Computing vibrational energy levels by solving linear equations using a tensor method with an imposed rank

Sangeeth Das Kallullathil[a] and Tucker Carrington Jr.[b]
Chemistry Department, Queen's University,
Kingston, Ontario K7L 3N6, Canada

(Dated: October 15, 2021)

Abstract

Present day computers do not have enough memory to store the high-dimensional tensors required when using a direct product basis to compute vibrational energy levels of a polyatomic molecule with more than about 5 atoms. One way to deal with this problem is to represent tensors using a tensor format. In this paper, we use CP format. Energy levels are computed by building a basis from vectors obtained by solving linear equations. The method can be thought of as a CP realization of a block inverse iteration method with multiple shifts. The CP rank of the tensors is fixed and the linear equations are solved with an Alternating Least Squares method. There is no need for rank reduction, no need for orthogonalization, and tensors with rank larger than the fixed rank used to solve the linear equations are never generated. The ideas are tested by computing vibrational energy levels of a 64-D bilinearly coupled model Hamiltonian and of acetonitrile(12-D).

[a] Electronic address: 17sdk2@queensu.ca
[b] Electronic address: Tucker.Carrington@queensu.ca

I. INTRODUCTION

The most general and systematic strategy for computing a vibrational spectrum from a potential energy surface (PES), without making approximations, is to expand wavefunctions in a basis and solve a matrix eigenvalue problem obtained from a variational method.[1–4] Implementing this strategy is computationally costly because using established methods both the computer time and the computer memory required increase rapidly with the size of the basis. A direct product basis, each of whose functions is a product of univariate functions,

$$\Phi_{i_1,\cdots,i_D}(q_1,...,q_D) =\varphi^{(1)}_{i_1}(q_1)\,\varphi^{(2)}_{i_2}(q_2)\ldots\varphi^{(D)}_{i_D}(q_D) \tag{1}$$

has the important advantage of simplicity. In this paper, D is the number of coordinates and the coordinates are q_c, $c = 1, \cdots, D$. The obvious disadvantage of a direct product basis is its size, n^D. n is a representative value of n_c, $c = 1, \cdots, D$, which is the number of 1-D basis functions for coordinate q_c. To simplify the notation we use n throughout the paper, but it is simple to use different n_c for different coordinates. Despite its size, it is possible to use a direct product basis by exploiting its structure to facilitate the evaluation of matrix-vector products (MVPs) and employing iterative methods to solve the matrix eigenvalue problem.[5–8] For molecules with more than five atoms, it becomes impossible to store (even) vectors with n^D components and one needs new ideas to compute spectra.

In this paper, we introduce a new tensor method for using iterative methods with a direct product basis. There is no need to store vectors with n^D components. To develop methods whose memory cost does not scale exponentially we have used what mathematicians call canonical polyadic (CP) format.[9–12] The first such method was the Reduced Rank Block Power Method (RRBPM).[13] It uses sum-of-product (SOP) basis functions, $F(q_1, ..., q_D)$, that are linear combinations of the functions of a direct product basis,

$$F(q_1,...,q_D) = \sum_{i_1=1}^{n}\ldots\sum_{i_D=1}^{n} F_{i_1,...,i_D}\;\varphi^{(1)}_{i_1}(q_1)\,\varphi^{(2)}_{i_2}(q_2)\ldots\varphi^{(D)}_{i_D}(q_D) \tag{2}$$

Each basis function is a sum of R products and it is not necessary to store $F_{i_1,...,i_D}$ as n^D numbers. Instead, one exploits

$$\mathbf{F} = \sum_{\ell=1}^{R}\bigotimes_{c=1}^{D}\mathbf{f}_{c,\ell} \tag{3}$$

and stores RD vectors $\mathbf{f}_{c,\ell}$. R is the number of terms in Eq. (3) and is called the rank. $\mathbf{f}_{c,\ell}$ has only n components. The memory cost scales linearly with D. In terms of components Eq. (3) is

$$F_{i_1,i_2,..,i_d} = \sum_{\ell=1}^{R} \prod_{c=1}^{D} f_{i_c}^{(c,\ell)} \tag{4}$$

The Multiconfiguration Time-dependent Hartree (MCTDH) method can also be considered a tensor method.[14–16] It, however, uses not CP format but Tucker format.[9] Because it uses Tucker format, the memory cost of an MCTDH calculation scales exponentially with D. The advantages of using CP format for electronic structure problems have also been recognized.[17] Multilayer-MCTDH uses basis functions that are products of multivariate functions.[18–20] It reduces the cost of an MCTDH calculation and is widely used when the PES is a SOP. The density matrix renormalization group (DMRG) method is often used to calculate a ground state energy (but not the corresponding wavefunction).[21] Related ideas have also been used to compute a small number of vibrational[22,23] or electronic[24] states. Larsson has used a DMRG-type optimization algorithm in conjunction with multilayer trees.[25]

There also exist non-tensor iterative methods that make variational calculations possible for molecules with more than five atoms. They all use a basis that is not a direct product. One class of such methods begins, at least conceptually, with a direct product basis and prunes away unnecessary basis functions. Some of these approaches work only if the PES has a special (e.g. SOP) form[26–29], and some work with a general PES[30–32]. Another class of such methods uses basis functions that are products of multivariate functions which are eigenfunctions of sub-problems.[33–37] For methods of both classes what is tricky is devising a scheme for evaluating MVPs. Selected configuration interaction methods have also been used successfully for large molecules.[38–41]

Tensor algorithms are only useful if the Hamiltonian operator has a compatible form. To use the RRBPM, the Hamiltonian must be a SOP. To use most MCTDH programs, the Hamiltonian must also be a SOP (however, there are non-SOP MCTDH calculations using Manthe's Correlation Discrete Variable Representation[42] or collocation[43]). If the Hamiltonian does not have the required form it can, at the cost of some extra calculations, be replaced by an operator that is approximately equal to the original Hamiltonian, but does have the required form.[44–48] For many molecules with more than five atoms, the best available PES is a SOP.[49]

Although the original RRBPM eliminates the memory problem one confronts when doing a variational calculation, it requires a lot of computer time. It is inefficient because the (shifted) power method requires many MVPs and because after each MVP it is necessary to reduce the rank of the output vector. The rank of the output vector is RT, the product of the rank of the input vector (R) and the number of terms in the SOP Hamiltonian, in this paper denoted by T. Several related, but more efficient, ideas have been introduced. One divides the full problem into a sequence of sub-problems each with a subset of the full D coordinates by imposing a tree structure.[50] This Hierarchical H-RRBPM was later improved by intertwining the matrix-vector products and the rank reduction algorithm.[51] Rakhuba et al. replaced the CP format used in the original RRBPM with the Tensor Train (TT) or Matrix Product State format and replaced the shifted power method eigensolver with a combination of LOBPCG (locally optimal block preconditioned conjugate gradient) and inverse iteration.[53] Because PESs for molecules with more than five atoms are often determined in SOP form, they convert a CP PES tensor to TT. In Larsson's Tree Tensor Network States method a tensor tree is used and the Schrödinger equation is solved with a DMRG-type optimization algorithm. In practice, he computes states one by one.[25]

Like the method of Rakhuba and Oseledets[53] our method is designed to simultaneously calculate all states in a given energy window. Compared to the RRBPM and the method of Rakhuba et al., the method of this paper has important advantages: no orthogonalization is necessary; no rank reduction ("rounding") is required. Rank reduction is the most time consuming step in most RRBPM calculations. In the RRBPM, rank reduction is required after each MVP, after each orthogonalization, and after each update.[13] The cost of the new method is much lower than the cost of the RRBPM. The method has two components: 1) eigenvalues and eigenvectors of the direct product representation of the Hamiltonian operator, which is here denoted $\mathbf{H}$, are determined by projecting into a basis of filtered vectors; 2) the filtered vectors are computed by exploiting the SOP character of the Hamiltonian operator and the direct product nature of the primitive basis. Component 1) somewhat resembles Block Inverse Iteration,[54] the Filter Diagonalization method,[55–58] and the Rational Krylov method[59]. The filtered vectors are designed so that each is a linear combination of a small number of eigenvectors of $\mathbf{H}$ whose corresponding eigenvalues are in some selected energy range. We call the method CP-MSBII. It is a CP implementation of a Multiple Shift Block Inverse Iteration Method. Rank reduction is not required because filtered vectors are computed by solving linear equations while constraining the rank of the solution. CP-MSBII is tested by computing energy levels of a 64-D bilinearly coupled model Hamiltonian and acetonitrile.

II. CONSTRAINING THE CP RANK OF THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS

Ideal filtered basis vectors would be determined (see section III) by solving,

$$(\mathbf{H} - \sigma\mathbf{I})\mathbf{F}^{ex} = \mathbf{s} \; . \tag{5}$$

Both $\mathbf{s}$ and $\mathbf{F}^{ex}$ are tensors with n^D components. In the CP-MSBII method (see section III), $\mathbf{s}$ is in CP format and has rank R. $\mathbf{H}$ is an $n^D \times n^D$ matrix representation of a SOP operator. Rather than computing $\mathbf{F}^{ex}$, we find approximate solutions, henceforth denoted $\mathbf{F}$, by replacing $\mathbf{F}^{ex}$ in Eq. (5) with $\mathbf{F}$ in Eq. (3) which is in CP format, and requiring that $\mathbf{F}$ have rank R. To determine basis vectors from which one can compute accurate eigenvalues (see section III) it is sufficient to solve Eq. (5) approximately. In this section, we explain how to compute $\mathbf{F}$ by minimizing $||(\mathbf{H} - \sigma\mathbf{I})\mathbf{F} - \mathbf{s}||$. An Alternating Least Squares (ALS) algorithm is used. The ALS solution is more accurate if the exact solution of Eq. (5) is low rank. This will be the case if $\mathbf{s}$ is a linear combination of a small number of the eigenvectors of $\mathbf{H}$ with the smallest eigenvalues. In section III we discuss choosing $\mathbf{s}$ to satisfy this requirement.

Using ALS, we sequentially determine $\boldsymbol{\mathcal{F}}_j$ for $j = 1, 2, \cdots, D$, keeping $\boldsymbol{\mathcal{F}}_{c \neq j}$ fixed. $\boldsymbol{\mathcal{F}}_c$ is defined so that its ℓth column, $\ell = 1, ..., R$, is $\mathbf{f}_{c,\ell}$ (see Eq. (3)). The number of elements in the ℓth column is n. $\boldsymbol{\mathcal{F}}_c$ is a collection of parameters for coordinate q_c. The columns of $\boldsymbol{\mathcal{F}}_c$ are not normalized. Computing $\boldsymbol{\mathcal{F}}_j$ for $j = 1, 2, \cdots, D$ constitutes one sweep. The most accurate possible ALS solution is obtained by doing many sweeps, however, we find that solutions determined with a single sweep are accurate enough.

The ALS algorithm we use is modelled on the one in Ref. 60. For any coordinate j, we define

$$\mathbf{W}_j^F(\boldsymbol{\mathcal{F}}_1, ..., \boldsymbol{\mathcal{F}}_{j-1}, \boldsymbol{\mathcal{F}}_{j+1}, ..., \boldsymbol{\mathcal{F}}_D) \in \mathbb{R}^{n^D \times nR} \; , \tag{6}$$

which depends on the parameters of all coordinates except $\boldsymbol{\mathcal{F}}_j$, so that

$$\mathbf{F} = \mathbf{W}_j^F vec(\boldsymbol{\mathcal{F}}_j) \ . \tag{7}$$

$vec(\boldsymbol{\mathcal{F}}_j)$ is a $nr \times 1$ column made by stacking the columns of $\boldsymbol{\mathcal{F}}_j$. It is straightforward to show that

$$\begin{aligned} \mathbf{W}_j^F(\boldsymbol{\mathcal{F}}_1, ..., \boldsymbol{\mathcal{F}}_{j-1}, \boldsymbol{\mathcal{F}}_{j+1}, ..., \boldsymbol{\mathcal{F}}_D) = (\mathbf{f}_{1,1} \otimes \cdots \otimes \mathbf{f}_{j-1,1} \otimes \mathbf{I}_j \otimes \mathbf{f}_{j+1,1} \otimes \cdots \otimes \mathbf{f}_{D,1} \Big| \\ \mathbf{f}_{1,2} \otimes \cdots \otimes \mathbf{f}_{j-1,2} \otimes \mathbf{I}_j \otimes \mathbf{f}_{j+1,2} \otimes \cdots \otimes \mathbf{f}_{D,2} \Big| \ , \\ \cdots \mathbf{f}_{1,R} \otimes \cdots \otimes \mathbf{f}_{j-1,R} \otimes \mathbf{I}_j \otimes \mathbf{f}_{j+1,R} \otimes \cdots \otimes \mathbf{f}_{D,R}) \end{aligned} \tag{8}$$

where $\mathbf{I}_j$ is the $n \times n$ identity matrix. The vertical lines in Eq. (8) divide slices of the matrix each of which is $n^D \times n$ matrix. $\hat{H}$ is a SOP operator

$$H(q_1, ..., q_D) = \sum_{t=1}^{T} \prod_{c=1}^{D} \hat{h}_c^t(q_c) \ . \tag{9}$$

In Eq. (9), $h_c^t(q_c)$ is a one-dimensional operator. $(\mathbf{H} - \sigma\mathbf{I})$ is a CP matrix,

$$(\mathbf{H} - \sigma\mathbf{I}) = \sum_{t=1}^{T+1} \bigotimes_{c=1}^{D} \mathbf{h}_c^t(q_c). \tag{10}$$

The $(T+1)$th term is $\sigma\mathbf{I}$. The right side of Eq. (5) is written

$$\mathbf{s} = \sum_{\ell=1}^{R} \bigotimes_{c=1}^{D} \mathbf{g}_{c,\ell} \ . \tag{11}$$

To determine $\boldsymbol{\mathcal{F}}_j$ we replace $\mathbf{F}^{ex}$ in Eq. (5) with $\mathbf{F}$ and then use Eq. (7) and multiply on the left by $(\mathbf{W}_j^F)^T$

$$(\mathbf{W}_j^F)^T(\mathbf{H} - \sigma\mathbf{I})\mathbf{W}_j^F vec(\boldsymbol{\mathcal{F}}_j) = (\mathbf{W}_j^F)^T\mathbf{s} \ . \tag{12}$$

Using Eq. (8), Eq. (10), Eq. (11), and $\mathbf{s} = \mathbf{W}_j^G \ vec(\boldsymbol{\mathcal{G}}_j)$ one finds a small linear system that can be solved for $vec(\boldsymbol{\mathcal{F}}_j)$,

$$\left[\sum_{t=1}^{T+1} \left(\bigodot_{c\neq j}^{D} (\boldsymbol{\mathcal{F}}_c^T \mathbf{h}_c^t \boldsymbol{\mathcal{F}}_c)\right) \otimes \mathbf{h}_j^t\right] vec(\boldsymbol{\mathcal{F}}_j) = \sum_{\ell''=1}^{R} \left(\bigodot_{c\neq j}^{D} (\boldsymbol{\mathcal{F}}_c^T \mathbf{g}_{c,\ell''})\right) \otimes \mathbf{g}_{j,\ell''} \ , \tag{13}$$

where $\bigodot$ denotes element wise (or Hadamard) multiplication.

When solving Eq. (13), $\boldsymbol{\mathcal{F}}_{c\neq j}$ are fixed. Eq. (13) is solved for $j = 1, 2, \cdots, D$. It has the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{A}$ is an $nR \times nR$ matrix. ALS converts a linear system with a $n^D \times n^D$

matrix into a set of linear problems with $nR \times nR$ matrices. To obtain the best possible solution, one would solve Eq. (13), for $j = 1, 2, \cdots, D$, N_{ALS} times, each time updating the $\boldsymbol{\mathcal{F}}_{c \neq j}$ that are fixed. Instead we solve only once for each j, i.e. $N_{ALS} = 1$ (see Section IV). If Eq. (13) is written $\mathbf{Ax} = \mathbf{b}$, then elements of $\mathbf{A}$ and $\mathbf{b}$ are

$$A(\ell', i'_j, \ell, i_j) = \sum_{t=1}^{T+1} \left(\prod_{c \neq j}^{D} (\mathbf{f}_{c,\ell'}^T \mathbf{h}_c^t \mathbf{f}_{c,\ell}) \right) h_{i'_j, i_j}^t, \tag{14}$$

and

$$b(\ell', i'_j) = \sum_{\ell''=1}^{R} \left[\prod_{c \neq j}^{D} (\mathbf{f}_{c,\ell'}^T \mathbf{g}_{c,\ell''}) \right] g_{i'_j}^{(j,\ell'')}. \tag{15}$$

To avoid calculating $\mathbf{A}$ and $\mathbf{b}$ from scratch for each j, we re-write these equations

$$A(\ell', i'_j, \ell, i_j) = \sum_{t=1}^{T+1} M_{\neq j}^t(\ell', \ell) h_{i'_j, i_j}^t, \tag{16}$$

$$b(\ell', i'_j) = \sum_{\ell''=1}^{R} P_{\neq j}(\ell', \ell'') \; g_{i'_j}^{(j,\ell'')} \;, \tag{17}$$

where

$$M_{\neq j}^t(\ell', \ell) = \prod_{c \neq j}^{D} (\mathbf{f}_{c,\ell'}^T \mathbf{h}_c^t \mathbf{f}_{c,\ell}), \tag{18}$$

and

$$P_{\neq j}(\ell', \ell'') = \prod_{c \neq j}^{D} (\mathbf{f}_{c,\ell'}^T \mathbf{g}_{c,\ell''}) \;. \tag{19}$$

Because it is small, we solve the linear system in Eq. (13) with methods of direct linear algebra. The cost scales as $(nR)^3$. It would also be possible to use a (preconditioned) iterative linear solver. When T is small iterative solvers might be less costly.

A pseudocode for the implementation of the method is given in algorithm 1. $\mathbf{M}_{\neq j}^t$ and $\mathbf{P}_{\neq j}$ do not need to be calculated from scratch for each j. They depend on $\boldsymbol{\mathcal{F}}_{c \neq j}$. After computing $\boldsymbol{\mathcal{F}}_j$, it is used to update $M_{\forall c}^t(\ell', \ell)$ and $P_{\forall c}(\ell', \ell'')$. The cost of constructing $\mathbf{M}_j^t$ for one j is n^2R+nR^2 and the cost of building $\mathbf{M}_{\forall j}^t$ is $D(n^2R+nR^2)$. Computing $\mathbf{A}$ therefore requires $(T+1)[D(n^2R+nR^2)+(nR)^2]$ operations. nR^2 is the cost of making one $\mathbf{P}_j$ matrix and $D(nR^2)$ is the cost of making a $\mathbf{P}_{\forall j}$ matrix. Computing $\mathbf{b}$ therefore costs $D(nR^2) + R^2n$. The algorithm requires the matrices,

$$M^t_{\forall c}(\ell', \ell) = \prod_{c=1}^{D} (\mathbf{f}^T_{c,\ell'} \mathbf{h}^t_c \mathbf{f}_{c,\ell}), \tag{20}$$

$$P_{\forall c}(\ell', \ell'') = \prod_{c=1}^{D} (\mathbf{f}^T_{c,\ell'} \mathbf{g}_{c,\ell''}), \tag{21}$$

$$M^t_j(\ell', \ell) = \mathbf{f}^T_{j,\ell'} \mathbf{h}^t_j \mathbf{f}_{j,\ell} \ , \tag{22}$$

and

$$P_j(\ell', \ell'') = \mathbf{f}^T_{j,\ell'} \mathbf{g}_{j,\ell''}. \tag{23}$$

$\mathbf{M}^t_{\neq j}$ and $\mathbf{P}_{\neq j}$ are calculated from

$$M^t_{\neq j}(\ell', \ell) = M^t_{\forall c}(\ell', \ell) / M^t_j(\ell', \ell) \tag{24}$$

and

$$P_{\neq j}(\ell', \ell'') = P_{\forall c}(\ell', \ell'') / P_j(\ell', \ell''). \tag{25}$$

III. MULTIPLE SHIFT BLOCK INVERSE ITERATION EIGENSOLVER

Our filtered vectors are computed by using ALS to solve successive systems of linear equations. A filtered vector, $\tilde{\mathbf{F}}^b_v$, is designed to approximate $(\mathbf{H} - \sigma^b_v \mathbf{I})^{-P} \mathbf{s}^b_v$. For example, when $P = 2$,

$$\tilde{\mathbf{F}}^b_v = (\mathbf{H} - \sigma^b_v \mathbf{I})^{-1} \mathbf{L}^b_v, \tag{26}$$

where

$$\mathbf{L}^b_v = \frac{\tilde{\mathbf{L}}^b_v}{\left\| \tilde{\mathbf{L}}^b_v \right\|}, \tag{27}$$

and

$$\tilde{\mathbf{L}}_v^b = (\mathbf{H} - \sigma_v^b \mathbf{I})^{-1} \mathbf{s}_v^b. \tag{28}$$

Both $\tilde{\mathbf{F}}_v^b$ and $\tilde{\mathbf{L}}_v^b$ are computed using the CP procedure of section II. The vectors are labelled by $b = 1, \cdots, B$ and by $v = 1, \cdots, S_b$. b labels a block of vectors. S_b is the number of vectors in a block. The σ_v^b values are referred to as shifts. $\mathbf{s}_v^b$ is a start vector. The start vector is in CP format and has rank R. In some cases $\mathbf{s}_v^b = \mathbf{s}_{v'}^b$. We normalize $\tilde{\mathbf{F}}_v^b$ to obtain $\mathbf{F}_v^b$ and build a basis from the $\mathbf{F}_v^b$. The basis vectors are not orthogonal. Once the filtered vectors are known, we project into the space the filtered vectors span, and solve a generalized eigenvalue problem,

$$\mathbf{H}^{\mathscr{F}} \mathbf{U} = \mathbf{S}^{\mathscr{F}} \mathbf{U} \mathbf{E} \ , \tag{29}$$

to determine eigenvalues and eigenvectors of $\mathbf{H}$. In Eq. (29), $\mathbf{H}^{\mathscr{F}} = \boldsymbol{\mathscr{F}}^T \mathbf{H} \boldsymbol{\mathscr{F}}$ and $\mathbf{S}^{\mathscr{F}} = \boldsymbol{\mathscr{F}}^T \boldsymbol{\mathscr{F}}$, and $\boldsymbol{\mathscr{F}}$ is the matrix whose columns are the BS filtered vectors $\mathbf{F}_v^b$. The cost of computing $\mathbf{S}^{\mathscr{F}}$ and $\mathbf{H}^{\mathscr{F}}$ scale as $B^2S^2DR^2n$ and $BS(DRTn^2 + BS(DR^2Tn))$, respectively. There is no need to orthogonalize vectors or reduce their rank. Rank reduction is by far the most time consuming step in the RRBPM of Ref. 13. The memory cost of the calculation is determined by the memory required to store the filtered basis: $BSDRn$.

If $\mathbf{F}_v^b$ were written as a linear combination of the eigenvectors of $\mathbf{H}$, the dominant contributions would be from eigenvectors whose corresponding eigenvalues are close to σ_v^b. The σ_v^b are distributed in a chosen energy range $[E_{min} – E_{max}]$ and the basis of $\mathbf{F}_v^b$ is therefore good for the purpose of computing energy levels in the range. We want the lowest energy levels, however, it is better not to begin the range at the energy of the minimum of the PES, but just below the ground state. In practice, the bottom of the range is slightly below the ground state of the harmonic Hamiltonian extracted from the full Hamiltonian by discarding all the coupling and anharmonicity.

The MSBII eigensolver is related to Filter Diagonalization[55–58], Inverse Iteration[54], and Rational Krylov methods[59]. We choose $\sigma_v^{b>1}$ so that their distribution is close to the distribution of the eigenvalues of $\mathbf{H}$ in the desired range and choose $\mathbf{s}_v^b$ reasonably close to eigenvectors of $\mathbf{H}$. This has two advantages: 1) the rank of $\mathbf{F}_v^b$ is low and therefore ALS solutions of the linear systems are accurate; 2) inner products of $\mathbf{F}_v^b$ with eigenvectors of $\mathbf{H}$ whose eigenvalues are not in the chosen energy range are small. The magnitude of the rank and the ALS error in 1) and the inner products in 2) can be reduced by increasing P or N_{ALS}. However, it is always helpful choose $\mathbf{s}_v^b$ that are close to eigenvectors of $\mathbf{H}$ and σ_v^b so that their distribution approximates the distribution of the eigenvalues of $\mathbf{H}$. The rank of $\mathbf{F}_v^b$ is low when it is a linear combination of a small number of exact eigenvectors, each of which is assumed to be low rank. The method will only work if the exact eigenvectors are low rank.

Consider first, $b = 1$. We choose S_1 $\sigma_v^{b=1}$ so that they are equally spaced and within the energy range. We use $\mathbf{s}_v^{b=1}$ that are nearly parallel to low rank vectors. There are L distinct $\mathbf{s}_v^{b=1}$. Each distinct $\mathbf{s}_t^{b=1}$ $t = 1, \cdots, L$ is the sum of a vector that corresponds to the tth energy level of the harmonic Hamiltonian referred to above and $R - 1$ terms each of which is an outer product of D of random vectors. Each component of the random vectors is a random number between 0 and 1 multiplied by 10^{-2}. It is important that the random numbers be much smaller than one. This ensures that each start vector has significant overlap with an exact eigenvector of $\mathbf{H}$ and hence low rank. S_1 is always chosen so that $S_1 > L$. This procedure for choosing $\mathbf{s}_v^{b=1}$ is cheap, but yields start vectors that are linear combinations of a small number of exact eigenvectors and hence low rank vectors. The method also works if each start vector is a linear combination of many exact eigenvectors, but in that case it is necessary to increase P (or N_{ALS}). $\sigma_v^{b=1}$ is linked to the $\mathbf{s}_t^{b=1}$ for which the corresponding harmonic level is closest. Some $\mathbf{s}_t^{b=1}$ are linked to more than one $\sigma_v^{b=1}$ ($S_1 > L$). $P = 2$ is used to obtain the basis $\mathscr{F}^1 = \mathbf{F}_v^{b=1}$, $v = 1, \cdots, S_1$. Eigenvalues computed in the $\mathscr{F}^1$ basis are not accurate. One way to improve the accuracy is to choose $P > 2$ and increase S_1 (see section IV). We find that it is less costly to increase B and to choose some of the $\sigma_v^{b=2}$ equal to $\mathscr{F}^1$ eigenvalues. If the distribution of the exact eigenvalues is non-uniform, the equally spaced shifts used for $b = 1$ are not ideal for two reasons. 1) The filtered vector that corresponds to a shift that is far from all exact eigenvalues will have small overlaps with all the exact eigenvectors and be a poor basis vector. 2) It is useful to have more shifts in dense regions of the spectrum. $P = 2$ is large enough to ensure that the distribution of the $\mathscr{F}^1$ eigenvalues is close to the distribution of the exact eigenvalues.

For $b = 2$, J of the $\sigma_v^{b=2}$ are set equal to the eigenvalues within the range obtained by solving the eigenvalue problem with the the basis $\mathscr{F}^1$. Additional $\sigma_v^{b=2}$ are chosen as follows. We identify the $S_2 - J$ closest pairs of eigenvalues obtained from the $\mathscr{F}^1$ basis and set $\sigma_v^{b=2}$ $v = J + 1, \cdots, S_2$ equal to the values half between these pairs. We add $\sigma_v^{b=2}$ in the densest region(s) of the spectrum because the density of shifts ought to be close to the density of the exact eigenvalues (which is close to the density of the eigenvalues obtained from the $\boldsymbol{\mathscr{F}}^1$ basis). We increase S_2 until eigenvalues of Eq. (29) converge. The distribution of the $\sigma_v^{b=2}$ is closer to the distribution of the exact eigenvalues of $\mathbf{H}$ than is the distribution of the $\sigma_v^{b=1}$. The S_2 $\mathbf{s}_v^{b=2}$, start vectors are made using the same procedure as was used for $\mathbf{s}_v^{b=1}$, but with different random numbers and $\mathbf{F}_v^{b=2}$, $v = 1, \cdots, S_2$, are computed with some chosen value of P. For $b = 3, \cdots, B$, $\sigma_v^b = \sigma_v^{b=2}$ and $\mathbf{s}_v^{b=2}$ are made using the same procedure as was used for $\mathbf{s}_v^{b=1}$, but with different random numbers, and the value of P is the same as for $b = 2$. We also did calculations by updating σ_v^b for $b = 3, \cdots, B$ (similar to the way we update $\sigma_v^{b=2}$) and discovered that the results differ negligibly from those computed with $\sigma_v^b = \sigma_v^{b=2}$.

Algorithm 1 CP-MSBII

```
Input : $E_{min}$ and $E_{max}$; shift values $\sigma_v^{b=1}$; start vectors $\mathbf{s}_v^b$, where $b = 1, \cdots, B$ and
$v = 1, \cdots, S_1$, as
        explained in the text
Output : $BS_B$ eigenvalues and eigenvectors of $\mathbf{H}$

  1. Find better shift values for $b = 2, 3, \cdots, B$

     (a) Loop over vectors in the block $b = 1$
         for $v = 1$ to $S_1$
     (b) Loop over the number of applications of the inverse
         for $p = 1$ to $2$
           i. Call $[\mathbf{F}_v^1]$ = CPII($\mathbf{s}_v^1$,$\sigma_v^1$)
          ii. $\mathbf{s}_v^1 \leftarrow \mathbf{F}_v^1$
     (c) Solve generalized eigenvalue problem
           i. Build matrix of filtered vectors, $\boldsymbol{\mathscr{F}} = [\mathbf{F}_1^1, ..., \mathbf{F}_{S_1}^1]$
          ii. Compute $\mathbf{H}^{\mathscr{F}} = \boldsymbol{\mathscr{F}}^T \mathbf{H} \boldsymbol{\mathscr{F}}$ and $\mathbf{S}^{\mathscr{F}} = \boldsymbol{\mathscr{F}}^T \boldsymbol{\mathscr{F}}$.
         iii. Solve $\mathbf{H}^{\mathscr{F}} \mathbf{U} = \mathbf{S}^{\mathscr{F}} \mathbf{U} \mathbf{E}$
          iv. $\sigma_1^2$,..., $\sigma_J^2 \leftarrow$ diag($\mathbf{E}$) (explained in the text)
           v. Add shifts $\sigma_{J+1}^2, \cdots$, between the closest elements of $\mathbf{E}$ (explained in the
              text)
```

```
2. Other blocks

   (a) Loop over the number of blocks
       for $b = 2$ to $B$
   (b) Loop over the vectors in the block
       for $v = 1$ to $S_b$
   (c) Loop over the number of application of the inverse($P$)
       for $p = 1$ to $P$
         i. Call [$\mathbf{F}_b^v$]= CPII($\mathbf{s}_v^b$,$\sigma_v^b$)
        ii. $\mathbf{s}_v^b \leftarrow \mathbf{F}_v^b$

3. Compute eigenvalues,

   (a) Collect the filtered vectors, $\boldsymbol{\mathscr{F}} = [\mathbf{F}_1^1, ..., \mathbf{F}_{S_B}^B]$
   (b) Compute $\mathbf{H}^{\mathscr{F}} = \boldsymbol{\mathscr{F}}^T \mathbf{H} \boldsymbol{\mathscr{F}}$ and $\mathbf{S}^{\mathscr{F}} = \boldsymbol{\mathscr{F}}^T \boldsymbol{\mathscr{F}}$.
   (c) Solve $\mathbf{H}^{\mathscr{F}} \mathbf{U} = \mathbf{S}^{\mathscr{F}} \mathbf{U} \mathbf{E}$
```

Function [$\mathbf{F}$] = CPII($\mathbf{s}$,σ)

```
1. Initialize

   (a) Assign $(\mathbf{f}_{1,\ell}, ..., \mathbf{f}_{D,\ell}) \leftarrow (\mathbf{g}_{1,\ell}, ..., \mathbf{g}_{D,\ell}) \quad \forall\ \ell$
   (b) Construct $\mathbf{M}_{\forall c}^t$ and $\mathbf{P}_{\forall c}$

2. Solve for CP parameters

   for $\alpha = 1$ to $N_{ALS}$
   for $j = 1$ to $D$

   (a) Compute $\mathbf{M}_{\neq j}^t : M_{\neq j}^t(\ell', \ell) = M_{\forall c}^t(\ell', \ell) / M_j^t(\ell', \ell) \quad \forall\ \ell, \ell'$
   (b) Compute $\mathbf{P}_{\neq j} : P_{\neq j}(\ell', \ell'') = P_{\forall c}(\ell', \ell'') / P_j(\ell', \ell'') \quad \forall\ \ell', \ell''$
   (c) Compute $\mathbf{A} : A(\ell', i_j', \ell, i_j) = \sum_{t=1}^{T+1} M_{\neq j}^t(\ell', \ell) h_j^t(i_j', i_j) \quad \forall\ \ell', i_j', \ell, i_j$
   (d) Compute $\mathbf{b} : b(\ell', i_j') = \sum_{\ell''=1}^{R} P_{\neq j}(\ell', \ell'')\ g_{i_j'}^{(j,\ell'')} \quad \forall\ \ell', i_j'$
   (e) Solve the linear system for $\mathbf{x} : A(\ell', i_j', \ell, i_j) x(\ell, i_j) = b(\ell', i_j') \quad \forall\ \ell, i_j$
```

```
    (f) Update $\mathbf{f}_{j,\ell}$ by replacing $f_{i_j}^{(j,\ell)} \leftarrow x(\ell, i_j) \quad \forall\, \ell, i_j$
    (g) Compute $\mathbf{M}_j^t : M_j^t(\ell', \ell) = \mathbf{f}_{j,\ell'}^T \mathbf{h}_j^t \mathbf{f}_{j,\ell} \quad \forall\, \ell, \ell', t$
    (h) Compute $\mathbf{P}_j : P_j(\ell', \ell'') = \mathbf{f}_{j,\ell'}^T \mathbf{g}_{j,\ell''} \quad \forall\, \ell'', \ell'$
    (i) Update $\mathbf{M}_{\forall c}^t$ with new parameters $M_{\forall c}^t(\ell', \ell) = M_{\neq j}^t(\ell', \ell) M_j^t(\ell', \ell) \quad \forall\, \ell, \ell', t$
    (j) Update $\mathbf{P}_{\forall c}$ with new parameters $P_{\forall c}(\ell', \ell'') = P_{\neq j}(\ell', \ell'') P_j(\ell', \ell'') \quad \forall\, \ell'', \ell'$
3. Normalize $\mathbf{F} \leftarrow \frac{\mathbf{F}}{\|\mathbf{F}\|}$
```

In summary, with the CP-MSBII method, one computes *all* states in a given energy range by solving one generalized eigenvalue problem. No vectors with rank larger than R are needed and there is no rank reduction. In contrast, vectors with rank larger than R and rank reduction are intrinsic elements of RRBPM calculations and the only way to use the RRBPM without storing vectors whose rank is larger than R requires (see section IID of Ref. 51 and Ref. 52) some additional calculations. Although many states are calculated, there is no need to orthogonalize vectors

IV. CHOOSING THE PARAMETERS

In practice, we choose a value of P and then select S_b and B so that energy levels are accurate. In most calculations we have done, P is about 5. The basis $\mathscr{F}$ is the union of B bases each with S_b vectors. Increasing B increases the size of the basis and therefore the accuracy of all the energies we compute. Accuracy can also be improved by making the basis vectors "better". This can be accomplished by increasing P or by increasing N_{ALS}. Using $B > 1$ is particularly helpful if one desires more than about 10 states. Increasing P or N_{ALS} reduces the number of exact eigenvectors that have large overlaps with a given basis vector. Doubling P is slightly more costly than doubling N_{ALS}, but for a given filtered vector, doubling P makes the filtered vector "sharper", i.e., it has significant overlap with fewer exact eigenvectors of $\mathbf{H}$. We therefore use $N_{ALS} = 1$ and $P > 1$. Increasing P decreases the value of B required to obtain accurate eigenvalues; decreasing P increases the value of B required to obtain accurate eigenvalues. If P is larger, S_b must be larger. If P is increased and S_b is not increased, it may happen that the inner products of some exact eigenvectors of $\mathbf{H}$ with *all* the filtered vectors are small. In this case, the corresponding eigenvalues will have large errors. If P is large and there are no shift values close enough to an exact eigenvalue, then it will be impossible to accurately calculate that eigenvalue. In practice, it is good to choose a large value of S_1. We want to "saturate" the energy range with shifts. If S_1 is too large solving the generalized eigenvalue problem might become difficult, but we have not encountered problems. When P is too small then the filtered vectors are too "broad", which means that a large number of filtered vectors and a large range would be required to compute accurate eigenvalues. Increasing R never increases errors in energy levels. However, if P, S_b, and B are not chosen correctly increasing R will not yield accurate levels.

V. RESULTS AND DISCUSSION

We have tested the CP-MSBII method by using it to compute energy levels of two Hamiltonians. The first represents 64 bilinearly coupled oscillators. It has been used to test other tensor methods and is a good test because by using normal coordinates it is possible to obtain an analytic equation for the energies.[13,50,53] The second is a 12-D normal-coordinate Hamiltonian for acetonitrile.[13,50,53,61]

A. 64-D Bilinearly coupled model Hamiltonian

It is not possible to store in memory a vector with n^{64} components and therefore to use a direct product basis for a 64-D Hamiltonian, it is imperative that vectors be stored in some tensor format. The Hamiltonian is

$$\mathbf{H} = \sum_{i=1}^{64} \frac{\omega_i}{2} \left(p_i^2 + q_i^2\right) + \sum_{i=1}^{64} \sum_{j>i}^{64} \alpha_{ij} q_i q_j \tag{30}$$

where $p_j = -\mathrm{i}\frac{\partial}{\partial q_i}$ and α_{ij} is a coupling constant. Following Ref. 13, we choose $\omega_i = \sqrt{i/2}$ and $\alpha_{ij} = 0.1$. There are 2144 terms in the Hamiltonian. The coupling constants are small, but shift energy levels significantly.

In Table I, the 52 lowest energy levels are reported. Parameter values are in Table II. $R = 10$ is sufficient for computing good approximations for all the first 52 levels. Increasing R to 30 improves the accuracy of most of the energies. Energies near the top of the chosen range are less accurate. Increasing R further to 50 improves accuracy of the energies slightly. Errors for energies near the top of the range are reduced more by increasing E_{max}. This is due to the fact that filtered vectors with shift values near the top of the window have significant overlaps with exact eigenvectors whose eigenvalues are outside the chosen range. Energies with similar errors were computed by Rakhuba et al.[53] using the Tensor Train format and and Tensor Train rank $R_{TT} = 15$. Inner products and matrix-vector products in a Tensor Train method scale as R_{TT}^2. Their cost is similar to that of their CP counterparts if $R_{TT}^2 = R_{CP}$. In our calculations, we use $R_{CP} = 10$ or 30. To do the calculation, we require less than 1 GB of memory (without a tensor format, storing a vector with 10^{64} requires $\sim 10^{55}$ GB).

TABLE I: The lowest vibrational energy levels of the 64-D bilinearly coupled model Hamiltonian

n	E_{exact}	$E_{\text{calc}}(R = 10)$	$\frac{E_{\text{calc}} - E_{\text{exact}}}{E_{\text{exact}}}$	$E_{\text{calc}}(R = 30)$	$\frac{E_{\text{calc}} - E_{\text{exact}}}{E_{\text{exact}}}$
0	121.620947675	121.624944552	3.29E-05	121.621022847	6.18E-07
1	122.292357688	122.296156022	3.11E-05	122.292517450	1.31E-06
2	122.585637797	122.600676444	1.23E-04	122.586064924	3.48E-06
3	122.810589928	122.829500585	1.54E-04	122.811655595	8.68E-06
4	122.963767701	122.970808029	5.73E-05	122.964708420	7.65E-06
5	123.000212841	123.042659907	3.45E-04	123.025834375	2.08E-04
6	123.167268924	123.187860433	1.67E-04	123.182855770	1.27E-04
7	123.257047811	123.286986894	2.43E-04	123.267171988	8.21E-05
8	123.318299395	123.439491265	9.83E-04	123.349524289	2.53E-04
9	123.457188689	123.497051491	3.23E-04	123.495528490	3.11E-04
10	123.481999941	123.527292763	3.67E-04	123.521399813	3.19E-04
11	123.550327920	123.613924214	5.15E-04	123.571185189	1.69E-04
12	123.586467177	123.626585991	3.25E-04	123.602569323	1.30E-04
13	123.635177715	123.679438354	3.58E-04	123.642043976	5.55E-05
14	123.671622854	123.714428880	3.46E-04	123.700174729	2.31E-04
15	123.707892200	123.760056703	4.22E-04	123.741427062	2.71E-04
16	123.775280050	123.806388473	2.51E-04	123.814687966	3.18E-04

Table I continued

n	E_{exact}	$E_{\text{calc}}(R = 10)$	$\frac{E_{\text{calc}}-E_{\text{exact}}}{E_{\text{exact}}}$	$E_{\text{calc}}(R = 30)$	$\frac{E_{\text{calc}}-E_{\text{exact}}}{E_{\text{exact}}}$
17	123.822743139	123.865527911	3.46E-04	123.847749617	2.02E-04
18	123.838678938	123.892514173	4.35E-04	123.865889920	2.20E-04
19	123.928457824	123.989597028	4.93E-04	123.971047148	3.44E-04
20	123.931985750	123.991758896	4.82E-04	123.991696643	4.82E-04
21	123.964902964	124.055356641	7.30E-04	124.034119090	5.58E-04
22	123.989709409	124.041523875	4.18E-04	124.048187436	4.72E-04
23	124.000232180	124.065808250	5.29E-04	124.049110448	3.94E-04
24	124.036370297	124.133319041	7.82E-04	124.087447038	4.12E-04
25	124.128598703	124.171365532	3.45E-04	124.172361270	3.53E-04
26	124.131959047	124.184557799	4.24E-04	124.177460464	3.67E-04
27	124.136493406	124.186369599	4.02E-04	124.195949823	4.79E-04
28	124.153409954	124.227969314	6.01E-04	124.235559150	6.62E-04
29	124.189855094	124.261156702	5.74E-04	124.238324082	3.90E-04
30	124.221737933	124.295122012	5.91E-04	124.263569410	3.37E-04
31	124.232838814	124.310919909	6.29E-04	124.310582943	6.26E-04
32	124.257877190	124.349109296	7.34E-04	124.328252058	5.66E-04
33	124.282989518	124.370241764	7.02E-04	124.368560923	6.89E-04
34	124.306587728	124.384819787	6.29E-04	124.372588950	5.31E-04
35	124.325805190	124.387249183	4.94E-04	124.376453552	4.07E-04
36	124.343032868	124.403494874	4.86E-04	124.391574203	3.90E-04
37	124.356911177	124.434326981	6.23E-04	124.422354037	5.26E-04
38	124.379302214	124.438318854	4.74E-04	124.442685352	5.10E-04
39	124.379478007	124.466253196	6.98E-04	124.466382313	6.99E-04
40	124.415725719	124.472106328	4.53E-04	124.506299774	7.28E-04
41	124.421878812	124.498230715	6.14E-04	124.518724305	7.78E-04
42	124.446690063	124.544776408	7.88E-04	124.529083139	6.62E-04
43	124.494153152	124.546455950	4.20E-04	124.582245015	7.08E-04
44	124.502882243	124.560177758	4.60E-04	124.597988163	7.64E-04

Table I continued

n	E_{exact}	$E_{\text{calc}}(R = 10)$	$\frac{E_{\text{calc}}-E_{\text{exact}}}{E_{\text{exact}}}$	$E_{\text{calc}}(R = 30)$	$\frac{E_{\text{calc}}-E_{\text{exact}}}{E_{\text{exact}}}$
45	124.507941648	124.569208533	4.92E-04	124.604172798	7.73E-04
46	124.510088951	124.647918981	1.11E-03	124.631952394	9.79E-04
47	124.515018042	124.651712243	1.10E-03	124.645035315	1.04E-03
48	124.546534091	124.753160978	1.66E-03	124.654108479	8.64E-04
49	124.551157299	124.763592405	1.71E-03	124.655684714	8.39E-04
50	124.587515706	124.832768310	1.97E-03	124.689328103	8.17E-04
51	124.599867837	124.859193684	2.08E-03	124.722703027	9.86E-04

TABLE II: Parameters for the calculation with the 64-D bilinearly coupled oscillator Hamiltonian

Parameter	Value
Rank	10/30
ω_i	$\sqrt{i/2}$
α_{ij}	0.1
n	10 $\forall n_c$
N_{ALS}	1
P	10
B	2
S_1	100
S_2	52
E_{min}	121.60
E_{max}	124.60

B. Acetonitrile(CH_3CN)

It is important to test the CP-MSBII method on more a realistic Hamiltonian and we have therefore computed the first 50 vibrational energy levels of acetonitrile. We use normal coordinates. The acetonitrile normal coordinate Hamiltonian is now a common test problem. Although the Hamiltonian is only 12-D, the acetonitrile calculations are more costly. The Hamiltonian is

$$\mathbf{H} = \frac{1}{2}\sum_{i=1}^{12}\omega_i\left(p_i^2+q_i^2\right)+\frac{1}{6}\sum_{i=1}^{12}\sum_{j=1}^{12}\sum_{k=1}^{12}\phi_{ijk}^{(3)}q_iq_jq_k+\frac{1}{24}\sum_{i=1}^{12}\sum_{j=1}^{12}\sum_{k=1}^{12}\sum_{l=1}^{12}\phi_{ijkl}^{(4)}q_iq_jq_kq_l\ . \quad (31)$$

The parameters are taken from Ref. 61 and are based on values reported in Ref. 62, which however does not give all the parameters required for a 12-D PES. The Hamiltonian operator has 323 terms: 12 kinetic energy terms, 12 quadratic potential terms, 108 cubic potential terms, and 191 quartic potential terms. q_5, q_6, and q_7, q_8, and q_9, q_{10}, and q_{11}, q_{12} are degenerate coordinates pairs.

We use the same direct product basis as Ref. 13. The energy levels are reported in Table III. The calculation parameters are in Table IV. As reference values, we use those from a pruned basis Smolyak quadrature calculation.[61]

Lower energy levels are more accurate, as was the case also for the 64-D Hamiltonian. Again, this is might be due in part to error caused by constraining the rank. The filtered vectors we use as basis vectors are certainly much better than the shifted power method vectors of the RRBPM. However, it is possible that the rank of the filtered vectors required to compute accurate energy levels is larger than the rank of the eigenvectors we wish to compute. Filtered vectors for shift values near the top of the chosen range have significant overlaps with exact eigenvectors outside the range and this will limit the accuracy of the largest eigenvalues. Eigenvalues in a dense region of the spectrum and degenerate eigenvalues tend to be less accurate; this is true throughout the energy range. It is for this reason that we put more shifts in dense regions. Because coupling is more important for acetonitrile than for the 64-D Hamiltonian, we use larger values of B, P, and R.

Thomas and Carrington used their H-RRBPM to compute vibrational levels of acetonitrile. Their results and those of this paper are of similar accuracy but they use a smaller rank. It is because H-RRBPM breaks the full problem into a sequence of lower-dimensional problems that the H-RRBPM rank is smaller. An advantage of the CP-MSBII is that it never requires tensors with rank larger than R. In the H-RRBPM, tensors of large rank must be stored and then their rank is reduced. Rakhuba and Oseledets have used their TT method for acetonitrile. Using $R_{TT} = 40$, they obtain results somewhat more accurate than ours. $R_{TT} = 40$ corresponds to a CP rank of 1600 and we are using only $R = 400$.

TABLE III: Vibrational energy levels (cm^{-1}) of CH$_3$CN

Level	Sym.	E_{ref}	Rank=20	E-E_{ref}	Rank=200	E-E_{ref}	Rank=400	E-E_{ref}
ZPVE		9837.4073	9837.726	0.318	9837.418	0.011	9837.410	0.003
ν_{11}	E	360.991	361.08	0.09	360.99	0.00	360.99	0.00
		360.991	361.09	0.10	360.99	0.00	360.99	0.00
$2\nu_{11}$	E	723.181	723.47	0.29	723.20	0.02	723.19	0.01
		723.181	723.51	0.33	723.20	0.02	723.19	0.01
$2\nu_{11}$	A_1	723.827	724.14	0.31	723.85	0.02	723.84	0.01
	A_1	900.662	900.97	0.31	900.70	0.04	900.67	0.00
ν_9	E	1034.126	1034.39	0.26	1034.17	0.04	1034.13	0.01
		1034.126	1034.49	0.36	1034.18	0.05	1034.14	0.01
$3\nu_{11}$	A_2	1086.554	1087.10	0.54	1086.63	0.08	1086.58	0.03
$3\nu_{11}$	A_1	1086.554	1087.12	0.56	1086.68	0.12	1086.58	0.03
$3\nu_{11}$	E	1087.776	1088.30	0.52	1087.86	0.08	1087.80	0.02
		1087.776	1088.37	0.60	1088.03	0.25	1087.80	0.02
$\nu_4 + \nu_{11}$	E	1259.882	1260.66	0.78	1259.91	0.03	1259.89	0.01
		1259.882	1260.67	0.78	1260.03	0.15	1259.90	0.02
ν_3	A_1	1388.973	1389.56	0.59	1389.18	0.21	1389.09	0.12
$\nu_9 + \nu_{11}$	E	1394.689	1395.28	0.59	1394.85	0.16	1394.75	0.06
		1394.689	1395.48	0.79	1394.92	0.23	1394.83	0.15
$\nu_9 + \nu_{11}$	A_2	1394.907	1395.67	0.77	1395.05	0.15	1394.96	0.06
$\nu_9 + \nu_{11}$	A_1	1397.687	1398.34	0.65	1397.90	0.21	1397.82	0.13
$4\nu_{11}$	E	1451.101	1451.95	0.85	1451.23	0.13	1451.18	0.08
		1451.101	1451.97	0.87	1451.25	0.15	1451.19	0.09
$4\nu_{11}$	E	1452.827	1453.66	0.83	1452.96	0.14	1452.89	0.06
		1452.827	1453.75	0.92	1452.99	0.17	1452.90	0.08
$4\nu_{11}$	A_1	1453.403	1454.31	0.91	1453.58	0.18	1453.47	0.07
ν_7	E	1483.229	1483.73	0.50	1483.55	0.32	1483.40	0.17
		1483.229	1483.76	0.53	1483.58	0.35	1483.41	0.18

Table III continued from previous page

Level	Sym.	E_{ref}	Rank=20	E-E_{ref}	Rank=200	E-E_{ref}	Rank=400	E-E_{ref}
$\nu_4 + 2\nu_{11}$	E	1620.222	1621.71	1.48	1620.64	0.42	1620.37	0.14
		1620.222	1621.77	1.54	1620.73	0.51	1620.42	0.20
$\nu_4 + 2\nu_{11}$	A_1	1620.767	1622.55	1.78	1621.33	0.57	1620.99	0.22
$\nu_3 + \nu_{11}$	E	1749.53	1750.56	1.03	1749.97	0.44	1749.79	0.26
		1749.53	1750.63	1.10	1750.17	0.64	1749.84	0.31
$\nu_9 + 2\nu_{11}$	A_1	1756.426	1757.69	1.26	1756.94	0.51	1756.69	0.27
$\nu_9 + 2\nu_{11}$	A_2	1756.426	1757.73	1.31	1756.97	0.54	1756.71	0.28
$\nu_9 + 2\nu_{11}$	E	1757.133	1758.30	1.17	1757.55	0.41	1757.32	0.19
		1757.133	1758.48	1.35	1757.60	0.47	1757.34	0.20
$\nu_9 + 2\nu_{11}$	E	1759.772	1760.87	1.10	1760.27	0.50	1760.07	0.30
		1759.772	1761.04	1.26	1760.37	0.60	1760.08	0.30
$2\nu_4$	A_1	1785.207	1787.08	1.88	1785.83	0.62	1785.45	0.25
$5\nu_{11}$	E	1816.799	1817.91	1.11	1817.22	0.42	1816.99	0.19
		1816.799	1817.96	1.16	1817.23	0.43	1817.00	0.20
$5\nu_{11}$	A_1	1818.952	1820.15	1.19	1819.26	0.30	1819.12	0.17
$5\nu_{11}$	A_2	1818.952	1820.19	1.24	1819.28	0.33	1819.12	0.17
$5\nu_{11}$	E	1820.031	1821.29	1.26	1820.36	0.33	1820.16	0.13
		1820.031	1821.41	1.37	1820.38	0.35	1820.19	0.16
$\nu_7 + \nu_{11}$	A_2	1844.258	1845.19	0.93	1844.80	0.54	1844.46	0.20
$\nu_7 + \nu_{11}$	E	1844.33	1845.24	0.91	1844.88	0.55	1844.59	0.26
		1844.33	1845.33	1.00	1844.75	0.42	1844.61	0.28
$\nu_7 + \nu_{11}$	A_1	1844.69	1845.70	1.01	1845.02	0.33	1844.95	0.26
$\nu_4 + \nu_9$	E	1931.547	1933.37	1.83	1931.98	0.44	1931.76	0.21
		1931.547	1933.54	1.99	1932.11	0.56	1931.78	0.23

TABLE IV: Parameters for the acetonitrile calculation

Parameter	Values
Rank	20/200/400
n_c	9 7 9 9 9 9 7 7 9 9 27 27
N_{ALS}	1
P	4
B	6
S_1	100
S_2	80
$E_{\min}$	9800
$E_{\max}$	11800

VI. CONCLUSION

It is costly to solve the Schrödinger equation to compute a vibrational spectrum. One problem is of course the required computer time. Another problem is memory cost. When the amount of memory needed to do a calculation is larger than that of the computer to which one has access, the calculation simply becomes impossible. The simplest approach to computing a spectrum requires storing the representation of the Hamiltonian in a basis and computing some of its eigenvalues. If N is the size of the basis then the memory cost scales as N^2. Using iterative methods, that require only storing vectors, the memory cost scales as only N.[8,63,64] There are good approaches for minimizing N which make it possible to use iterative methods for molecules with as many as about six atoms. They all require using a basis that is not a direct product of univariate functions. However, even a memory cost that scales as N is debilitating.

In this paper,we propose a new tensor method, the CP-MSBII. It uses a direct product basis. Tensor methods reduce the memory required to store wavefunctions by representing them (and all vectors used to compute them) with what is called a tensor format. The memory cost scales linearly with D. Naturally other issues arise: 1) the algorithms become more complicated; 2) in some cases, the computer time (not memory) required for the calculation is large.

CP-MSBII uses CP format, i.e. basis functions are simple sums of products. Previous tensor methods[13,50–53] designed to simultaneously compute many states require orthogonalizing tensors and reducing the rank of tensors. The CP-MSBII does not. It uses a basis of non-orthogonal filtered vectors and solves a (small) generalized eigenvalue problem. To generate the filtered vectors, we must apply $(\sigma\mathbf{I} - \mathbf{H})^{-1}$ to vectors. The number of applications is orders of magnitude smaller than the number of $\mathbf{H}$ applications required to obtain eigenvalues of similar accuracy from the shifted block power method. The cost of an RRBPM calculation is mostly due to the cost of rank reduction which is not necessary when using the CP-MSBII. Not needing to reduce the rank also has the important advantage that there is no need to store tensors whose rank is larger than R. To make standard inverse iteration work, Rakhuba and Oseledets[53] needed to do a locally optimal block preconditioned conjugate gradient (LOBPCG) calculation first, to prepare initial vectors. The CP-MSBII works as is, there is no need to combine two eigensolvers.

The accuracy of the energy levels computed with the CP-MSBII depends mostly on the choice of three parameters: B, P, N_{ALS}. P and N_{ALS} determine the "quality" of the basis vectors. B determines the number of basis vectors. We find that increasing N_{ALS} is not as effective as increasing P and set $N_{ALS} = 1$. When P is larger, a smaller B is required and vice versa. We must also choose the rank R and the range in which energy levels are to be determined and of course the size of the direct product basis. We used the CP-MSBII method to calculate accurate vibrational energy levels of a 64-D model Hamiltonian model and a 12-D acetonitrile Hamiltonian. It could also be used for molecules with $D > 12$.

We have demonstrated that the CP-MSBII method is a good way of computing the lowest vibrational energy levels of large molecules. Often one does not want only the lowest levels or does not want to calculate *all* energy levels up to and including those of interest. It should be possible to use the CP-MSBII method also for energy levels in a window in the middle of the spectrum. Regardless of where the desired levels are, it ought to be possible to speed up CP-MSBII calculations by using a preconditioned iterative linear solver to make the filtered vectors and/or by using it in conjunction with a hierarchical approach.[50]

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgements

The financial support of the Natural Sciences and Engineering Research Council is gratefully acknowledged. We thank Compute Canada for providing access to its computers. TC thanks Mike Espig for discussions about ALS and sending him Ref. 60. We thank Phillip Thomas and Robert Wodraszka for comments on the manuscript.

[1] G. D. Carney, L. L. Sprandel, and C. W. Kern, Adv. Chem. Phys. **37**, 305 (1978).

[2] J. Tennyson, Comput. Phys. Rep. **4**, 1 (1986).

[3] S. Carter and N. C. Handy, Comput. Phys. Commun. **51**, 49 (1988).

[4] T. Carrington, Jr., J. Chem. Phys. **146**, 120902 (2017).

[5] M. J. Bramley and T. Carrington, J. Chem. Phys. **99**, 8519 (1993).

[6] H.-G. Yu and J. T. Muckerman, J. Mol. Spectrosc. **214**, 11 (2002).

[7] J. C. Light and T. Carrington, Adv. Chem. Phys. **114**, 263 (2000).

[8] A. G. Császár, C. Fábri, T. Szidarovszky, E. Mátyus, T. Furtenbacher, and G. Czakó, Phys. Chem. Chem. Phys. **14**, 1085 (2012).

[9] T. G. Kolda and B. W. Bader, SIAM. Rev. **51**, 455 (2009).

[10] H. L. Hitchcock, J. Math. Phys. **6**, 164 (1927).

[11] G. Beylkin and M. J. Mohlenkamp, SIAM J. Sci. Comput. **26**, 2133 (2005).

[12] G. Beylkin and M. J. Mohlenkamp, PNAS **99**, 10246 (2002).

[13] A. Leclerc and T. Carrington, J. Chem. Phys. **140**, 174111 (2014).

[14] H.-D. Meyer, U. Manthe, and L. S. Cederbaum, Chem. Phys. Lett. **165**, 73 (1990).

[15] U. Manthe, H.-D. Meyer, and L. S. Cederbaum, J. Chem. Phys. **97**, 3199 (1992).

[16] M. H. Beck, A. Jaeckle, G. A. Worth, and H. D. Meyer, Phys. Rep. **324**, 1 (2000).

[17] J. Jerke and B. Poirier, J. Chem. Phys. **148**, 104101 (2018).

[18] H. Wang and M. Thoss, J. Chem. Phys. **119**, 1289 (2003).

[19] U. Manthe, J. Chem. Phys. **128**, 164116 (2008).

[20] O. Vendrell and H.-D. Meyer, J. Chem. Phys. **134**, 044135 (2011).

[21] U. Schollwöck, Ann. Phys. **326**, 96 (2011).

[22] A. Baiardi, C. J. Stein, V. Barone, and M. Reiher, J. Chem. Theory Comput. **13**, 3764 (2017).

[23] A. Baiardi, C. J. Stein, V. Barone, and M. Reiher, J. Chem. Phys. **150**, 094113 (2019).

[24] S. Wouters, W. Poelmans, P. W. Ayers, and D. V. Neck, Comput. Phys. Commun. **185**, 1501 (2014).

[25] H. R. Larsson, J. Chem. Phys. **151**, 204102 (2019).

[26] H. R. Larsson, B. Hartke, and D. J. Tannor, J. Chem. Phys. **145**, 204108 (2016).

[27] L. Halonen, D. W. Noid, and M. S. Child, J. Chem. Phys. **78**, 2083 (1983).

[28] X.-G. Wang and T. Carrington, Jr., J. Phys. Chem. A **105**, 2575 (2001).

[29] B. Poirier, J. Theor. Comput. Chem. **2**, 65 (2003).

[30] G. Avila and T. Carrington, Jr., J. Chem. Phys. **137**, 174108 (2012).

[31] G. Avila and T. Carrington, Jr., J. Chem. Phys. **131**, 174103 (2009).

[32] G. Avila and T. Carrington, Jr., J. Chem. Phys. **143**, 214108 (2015).

[33] M. J. Bramley and T. Carrington, J. Chem. Phys. **101**, 8494 (1994).

[34] X.-G. Wang and T. Carrington, Jr., J. Chem. Phys. **117**, 6923 (2002).

[35] X.-G. Wang and T. Carrington, Jr., J. Chem. Phys. **148**, 074108 (2018).

[36] H.-S. Lee and J. C. Light, J. Chem. Phys. **120**, 4626 (2004).

[37] P. M. Felker and Z. Bacic, J. Chem. Phys. **152**, 124103 (2020).

[38] A. U. Bhatty and K. R. Brorsen, Mol. Phys. **119**, e1936250 (2021).

[39] J. H. Fetherolf and T. C. Berkelbach, J. Chem. Phys. **154**, 074104 (2021).

[40] N. Gohaud, D. Bégué, C. Darrigan, and C. Pouchan,, J. Comput. Chem. **26**, 743 (2005).

[41] P. Cassam-Chenai, and J. Liévin, Int. J. Quantum Chem. **93**, 245 (2003).

[42] U. Manthe, J. Chem. Phys. **105**, 6989 (1996).

[43] R. Wodraszka and T. Carrington, Jr., J. Chem. Phys. **148**, 044115 (2018).

[44] A. Jäckle and H.-D. Meyer, J. Chem. Phys. **104**, 7974 (1996).

[45] M. Schröder, J. Chem. Phys. **152**, 024108 (2020).

[46] S. Manzhos and T. Carrington, Jr., J. Chem. Phys. **125**, 194105 (2006).

[47] E. Pradhan and A. Brown, J. Mol. Spectrosc. **330**, 158 (2016).

[48] W. Koch and D. H. Zhang, J. Chem. Phys. **141**, 021101 (2014).

[49] M. B. Gardner, B. R. Westbrook, R. C. Fortenberry, and T. J. Lee, Spectrochim. Acta A Mol. Biomol. Spectrosc. **248**, 119184 (2020).

[50] P. S. Thomas and T. Carrington, J. Chem. Phys. **119**, 13074 (2015).

[51] P. S. Thomas and T. Carrington, J. Chem. Phys. **146**, 204110 (2017).

[52] P. S. Thomas, T. Carrington, J. Agarwal, and H. F. Schaefer, J. Chem. Phys. **149**, 064108 (2018).

[53] M. Rakhuba and I. Oseledets, J. Chem. Phys. **145**, 124101 (2016).

[54] F. Chatelin, *Eigenvalues of Matrices, Revised Edition*, SIAM, Philadelphia, 2012).

[55] M. R. Wall and D. Neuhauser, J. Chem. Phys. **102**, 8011 (1995).

[56] V.A. Mandelshtam and H.S. Taylor, J. Chem. Phys. **106**, 5085 (1997).

[57] V.A. Mandelshtam, J. Chem. Phys. **108**, 24 (1998).

[58] H. Zhang and S. C. Smith, Phys. Chem. Chem. Phys. **3**, 2282 (2001).

[59] A. Ruhe, Linear Algebra Appl. **58**, 391 (1984).

[60] M. Espig, W. J. Hackbusch, and A. Khachatryan *On the Convergence of Alternating Least Squares Optimisation in Tensor Format Representations* (2015).

[61] G. Avila and T. Carrington, J. Chem. Phys. **134**, 054126 (2011).

[62] D. Bégué, P. Carbonnière, and C. Pouchan, J. Phys. Chem. A **109**, 4611 (2005).

[63] T. Carrington, Jr., Adv. Chem. Phys. **163**, 217–243 (2018).

[64] J. Brown, X.-G. Wang, R. Dawes, and T. Carrington, Jr., J. Chem. Phys. **136**, 134306 (2012).