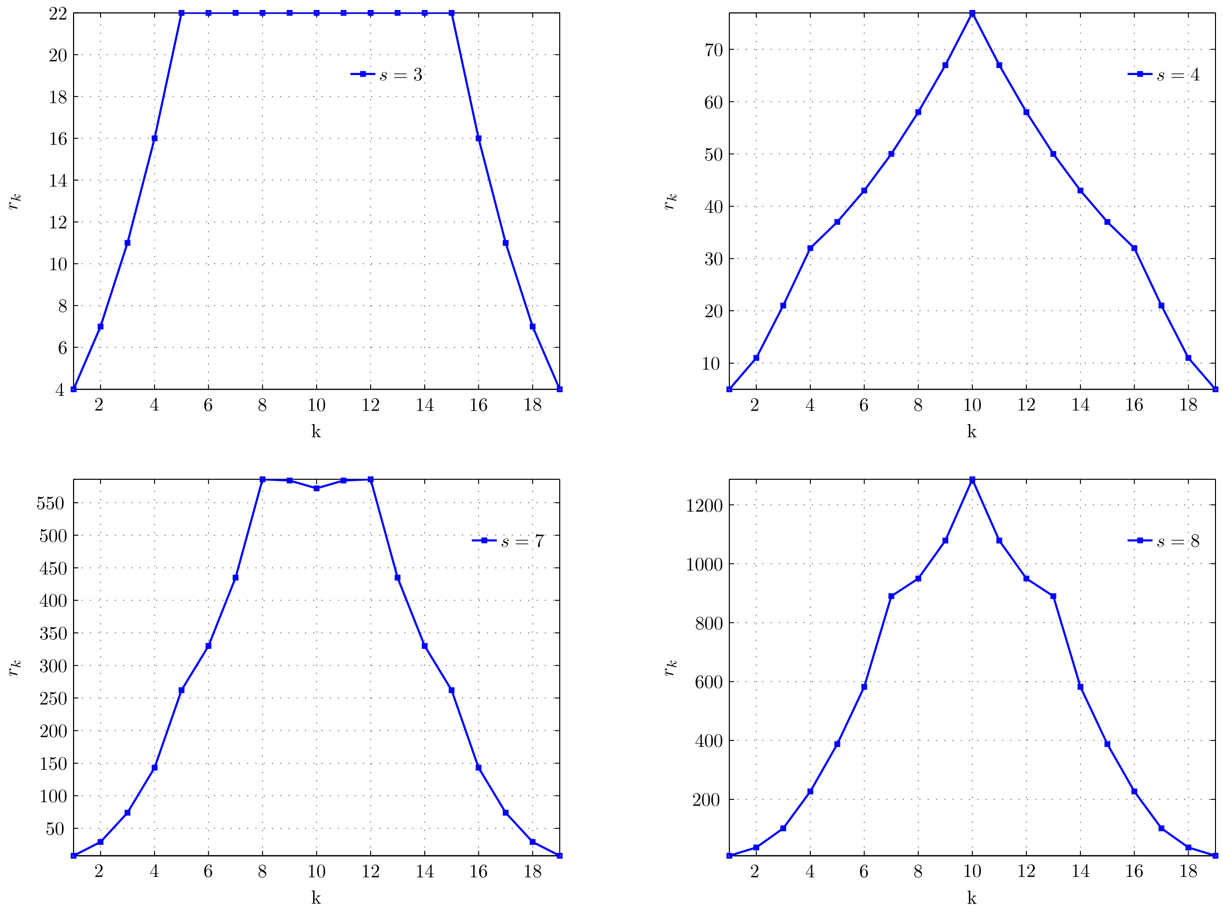

Figure 4.1: Rank bounds (4.1) for $f = 20$ and different values of s versus the mode number k

Lemma 4.4 *The function* $S(k)$ *in* (4.1) *is a polynomial function on the intervals* $[k_q, k_{q+1}]$, $q = 0, 1, \ldots, q_{\max},$, *where*

$$q_{\max} = \lceil \frac{s}{2} \rceil, k_0 = 0$$

and k_q *is the unique solution of the polynomial equation*

$$P(k_q, s - q) = P(f - k_q, q).$$

Moreover, on the q*-th interval*

$$S(k) = P(k + 1, s - q) + P(f - k + 1, q).$$

Proof. The strict proof is rather technical and is omitted to the save space. The idea of the proof is simple. It is sufficient to consider pairs of terms in the definition of $S(k)$ and accurately determine their form and points of discontinuity. Then the final result comes from the recurrence relation for the $P(s, k)$. ■

For the important cases of cubic and quartic potentials, the exact form of $S(k)$ is as follows.

Corollary 4.5 *For a homogeneous cubic potential,*

$$S(k) = \begin{cases} \frac{(k+1)(k+2)}{2} + f - k + 1, & k \leq k_1, \\ f + 2, & k_1 \leq k \leq \frac{f}{2}, \end{cases}$$

where k_1 *solves*

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

The maximal rank is then $f+2$.

Corollary 4.6 *For a homogeneous quartic potential,*

$$S(k) = \begin{cases} \frac{(k+1)(k+2)(k+3)}{6} + f - k, & k \leq k_1, \\ \frac{(k+1)(k+2)}{2} + \frac{(f-k+1)(f-k+2)}{2}, & k_1 \leq k \leq \frac{f}{2}, \end{cases}$$

where k_1 *solves the equation*

$$\frac{k(k+1)(k+2)}{6} = f - k.$$

The maximal rank occurs for $k = \frac{f}{2}$ *and is equal to the integer part of* $\frac{(f+2)(f+4)}{4}$.

Corollary 4.7 *In the setting of Theorem 4.3 we have: for an even* s *maximum of* $S(k)$ *occurs at* $k = \frac{f}{2}$ *and it is equal to*

$$2P\Big(\frac{f}{2}+1, \frac{s}{2}\Big);$$

for an odd s *maximum of* $S(k)$ *occurs at* k^* *and* $f - k^*$, *where* k^* *solves the polynomial equation*

$$P\Big(k^*, \frac{s+1}{2}\Big) = P\Big(f - k^*, \frac{s-1}{2}\Big).$$

In this case, the maximal value of $S(k)$ *is then given by*

$$P\Big(k^* + 1, \tfrac{s+1}{2}\Big) + P\Big(f - k^* + 1, \tfrac{s-1}{2}\Big) =$$
$$= P\Big(f - k^*, \tfrac{s-1}{2}\Big) \cdot \left(\frac{k^* + \frac{s+1}{2}}{k^*} + \frac{f - k^* + \frac{s-1}{2}}{f - k^*}\right) \leq P\Big(f-1, \tfrac{s-1}{2}\Big)\left(\tfrac{s+1}{2} + \tfrac{s-1}{2(f-1)}\right).$$

In the following proposition uniform grid is assumed.

Proposition 4.8 *For a general homogeneous polynomial potential of form*

$$V(q_1, \ldots, q_f) = \sum_{i_1, \ldots, i_s = 1}^{f} a(i_1, \ldots, i_s) \prod_{k=1}^{s} q_{i_k},$$

$$\mathrm{rank}_{QTT}(V) \leq 2(s+1)\mathrm{rank}_{TT}(V).$$

Proof. It is sufficient to notice that the TT-decomposition obtained in the proof of Proposition 4.3 gives rise to the functional TT decomposition of form

$$V(q_1, \ldots, q_f) = \nu_1(q_1)V_2(q_2) \cdot \ldots \cdot V_{f-1}(q_{f-1})\nu_f(q_f),$$

where V_p for $p = 2, \ldots, f-1$ are $r_{p-1} \times r_p$ matrices with elements which are polynomials of degree at most s, and r_p are TT-ranks of V. After discretization in the variable q_p with 2^k points a discrete representation for V_p as a $r_{p-1} \times 2 \times 2 \times \ldots \times 2 \times r_p$ tensor is obtained, and analogously to the proof in the scalar case (Theorem 6 in [41]) it can be shown that for the matrix polynomial case the QTT-ranks are bounded by $(r_{p-1} + r_p)(s+1)$. ■

The following Hypothesis summarized the results of our numerical experiments for the particular values of s.

Hypothesis 4.9 *If the uniform grid is used the following rank estimates hold.*

1. *For a general quadratic potential*

$$V(q_1, \dots, q_f) = \sum_{ij}^{f} a_{ij} q_i q_j,$$

$\mathrm{rank}_{\mathrm{QTT}}(V) \leq f + 1.$

2. *For a general cubic potential*

$$V(q_1, \dots, q_f) = \sum_{ijk}^{f} a_{ijk} q_i q_j q_k,$$

$\mathrm{rank}_{\mathrm{QTT}}(V) \leq f + 1.$

3. *For a general quartic potential*

$$V(q_1, \dots, q_f) = \sum_{ijkl}^{f} a_{ijkl} q_i q_j q_k q_l,$$

$\mathrm{rank}_{\mathrm{QTT}}(V) \leq f(f + 1).$

However, these are upper estimates for general coefficients of polynomials. For particular potentials the QTT-ranks can be much smaller.

Lemma 4.10 *For the Henon-Heiles potential [37] of form*

$$V(q_1, \dots, q_f) = \frac{1}{2} \sum_{k=1}^{f} q_k^2 + \lambda \sum_{k=1}^{f-1} \left(q_k^2 q_{k+1} - \frac{1}{3} q_k^3 \right), \tag{4.8}$$

the QTT-ranks are bounded by 4 *(the TT-ranks are bounded by* 3*).*

Proof. Since the maximal QTT-rank of discretized monomial q^2 is 3, the result for harmonic potential follows from the Theorem 2 of [41]. To get a decomposition for the Henon-Heiles potential, first separate q_1:

$$V = \begin{pmatrix} -\frac{\lambda}{3} q_1^3 + \frac{1}{2} q_1^2 & \lambda q_1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ q_2 \\ V_2(q_2, \dots, q_f) \end{pmatrix},$$

where $V_2(q_2, \dots, q_f)$ is the Henon-Heiles potential from $q_2, \dots, q_f$. Separation of q_2 gives

$$V = \begin{pmatrix} -\frac{\lambda}{3} q_1^3 + \frac{1}{2} q_1^2 & \lambda q_1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ q_2 & 0 & 0 \\ \frac{1}{2} q_2^2 - \frac{\lambda}{3} q_2^3 & q_2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ q_3 \\ V_3(q_3, \dots, q_f) \end{pmatrix},$$

which justifies the general structure of the FTT cores at the subsequent steps: they are 3×3 matrices

$$G_k(q_k) = \begin{pmatrix} 1 & 0 & 0 \\ q_k & 0 & 0 \\ \frac{1}{2} q_k^2 - \frac{\lambda}{3} q_k^3 & q_k & 1 \end{pmatrix},$$

thus TT-ranks are equal to 3. To obtain a QTT-decomposition, one should consider a binary representation of q_k, $q_k = a + h(\sum_{s=0}^{L-1} i_s 2^{s-1})$, where a is the start of the interval where q_k is defined, h is a step size, and i_s take values 0 and 1 (for simplicity, index k is omitted, of course a, h, i_s will depend on k). Introduces new variables

$$x_s = \frac{a}{d} + h i_s 2^{s-1},$$

then

$$q_k = x_1 + \ldots + x_L.$$

Estimation of the QTT-ranks is reduced to the separation of indices in a block parameter-dependent matrix

$$G(q) = \begin{pmatrix} 1 & 0 & 0 \\ q & 0 & 0 \\ \frac{1}{2}q^2 - \frac{\lambda}{3}q^3 & q & 1 \end{pmatrix}.$$

To separate x_1 (the first QTT mode) consider q as

$$q = x_1 + \nu,$$

where $\nu = x_2 + \ldots + x_L$. A simple algebra shows that

$$G(x_1 + \nu) = G_1(x_1) \begin{pmatrix} 1 & 0 & 0 \\ \nu & 0 & 0 \\ \nu^2 & 1 & 0 \\ \nu^3 & \nu & 1 \end{pmatrix},$$

thus the first rank is bounded by 4. At the next step, $\nu = x_2 + w$, and

$$\begin{pmatrix} 1 & 0 & 0 \\ \nu & 0 & 0 \\ \nu^2 & 1 & 0 \\ \nu^3 & \nu & 0 \end{pmatrix} = G_2(x_2) \begin{pmatrix} 1 & 0 & 0 \\ w & 0 & 0 \\ w^2 & 1 & 0 \\ w^3 & w & 1 \end{pmatrix},$$

and so on for $x_3, \ldots, x_L$. Thus the upper bound for the QTT-ranks of the Henon-Heiles potential is 4. ■

4.2 How to get the QTT representation numerically

From the previous subsection we know, that the polynomial PES can be well-approximated in the QTT-format. The question is how to get such decompositions numerically. A general framework can be based on the interpolation of the tensor at certain points. Suppose it is known that V has low QTT-ranks, but only a subroutine that allows a computation of any prescribed element of a tensor is given. Then V can be recovered by the TT-cross interpolation formula [44]. When V is given analytically, a more simple approach is applicable. Often, V can be represented as a separable expansion of form

$$V(q_1, \ldots, q_f) \approx \sum_{\alpha=1}^{R} h_1(q_1, \alpha) \ldots h_f(q_f, \alpha), \tag{4.9}$$

but with large number of terms R. This is true for the polynomial PES, but we will consider an approximation of a 6-dimensional PES for HONO, which is defined through a certain analytic expansion, and it can also be put into the form (4.9). The conversion of V defined by (4.9), into the QTT-format is performed using the following steps. Each summand is converted into the QTT-format, by converting one-dimensional functions $h_k(q_1, \alpha), k = 1, \dots, f$ into the QTT-format using either the compression of the full array to the TT-format [42], or in the case of certain functions, like polynomial or sine functions, by using known analytical QTT-representations [41]. Then their Kronecker product is computed by

$$\mathbf{V}_\alpha = \mathbf{A}_1 \times \mathbf{A}_2 \times \dots \mathbf{A}_f,$$

which is a Lf-dimensional tensor. In fact, no operations are performed and the Kronecker product reduces to the concatenation of cores. Thus, the required tensor $\mathbf{V}$ in the QTT-format is now given as a sum of R tensors in the QTT-format:

$$\mathbf{V} = \sum_{\alpha=1}^{R} \mathbf{V}_\alpha.$$

These additions can be performed in the QTT-format directly, yielding a tensor with rank equal to $R\,(\max_\alpha \operatorname{rank}_{\mathrm{QTT}}(\mathbf{V}_\alpha))$. However, R can be very large, and to avoid rank growth, $\mathbf{V}_\alpha$ are added one-by-one approximately by using the following scheme:

$$\mathbf{V}^{(1)} := \mathbf{V}_1, \quad \mathbf{V}^{(k+1)} = T_\varepsilon(\mathbf{V}^{(k)} + \mathbf{V}_k),$$

where T_ε is the rounding operator with relative accuracy ε in the QTT-format. If we assume, that the QTT-ranks of the intermediate tensors $\mathbf{V}^{(k)}$ are or order r, then the complexity of the algorithm is $\mathcal{O}(Rdfr^3)$. If no intermediate compression is used, the complexity would be cubic in R and is unacceptable.

To see how algorithm works, consider the following example, which was kindly provided to us by H.-D. Meyer and is described in details in [47]. It is a 6-dimensional system (i.e., $f = 6$), with the potential of form

$$\begin{aligned} V(R_1, R_2, R_3, \theta_1, \theta_2, \tau) &= S_0 + e^{-D(R_1, R_2, R_3, \theta_1, \theta_2)} \\ &\times \sum_{ijklmn} c_{ijklmn} Q_1^i Q_2^j Q_3^k Q_4^l Q_5^m \cos(n Q_6), \end{aligned}$$

with

$$D = \sum_{i=1}^{3} d_i (R_i - R^{\mathrm{ref}})_i^2 + \sum_{j=1}^{2} d_{j+3} (\theta_j - \theta^{\mathrm{ref}_j})^2,$$

and

$$Q_{1/2/3} = 1 - e^{-0.7(R_{1/2/3} - R^{\mathrm{ref}}_{1/2/3})}, Q_{4/5} = \theta_{1/2} - \theta^{\mathrm{ref}}_{1/2}, Q_6 = \tau - \tau^{\mathrm{ref}}.$$

As discussed before, to approximate it in the QTT-format, each normal coordinate is put into a segment, and a uniform grid with 2^L points is introduced, and the QTT-approximation is computed via add-and-compress method. In variablesQ_1, ..., Q_5, V is a polynomial, whereas in Q_6 it is a trigonometric function, and these univariate functions can be decomposed into the QTT-format with uniform rank bounds in grid size, see [41]. To check the accuracy, an approximation of the potential with $\varepsilon = 10^{-12}$ is computed, and then used as a "true" potential. The computational time for the largest example ($n = 2^{10}$) was about 1 minute to get the approximation with the relative accuracy 10^{-12} in the Frobenius norm. It can be seen, that the approximation of this potential energy surface is very good with the QTT-ranks independent of the dimension and also depending logarithmically on the accuracy, so the QTT-format for this PES looks very promising (see Table 4.2).

n	rank(10^{-2})	rank(10^{-4})	rank(10^{-7})
2^4	3.2	7.3	12.0
2^5	3.3	7.9	12.5
2^6	3.4	8.0	12.6
2^7	3.4	8.1	12.7
2^8	3.4	8.1	12.7
2^9	3.7	8.0	12.7
2^{10}	3.6	8.0	12.7

Table 4.2: Rank of the HONO potential for different accuracies.

5 Solution of eigenvalue problem by DMRG method

5.1 Formulation as nonlinear optimization problem

Now, after the PES is transformed into the QTT-format, ($\Delta_{(f)}$ is already in the QTT-format) we can safely assume that the Hamiltonian H in (2.1) is also in the QTT-format. From now on we also forget about the fact that actual dimension of the tensor is equal to Lf and denote the dimensions of the space by d, since our results apply to any matrix that is represented in the QTT-format and are purely algebraic. So,

$$H(i_1, \ldots, i_d, j_1, \ldots, j_d) = H_1(i_1, j_1) \ldots H_d(i_d, j_d), \tag{5.1}$$

where

$$H_k(i_k, j_k) = H_k(\alpha_{k-1}, i_k, j_k, \alpha_k)$$

are the cores of the QTT-representation of H, and

$$1 \leq \alpha_k \leq R_k, k = 1, \ldots, d,$$

and

$$1 \leq i_k, j_k \leq n_k.$$

For the QTT-format $n_k = 2$, but the results are also applicable to other mode sizes. We want to find an eigenvector ψ that solves

$$H\psi = E\psi. \tag{5.2}$$

Since the complexity of solving (5.2) in the full space is exponential in the dimension, the vector ψ is sought in the set of QTT-structured tensors with small QTT-ranks. The vector ψ can be naturally associated with a d-dimensional tensor Ψ with elements $\Psi(i_1, i_2, \ldots, i_d)$. This tensor is then approximated by another tensor in the TT-format:

$$\Psi(i_1, \ldots, i_d) \approx G_1(i_1) \ldots G_d(i_d). \tag{5.3}$$

Let us denote this class of tensors (with the QTT-ranks bounded by some number R) by $\mathcal{S}$. What are equations for parameters, defining representation? A natural way is to formulate (5.2) as a minimization problem, and replace the minimization over the whole space (which contains n^d variables) by the minimization over $\psi \in \mathcal{S}$. This yields a nonlinear minimization problem, but with fewer parameters. For the eigenvalue problem (5.2) the standard way is to minimize the Rayleigh quotient:

$$\psi = \arg\min(H\psi, \psi), \quad (\psi, \psi) = 1,$$

so we seek for the minimizer of the Rayleigh quotient for the QTT-structured tensors:

$$\psi = \arg\min(H\psi, \psi), \quad \psi : (\psi, \psi) = 1, \quad \psi \in \mathcal{S} \tag{5.4}$$

The parameter R here controls the memory, required for the storage of the approximate solution, and has to be chosen as a compromise between accuracy and storage.

How to solve (5.4)? One can use general-purpose minimization approaches. This requires computation of the function values and its gradients. This is an interesting and important research direction, but it has several drawbacks. First of all, it does not take into account directly a special structure of the functional. For this particular problem one can hope that specialized methods can be developed.

The second problem is not that obvious, but is encountered by many people that tried to solve high-dimensional problems in tensor formats: the selection of ranks. There are $(d-1)$ TT-ranks, and they have to be selected somehow, and an adaptive procedure should be available, since a simple guess-and-try method is not always the best choice. For example, the alternating least squares method (ALS), which is very popular in multilinear algebra (especially for the computation of the canonical decomposition) can be implemented to solve (5.4) in the following way. If we fix all the cores, except G_k, then the minimization over G_k becomes a quadratic optimization, and can be solved as a "small" eigenvalue problem. In this case, all r_k have to be specified, and it is unclear, how to do this.

Our experiments show that the ALS method can converge slowly. However, it can be modified for the QTT-format (when all modes sizes n_k are small) to solve both problems: how to determine the QTT-ranks given only the required accuracy ε, and how to speed up the convergence. A simple modification is to minimize over two adjacent cores, G_k and G_{k+1}. This approach originates from the solid state physics in the study of quantum spin systems, where the representations of form (5.3) are used for quite a long time [45, 8]. The wavefunctions of spin systems depend can be considered as tensors with small modes size, usually $n_k = 2$ or $n_k = 4$. The numerical modelling of quantum spin systems thus leads to algebraically the same problem, as we consider in this paper, but with an important difference: The Hamiltonians are usually given in form, that is analogous to the canonical format, and we use the QTT-format for matrices also.

The minimization over a pair of adjacent cores leads to the DMRG (Density Matrix Renormalization Group) algorithm, first proposed by White [51], and its variational nature was understood in [45]. The DMRG-type algorithms for the eigenvalue problems were also recently considered in [48]. The DMRG algorithm is, in fact, a modified Gauss-Seidel method (i.e. the minimization is over G_k and G_{k+1}, then over G_{k+1} and G_{k+2} and so on) but it has exceptional convergence properties, which are uncommon to the ALS algorithms. This an experimental fact, and no convergence theory is available by now.

5.2 DMRG algorithm

How does the DMRG look like? If all cores except G_k and G_{k+1} are fixed, we are left with a "small" minimization problem. The optimization problem in G_k and G_{k+1} is still nonlinear. To make it linear, a new *superblock* is introduced:

$$W(i_k, i_{k+1}) = G_k(i_k)G_{k+1}(i_{k+1}),$$

which can be represented as a $r_{k-1} \times n_k \times n_{k+1} \times r_{k+2}$ tensor. This is equivalent to the merge of two modes, k and $k+1$ in the TT-representation, which leads to a $n_1 \times \ldots \times n_{k-1} \times (n_k n_{k+1}) \times n_{k+2} \ldots \times n_d$ tensor, and the "middle" core is optimized. The problem in W is quadratic and can be reduced to the eigenvalue problem for W. The initial eigenvalue problem had an important constraint, $||\psi|| = 1$. This constraint is trivially imposed on W also due to the properties of the TT-format. If, before the minimization, we ensure that in the TT-representation of ψ (5.3) the fixed cores $G_s, s = 1, \ldots, k-1$ are *left-orthogonal*, i.e.

$$\sum_{i_s} G_s^\top(i_s)G_s(i_s) = I_{r_s},$$

and the cores $G_s, s = k+2, \ldots, d$ are *right-orthogonal*, i.e.

$$\sum_{i_s} G_s(i_s)G_s^\top(i_s) = I_{r_{s-1}},$$

then, as in the TT-rounding procedure [39], it can be shown that for a such orthogonalized representation the norm of ψ is the norm of the cores that we optimize, i.e. in the matrix form

$$\sqrt{\sum_{i_k, i_{k+1}} ||W(i_k, i_{k+1})||_F^2} = ||\psi|| = 1. \tag{5.5}$$

After W is obtained from the "local" eigenvalue problem, the solution W is approximated to recover G_k and G_{k+1}.

$$W(i_k, i_{k+1}) \approx G_k(i_k)G_{k+1}(i_{k+1}), \tag{5.6}$$

with some prescribed accuracy ε. It is called the *decimation step.*To compute the approximation (5.6) one SVD is required. The equation (5.6) in the index form is

$$W(\alpha_{k-1}, i_k, i_{k+1}, \alpha_{k+1}) \approx \sum_{\alpha_k} G_k(\alpha_{k-1}, i_k, \alpha_k)G_{k+1}(\alpha_k, i_{k+1}, \alpha_{k+1}),$$

and it can be computing via the reshaping of W into a $r_{k-1}n_k \times n_{k+1}r_{k+1}$ matrix and computing its ε-truncated SVD. The rank r_k is computed here from the accuracy parameter ε, and is determined adaptively, which is a big advantage over the standard ALS approach.

Then, the whole DMRG step (called sweep) consists of the following. First, the cores $G_3, \ldots, G_d$ are made right-orthogonal, then we optimize for G_1, and G_2, then for G_2 and G_3 and so on, keeping the cores $G_1, \ldots, G_{k-1}$ left-orthogonal so that (5.5) can be used. After the last core is reached, the sweep can be made from right to left [3], and the process continues until the convergence (which can be detected either by the stabilization of the Rayleigh quotient, or by the stabilization of the ψ itself).

[3] One can use either two-sided DMRG or one-sided DMRG, when from right-to-left we only orthogonalize cores of the solution. It is not clear to us which method is better.

5.3 Local eigenvalue problem

Now we have to obtain an eigenvalue problem for W. Ψ is represented as

$$\Psi(i_1, \dots, i_d) = G_1(i_1) \dots G_{k-1}(i_{k-1}) W(i_k, i_{k+1}) G_{k+2}(i_{k+2}) \dots G_d(i_d).$$

In order to the simplify the expression for $(H\psi, \psi)$, recall how two basic operations in the TT-format look like: the matrix-by-vector product and the scalar product. The matrix-by-vector product

$$y = H\psi,$$

if H and ψ are in the TT-format, is also in the TT-format with the cores [39]

$$Y_k(i_k) = \sum_{j_k} \Big(H_k(i_k, j_k) \otimes G_k(j_k) \Big).$$

For two tensors $\mathbf{A}$ and $\mathbf{B}$ their scalar product

$$\sum_{i_1,\dots,i_d} A(i_1, \dots, i_d) B(i_1, \dots, i_d)$$

can be computed in two steps. First, the Hadamard (elementwise) product $\mathbf{C} = \mathbf{A} \circ \mathbf{B}$ is computed, its cores are

$$C_k(i_k) = A_k(i_k) \otimes B_k(i_k),$$

and then the sum of all elements of C_k is computed via the contraction with the tensor of all ones. Summing everything up, the following representation for $(H\psi, \psi)$ is obtained:

$$(H\psi, \psi) = \Gamma_1 \Gamma_2 \dots \Gamma_{k-1} \cdot \Gamma \cdot \Gamma_{k+2} \Gamma_d,$$

where

$$\Gamma_s = \sum_{i_s, j_s} H_s(i_s, j_s) \otimes G_s(i_s) \otimes G_s(j_s), \quad s = 1, \dots d, s \neq k, s \neq k+1,$$

and

$$\Gamma = \sum_{i_k, i_{k+1}, j_k, j_{k+1}} H_k(i_k, j_k) H_{k+1}(i_{k+1}, j_{k+1}) \otimes W(i_k, i_{k+1}) \otimes W(j_k, j_{k+1}).$$

From its definition it can be seen, that Γ_s is a matrix of size $R_s r_s^2 \times R_{s+1} r_{s+1}^2$ and can be indexed (in principle) as a 6-dimensional array, and Γ (corresponding to the merged core) is a matrix of size $R_{k-1} r_{k-1}^2 \times R_{k+1} r_{k+1}^2$.

The expression for the Rayleigh quotient can be simplified. Γ_1 is indeed a row vector (since $r_0 = R_0 = 1$) and the product

$$p_k = \Gamma_1 \dots \Gamma_{k-1}$$

is also a row vector of length $R_{k-1} r_{k-1}^2$, which can be considered as a three-dimensional array $P_{k-1}(\alpha, \beta, \gamma)$. The same is for the product

$$q_k = \Gamma_{k+2} \dots \Gamma_d,$$

which is a column vector of length $R_{k+1} r_{k+1}^2$. The cost of computing p_{s+1} when p_s is known is the cost of a evaluating matrix-by-vector product

$$p_s \Gamma_s = p_s \sum_{i_s, j_s} H_s(i_s, j_s) \otimes G_s(i_s) \otimes G_s(j_s) = \sum_{i_s, j_s} w(i_s, j_s),$$

where

$$w(i_s, j_s) = p_s\Big(H_s(i_s, j_s) \otimes G_s(i_s) \otimes G_s(j_s)\Big).$$

The computation of $w(i_s, j_s)$ for each fixed pair (i_s, j_s) is a multiplication of a tensor rank-1 matrix by a full vector. To estimate the complexity, assume that $r_s \sim r, R_s \sim R$, then it is well known that such product can be realized in $\mathcal{O}(Rr^3)$ operations, and the total cost for the computation of p_s is $\mathcal{O}(n^2Rr^3)$ operations. Since $n = 2$ in our case, the complexity is linear in the matrix rank and cubic in the rank of the solution. The same is for q_k. After p_k and q_k are computed (recall that we assumed that the corresponding cores are fixed) the expression for the Rayleigh quotient becomes

$$(H\psi, \psi) = p_k\Big(\sum_{i_k, i_{k+1}, j_k, j_{k+1}} H_k(i_k, j_k)H_{k+1}(i_{k+1}, j_{k+1}) \otimes W(i_k, i_{k+1}) \otimes W(j_k, j_{k+1})\Big)q_k.$$

Now let us for a moment use simpler notations. As noted above, the DMRG optimization step is equivalent to the ALS approach applied to the tensor with modes $k, k+1$ merged, so let us consider (i_k, i_{k+1}) as one long index (denote it by i) and (j_k, j_{k+1}) — as another (denote it by j). Our main goal is to solve the optimization problem for W, so and this local problem will be described in more details. The function to be minimized has form

$$f(W) = p\Big(\sum_{i,j} M(i,j) \otimes W(i) \otimes W(j)\Big)q,$$

where $M(i,j), i,j = 1,\dots,m$ are $R_1 \times R_2$ matrices, $W(i), i = 1,\dots,m$ are $r_1 \times r_2$ matrices, p is a row vector of length $R_1r_1^2$ and q is a column vector of length $R_2r_2^2$. For our case $m = n^2 = 4$, i.e., is small compared to r_1, r_2, R_1, R_2. Now we have the following optimization problem:

$$p\Big(\sum_{i,j} M(i,j) \otimes W(i) \otimes W(j)\Big)q \to \min, \quad \sum_i ||W(i)||_F^2 = 1. \tag{5.7}$$

The problem (5.7) is a quadratic optimization problem in W, thus it can be reduced to the eigenvalue problem for W, which has form

$$\widehat{M}w = Ew, ||w|| = 1,$$

and w is the tensor W transformed into the "long vector" (in MATLAB notation, $w = W(:)$). In our case, W contains $4r^2$ elements ($n = 2$ is assumed), thus the matrix $\widehat{M}$ is $4r^2 \times 4r^2$. If it is formed as a full matrix, the TT-ranks no more than 50 can be chosen, since $4r^2 > 10000$ for $r = 50$, and in practice one can encounter TT-ranks of order several hundreds. However, the storage of $\widehat{M}$ requires only the storage of $M(i,j)$ which is only $16r^2$ memory cells, and moreover, the matrix-by-vector product $\widehat{M}w$ can be computed in $\mathcal{O}(Rr^3 + R^2r^2)$ operations. Thus an iterative method for the computation of the lowest eigenvalue can be used. In the DMRG calculations, usually Jacobi-Davidson algorithms or Lanczos algorithms are employed. We will use Locally Optimal Block Preconditioned Conjugate Gradient Algorithm (LOBPCG) by Knyazev [29] for this task. To select initial approximation to the eigenvector for the local problem, an eigenvector from the previous DMRG sweep is used:

$$\widehat{W} = G_{k-1}(i_{k-1})G_k(i_k).$$

The computational scheme is summarized in Algorithm 1, and the matrix-by-vector product subroutine — in Algorithm 2. To describe the matrix-by-vector product, we have to use index notations.

Algorithm 1 DMRG algorithm

```
Require: Matrix $H$ in QTT format, accuracy $\varepsilon$, Initial approximation to eigenvector $\psi$ in QTT format.
Ensure: Improved approximation $\psi$ to eigenvector
not_converged = .true.
while not_converged = .true. do
    not_converged = .false.
    {right-to-left sweep}
    $R := [1], \Psi_d = [1]$.
    for $k = d$ to 3 step $-1$ do
        $G_k(i_k) := G_k(i_k)R$.
        $[G_k(\beta_{k-1}; i_k\beta_k), R(\alpha_{k-1}, \beta_{k-1})] := QR_{\mathrm{rows}}(G_k(\alpha_{k-1}; i_k\beta_k))$.
        {Compute new q }
        $w(i_k, j_k) := \Big(H_k(i_k, j_k) \otimes G_k(i_k) \otimes G_k(j_k)\Big) q_k$.
        $q_{k-1} := \sum_{i_k, j_k} w(i_k, j_k)$.
    end for
    {left-to-right sweep, main cycle}
    for $k = 1$ to $k = d - 1$ do
        {Compute approximation from previous step $\widehat{W}$ to the solution of (5.7)}
        $W(i_k, i_{k+1}) = G_k(i_k)G_k(i_{k+1})$.
        Solve (5.7) using LOBPCG with $\widehat{W}$ as initial vector and obtain new $W$
        {Check convergence}
        If $||W - \widehat{W}||_F > \varepsilon ||W||_F$ not_converged=.true..
        {Truncation} Compute truncated SVD of $W(\alpha_{k-1} i_k; i_{k+1} \alpha_{k+1})$
            $$W(i_k, i_{k+1}) \approx \widehat{G}_k(i_k)\widehat{G}_{k+1}(i_{k+1}), \quad \text{and } \widehat{G}_k, \widehat{G}_{k+1} \text{ are left-orthogonal,}$$
        and truncation error is bounded by $\frac{\varepsilon}{\sqrt{d-1}}||W||_F$.
        $G_k := \widehat{G}_k, G_{k+1} := \widehat{G}_{k+1}$.
        {Compute new $p_k$}
        $w(i_k, j_k) := p_k \Big(H_k(i_k, j_k) \otimes G_k(i_k) \otimes G_k(j_k)\Big)$
        $p_{k+1} := \sum_{i_k, j_k} w(i_k, j_k)$.
    end for
    {If two-sided DMRG is used, perform sweep from right-to-left}
end while
```

Algorithm 2 Fast matrix-by-vector product in DMRG

```
Require: $\widehat{M}$ specified implicitly by
    $H_k(\alpha_{k-1}, i_k, j_k, \alpha_k), H_{k+1}(\alpha_k, i_{k+1}, j_{k+1}, \alpha_{k+1}) p_{k-1}(\alpha_{k-1}, \beta_{k-1}, \gamma_{k-1}), q_{k+1}(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})$,
    vector $w$ indexed as $W(\beta_{k-1}, j_k, j_{k+1}, \beta_{k+1})$,
Ensure: Matrix-by-vector product $Y = \widehat{M}w = Y(\gamma_{k-1}, i_k, i_{k+1}, \gamma_{k+1})$.
 1: Preprocessing
    $\widetilde{H}_k(\beta_{k-1}, \gamma_{k-1}, i_k, j_k, \alpha_k) = \sum_{\alpha_{k-1}} H_k(\alpha_{k-1}, i_k, j_k, \alpha_k) p_{k-1}(\alpha_{k-1}, \beta_{k-1}, \gamma_{k-1})$
                                        Complexity: $\mathcal{O}(n^2 R^2 r^2)$
    $\widetilde{H}_{k+1}(\alpha_k, i_{k+1}, j_{k+1}, \beta_{k+1}, \gamma_{k+1}) = \sum_{\alpha_{k+1}} H_{k+1}(\alpha_k, i_{k+1}, j_{k+1}, \alpha_{k+1}) q_{k+1}(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})$
                                        Complexity: $\mathcal{O}(n^2 R^2 r^2)$
 2: {Matvec}
 3: $Y(\gamma_{k-1}, i_k, \alpha_k, j_{k+1}, \beta_{k+1}) := \sum_{\beta_{k-1}, j_k} \widetilde{H}_k(\beta_{k-1}, \gamma_{k-1}, i_k, j_k, \alpha_k) Y(\beta_{k-1}, j_k, j_{k+1}, \beta_{k+1})$
                                        Complexity: $\mathcal{O}(n^3 R r^3)$
 4: $Y(\gamma_{k-1}, \beta_{k+1}, i_{k+1}, \alpha_{k+1}) :=$
    $:= \sum_{\alpha_k, j_{k+1}, \beta_{k+1}} \widetilde{H}_{k+1}(\alpha_k, i_{k+1}, j_{k+1}, \beta_{k+1}, \gamma_{k+1}) Y(\gamma_{k-1}, i_k, \alpha_k, j_{k+1}, \beta_{k+1})$
                                        Complexity: $\mathcal{O}(n^3 R r^3)$
```

The total complexity of Algorithm 2 is estimated as $\mathcal{O}(n^2 R^2 r^2)$ operations at the preprocessing step and $\mathcal{O}(n^2 R r^3)$ for all subsequent matrix-by-vector products. All these steps are implemented using matrix-by-matrix products, which leads to high efficiency of such operations provided that the optimized BLAS libraries are used.

6 Numerical experiments

6.1 Model 3-body problem

As a first example, consider model 3-body problem [6]. The Hamiltonian has the form

$$H = H_x + H_y + H_z + V_c(x, y, z), \tag{6.1}$$

where

$$\begin{aligned} H_x &= \frac{1}{2}\left(-\frac{\partial^2}{\partial x^2} + w_x^2 x^2 + 2\lambda\eta x^3\right), \\ H_y &= \frac{1}{2}\left(-\frac{\partial^2}{\partial y^2} + w_y^2 y^2 + 2\mu\zeta y^3\right), \\ H_z &= \frac{1}{2}\left(-\frac{\partial^2}{\partial z^2} + w_z^2 z^2\right), \end{aligned}$$

and

$$V_c = \lambda x y^2 + \mu y z^2.$$

The values of the parameters were chosen as in [6]. They are $w_x^2 = 0.49, w_y^2 = 1.69, w_z^2 = 1.0, \lambda = \mu = -0.10$ and $\nu = \zeta = 0.10$. We discretize the problem in a cube $[-a, a]^3$ with sufficiently large a, on a uniform grid with $n = 2^L$ points in each direction.

The potential is a polynomial of the third order, so our estimates give us a rank bound 12 for the diagonal part, plus 4 for the kinetic part, so the upper bound for the rank is 16. To compute the solution, we first set up the DMRG-method with a low accuracy, wait until it converges and then increase the accuracy. Three thresholds are used: $\varepsilon = 10^{-3}, \varepsilon = 10^{-5}, \varepsilon = 10^{-7}$, and the solution of each problem gives a starting vector for the solution of the next one. We need a reference solution and a reference eigenvalue eigenvalue to measure the accuracy. As a reference solution, we use the solution computed with the following parameters: $d = 13, a = 7, \varepsilon = 10^{-10}$. The computed reference solution is $E^* = 1.4895990$. The convergence of the eigenvalues is shown in Table 6.1.

n	$E(10^{-2}) - E^*$	$E(10^{-4}) - E^*$	$E(10^{-7}) - E^*$
2^4	-0.0688173	-0.0689082	-0.0689082
2^5	-0.0163878	-0.0176922	-0.0176922
2^6	-0.0030445	-0.0045148	-0.0045155
2^7	0.0004258	-0.0005373	-0.0011436
2^8	0.0025507	0.0003197	-0.0002879
2^9	0.0238319	0.0005368	-0.0000715
2^{10}	0.0244351	0.0005914	-0.0000173

Table 6.1: Accuracy of computed eigenvalues

This shows, that the parameter ε has to be chosen adaptively with the grid size n: the larger is n is, the higher the accuracy should be.

6.2 Henon-Heiles potential

The proposed approach was tested on the Henon-Heiles potential (4.8). First, the matrix was approximated in the QTT-format by using add-and-compress algorithm. Then, DMRG method (Algorithm 1) was applied. The grid size was set to 128, and the number of degrees of freedom f is varying from 4 to 256. In Figure 6.1 the time required to approximate the matrix is presented, in Figure 6.2 the dependence of maximal and effective ranks on the number of degrees of freedom is given. The Figure 6.3 shows, how the QTT-ranks of the solution look like. Note that the minima of this oscillating picture correspond to the TT-ranks, i.e. when no quantization is used, and for the new (virtual) dimensions the QTT-ranks are larger but still acceptable.

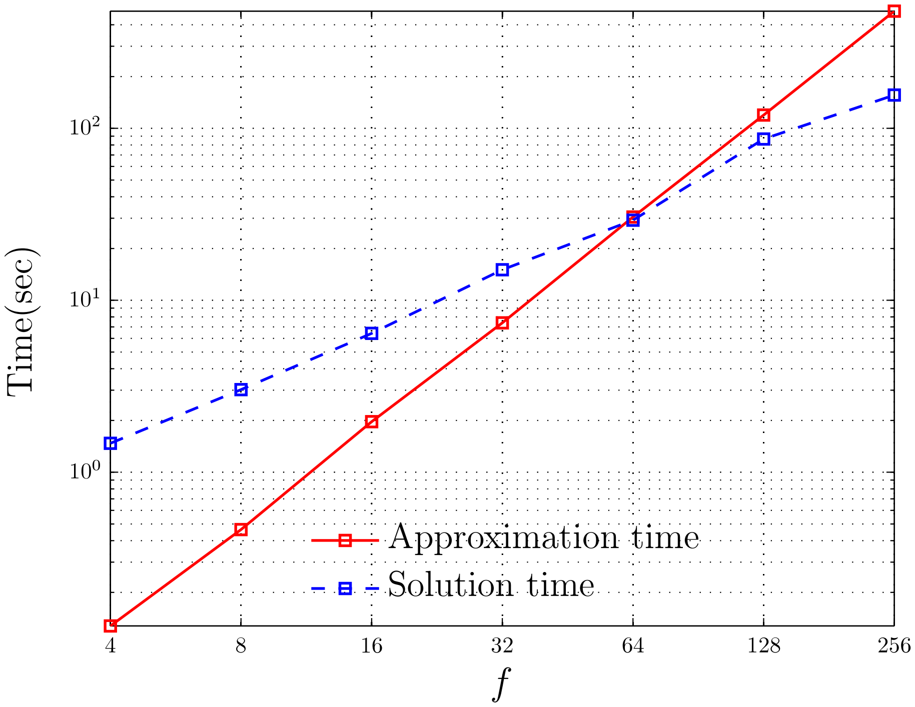

Figure 6.1: Solution and approximation timings for Henon-Heiles potentials with $d = 7$ and f is varying, $\varepsilon = 10^{-6}$

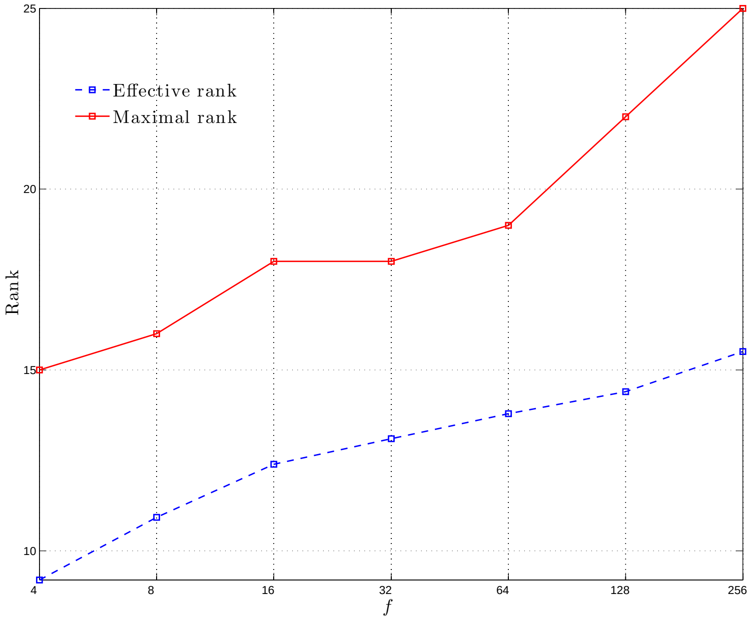

Figure 6.2: Maximal and effective rank behaviour, $d = 7$, f is varying, $\varepsilon = 10^{-6}$

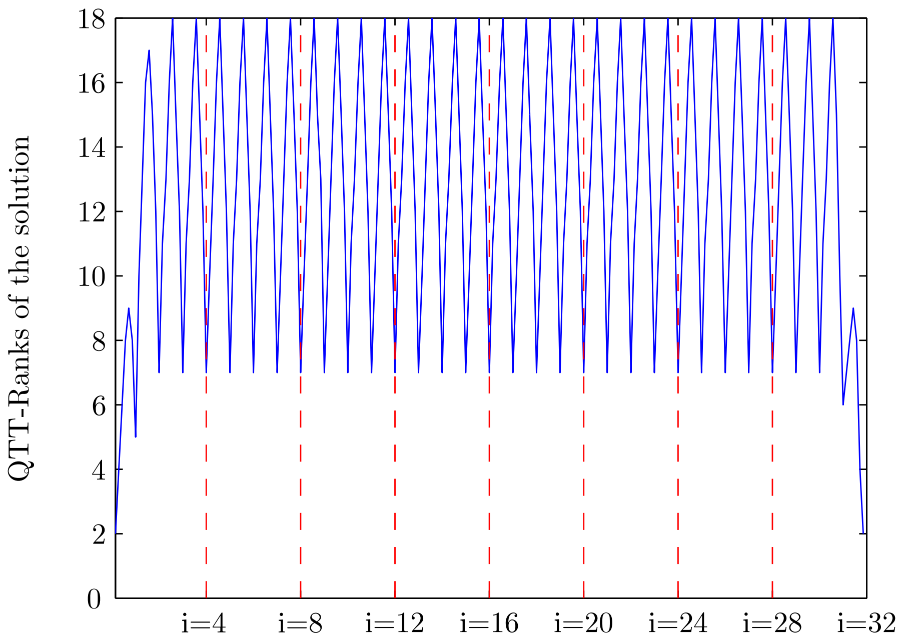

Figure 6.3: QTT-ranks for the solution with f = 32, d = 7 for different modes, vertical lines correspond to "physical" modes

7 Conclusions and future work

In this paper we presented a numerical algorithm for the calculation of the vibrational ground state of the high-dimensional molecular Schrödinger operator. It is based on two ingredients:

- QTT-representation of the potential energy surface (PES)
- DMRG algorithm for the minimization of the Rayleigh quotient in the QTT-format.

For the polynomial PES the estimates on TT and QTT ranks were obtained, and they are polynomial in the number of degrees of freedom. The DMRG algorithm was initially designed for handling quantum spin systems. However, quantization of the solution of the high-dimensional molecular Schrödinger equation and the representation of the solution in the QTT-format leads literally to the same problem, but now the binary indices correspond not to spins, but to the binary digits of the grid point index.

For a HONO PES, having an analytical representation but with many terms we showed, that it can be well-approximated in the QTT-format with very small ranks. Similar results have been obtained for the Henon-Heiles potential.

Being a block relaxation with overlapping, the DMRG method showed a very good convergence properties, which are uncommon for the ALS method, making the DMRG approach very promising for solving optimization problems in the QTT-format. The experiments on a model 3-body problem and on a benchmark high-dimensional Henon-Heiles potential show the effectiveness of our approach.

In the next steps we plan to apply the QTT+DMRG approach to solve problems with more complicated PES, and to improve its robustness by using multigrid ideas.

The computation of several smallest vibrational states is very important, so one has to design a block version of the DMRG in the QTT-format. A prototype realization in MATLAB has been already developed by us, and it works good.

For the quantum molecular dynamics the time evolution is of great importance. For this case, one has to use a time-dependent version of the DMRG in the combination with the TT-format, and this is a separate study. However, for all these problems, the approximation of the Hamiltonian in the QTT-format seems to be a good idea.

To conclude, we hope that the ideas of the QTT with the help of the advanced optimization algorithm (DMRG) will lead to very efficient numerical scheme for solving quantum molecular dynamics problems with a large number of degrees of freedom.

Acknowledgements. The authors gratefully acknowledge Prof. H.-D. Meyer (University of Heidelberg) for providing practically interesting examples of PES and for stimulating discussions, and to Prof. R. Schneider (TU Berlin) for motivating discussions on the topic of the DMRG and its relation to the TT-format.

References

[1] J. Ballani, L. Grasedyck, and M. Kluge, *Black box approximation of tensors in Hierarchical Tucker format*, Preprint 57, MPI MIS, Leipzig, 2010.

[2] J. M. Bowman, *The self-consistent-field approach to polyatomic vibrations*, Acc. Chem. Res., 19 (1986), pp. 202–208.

[3] J. M. Bowman, J. S. Bittman, and L. B. Harding, *Ab initio calculations of electronic and vibrational energies of HCO and HOC*, J. Chem. Phys., 85 (1986), p. 911.

[4] S. Carter, S. J. Culik, and J. M. Bowman, *Vibrational self-consistent field method for many-mode systems: A new approach and application to the vibrations of CO adsorbed on Cu(100)*, J. Chem. Phys., 107 (1997), p. 10458.

[5] G. M. Chaban, J. O. Jung, and R. B. Gerber, *Ab initio calculation of anharmonic vibrational states of polyatomic systems: Electronic structure combined with vibrational self-consistent field*, J. Chem. Phys., 111 (1999), p. 1823.

[6] K. Christoffel, *Investigations of self-consistent field, scf ci and virtual stateconfiguration interaction vibrational energies for a model three-mode system*, Chem. Phys. Lett., 85 (1982), pp. 220–224.

[7] A. Falco and W. Hackbusch, *On minimal subspaces in tensor representation*, Preprint 70, MPI MIS, Leipzig, 2010.

[8] M. Fannes, B. Nachtergaele, and R. Werner, *Finitely correlated states on quantum spin chains*, Communications in Mathematical Physics, 144 (1992), pp. 443–490.

[9] I. Gavrilyuk and B. Khoromskij, *Quantized-TT-Cayley transform for computing the dynamics and the spectrum of high-dimensional Hamiltonians*, Comput. Methods in Appl. Math., 11 (2011), pp. 273–290.

[10] L. Grasedyck, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2029–2054.

[11] L. Grasedyck and W. Hackbusch, *An introduction to hierarchical ($\mathcal{H}$-) and TT-rank of tensors with examples*, Comput. Meth. Appl. Math., 3 (2011), pp. 291–304.

[12] W. Hackbusch and S. Kühn, *A new scheme for the tensor representation*, J. Fourier Anal. Appl., 15 (2009), pp. 706–722.

[13] S. Holtz, T. Rohwedder, and R. Schneider, *On manifolds of tensors of fixed TT–rank*, Numer. Math., (2011).

[14] T. Huckle, K. Waldherr, and T. Schulte-Herbruggen, *Computations of quantum tensor networks*, Preprint , TU München, München, 2010.

[15] J. O. Jung and R. B. Gerber, *Vibrational wave functions and energy levels of large anharmonic clusters: A vibrational SCF study of (Ar)13*, J. Chem. Phys., 105 (1996), p. 10682.

[16] ——, *Vibrational wave functions and spectroscopy of (H2O)n, n=2,3,4,5: Vibrational self-consistent field with correlation corrections*, J. Chem. Phys., 105 (1996), p. 10332.

[17] V. Kazeev and B. N. Khoromskij, *Explicit low-rank QTT representation of Laplace operator and its inverse*, Preprint 75, MPI MIS, Leipzig, 2010.

[18] V. Khoromskaia, B. N. Khoromskij, and R. Schneider, *QTT representation of the Hartree and exchange operators in electronic structure calculations*, Comput. Meth. Appl. Math, 11 (2011), pp. 327–341.

[19] B. Khoromskij and C. Schwab, *Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs*, SIAM J. Sci. Comput., 33 (2011), pp. 1–25.

[20] B. N. Khoromskij, *Structured rank-*$(r_1, \ldots, r_d)$ *decomposition of function-related operators in* $\mathbb{R}^d$, Comput. Meth. Appl. Math, 6 (2006), pp. 194–220.

[21] ——, *Tensor-structured preconditioners and approximate inverse of elliptic operators in* $\mathbb{R}^d$, Constr. Approx, (2009), pp. 599–620.

[22] ——, *Introduction to tensor numerical methods in scientific computing*, Preprint, Lecture Notes 06-2011, University of Zürich, 2010.

[23] ——, $\mathcal{O}(d \log n)$*–Quantics approximation of* N–d *tensors in high-dimensional numerical modeling*, Constr. Appr., 34 (2011), pp. 257–280.

[24] ——, *Tensor-structured numerical methods in scientific computing: survey on recent advances*, Chemometr. Intell. Lab. Syst., (2011).

[25] B. N. Khoromskij and V. Khoromskaia, *Multigrid accelerated tensor approximation of function related multidimensional arrays*, SIAM J. Sci. Comput., 31 (2009), pp. 3002–3026.

[26] B. N. Khoromskij, V. Khoromskaia, and H.-J. Flad., *Numerical solution of the Hartree–Fock equation in multilevel tensor-structured format*, SIAM J. Sci. Comput., 33 (2011), pp. 45–65.

[27] B. N. Khoromskij and I. V. Oseledets, *Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs*, Comput. Meth. Appl. Math., 10 (2010), pp. 376–394.

[28] ——, *QTT-approximation of elliptic solution operators in high dimensions*, Rus. J. Numer. Anal. Math. Model., 26 (2011), pp. 303–322.

[29] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[30] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.

[31] B. Kuhn, T. Rizzo, D. Luckhaus, M. Quack, and M. Suhm, *A new six-dimensional analytical potential up to chemically significant energies for the electronic ground state of hydrogen peroxide*, J. Chem. Phys., 111 (1999), p. 2565.

[32] C. Lubich, *From quantum to classical molecular dynamics: reduced models and numerical analysis*, EMS, Zürich, 2008.

[33] U. Manthe, *A multilayer multiconfigurational time-dependent Hartree approach for quantum dynamics on general potential energy surfaces.*, J. Chem. Phys., 128 (2008), p. 164116.

[34] S. Manzhos and T. Carrington Jr, *Using redundant coordinates to represent potential energy surfaces with lower-dimensional functions*, J. Chem. Phys., 127 (2007), p. 014103.

[35] H.-D. Meyer, F. Gatti, and G. A. Worth, *Multidimensional Quantum Dynamics: MCTDH Theory and Applications*, Wiley-VCH, Weinheim, 2009.

[36] H.-D. Meyer, U. Manthe, and L. Cederbaum, *The multi-configurational time-dependent Hartree approach*, Chem. Phys. Lett., 165 (1990), pp. 73–78.

[37] M. Nest and H.-D. Meyer, *Benchmark calculations on high-dimensional Henon-Heiles potentials with the multi-configuration time dependent Hartree (MCTDH) method*, J. Chem. Phys., 117 (2002), p. 10499.

[38] I. Oseledets and E. Tyrtyshnikov, *Recursive decomposition of multidimensional tensors*, Doklady Math., 427 (2009), pp. 14–16.

[39] I. V. Oseledets, *Compact matrix form of the* d*-dimensional tensor decomposition*, Preprint 2009-01, INM RAS, Moscow, 2009.

[40] ——, *Approximation of* $2^d \times 2^d$ *matrices using tensor decomposition*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2130–2145.

[41] ——, *Constructive representation of functions in tensor formats*, Preprint 2010-04, INM RAS, Moscow, 2010.

[42] ——, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.

[43] I. V. Oseledets and E. E. Tyrtyshnikov, *Breaking the curse of dimensionality, or how to use svd in many dimensions*, SIAM J. Sci. Comput., 31 (2009), pp. 3744–3759.

[44] ——, *TT-cross algorithm for the approximation of multidimensional arrays*, Linear Algebra Appl., 432 (2010), pp. 70–88.

[45] S. Östlund and S. Rommer, *Thermodynamic limit of density matrix renormalization*, Phys. Rev. Lett., 75 (1995), pp. 3537–3540.

[46] H. Partridge and D. Schwenke, *The determination of an accurate isotope dependent potential energy surface for water from extensive ab initio calculations and experimental data*, J. Chem. Phys., 106 (1997), p. 4618.

[47] F. Richter, M. Hochlaf, P. Rosmus, F. Gatti, and H.-D. Meyer, *A study of the mode-selective trans–cis isomerization in HONO using ab initio methodology.*, J. Chem. Phys., 120 (2004), pp. 1306–17.

[48] T. Rohwedder, S. Holtz, and R. Schneider, *The alternation least squares scheme for tensor optimisation in the TT-format*, Preprint DFG-Schwerpunktprogramm 1234 71, 2010.

[49] A. Roitberg, R. B. Gerber, R. Elber, and M. Ratner, *Anharmonic wave functions of proteins: quantum self-consistent field calculations of BPTI*, Science, 268 (1995), pp. 1319–1322.

[50] G. Vidal, *Efficient classical simulation of slightly entangled quantum computations*, Phys. Rev. Lett., 91 (2003).

[51] S. R. White, *Density-matrix algorithms for quantum renormalization groups*, Phys. Rev. B, 48 (1993), pp. 10345–10356.

[52] S. R. White and D. Huse, *Numerical renormalization-group study of low-lying eigenstates of the antiferromagnetic S=1 Heisenberg chain*, Phys. Rev. B, 48 (1993), pp. 3844–3852.