Systematically expanding nondirect product bases within the pruned multi-configuration time-dependent Hartree (MCTDH) method: A comparison with multi-layer MCTDH

Robert Wodraszka^{a)} and Tucker Carrington, Jr.^{b)}

Department of Chemistry, Queen's University, Kingston, Ontario K7L 3N6, Canada

(Received 28 March 2017; accepted 28 April 2017; published online 17 May 2017)

We propose a pruned multi-configuration time-dependent Hartree (MCTDH) method with systematically expanding nondirect product bases and use it to solve the time-independent Schrödinger equation. No pre-determined pruning condition is required to select the basis functions. Using about 65 000 basis functions, we calculate the first 69 vibrational eigenpairs of acetonitrile, CH₃CN, to an accuracy better than that achieved in a previous pruned MCTDH calculation which required more than 100 000 basis functions. In addition, we compare the new pruned MCTDH method with the established multi-layer MCTDH (ML-MCTDH) scheme and determine that although ML-MCTDH is somewhat more efficient when low or intermediate accuracy is desired, pruned MCTDH is more efficient when high accuracy is required. In our largest calculation, the vast majority of the energies have errors smaller than 0.01 cm⁻¹. *Published by AIP Publishing.* [<http://dx.doi.org/10.1063/1.4983281>]

I. INTRODUCTION

The (ro-)vibrational Schrödinger equation is most often solved by representing wavefunctions in a basis.^{1–7} When D , the number of dimensions, is less than about ten, it is possible to use a direct (or tensor) product basis.^{8–12} Direct product bases (DPBs) are conceptually simple and, more importantly, they make it possible to efficiently evaluate matrix-vector products (MVPs) when iterative methods are used to solve the Schrödinger equation. The CPU cost scales as N^{D+1} and the memory cost scales as N^D , where N is a representative number of 1D functions.⁶ Although when using an iterative method, there is no need to manipulate, store, or even calculate huge matrices, both the memory and the CPU costs of a DPB/iterative calculation nevertheless scale exponentially.

There are several strategies to mitigate this “curse of dimensionality.”¹³ First, it is often possible to remove many of the product functions from a direct product basis, without degrading the accuracy of the calculation. The resulting reduced or pruned basis is a nondirect product.^{14–25} The efficiency can be improved by carefully selecting the 1D functions used to build the DPB.^{20,26–28} Second, contraction techniques can be employed to reduce the size of the basis. To use the simplest contraction technique, one solves the Schrödinger equation in a basis of products of functions that are themselves solutions for subsets of coordinates.^{3,29–35} Third, one can reduce the size of a DPB by using n ($n \ll N$) 1D functions which are variationally optimised. The multi-configuration time-dependent Hartree (MCTDH) method^{36–38} employs this approach and the corresponding 1D functions are called single-particle functions (SPFs). As its name indicates, MCTDH was originally designed to be used in a time-dependent framework;

however, it is also possible to exploit its optimisation strategy to solve the time-independent Schrödinger equation.

All of the aforementioned strategies have advantages and disadvantages and it is therefore quite natural to combine them to extend their applicability. The multi-layer version of MCTDH (ML-MCTDH)^{39–41} was introduced to deal with the exponential scaling (e.g., the memory cost of standard MCTDH is $\mathcal{O}(n^D)$). ML-MCTDH shares with contraction the idea of using basis functions that are products of multi-dimensional functions (in this case, SPFs). The multi-dimensional SPFs are recursively represented in multi-dimensional SPFs of lower dimensionality. This results in a tree-like representation of the wavefunction. It is most efficient if coordinates are grouped so that coordinates within a group are more strongly coupled than coordinates in different groups. Mathematicians denote this type of wavefunction decomposition a hierarchical Tucker format.^{42,43} For simple Hamiltonians, ML-MCTDH makes it possible to study the dynamics of systems with hundreds or even thousands of degrees of freedom.^{44–46} Although it does make solving the Schrödinger equation possible, it is not clear how costly it is to obtain accurate solutions with ML-MCTDH. Clearly, the relative advantage of ML-MCTDH is less important if coupling is strong or propagation times are long. To calculate a rate constant or a poorly resolved spectrum, great accuracy is not necessary. ML-MCTDH has other drawbacks. First, designing an efficient tree, especially for complex systems, is difficult. Second, the complicated, coupled, non-linear differential equations one must solve to use ML-MCTDH are challenging to program and may be costly to solve. The coefficients for *all* of the nodes are coupled.

Recently, we devised a pruned MCTDH (P-MCTDH) approach that should also extend the applicability of MCTDH.⁴⁷ It combines the first and the third of the aforementioned techniques and exploits the fact that in a natural orbital (NO) basis many of the n^D product functions employed in the

^{a)}Electronic mail: Robert.Wodraszka@chem.queensu.ca

^{b)}Electronic mail: Tucker.Carrington@queensu.ca

standard MCTDH method can be discarded without degrading the accuracy of the results. It has been shown that although the resulting basis is a nondirect product, the sums one must evaluate to use it in conjunction with MCTDH can nevertheless be done sequentially. By calculating vibrational eigenpairs of acetonitrile, CH_3CN , a 12D problem, we have demonstrated that the Schrödinger equation can be solved very accurately with a pruned basis that is orders of magnitude smaller than the standard MCTDH basis with which one could achieve the same accuracy. P-MCTDH of Ref. 47 also has a key drawback: the constraint used to prune the basis does not necessarily retain the most important basis functions and it is unclear how to generalize it so that it can be applied to more strongly coupled molecules or non-spectroscopic problems.

In this paper, we develop a general method for choosing which product functions to put into the basis when using P-MCTDH. With the resulting nondirect product basis and P-MCTDH, we compute energy levels of CH_3CN . CH_3CN levels are also calculated with ML-MCTDH. Computational parameters are chosen so that the ML-MCTDH and P-MCTDH levels are about equally accurate and the cost of the two calculations is compared. In our previous study,⁴⁷ an algebraic expression was used to determine which functions to include in the basis. In this paper we, instead, do a series of eigenvalue calculations and after each one we expand the size of the basis. The expanded basis includes functions with multi-indices that are not in the previous basis. The multi-indices added are chosen using a proximity criterion. This strategy allows one to employ P-MCTDH in a black-box type fashion because the only information one has to provide is the initial basis; the basis will subsequently expand automatically. Moreover, the final basis size ought to be independent of the initial basis, provided that it does not include unnecessary functions. A similar idea has recently been used in a different context.⁴⁸

A basis whose size is increased using a proximity criterion has no obvious structure. A basis with structure is one for which a list of basis functions can be generated by executing a sequence of nested loops. For decades the structure of a DPB has been used to efficiently evaluate MVPs.^{6,8,9,49–52} It is also possible to exploit the structure of a basis pruned with a pre-determined condition.^{19,24} With an expanding basis, because there is no algebraic pruning condition and no obvious structure, to do MVPs, one needs to use multi-index lists and have some means of determining the position of a given multi-index in the list. To deal with identity operators in many of the terms in the Hamiltonian, we proposed, in Ref. 47, a method for doing MVPs that uses such lists. It can also be used with the expanding basis idea. In this paper, we reduce the cost of MVPs by using a three-nested-loop structure.⁵³

II. REPRESENTING WAVEFUNCTIONS

A. Pruned MCTDH (P-MCTDH)

In the recently introduced pruned MCTDH approach,⁴⁷ D -dimensional wavefunctions $\Psi_s(x_1, x_2, \dots, x_D, t)$ are represented in a *nondirect* product basis of one-dimensional time-dependent NOs $\varphi_{i_k}^{(k)}(x_k, t)$. The NOs are themselves

expanded in a time-independent orthonormal basis with functions $\chi_{i_k}^{(k)}(x_k)$. The complete *ansatz* is

$$\Psi_s(x_1, x_2, \dots, x_D, t) = \sum_{\substack{1 \leq i_k \leq n_k \forall k=1 \dots D \\ f(i_1, i_2, \dots, i_D) \leq b}} A_{i_1, i_2, \dots, i_D; s}(t) \times \varphi_{i_1}^{(1)}(x_1, t) \varphi_{i_2}^{(2)}(x_2, t) \dots \varphi_{i_D}^{(D)}(x_D, t), \quad (1)$$

$$\varphi_{i_k}^{(k)}(x_k, t) = \sum_{j=1}^{N_k} c_{j, i_k}^{(k)}(t) \chi_j^{(k)}(x_k) \quad \forall k = 1 \dots D. \quad (2)$$

We refer to $(i_1, i_2, \dots, i_D)$ as a multi-index. We are using the state-averaged formalism,⁵⁴ which is advantageous whenever one wishes to compute more than one wavefunction. In contrast to the original MCTDH approach,^{36,37} only combinations of basis function indices $(i_1, i_2, \dots, i_D)$ which satisfy the condition $f(i_1, i_2, \dots, i_D) \leq b$ are allowed. Choosing $n_k = b + 1 \forall k$ and the simple restriction $\sum_k (i_k - 1) \leq b$ reduces the size of the basis from $(b + 1)^D$ (direct product case) to $\frac{(b+D)!}{b!D!}$. Thus, the cost is significantly decreased when D is large, due mainly to the $D!$ factorial in the denominator.

Optimised SPFs, obtained from the Dirac-Frenkel variational principle $\langle \delta \Psi | i\hbar \partial_t \Psi - \hat{H} \Psi \rangle = 0$,^{55–57} have been derived for a pruned MCTDH method designed for fermionic⁵⁸ and bosonic⁵⁹ systems. Analogous equations could also be used for a P-MCTDH approach for vibrational problems. It is important to note that what is referred to as the gauge, $\mathbf{g}^{(k)} = \mathbf{c}^{(k)\dagger} \mathbf{c}^{(k)}$, cannot be, as in MCTDH with a DPB, chosen arbitrarily but is determined by the Dirac-Frenkel variational principle itself. In standard MCTDH, a set of orthonormal SPFs, $\{\varphi_{i_k}^{(k)}(x_k, t)\}_{i_k=1 \dots n_k}$, can be replaced by another obtained from it with a unitary transformation $\mathbf{U}^{(k)}$ and identical results are obtained if the coefficient tensor $A_{i_1, i_2, \dots, i_D; s}(t)$ is transformed with the inverse of the same transformation. This is not true in P-MCTDH, where it is important to use SPFs for which many of the $A_{i_1, i_2, \dots, i_D; s}(t)$ are small.

To simplify the numerical solution of the P-MCTDH differential equations, we do not use SPFs that are optimised for a particular pruning strategy. Instead, as in previous papers,^{47,60} we use natural orbitals (NOs). We can then solve the differential equations for $A_{i_1, i_2, \dots, i_D; s}(t)$ and $\mathbf{c}^{(k)}(t)$ employing established numerical integrators^{61,62} (imposing the standard gauge $\mathbf{g}^{(k)} = 0$). After each integration step, we transform back to the NO representation employing the appropriate $\mathbf{U}^{(k)}$.

B. Multi-layer MCTDH (ML-MCTDH)

ML-MCTDH is another strategy for reducing the cost of MCTDH calculations. Standard MCTDH represents wavefunctions in a basis of products of 1D functions. If instead one uses multidimensional SPFs, then fewer SPFs are required. This is called mode combination.^{63,64} Mode combination incorporates the effect of coupling within its multidimensional SPFs. When there are about a dozen coordinates and the multidimensional SPFs depend on 3 or 4 coordinates, mode combination can be very efficient. It is less useful for problems with more coordinates because the calculation of multidimensional SPFs that depend on many coordinates is costly. When this

is implemented recursively, one obtains ML-MCTDH. Each node of a ML-MCTDH tree corresponds to SPFs that depend on some subset of the full set of coordinates. The SPFs of a node are represented as linear combinations of products of the children of the node.^{39–41} The original physical coordinates, x_k , $k = 1 \dots D$, are organized into logical coordinates, $q_m^{(l;\dots)}$, $m = 1 \dots d_l, \dots$. The superscript l indicates the layer of the tree and the numbers denoted by $\dots$ determine a specific node of the tree. The ML-MCTDH wavefunctions are

$$\Psi_s(q_1^{(1)}, q_2^{(1)}, \dots, q_d^{(1)}, t) = \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} A_{i_1, \dots, i_d; s}^{(1)}(t) \times \varphi_{i_1}^{(1;1)}(q_1^{(1)}, t) \dots \varphi_{i_d}^{(1;d)}(q_d^{(1)}, t). \quad (3)$$

These functions or $A_{i_1, \dots, i_d; s}^{(1)}(t)$ are at the top of the tree. $\varphi_{i_k}^{(1;k)}(q_k^{(1)}, t)$, $k = 1 \dots d$, can be multi-dimensional. Consider, for example, $\varphi_{i_1}^{(1;1)}(q_1^{(1)}, t)$. It could be one-dimensional, e.g., $q_1^{(1)} = x_1$, or it could be multi-dimensional, e.g., $q_1^{(1)} = (x_1, x_2, x_3)$. In the first case, $\varphi_{i_1}^{(1;1)}(x_1, t)$ would itself be represented in a primitive basis according to Eq. (2) and the expansion with respect to x_1 would be finished (one would reach the bottom of the tree). In the second case, one could expand $\varphi_{i_1}^{(1;1)}(q_1^{(1)}, t) = \varphi_{i_1}^{(1;1)}(x_1, x_2, x_3, t)$ as

$$\varphi_{i_1}^{(1;1)}(x_1, x_2, x_3, t) = \sum_{j_1=1}^{n_{1,1}} \sum_{j_2=1}^{n_{1,2}} \sum_{j_3=1}^{n_{1,3}} A_{j_1 j_2 j_3; i_1}^{(2;1)} \varphi_{j_1}^{(2;1,1)}(x_1, t) \times \varphi_{j_2}^{(2;1,2)}(x_2, t) \varphi_{j_3}^{(2;1,3)}(x_3, t), \quad (4)$$

that is, as a standard MCTDH wavefunction, cf. Eq. (1). Then after representing the 1D $\varphi_{j_k}^{(2;1,k)}(x_k, t)$, $k = 1 \dots 3$, in primitive bases, one would reach the bottom of the tree. Or one could introduce another layer and split the coordinates (x_1, x_2, x_3) so that, e.g., $q_1^{(2;1)} = x_1$ and $q_2^{(2;1)} = (x_2, x_3)$. Then

$$\varphi_{i_1}^{(1;1)}(x_1, x_2, x_3, t) = \sum_{j_1=1}^{n_{1,1}} \sum_{j_2=1}^{n_{1,2}} A_{j_1 j_2; i_1}^{(2;1)} \varphi_{j_1}^{(2;1,1)}(x_1, t) \times \varphi_{j_2}^{(2;1,2)}(x_2, x_3, t). \quad (5)$$

$\varphi_{j_2}^{(2;1,2)}(x_2, x_3, t)$ would then again be expanded in the standard MCTDH fashion. These examples convey the main idea. It can be extended to any number of coordinates and layers. A detailed discussion and the general derivation of the working equations obtained by employing the Dirac-Frenkel principle can be found elsewhere.⁴⁰

III. SOLVING THE TIME-INDEPENDENT SCHRÖDINGER EQUATION

MCTDH was originally devised to solve the time-dependent Schrödinger equation.^{36,37} Nonetheless, because MCTDH is an algorithm for applying $e^{-it\hat{H}}$ to functions, it can be used to determine solutions of the time-independent Schrödinger equation by replacing $t \rightarrow -i\beta$, $\beta \in \mathbb{R}_{>0}$. MCTDH can be viewed as a means of converting a function in the Tucker format to another function also in the Tucker format,

$$e^{-\beta\hat{H}}\Psi^{(\text{Tucker})}(0) = \Psi^{(\text{Tucker})}(\beta). \quad (6)$$

When expanded in terms of eigenfunctions of $\hat{H}$, the function on the RHS of Eq. (6) has larger components for low-lying eigenfunctions (and smaller components for high-lying eigenfunctions). Repeatedly applying $e^{-\beta\hat{H}}$ will push a function into the space spanned by the lowest wavefunctions. For the same reasons, ML-MCTDH used with $t \rightarrow -i\beta$ pushes a function in a hierarchical Tucker format towards the space spanned by the lowest wavefunctions, while maintaining the hierarchical Tucker format. If $e^{-\beta\hat{H}}$ is applied many times, one obtains the ground state. The larger the value of β , the smaller is the number of iterations required to achieve converged energy levels. However, increasing β also increases the CPU cost of each MCTDH propagation step.

A. (Block) power method

Obviously, we wish to compute more than just the ground state. One way to do this is to use a block power method.⁶⁵ This requires doing a series of iterations each of which consists of an application of $e^{-\beta\hat{H}}$ to the functions in a block and orthogonalization of the resulting vectors with the Gram-Schmidt procedure. The orthogonalization is greatly facilitated by using the state-averaged MCTDH approach⁵⁴ because there is a common SPF basis. With the block power method, one simultaneously calculates n_{states} eigenstates of the Hamiltonian $\hat{H}$ using a block with n_{states} functions.

B. (Block) improved relaxation

For the purpose of computing dozens of bound states with MCTDH, the block variant⁶⁶ of improved relaxation⁶⁷ is more efficient than the block power method. The working equations of improved relaxation are derived from a time-independent variational principle. Like a power method calculation, an improved relaxation calculation also involves a series of iterations. If the goal is to compute a single state, in one iteration, SPFs are propagated with MCTDH, and rather than propagating the A tensor, as is usually done in MCTDH, it is set equal to an eigenvector of the Hamiltonian matrix $\mathbf{H}$, i.e., $\hat{H}$ represented in the SPF basis,

$$\mathbf{H}\mathbf{A} = E\mathbf{A}. \quad (7)$$

The $\mathbf{A}$ tensor corresponding to the wavefunction that has the largest overlap with the initial wavefunction is taken for the subsequent iteration. Eigenvectors are typically calculated using the Davidson algorithm⁶⁸ with a diagonal preconditioner. This machinery can also be applied to a block of vectors and combined with state-averaged MCTDH to obtain n_{states} eigenpairs simultaneously. We use a block-Davidson routine extracted from the Heidelberg MCTDH package.⁸⁸ In the rest of the paper, we use the block power method of Subsection III A in conjunction with both ML-MCTDH and P-MCTDH and the improved relaxation method of Subsection III B in conjunction with P-MCTDH. We therefore drop the t variable from the SPFs. We do not replace it with β .

IV. USING AN EXPANDING NONDIRECT PRODUCT BASIS WITH MCTDH

In the P-MCTDH method of Ref. 47, a nondirect product NO basis is used. An obvious way to do this is to impose a

pruning condition, as indicated below the summation sign in Eq. (1). For the calculation of vibrational energy levels of a molecule whose potential energy surface (PES) has a single deep well, we found that

$$b \geq f(i_1, i_2, \dots, i_D) = \sum_{k=1}^D \alpha_k (i_k - 1) \quad (8)$$

works well. The same pruning condition is commonly used with 1D harmonic oscillator basis functions.^{1,15,24,69} We have shown that using Eq. (8) to prune an NO basis within the P-MCTDH approach,⁴⁷ it is possible to obtain accurate results for acetonitrile, CH₃CN.

There is no reason to think that with NOs Eq. (8) would also work for molecules undergoing large amplitude motion, molecules falling apart, etc. In this paper, the basis we use has the potential to work even when a harmonic approximation is poor. The basis of this paper is not obtained (even conceptually) by starting with a large direct product basis and removing functions. Instead, we start with a small basis, each of whose functions is associated with some multi-index, and we expand it by adding products of NOs associated with multi-indices that are not in the start basis. In the expanded basis, functions labelled by multi-indices that also label functions in the start basis differ slightly from their counterparts in the start basis. They differ because NOs for coordinate x_k depend on all of the functions in the basis. The expanded basis is expanded again and again until convergence is achieved. Of course, even if our algorithm for expanding the basis is not optimal, more accurate results can always be obtained by using a larger basis. In that sense a good expansion scheme is better only because it is more efficient. A similar expanding basis idea has been used with phase-space localized and harmonic oscillator basis functions.^{48,70,71} Our implementation is close to that of Ref. 48. However, in our case, when the basis size is increased, the definition of all of the functions in the basis changes. If the A tensor elements that correspond to new multi-indices are small, then functions in the start and expanded bases with the same multi-indices are close.

We start with a small (NO) nondirect product basis, e.g., by choosing $\alpha_k = 1 \forall k$ and a small threshold b in Eq. (8). Converged (with respect to the number of iterations) eigenvalues are computed with the P-MCTDH method. We then sort product NO functions in the start basis according to the importance measure

$$O_{(i_1, i_2, \dots, i_D)} = \sum_{s=1}^{n_{\text{states}}} |\langle \varphi_{i_1}^{(1)} | \langle \varphi_{i_2}^{(2)} | \dots \langle \varphi_{i_D}^{(D)} | \Psi_s \rangle|^2, \quad (9)$$

where Ψ_s is the s -th eigenfunction obtained with the start basis. To expand the basis, we add the D multi-indices that

are “nearest neighbours” of the multi-index $(i_1, i_2, \dots, i_D)$ in the basis with the largest value of $O_{(i_1, i_2, \dots, i_D)}$. Each neighbour is obtained by augmenting one of the indices, i.e., $(i_1, \dots, i_k + 1, \dots, i_D) \forall k$. This process is repeated with the second, third, $\dots$ largest $O_{(i_1, i_2, \dots, i_D)}$ until the number of functions added to the basis exceeds 5%.

The basis expansion can be implemented with or without fixing the maximum number (n_k in Eq. (1)) of 1D SPFs for each coordinate. Often it is possible, by inspecting the occupation numbers of the corresponding NOs, to estimate the number of 1D SPF bases required for convergence. If the n_k are fixed, then one omits neighbours whose i_k values exceed the maximum. If n_k are not fixed, then the basis grows completely automatically in a black-box fashion. We present results for both approaches.

NOs are a good choice for the 1D functions of a general expanding basis. Ideal is to have a basis in which, e.g., for a 3D problem, A_{i_1, i_2, i_3} for multi-indices added to the basis become smaller and smaller as the basis size grows. In the harmonic case,⁴⁸ the expanding basis idea works because harmonic functions for which i_1, i_2, i_3 are all close have similar zeroth order energies and, at least for a Taylor series potential, only nearby values of i_1, i_2, i_3 are coupled. For a NO basis, it also works well because

$$|A_{i_1, i_2, i_3}|^2 \leq \min(\lambda_{i_1}^{(1)}, \lambda_{i_2}^{(2)}, \lambda_{i_3}^{(3)}) \quad (10)$$

ensures that increasing the i_k ($k = 1, 2, 3$) with the smallest $\lambda_{i_k}^{(k)}$ (the diagonal elements of the density matrix in coordinate k) decreases A_{i_1, i_2, i_3} . For a pair i_1, i_2 (i_1, i_3 ; i_2, i_3), if i_3 (i_2 ; i_1) is large enough, then A_{i_1, i_2, i_3} will decrease as i_3 (i_2 ; i_1) increases. This means that, in general, A_{i_1, i_2, i_3} on the outside will be smaller than A_{i_1, i_2, i_3} on the inside. It also motivates expanding the basis by adding neighbours.

A. Evaluating matrix-vector products

We must solve the MCTDH working equations in the expanded basis. This is non-trivial because it is a nondirect product basis. Our procedure works for a Hamiltonian in sum-of-products (SOPs) form, i.e.,

$$\hat{H} = \sum_{g=1}^{n_{\text{term}}} \hat{h}^{(1,g)} \otimes \hat{h}^{(2,g)} \otimes \dots \otimes \hat{h}^{(D,g)}. \quad (11)$$

Each factor only depends on a single coordinate x_k , $k = 1 \dots D$. The most time-consuming part of the calculation is the evaluation of matrix-vector products. When a direct product basis is employed, matrix-vector products are usually evaluated by doing sums *sequentially*,^{8,49,72}

$$A'_{i'_1, i'_2, \dots, i'_D; s} = \sum_{i_D=1}^{n_D} h_{i_D, i'_D}^{(D,g)} \dots \sum_{i_2=1}^{n_2} h_{i_2, i'_2}^{(2,g)} \underbrace{\sum_{i_1=1}^{n_1} h_{i_1, i'_1}^{(1,g)} A_{i_1, i_2, \dots, i_D; s}}_{=A_{i'_1, i_2, \dots, i_D; s}^{(1,g)}} \quad \forall g = 1 \dots n_{\text{term}}, s = 1 \dots n_{\text{states}}. \quad (12)$$

$\underbrace{\hspace{10em}}_{=A_{i'_1, i'_2, \dots, i'_D; s}^{(2,g)}}$

The underbraces indicate that for each s and for each term g , the sums are evaluated factor-wise and intermediate results are stored in $A_{i'_1, \dots, i'_k, \dots, i_D; s}^{(k, g)}$, $k = 1 \dots D - 1$. The cost of Eq. (12) scales as Dn^{D+1} , with n being a representative number of 1D basis functions. The 1D matrix elements are $h_{i'_k, i_k}^{(k, g)} = \langle \varphi_{i'_k}^{(k)} | \hat{h}^{(k, g)} | \varphi_{i_k}^{(k)} \rangle$. They are inexpensive to compute.

It is known¹⁹ that when using a nondirect product pruned basis determined by a pruning condition of the form Eq. (8), it is also possible to evaluate sums sequentially. This is due to the fact that although the basis is pruned, it still has exploitable structure. In the pruned case, the unprimed and primed indices are restricted among themselves. This means that the upper limits of the sums will depend on all preceding indices, i.e., $n_k = n_k(i_{k+1}, i_{k+2}, \dots, i_D)$. We used these ideas in Ref. 47. In an earlier paper, Ref. 60, the range of some of the loops was not restricted. Restricting the ranges of the primed and unprimed indices is straightforward.

One way to evaluate the matrix-vector product in the expanded basis is to use the mapping method of Refs. 22 and 48. In those papers, all the intermediate vectors are the same length and all $\hat{h}^{(k, g)}$ are represented in the full basis. It is also possible to use Eq. (12) and intermediate vectors of different lengths. To do this, one must impose a structure on the basis. We use the following notation. The input vector is labelled by multi-indices that are the components of J_0 . We use anti-lexicographic ordering. $A_{i'_1, i_2, \dots, i_D; s}^{(1, g)}$ is labelled by multi-indices that are the components of a list we denote J_1 . The structure is imposed by choosing J_1 so that it is a Cartesian product of all i'_1 values that occur in J_0 and the $\{i_2, i_3, \dots, i_D\}$ multi-indices that occur in J_0 . We divide the elements of J_0 into two sets $I_1^{(\text{left})}$ and $I_1^{(\text{right})}$,

$$\underbrace{(i'_1)}_{\in I_1^{(\text{left})}} \underbrace{(i_2, i_3, \dots, i_D)}_{\in I_1^{(\text{right})}} \in J_0, \quad (13)$$

and construct J_1 as the Cartesian product

$$J_1 = I_1^{(\text{left})} \times I_1^{(\text{right})}. \quad (14)$$

J_2 is made similarly by dividing J_0 according to

$$\underbrace{(i'_1, i'_2)}_{\in I_2^{(\text{left})}} \underbrace{(i_3, i_4, \dots, i_D)}_{\in I_2^{(\text{right})}} \in J_0 \quad (15)$$

and defining J_2 by

$$J_2 = I_2^{(\text{left})} \times I_2^{(\text{right})}. \quad (16)$$

Other J_k are defined in the same way. Note that J_1 contains multi-indices that are not in J_0 . There is a set of J_k for each term g . J_k (with $0 < k < D$) define intermediate bases that have enough structure that it is possible to evaluate matrix-vector products with Eq. (12). In Ref. 47, the structure necessary to use Eq. (12) is a natural consequence of the pruning condition. When using the expanded basis, the structure must be imposed.

In this paper, we build and use lists of multi-indices that are based on J_k . This reduces the cost because most (all in our case) terms in typical SOP Hamiltonians contain identity operators for some coordinates. Using lists of multi-indices, it is possible to obviate sums over indices of identity matrices. To do this, we modify J_k , as explained in Ref. 47, by removing from the J_k sets defined in the previous paragraph elements that are not required. With the current code, it is necessary that the multi-index lists have no holes in an individual index. For a 3D example, this means that (2, 1, 3) could not be followed by (2, 3, 3); the hole would have to be filled by adding (2, 2, 3). However, this “repair” can easily be accomplished when expanding the basis.

In this paper, to evaluate the matrix-vector product, we use the loop structure suggested by Larsson *et al.*⁵³ To illustrate the idea, consider the loops required to apply $h_{i'_k, i_k}^{(k)}$ to $A_{i'_1, \dots, i'_{k-1}, i_k, \dots, i_D}^{(k-1)}$. For each $i'_1, i'_2, \dots, i'_{k-1}$ and for each $i_{k+1}, \dots, i_D$, one must sum over i_k for all i'_k . In Ref. 47, we do this with four loops. By re-ordering the indices, Larsson *et al.* use three loops. The sets of multi-indices of the input and output vectors are J_{k-1} and J_k . The application of $h_{i'_k, i_k}^{(k)}$ is done,

$$\underbrace{i'_k, i'_{k-1}, \dots, i'_1, i_{k+1}, \dots, i_D}_{\in J_k} \leftarrow h_{i'_k, i_k}^{(k)} \underbrace{i_k, i'_{k-1}, \dots, i'_1, i_{k+1}, \dots, i_D}_{\in J_{k-1}}. \quad (17)$$

The indices of $h_{i'_k, i_k}^{(k)}$ appear at the front of the input (i_k) and output (i'_k) lists. This allows one to use a three-nested-loop structure

```

A(k) ← 0
ck-1 ← 0
ck ← 0
loop over multind = 1 ... |Ik|
  loop over ik = 1 ... maxk-1(multind)
    ck-1 ← ck-1 + 1
    loop over i'k = 1 ... maxk(multind)

```

$$A_{c_k + i'_k}^{(k)} \leftarrow A_{c_k + i'_k}^{(k)} + h_{i'_k, i_k}^{(k)} \cdot A_{c_{k-1}}^{(k-1)}$$

```

end loop
end loop
ck ← ck + maxk(multind)
end loop. \quad (18)

```

The two multi-dimensional A -tensors (input and output) are stored as vectors with a single index. As the matrix-vector product is calculated, counters, c_{k-1} and c_k , that determine the position of elements in the vectors are computed. The ranges of the two inner loops over i_k and i'_k depend on *multind*.

TABLE I. The nondirect product basis sizes used in P-MCTDH calculations.

Basis	# Nondirect product basis functions	Ratio direct/nondirect product basis functions
$B_{\text{initial}}^{\text{P}}$	6 953	16 966
$B_{\text{inter}}^{\text{P}}$	28 929	4 078
$B_{\text{final}}^{\text{P}}$	65 062	1 813

fixed for both the ML-MCTDH and the P-MCTDH calculations. We use the same tree used in Ref. 75, where it worked well. Although there may be better choices, this tree ought to work well with ML-MCTDH.

For each of the three ML-MCTDH calculations, we find an expanded basis large enough that the P-MCTDH energies are about as accurate as the ML-MCTDH energies. These three bases are designated, $B_{\text{initial}}^{\text{P}}$, $B_{\text{inter}}^{\text{P}}$, and $B_{\text{final}}^{\text{P}}$. Their sizes are reported in Table I. These bases are all assembled imposing fixed values of n_k . We do another set of P-MCTDH calculations without fixing the n_k values. These are our most accurate P-MCTDH calculations and about as accurate as those of Refs. 73 and 76. The parameters of the largest calculation are given in Table II.

B. Results

ML-MCTDH energies computed with the $B_{\text{final}}^{\text{ML}}$ tree are about as accurate as those obtained with P-MCTDH and the $B_{\text{final}}^{\text{P}}$ basis; see Table III, where we also include results from the largest calculation of Ref. 47. With $\sim 65\,000$ basis functions, the P-MCTDH errors are less than about 0.1 cm^{-1} . In Ref. 47, where the basis was pruned with Eq. (8), more than $100\,000$ basis functions were required. The ML-MCTDH maximum error is similar but about twice as high as the P-MCTDH error. The mean ML- and P-MCTDH errors are very close. It is not our purpose to compare the accuracy of the two methods. Both will give numerically exact results if the basis sizes are increased. We could certainly have obtained more accurate ML-MCTDH levels by further increasing the

TABLE II. Parameters of the largest P-MCTDH calculation employing varying n_k .

Mode	NOs
ω_1	5
ω_2	6
ω_3	5
ω_4	7
ω_5	6
ω_6	6
ω_7	6
ω_8	6
ω_9	7
ω_{10}	8
ω_{11}	11
ω_{12}	11
# Direct product basis functions	9 220 780 800
# Nondirect product basis functions	164 242
Ratio direct/nondirect	56 141

TABLE III. In columns II and III, we report errors of the $B_{\text{final}}^{\text{ML}}$ ML-MCTDH and $B_{\text{final}}^{\text{P}}$ P-MCTDH calculations, both calculations use the block power method. Column IV contains errors for a $B_{\text{final}}^{\text{P}}$ P-MCTDH calculation with block improved relaxation. The maximum and mean errors are in cm^{-1} . The numbers in the last column are included for comparison.

	$B_{\text{final}}^{\text{ML}}$	$B_{\text{final}}^{\text{P}}$	$B_{\text{final}}^{\text{P}}$	Pruned MCTDH, Ref. 47
Maximum error	0.230	0.119	0.114	0.141
Mean error	0.054	0.046	0.042	0.063
Computation time	>5 weeks	~ 3 weeks	~ 2 days	~ 4 weeks

node basis sizes. However, to achieve high accuracy, it is necessary to include NOs with occupation numbers as low as 10^{-7} and there are many that are about equally important. It is more important to compare computation times. The ML-MCTDH calculation takes much longer than the P-MCTDH calculation. This is due in part to the fact that when using ML-MCTDH, more iterations (i.e., applications of $e^{-\beta\hat{H}}$) are required. More iterations are required in part because, due to small eigenvalues of density matrices, an MCTDH propagation may be inaccurate at short times. At longer times, “self-healing” ensues and accurate results are obtained.⁷⁷ It is known that using many layers exacerbates this problem.⁷⁷ After about 5 weeks, the highest eigenenergy was about 3 cm^{-1} above the reference value. We decided to stop the calculation at this point. In the $B_{\text{final}}^{\text{ML}}$ column, we report errors for converged results obtained by using wavefunctions of the previous calculation to start the final ML-MCTDH calculation. Without using special tricks^{77,78} to decrease the ratio of the ML time to the P time, it seems that ML-MCTDH is at least as costly as P-MCTDH. To demonstrate that the absolute numerical efficiency can be increased, we also present results for $B_{\text{final}}^{\text{P}}$ using the block improved relaxation approach. As shown in Table III, the computation time is significantly lowered, from about 3 weeks to 2 days. Since all of the basis functions used in the calculations are real, a further gain in efficiency (and reduction in memory use) could be achieved by modifying the P-MCTDH code to use real (instead of complex) arithmetic.

The efficiencies of ML-MCTDH and P-MCTDH in an intermediate accuracy regime are compared in Table IV, where we present ML-MCTDH energies obtained with the tree of Fig. 2, $B_{\text{inter}}^{\text{ML}}$, and P-MCTDH energies with similar accuracy, obtained with $B_{\text{inter}}^{\text{P}}$. In this intermediate accuracy regime, the P-MCTDH calculation takes *more* time than the ML-MCTDH calculation but is somewhat more accurate. This supports the view that ML-MCTDH is a powerful method when low or

TABLE IV. As Table III, but for $B_{\text{inter}}^{\text{ML}}$ and $B_{\text{inter}}^{\text{P}}$.

	$B_{\text{inter}}^{\text{ML}}$	$B_{\text{inter}}^{\text{P}}$
Maximum error	0.903	0.496
Mean error	0.260	0.241
Computation time	~ 7 days	~ 11 days

TABLE V. (Continued.)

Assignment	Symmetry	E_{ref}	E_P	$E_P - E_{\text{ref}}$
$\omega_3 + 2 \omega_{11}$	E	2111.377	2111.374	-0.003
		2111.379	2111.374	-0.005
$\omega_3 + 2 \omega_{11}$	A_1	2112.294	2112.292	-0.002
		2119.317	2119.321	0.004
$\omega_9 + 3 \omega_{11}$	E	2119.317	2119.321	0.004
		2120.529	2120.535	0.006
$\omega_9 + 3 \omega_{11}$	E	2120.530	2120.535	0.005
		2120.897	2120.903	0.006
$\omega_9 + 3 \omega_{11}$	A_2	2122.835	2122.827	-0.008
		2122.838	2122.828	-0.010
$\omega_9 + 3 \omega_{11}$	E	2122.838	2122.828	-0.010
		2123.300	2123.294	-0.006
$\omega_9 + 3 \omega_{11}$	A_1	2123.300	2123.294	-0.006

VI. CONCLUSION

Despite the many virtues of MCTDH, the CPU and memory costs of a standard MCTDH calculation both scale exponentially with D , the dimension of the problem. ML-MCTDH reduces both the costs by representing wavefunctions in a hierarchical format. ML-MCTDH makes it possible to do calculations on systems with hundreds or thousands of coordinates. It is rapidly becoming the method of first resort for large quantum dynamics calculations. In this paper, we generalize a P-MCTDH method we proposed in Ref. 47 and compare it with ML-MCTDH. We find that, for CH_3CN , when high accuracy is required, P-MCTDH is more efficient than current implementations of ML-MCTDH and that when intermediate accuracy is sufficient, ML-MCTDH is slightly more efficient.

To generalize P-MCTDH, we introduce a new procedure for choosing which and how many product NO basis functions to include in the basis. In this paper, we employ systematically expanding nondirect product bases. In contrast to our previous work,⁴⁷ the basis size is not determined by an algebraic expression. Instead we begin with a start basis and (automatically) expand it employing a proximity criterion. This strategy is not based on the quality of a harmonic approximation and it should work well even when coupling is important. Using the expanding basis, accurate results can be obtained with fewer basis functions than when using the algebraic pruning condition of Ref. 47. Combining the P-MCTDH approach with block improved relaxation significantly reduces the computation times. Furthermore, a more efficient three-loop structure for evaluating matrix-vector products has been implemented.⁵³

In principle, one can compute numerically exact results with ML-MCTDH, why then is it useful to consider alternatives? First, to use ML-MCTDH, one must choose a wavefunction tree. This can be tricky and the accuracy and efficiency of a ML-MCTDH calculation depend on the choice of the tree. Second, P-MCTDH is conceptually simpler. The non-linearity of the equations is easier to manage. In ML-MCTDH, differential equations for $A^{(l,\dots)}$ in all of the layers must be simultaneously satisfied. Third, ML-MCTDH calculations are much simpler if the Hamiltonian has a special form. ML-MCTDH is usually used with a SOP Hamiltonian, but it is also possible to put the Hamiltonian into a ML format.⁷⁹ Although, it is possible to massage a general PES into SOP form,^{80,81} it would be nice to obviate this step. One way to do this is to use a

ML-CDVR (multi-layer correlation DVR) approach.⁸² It is based on a simultaneous diagonalization⁸³ and is intrinsically approximate. Our present implementation of P-MCTDH also relies on the Hamiltonian having a special form. However, because P-MCTDH is relatively simple, it might be possible to merge it with accurate Smolyak quadrature and interpolation schemes that exist^{84–86} and that have recently been used to solve the Schrödinger equation with pruned bases.^{23,24,87} This would make it possible to use nondirect product NO bases with general PESs.

ACKNOWLEDGMENTS

The financial support of the Natural Sciences and Engineering Research Council is gratefully acknowledged.

- ¹S. Carter and N. C. Handy, *Comput. Phys. Rep.* **5**, 117 (1986).
- ²J. Tennyson, *Comput. Phys. Rep.* **4**, 1 (1986).
- ³Z. Bacic and J. C. Light, *Annu. Rev. Phys. Chem.* **40**, 469 (1989).
- ⁴E. L. Sibert, *Int. Rev. Phys. Chem.* **9**, 1 (1990).
- ⁵T. Carrington, Jr., in *Encyclopedia of Computational Chemistry*, edited by P. R. Schleyer (John Wiley & Sons, Ltd., West Sussex, 1998), pp. 3157–3166.
- ⁶J. C. Light and T. Carrington, Jr., *Adv. Chem. Phys.* **114**, 263 (2000).
- ⁷J. Bowman, T. Carrington, Jr., and H.-D. Meyer, *Mol. Phys.* **106**, 2145 (2008).
- ⁸M. J. Bramley and T. Carrington, Jr., *J. Chem. Phys.* **99**, 8519 (1993).
- ⁹H.-G. Yu and J. T. Muckerman, *J. Mol. Spectrosc.* **214**, 11 (2002).
- ¹⁰G. Czako, T. Furtenbacher, A. G. Császár, and V. Szalay, *Mol. Phys.* **102**, 2411 (2004).
- ¹¹E. Mátyus, G. Czako, and A. G. Császár, *J. Chem. Phys.* **130**, 134112 (2009).
- ¹²E. Mátyus, J. Šimunek, and A. G. Császár, *J. Chem. Phys.* **131**, 074106 (2009).
- ¹³R. E. Bellman, *Adaptive Control Processes: A Guided Tour* (Princeton University Press, Princeton, 1961).
- ¹⁴R. J. Whitehead and N. C. Handy, *J. Mol. Spectrosc.* **59**, 459 (1976).
- ¹⁵G. D. Carney, *Adv. Chem. Phys.* **37**, 305 (1978).
- ¹⁶L. Halonen, D. W. Noid, and M. S. Child, *J. Chem. Phys.* **78**, 2803 (1983).
- ¹⁷J. M. Bowman, S. Carter, and X. Huang, *Int. Rev. Phys. Chem.* **22**, 533 (2003).
- ¹⁸A. Maynard, R. E. Wyatt, and C. Iung, *J. Chem. Phys.* **106**, 9483 (1997).
- ¹⁹X.-G. Wang and T. Carrington, Jr., *J. Phys. Chem. A* **105**, 2575 (2001).
- ²⁰R. Dawes and T. Carrington, Jr., *J. Chem. Phys.* **122**, 134101 (2005).
- ²¹R. Dawes and T. Carrington, Jr., *J. Chem. Phys.* **124**, 054102 (2006).
- ²²J. Cooper and T. Carrington, Jr., *J. Chem. Phys.* **130**, 214110 (2009).
- ²³G. Avila and T. Carrington, Jr., *J. Chem. Phys.* **131**, 174103 (2009).
- ²⁴G. Avila and T. Carrington, Jr., *J. Chem. Phys.* **134**, 054126 (2011).
- ²⁵G. Avila and T. Carrington, Jr., *J. Chem. Phys.* **135**, 064101 (2011).
- ²⁶B. Poirier, *J. Theor. Comput. Chem.* **2**, 65 (2003).
- ²⁷A. Shimshovitz and D. J. Tannor, *Phys. Rev. Lett.* **109**, 070402 (2012).
- ²⁸J. Brown and T. Carrington, Jr., *Phys. Rev. Lett.* **114**, 058901 (2015).
- ²⁹J. R. Henderson and J. Tennyson, *Chem. Phys. Lett.* **173**, 133 (1990).
- ³⁰M. J. Bramley and T. Carrington, Jr., *J. Chem. Phys.* **101**, 8494 (1994).
- ³¹X.-G. Wang and T. Carrington, Jr., *J. Chem. Phys.* **117**, 6923 (2002).
- ³²H.-G. Yu, *J. Chem. Phys.* **117**, 8190 (2002).
- ³³X.-G. Wang and T. Carrington, Jr., *J. Chem. Phys.* **119**, 101 (2003).
- ³⁴X.-G. Wang and T. Carrington, Jr., *J. Chem. Phys.* **129**, 234102 (2008).
- ³⁵X.-G. Wang and T. Carrington, Jr., *J. Chem. Phys.* **138**, 104106 (2013).
- ³⁶H.-D. Meyer, U. Manthe, and L. S. Cederbaum, *Chem. Phys. Lett.* **165**, 73 (1990).
- ³⁷U. Manthe, H.-D. Meyer, and L. S. Cederbaum, *J. Chem. Phys.* **97**, 3199 (1992).
- ³⁸H.-D. Meyer, F. Gatti, and G. A. Worth, *Multidimensional Quantum Dynamics: MCTDH Theory and Applications* (Wiley-VCH, Weinheim, 2009).
- ³⁹H. Wang and M. Thoss, *J. Chem. Phys.* **119**, 1289 (2003).
- ⁴⁰U. Manthe, *J. Chem. Phys.* **128**, 164116 (2008).
- ⁴¹O. Vendrell and H.-D. Meyer, *J. Chem. Phys.* **134**, 044135 (2011).
- ⁴²W. Hackbusch and S. Kühn, *J. Fourier Anal. Appl.* **15**, 706 (2009).
- ⁴³L. Grasedyck, *SIAM J. Matrix Anal. Appl.* **31**, 2029 (2010).

- ⁴⁴H. Wang and M. Thoss, *J. Phys. Chem. A* **111**, 10369 (2007).
- ⁴⁵I. Kondov, M. Čížek, C. Benesch, H. Wang, and M. Thoss, *J. Phys. Chem. C* **111**, 11970 (2007).
- ⁴⁶H. Wang and M. Thoss, *New J. Phys.* **10**, 115005 (2008).
- ⁴⁷R. Wodraszka and T. Carrington, Jr., *J. Chem. Phys.* **145**, 044110 (2016).
- ⁴⁸J. Brown and T. Carrington, Jr., *J. Chem. Phys.* **145**, 144104 (2016).
- ⁴⁹U. Manthe and H. Köppel, *J. Chem. Phys.* **93**, 345 (1990).
- ⁵⁰R. Chen, G. Ma, and H. Guo, *J. Chem. Phys.* **114**, 4763 (2001).
- ⁵¹P. Sarkar, N. Poulin, and T. Carrington, Jr., *J. Chem. Phys.* **110**, 10269 (1999).
- ⁵²N. M. Poulin, M. J. Bramley, T. Carrington, Jr., H. G. Kjaergaard, and B. R. Henry, *J. Chem. Phys.* **104**, 7807 (1996).
- ⁵³H. R. Larsson, B. Hartke, and D. J. Tannor, *J. Chem. Phys.* **145**, 204108 (2016).
- ⁵⁴U. Manthe, *J. Chem. Phys.* **128**, 064108 (2008).
- ⁵⁵P. A. M. Dirac, *Math. Proc. Cambridge Philos. Soc.* **26**, 376 (1930).
- ⁵⁶J. Frenkel, *Wave Mechanics* (Clarendon Press, Oxford, 1934).
- ⁵⁷C. Lubich, *From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis* (European Mathematical Society, Zurich, 2008).
- ⁵⁸D. J. Haxton and C. W. McCurdy, *Phys. Rev. A* **91**, 012509 (2015).
- ⁵⁹C. Lévesque and L. B. Madsen, *New J. Phys.* **19**, 043007 (2017).
- ⁶⁰G. A. Worth, *J. Chem. Phys.* **112**, 8322 (2000).
- ⁶¹M. H. Beck and H.-D. Meyer, *Z. Phys. D* **42**, 113 (1997).
- ⁶²U. Manthe, *Chem. Phys.* **329**, 168 (2006).
- ⁶³A. Raab, G. A. Worth, H.-D. Meyer, and L. S. Cederbaum, *J. Chem. Phys.* **110**, 936 (1999).
- ⁶⁴M. H. Beck, A. Jäckle, G. A. Worth, and H.-D. Meyer, *Phys. Rep.* **324**, 1 (2000).
- ⁶⁵G. H. Golub and C. F. Van Loan, *Matrix Computations* (The Johns Hopkins University Press, 1996).
- ⁶⁶L. Doriol, F. Gatti, C. Iung, and H.-D. Meyer, *J. Chem. Phys.* **129**, 224109 (2008).
- ⁶⁷H.-D. Meyer, F. L. Quere, C. Leonard, and F. Gatti, *Chem. Phys.* **329**, 179 (2006).
- ⁶⁸E. R. Davidson, *J. Comput. Phys.* **17**, 87 (1975).
- ⁶⁹L. Halonen and M. S. Child, *Comput. Phys. Commun.* **51**, 173 (1988).
- ⁷⁰A. Shimshovitz, Z. Bačić, and D. J. Tannor, *J. Chem. Phys.* **141**, 234106 (2014).
- ⁷¹J. Brown and T. Carrington, Jr., *J. Chem. Phys.* **143**, 044104 (2015).
- ⁷²T. Carrington, Jr., "Using iterative eigensolvers to compute vibrational spectra," *Adv. Chem. Phys.* (to be published).
- ⁷³M. Rakhuba and I. Oseledets, *J. Chem. Phys.* **145**, 124101 (2016).
- ⁷⁴D. Bégué, P. Carbonnière, and C. Pouchan, *J. Phys. Chem. A* **109**, 4611 (2005).
- ⁷⁵P. S. Thomas and T. Carrington, Jr., *J. Phys. Chem. A* **119**, 13074 (2015).
- ⁷⁶G. Avila and T. Carrington, Jr., "Pruned bases that are compatible with iterative eigensolvers and general potentials: New results for CH₃CN," *Chem. Phys.* **482**, 3–8 (2017).
- ⁷⁷U. Manthe, *J. Chem. Phys.* **142**, 244109 (2015).
- ⁷⁸C. Lubich, *Appl. Math. Res. Express* **2015**, 311.
- ⁷⁹F. Otto, *J. Chem. Phys.* **140**, 014106 (2014).
- ⁸⁰A. Jäckle and H.-D. Meyer, *J. Chem. Phys.* **104**, 7974 (1996).
- ⁸¹S. Manzhos and T. Carrington, Jr., *J. Chem. Phys.* **129**, 224104 (2008).
- ⁸²U. Manthe, *J. Chem. Phys.* **130**, 054109 (2009).
- ⁸³R. Dawes and T. Carrington, Jr., *J. Chem. Phys.* **121**, 726 (2004).
- ⁸⁴S. A. Smolyak, *Sov. Math. Dokl.* **4**, 240 (1963).
- ⁸⁵F.-J. Delves and W. Schempp, *Pitman Research Notes in Mathematics* (Longman, New York, 1989), Vol. 230.
- ⁸⁶H.-J. Bungartz and M. Griebel, *Acta Numer.* **13**, 147 (2004).
- ⁸⁷G. Avila and T. Carrington, *J. Chem. Phys.* **143**, 214108 (2015).
- ⁸⁸See <http://mctdh.uni-hd.de> for Heidelberg MCTDH package.