A pruned collocation-based multi-configuration time-dependent Hartree approach using a Smolyak grid for solving the Schrödinger equation with a general potential energy surface

Robert Wodraszka\textsuperscript{a} and Tucker Carrington Jr.\textsuperscript{b}

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

(Dated: March 18, 2019)

Abstract

Standard multi-configuration time-dependent Hartree (MCTDH) calculations use a direct product basis and rely on the potential being a sum of products (SOPs). The size of the direct product MCTDH basis scales exponentially with the number of atoms. Accurate potentials may not be SOPs. We introduce an MCTDH approach that uses a pruned basis and a collocation grid. Pruning the basis significantly reduces its size. Collocation makes it possible to do calculations using a potential that is not a SOP. The collocation point set is a Smolyak grid. Strategies using pruned MCTDH bases already exist, but they work only if the potential is a SOP. Strategies for using MCTDH with collocation also exist, but they work only if the MCTDH basis is a direct product. In this paper we combine a pruned basis with collocation. This makes it possible to mitigate the direct-product basis size problem and do calculations when the potential is not a SOP. Because collocation is used, there are no integrals and no need for quadrature. All required matrix-vector products can be evaluated sequentially. We use nested sets of collocation points and hierarchical basis functions. They permit efficient inversion of the (large) matrix whose elements are basis functions evaluated at points, which is necessary to transform values of functions at points to basis coefficients. The inversion technique could be used outside of chemical physics. We confirm the validity of this new pruned, collocation-based (PC-)MCTDH approach by calculating the first 50 vibrational eigenenergies of CH$_2$NH.

\textsuperscript{a} Electronic address: Robert.Wodraszka@chem.queensu.ca
\textsuperscript{b} Electronic address: Tucker.Carrington@queensu.ca
I. INTRODUCTION

A direct product (sometimes called a tensor product) basis is often used to solve the Schrödinger equation required to study the motion of nuclei because it facilitates the evaluation of matrix-vector products (MVPs)\textsuperscript{1–5}. Better bases can be made from eigenfunctions of Hamiltonians that depend on a subset of the coordinates\textsuperscript{6–13} and by pruning a direct product basis,\textsuperscript{14–25} however, the corresponding MVPs are more complicated.\textsuperscript{8,19,22,23,26} A direct product basis has the disadvantage that its size scales exponentially with $D$, the dimensionality of the problem. As a result, the computer memory required to store vectors, the cost of a single MVP, and the required number of MVPs all become large as $D$ increases. These problems are important for systems with more than 5 atoms. Due in part to their generality and simplicity, direct product methods are nevertheless popular. If one decides to use a direct product basis then it makes sense to optimise the basis functions to reduce the size of the basis. Optimisation of a direct product basis is the key idea of the multi-configuration time-dependent Hartree (MCTDH) method.\textsuperscript{27–29} It is best when coupling is not too strong and when it is not necessary to propagate for a long time. Standard MCTDH basis functions are products of optimised univariate functions, which are called single-particle functions (SPFs). The optimisation appreciably reduces the number of functions required in each dimension in many practical applications.

The standard MCTDH wavefunction is,

$$\Psi (x_1, \ldots, x_D, t) = \sum_{i_1=1}^{n_1} \ldots \sum_{i_D=1}^{n_D} A_{i_1 \ldots i_D} (t) \varphi^{(1)}_{i_1} (x_1, t) \varphi^{(2)}_{i_2} (x_2, t) \ldots \varphi^{(D)}_{i_D} (x_D, t),$$

where the SPFs are,

$$\varphi^{(k)}_{i_k} (x_k, t) = \sum_{j_k=1}^{N_k} c^{(k)}_{j_k,i_k} (t) \chi^{(k)}_{j_k} (x_k) \quad \forall k = 1 \ldots D.$$ \hspace{1cm} (2)

Standard MCTDH has two important deficiencies. 1) Although the MCTDH direct product basis is smaller than the direct product basis made from the primitive univariate bases $\chi^{(k)}_{j_k}$ with which the SPFs are computed ($n_k < N_k$), its size increases exponentially with $D$. 2) To calculate matrix elements in the MCTDH basis, it is almost always necessary to make approximations.

Using newer ideas, it is possible to generalise the MCTDH approach to mitigate the first deficiency. Some of these newer ideas make use of optimised multivariate SPFs: mode
combination (MC),\textsuperscript{30} and multi-layer (ML-)MCTDH\textsuperscript{31–33}. Another newer approach is to use a nondirect product basis made by pruning the MCTDH direct product basis. This is the pruned MCTDH method (P-MCTDH).\textsuperscript{34–37} MC, ML-MCTDH, and P-MCTDH all reduce the cost of MCTDH calculations by orders of magnitude.

Standard MCTDH, MC, ML-MCTDH, and P-MCTDH can only be used to solve the Schrödinger equation if it is possible to compute matrix elements in the SPF basis and are therefore hampered by the second deficiency. It is common to compute matrix elements by replacing the potential with a sum-of-products approximation (the exact kinetic energy operator is often a sum of products).\textsuperscript{38} In a sum of products (SOP) potential, each term is a product of operators acting on a single coordinate. When the potential is a SOP and the basis is a direct product, every matrix element of the potential is a product of 1-D integrals and MVPs can be efficiently evaluated by doing sums \textit{sequentially},\textsuperscript{1,39} i.e., factor by factor for each of the terms in the Hamiltonian. Each of the factors, when represented in the 1-D basis for the corresponding coordinate, is a small matrix. If $n = 10$, then the small matrices are $\sim 10 \times 10$. $n$ is a representative value of $n_k$. There is no need to calculate, store, or manipulate huge matrices with $n^{2D}$ elements. It has been shown that the sequential summation idea also works with SOP Hamiltonians and nondirect product bases.\textsuperscript{19}

Unfortunately, \textit{ab initio} potential energy surfaces (PESs) are often not SOPs, at least not in coordinates suitable for quantum dynamics calculations. Consequently, several approaches have been proposed to convert such PESs into SOP form.\textsuperscript{40–46} The approximation introduced by converting is a manifestation of the second deficiency. Most MCTDH calculations have been done with a SOP made using either the Potfit algorithm\textsuperscript{40} or a Potfit extension.\textsuperscript{44,47,48} To use the original Potfit algorithm, it is necessary to store the PES on a direct product primitive grid and this is not possible for systems with more than about 7 coordinates. Neural network SOP PESs can be made without storing potential values on a direct product grid.\textsuperscript{41,49} Regardless of how the SOP PES is generated, the cost of an MCTDH calculation with a SOP PES scales linearly with the number of terms and for large systems the number of terms is unavoidably large. It is clearly desirable to develop ideas that make it possible to use general \textit{ab initio} PESs, without making approximations, and thereby to overcome the second deficiency.

Manthe’s correlation discrete variable representation (CDVR) does make it possible to do MCTDH calculations with a general PES. The CDVR is an intermediate representa-
tion used to compute matrix elements in a basis of products of SPFs.\textsuperscript{50} A conventional (time-dependent) DVR is not sufficient because the SPF bases that are large enough to represent wavefunctions are not large enough that replacing the matrix representation of the PES in the multi-dimensional DVR with a diagonal matrix is a good approximation. The CDVR therefore uses a "correction". There is evidence that accurate results can be obtained using the CDVR and standard MCTDH.\textsuperscript{51–53} Results obtained from 1-D CDVR functions are as good as the SPF basis and the product approximation.\textsuperscript{2,54} CDVR functions that depend on more than one coordinate are obtained by finding a transformation matrix that nearly simultaneously diagonalises (SD)\textsuperscript{55} non-commuting matrices.\textsuperscript{56–58} Because nearly diagonal matrices are replaced with diagonal matrices, an SD CDVR has an additional approximation. This is another manifestation of the second deficiency. Difficulties with SD are exacerbated by numerical problems caused by needing to use unoccupied SPFs in ML-MCTDH calculations.\textsuperscript{59} Because the CDVR is time-dependent, using it requires evaluating the PES at each time step. Repeatedly evaluating the potential takes about 90\% of the computer time in a typical CDVR calculation.\textsuperscript{60}

In this paper, we present a method that mitigates both of the MCTDH deficiencies. To do this, we have used P-MCTDH, which reduces the size of the basis, with collocation, which obviates the need for a SOP potential. Collocation is a technique for solving the Schrödinger equation that completely avoids integrals. To use collocation, one demands that a finite representation of solutions of the Schrödinger equation exactly satisfy the Schrödinger equation at a set of collocation points. Collocation is reminiscent of interpolation, but rather than approximating a function that is know at points, one is determining wavefunctions that satisfy the Schrödinger equation at points. Collocation is sometimes called the pseudospectral method,\textsuperscript{61} but note that it does not require transforming\textsuperscript{62} from the basis (grid) to the grid (basis) and then back to the basis (grid) and does not use DVR basis functions that are nonzero at one point and zero at all other points. The standard collocation eigenvalue equation for a 1-D Schrödinger equation is (atomic units are used throughout the paper),\textsuperscript{63}

\[
\left[ -\frac{1}{2m} \mathbf{B}'' + \mathbf{V}^d \mathbf{B} \right] \mathbf{U} = \mathbf{BUE},
\]

where $\mathbf{B}''$ is the matrix whose elements are second derivatives of basis functions evaluated at collocation points, $\mathbf{V}^d$ is a diagonal matrix of values of the potential at collocation points, and $\mathbf{B}$ is the matrix whose elements are basis functions evaluated at collocation points.
Eq. (3) can be converted, by left multiplying by $B^TW$, where $W$ is a diagonal matrix of weights, into an eigenvalue equation that is equivalent to the one obtained by using a variational method and evaluating matrix elements of the PES, the KEO, and the overlap matrix by quadrature. Note, however, that in most variational calculations (regardless of whether DVR or FBR is used) matrix elements of the KEO are exact (i.e., "variational basis representation") and not computed by quadrature.² Beside the key benefit that no quadrature is required, an important advantage of collocation is that a sufficiently complete basis makes the choice of collocation points immaterial;⁶⁴ accurate results can (disregarding numerical subtleties), in principle, be obtained from any set of points provided it is possible to represent the desired wavefunctions exactly as linear combinations of the basis functions.

The collocation-based MCTDH (C-MCTDH) equations we recently presented⁶⁵ make it possible, like the CDVR, to use MCTDH with a general PES. The C-MCTDH working equations were derived using a (Petrov-)Galerkin-like strategy, which, in the standard MCTDH case, is equivalent to the more common Dirac-Frenkel variational principle.⁶⁶–⁶⁸ We have shown that, at least for some problems, one is able to attain accurate eigenenergies with C-MCTDH using either collocation points that are determined as the SPF basis evolves (like CDVR time-dependent points)⁴⁸ or points that are chosen at the beginning and held constant during the calculation. The latter procedure saves a lot of computer time because the PES has to be evaluated at only one set of points.

C-MCTDH is thus a means of mitigating deficiency 2). It obviates the need to use a SOP PES, but requires using a direct product basis and a direct product collocation grid. P-MCTDH, on the other hand, uses a nondirect product basis, orders of magnitude smaller than the direct product basis from which it is extracted, and is a means of mitigating deficiency 1), but requires using a SOP PES. In this paper we use C-MCTDH and P-MCTDH together to deal with both deficiencies. This is accomplished by using a special sparse-gird or Smolyak interpolant.
II. MCTDH USING A DIRECT PRODUCT BASIS

A. Standard MCTDH

In its original and simplest form, the multi-dimensional MCTDH wavefunction \( \Psi(x_1, \ldots, x_D, t) \) is expanded in a time-dependent direct (tensor) product basis, see Eq. (1) and Eq. (2). The primitive time-independent orthonormal basis functions \( \chi_{jk}^{(k)}(x_k) \) are often DVR functions made from weighted orthogonal polynomials (Hermite, Legendre, etc.). The MCTDH wavefunction representation is known in the mathematical literature as the Tucker format.\(^{69}\) The SPFs are optimal at all times and therefore evolve in time.

To solve the Schrödinger equation with the above wavefunction ansatz, differential equations for the tensor \( A_{i_1, \ldots, i_D}(t) \) and for the (rectangular) matrices \( c^{(k)} \) were derived by employing the Dirac-Frenkel variational principle.\(^{27,28}\) Using orthonormal SPFs and the standard gauge condition, \( \langle \varphi_{ik}^{(k)}(t) | \varphi_{ik}^{(k)}(t) \rangle = 0 \), the following set of equations was obtained:

\[
\begin{align*}
    i \dot{A}_{i_1, \ldots, i_D}(t) &= \sum_{i_1=1}^{n_1} \ldots \sum_{i_D=1}^{n_D} H_{i_1, \ldots, i_D} A_{i_1, \ldots, i_D}(t) \quad (4) \\
    \sum_{i_k=1}^{n_k} \rho_{ik}^{(k)} i |\hat{\varphi}_{ik}^{(k)}(t)\rangle &= \left( \hat{1} - \hat{P}^{(k)} \right) \sum_{i_k=1}^{n_k} \hat{H}_{ik}^{(k)} |\varphi_{ik}^{(k)}(t)\rangle \quad \forall k = 1 \ldots D. \quad (5)
\end{align*}
\]

These equations contain: 1) elements of a Hamiltonian matrix, \( H_{i_1, \ldots, i_D, \ldots, i_D}^{i_1, \ldots, i_D} = \langle \varphi_i^{(1)}(t) | \ldots | \varphi_{i_D}^{(D)}(t) \rangle \hat{H} |\varphi_i^{(1)}(t)\rangle \ldots |\varphi_{i_D}^{(D)}(t)\rangle \); 2) density matrices, \( \rho_{ik}^{(k)} = \sum_{i_{-k} \ldots i_D} A_{i_1, \ldots, i_k, \ldots, i_D}^{*}(t) A_{i_1, \ldots, i_k, \ldots, i_D}(t) \), where \(-k\) means that the \( k\)-th index is omitted from the summation; 3) the projection operators, \( \hat{P}^{(k)} = \sum_{i_k} |\varphi_{ik}^{(k)}(t)\rangle \langle \varphi_{ik}^{(k)}(t)| \); and 4) the mean-field matrices, \( \hat{H}_{ik}^{(k)} = \sum_{i_1, \ldots, i_D} A_{i_1, \ldots, i_k, \ldots, i_D}^{*}(t) \left( \bigotimes_{l \neq k} |\varphi_{i_l}^{(l)}(t)\rangle \right) \hat{H} \left( \bigotimes_{l \neq k} |\varphi_{i_l}^{(l)}(t)\rangle \right) A_{i_1, \ldots, i_D}(t) \). Equations for the \( c^{(k)} \) matrices are obtained from Eq. (5) by putting \( |\chi_{jk}^{(k)}\rangle \) on the left and inverting the density matrix. Eqs. (4) and (5) are coupled and non-linear because \( H_{i_1, \ldots, i_D}^{i_1, \ldots, i_D} \) depends on the SPFs and \( \hat{H}_{ik}^{(k)} \) depends on the \( A \) tensor. Specially adapted integrators (constant mean-field, CMF) have been introduced, which, on a short time scale, approximately decouple Eqs. (4) and (5) and thus efficiently integrate the differential equations. In this paper, we employ the CMF2 scheme of Ref. 70.
B. Collocation-based MCTDH (C-MCTDH)

The standard MCTDH equations are usually derived from the wavefunction ansatz in Eqs. (4) and (5) by using the Dirac-Frenkel variational principle. It is not obvious how to use the Dirac-Frenkel variational principle to derive C-MCTDH equations. In Ref. 65, we propose a (Petrov-)Galerkin-type procedure that is equivalent to the Dirac-Frenkel variational principle when both the basis functions in the ansatz and the ”test” functions are SPFs, but can also be used with collocation, when ”test” functions are Dirac delta functions. In the direct product case, the C-MCTDH equations are:

$$i \dot{A}_{i_1,\ldots,i_D}(t) = \sum_{a_1,\ldots,a_D} \sum_{j_1,\ldots,j_D} \left[ B^{(1)}_{a_1,\ldots,a_D}^{-1} \right]_{i_1,a_1} \cdots \left[ B^{(D)}_{a_D}^{-1} \right]_{j_D,a_D} H_{a_1,\ldots,a_D} A_{j_1,\ldots,j_D}(t)$$

and

$$\sum_{i_k=1}^{n_k} \mathcal{B}^{(k)}_{i_k,i_k} \left| \varphi^{(k)}_{i_k}(t) \right\rangle = \left\{ 1 - \sum_{j_k,a_k=1}^{n_k} \left| \varphi^{(k)}_{j_k}(t) \right\rangle \left\langle B^{(k)}_{j_k,a_k} \right| \varphi^{(k)}_{i_k}(t) \right\} \sum_{i_k=1}^{n_k} \mathcal{H}^{(k)}_{i_k,i_k} \left| \varphi^{(k)}_{i_k}(t) \right\rangle .$$

Here, the $r^{(k)}_{a_k}(t)$ denote the (they might be time-dependent) collocation points and the matrix elements

$$B^{(k)}_{i_k,i_k} = \varphi^{(k)}_{i_k} \left( r^{(k)}_{a_k}(t) , t \right) = \left\langle r^{(k)}_{a_k}(t) | \varphi^{(k)}_{i_k}(t) \right\rangle$$

are values of the SPFs at those points. Except for the inverses of the $B^{(k)}$ matrices in Eq. (6), the above equations have exactly the same form as the standard MCTDH equations (4) and (5). However, the definitions of the C-MCTDH projector $\hat{P}^{(k)}$, Hamiltonian matrix, and mean-field matrices are not the standard MCTDH definitions. The latter two are

$$H_{i_1,\ldots,i_D} = \left\langle r^{(1)}_{a_1}(t) | \cdots | r^{(D)}_{a_D}(t) \right| \hat{H} \left| \varphi^{(1)}_{i_1}(t) \right\rangle \cdots \left| \varphi^{(D)}_{i_D}(t) \right\rangle$$

and

$$\mathcal{H}^{(k)}_{i_k,i_k} = \sum_{i'_1,\ldots,i'_D} \sum_{a_1,\ldots,a_D} \sum_{i_1,\ldots,i_D} \left( \prod_{l \neq k} \mathcal{B}^{(l)}_{i_l,a_l}^{-1} \right) \left| \otimes_{l \neq k} \left\langle r^{(l)}_{a_l}(t) \right| \right\rangle \hat{H} \left( \otimes_{l \neq k} \varphi^{(l)}_{i_l}(t) \right) A_{i_1,\ldots,i_D}.$$
which are non-Hermitian. The C-MCTDH density matrices are

\[
\rho_{i_k, i_k}^{(k)} = \sum_{j_1, \ldots, j_D} A_{j_1, \ldots, i_k}^{*} A_{j_1, \ldots, i_k}^{j_1, \ldots, j_D} .
\]

Explicit equations for the \(c^{(k)}\) matrices (cf. Eq. (2)) are derived by projecting Eq. (7) from the left with \(\langle \tilde{r}^{(k)}_{a_k} \rangle\), where \(\tilde{r}^{(k)}_{a_k}\) is one of \(N_k\) fine-grid collocation points:

\[
\sum_{i_k=1}^{n_k} \rho^{(k)}_{i_k, i_k} c^{(k)}_{m_k, i_k} (t) = \sum_{\alpha_k=1}^{N_k} \sum_{i_k=1}^{n_k} \sum_{m_k=1}^{n_k} \left[ B^{(k)-1} \right]_{m_k, a_k} \langle \tilde{r}^{(k)}_{a_k} | \tilde{H}^{(k)}_{i_k, i_k} | \chi^{(k)}_{m_k} \rangle c^{(k)}_{m_k, i_k} (t)
\]

\[
- \sum_{j_k=1}^{n_k} \sum_{a_k=1}^{n_k} \sum_{i_k=1}^{n_k} \sum_{m_k=1}^{n_k} c^{(k)}_{m_k, j_k} \left[ B^{(k)-1} \right]_{j_k, a_k} \langle \tilde{r}^{(k)}_{a_k} (t) | \tilde{H}^{(k)}_{i_k, i_k} | \chi^{(k)}_{m_k} \rangle c^{(k)}_{m_k, i_k} (t) .
\]

The fine-grid points we use are the Gauss quadrature points corresponding to the time-independent basis of Eq. (2). The matrix elements \(\tilde{B}^{(k)}_{a_k, m_k} = \langle \tilde{r}^{(k)}_{a_k} | \chi^{(k)}_{m_k} \rangle\) are values of the time-independent primitive basis functions at the fine-grid collocation points \(\tilde{r}^{(k)}_{a_k}\).

The most compelling advantage of C-MCTDH is the fact that matrix elements of the PES and the KEO can be easily and exactly computed without first forcing operators into SOP form. Note that, \(\langle r^{(1)}_{a_1} (t) \ldots r^{(D)}_{a_D} (t) | V (x_1, \ldots, x_D) | \varphi^{(1)}_{i_1} (t) \ldots | \varphi^{(D)}_{i_D} (t) \rangle = V \left( r^{(1)}_{a_1} (t), \ldots, r^{(D)}_{a_D} (t) \right) \prod_k B^{(k)}_{a_k, i_k},\) which is exact, no DVR-like diagonal approximation is necessary. In a similar fashion, coordinate-dependent coefficients in the KEO (e.g., the exact Watson KEO\textsuperscript{71} has complicated coordinate-dependent coefficients) can simply be evaluated at points. Exact and simple matrix elements would not be a significant advantage if it were not possible to compute matrix-vector products by doing sums sequentially. As explained in Ref. 65, in the direct product case, this is possible. In this paper, we use C-MCTDH, but with a nondirect product, pruned basis. It is imperative that we develop ideas that make it possible to evaluate sums sequentially when the basis is not a direct product.

III. C-MCTDH USING A NONDIRECT PRODUCT BASIS AND A SMOLYAK GRID

A. Equations for calculating the \(A\) tensor and the collocation point set

It is rather straightforward to write the equations we must solve to use a pruned basis. The equation for the \(A\) coefficients is the same as Eq. (6), except the sum over basis indices
is restricted, because of the pruning,
\[
\sum_{g(i_1,\ldots,i_D)\leq H} i \dot{A}_{i_1,\ldots,i_D}(t) \prod_{k=1}^{D} \varphi^{(k)}_{i_k}(r^{(k)}_{a_k}(t), t) = \sum_{g(i_1,\ldots,i_D)\leq H} \langle r^{(1)}_{a_1}(t) | \ldots | r^{(D)}_{a_D}(t) | \hat{H} | \varphi^{(1)}_{i_1}(t) \rangle \ldots | \varphi^{(D)}_{i_D}(t) \rangle A_{i_1,\ldots,i_D}. \tag{13}
\]

The function \( g(i_1, \ldots, i_D) \) determines which multi-indices are included in the basis. In this paper we use,
\[
g(i_1, \ldots, i_D) = \sum_{k=1}^{D} (i_k - 1). \tag{14}
\]

In Refs. 34,36,72, we used more general pruning conditions. Note that the basis indices \( i_k = 1, \ldots n_k, k = 1, \ldots D \) start at 1. Pruned C-MCTDH (PC-MCTDH) will work best if, instead of the SPFs defined in Eq. (7), the \( \varphi^{(k)}_{i_k}(t) \) are natural orbitals (NOs).\(^{73,74}\) NOs are linear combinations of the SPFs of Eq. (7). The transformation between the SPF basis and the NO basis is given by the eigenvectors of a density matrix like Eq. (11), but obtained after orthogonalising the SPFs. In the rest of this paper, \( \varphi^{(k)}_{i_k}(t) \) is a NO. We shall write Eq. (13),
\[
\sum_{g(i_1,\ldots,i_D)\leq H} B_{a_1,j_1}^{(1)} \ldots B_{a_D,j_D}^{(D)} i \dot{A}_{i_1,\ldots,i_D}(t) = \sum_{g(j_1,\ldots,j_D)\leq H} H_{a_1,\ldots,a_D} A_{j_1,\ldots,j_D}(t), \tag{15}
\]

where
\[
H_{a_1,\ldots,a_D} = K_{a_1,\ldots,a_D} + V_{a_1,\ldots,a_D} \tag{16}
\]

and
\[
V_{a_1,\ldots,a_D} = V \left( r^{(1)}_{a_1}(t), \ldots, r^{(D)}_{a_D}(t) \right) B_{a_1,j_1}^{(1)} \ldots B_{a_D,j_D}^{(D)}. \tag{17}
\]

\( V \) is the matrix for the PES, and \( V \left( r^{(1)}_{a_1}(t), \ldots, r^{(D)}_{a_D}(t) \right) \) is the PES evaluated at a collocation point. \( K \) is the matrix for the KEO,
\[
K_{a_1,\ldots,a_D} = \hat{K} \varphi^{(1)}_{i_1} \varphi^{(2)}_{i_2} \ldots \varphi^{(D)}_{i_D} \bigg|_{a_1,\ldots,a_D}, \tag{18}
\]

which is easily calculated even when terms in \( \hat{K} \) have coordinate dependent coefficients. To use Eq. (15), we must specify a set of collocation points that is as large as the pruned basis.
To solve it, we will invert $B$, a matrix whose rows are labelled by $a_1, \ldots, a_D$, whose columns are labelled by $i_1, \ldots, i_D$, and whose elements are

$$B_{a_1, \ldots, a_D}^{i_1, \ldots, i_D} = B_{a_1, i_1}^{(1)} \cdots B_{a_D, i_D}^{(D)} \quad \forall g(a_1, \ldots, a_D) \leq H \quad \text{and} \quad g(i_1, \ldots, i_D) \leq H .$$

(19)

As collocation points, we choose the points generated by using a Smolyak recipe for evaluating multi-dimensional quadratures in the pruned basis. Good quadrature points are good collocation points because energies computed from collocation are equal to energies computed by using a variational method and evaluating all matrix elements, also those of the KEO and the overlap matrix, by quadrature. The input for Smolyak’s prescription is a sequence of 1-D grids for each of the coordinates. Smolyak’s quadrature equation has a sum over contributing grids, each of which is a tensor product of 1-D grids that are members of the sequences. In the special case that the 1-D sequences are nested, a Smolyak quadrature can be evaluated by summing over points in the set that is the union of the contributing grids. This makes it possible to do MVPs by summing sequentially, without summing over contributing grids. In this paper, for all the coordinates, the sequences of 1-D points have 1, 2, 3... points. We refer to the 1-D grid with 1 point as the level $\ell = 1$ grid, the 1-D grid with 2 points as the level $\ell = 2$ grid, etc. $m_\ell$ is the number of points in the level $\ell$ grid. Although we use $m_\ell = \ell$, different choices are possible. A grid is nested when all points in the grid for level $\ell$ are also in the grid for level $\ell + 1$. Our collocation point set is the union grid. Most mathematicians and physicists use Smolyak (sparse grid) equations with a sum over levels. By using the union grid, it is possible to avoid the sum over levels, both for quadrature and for interpolation/collocation.

The $A$ tensor is determined from Eq. (15), by multiplying both sides with $B^{-1}$,

$$i \dot{A}_{i_1, \ldots, i_D}(t) = \sum_{a_1, a_2, \ldots, a_D} [B^{-1}]_{i_1, i_2, \ldots, i_D}^{a_1, a_2, \ldots, a_D} \sum_{g(j_1, \ldots, j_D) \leq H} H_{a_1, \ldots, a_D}^{j_1, \ldots, j_D} A_{j_1, \ldots, j_D}(t) ,$$

(20)

where $H_{a_1, \ldots, a_D}^{j_1, \ldots, j_D}$ is given in Eq. (16) and Eq. (17) and $\sum_{a_1, a_2, \ldots, a_D}$ is over all the points of the union grid. To use this equation, we must develop tools for applying $B$ and $K$ to a basis vector and $B^{-1}$ to a grid vector. These MVPs are more efficient, if one uses a "hierarchical" basis.
Hierarchical functions are made from a 1-D basis set which is divided into levels and a set of points associated with the levels. A recipe for making hierarchical functions is given in Ref. 72. In this paper, we have one function in level 1, two functions in level 2 etc. If the functions (natural orbitals) are \( \varphi_{i_k}^{(k)} \), then in level 1, we have \( \varphi_{1_k}^{(k)} \), in level 2 we have \( \left( \varphi_{1_k}^{(k)}, \varphi_{2_k}^{(k)} \right) \), in level 3 we have \( \left( \varphi_{1_k}^{(k)}, \varphi_{2_k}^{(k)}, \varphi_{3_k}^{(k)} \right) \), etc. Correspondingly, in level 1, we have the point \( r_{1_k}^{(k)} \), in level 2 we have the points \( \left( r_{1_k}^{(k)}, r_{2_k}^{(k)} \right) \), in level 3 we have \( \left( r_{1_k}^{(k)}, r_{2_k}^{(k)}, r_{3_k}^{(k)} \right) \), etc. The hierarchical functions are

\[
\tilde{\varphi}_{i_k}^{(k)}(x_k) = \sum_{j_k=1}^{i_k} \tilde{A}_{i_k,j_k} \varphi_{j_k}^{(k)}(x_k),
\]

where \( \tilde{A}_{i_k,j_k} \) is chosen so that \( \tilde{\varphi}_{i_k}^{(k)}(r_{a_k}^{(k)}) = 0 \) \( \forall \ a_k < i_k \) and \( \tilde{A}_{i_k,i_k} = 1 \). Note that \( \tilde{\varphi}_{i_k}^{(k)}(x_k) \) depend on the interpolation points and \( \varphi_{i_k}^{(k)}(x_k) \) do not. The 1-D points must be chosen so that \( \tilde{B}^{(k)} \) is not singular, or near-singular, where \( \tilde{B}_{a_k,i_k}^{(k)} = \tilde{\varphi}_{i_k}^{(k)}(r_{a_k}^{(k)}) \); in our experience that is never a problem.

Mathematicians using sparse grid methods often use piecewise linear basis functions, \( \theta^{(\ell)}_j \), that have the property that functions in level \( \ell \) are equal to zero at all points in levels 1, 2, \ldots, \( \ell - 1 \). They call these piecewise linear functions hierarchical. The mathematicians’ hierarchical functions are importance-ordered, i.e., when the functions they wish to approximate are expanded in the \( \theta^{(\ell)}_j \) basis, the coefficients of functions labelled by \( \ell \) are larger than those of functions labelled by \( \ell' > \ell \). The \( \varphi_{i_k}^{(k)} \) natural orbitals used in this paper are also importance-ordered, but they are not zero at points in previous levels. In this paper (and in Ref. 72) we call the \( \tilde{\varphi}_{i_k}^{(k)}(x_k) \) functions hierarchical because they, like the hierarchical piecewise linear functions, have the property that functions in level \( \ell \) are equal to zero at points in levels 1, 2, \ldots, \( \ell - 1 \). We designate this the zero-at-points-in-previous-levels (ZAPPL) property. Using the prescription of Eq. (21), one can make a basis with the ZAPPL property that spans the same space as \emph{any} \( \varphi_{i_k}^{(k)} \) basis. In Ref. 72, the KEO is applied to a grid vector by first transforming it to the hierarchical basis and the potential is applied directly to a grid vector. In this paper, the KEO is applied directly to a basis vector and the potential is applied to a basis vector by transforming the vector to the grid using the \( \tilde{B} \) matrix.

The hierarchical basis functions are important for two reasons. First, they simplify computing MVPs with \( \tilde{B} \) (see Eq. (17)). Second, they simplify computing MVPs with \( \tilde{B}^{-1} \),
which are required to solve Eq. (20). Hereafter, we shall drop the tildes: \( \varphi_{i_k}^{(k)}(x_k) \) basis functions and \( B^{(k)} \) matrices are always hierarchical.

C. Computing matrix-vector products with \( B \)

To evaluate a MVP with \( B \), one must loop over all of the points on the union grid. This can be done, as shown in Ref. 23, whenever the 1-D grids are nested, without considering the small tensor product grids that together compose the union grid, by executing \( D \) loops, one for each coordinate. The maximum values of each of the \( a_k \) loops are:

\[
\begin{align*}
a_{D}^\text{max} &= H + 1 \\
a_{D-1}^\text{max} &= H - (a_D - 1) + 1 \\
&\vdots \\
a_{2}^\text{max} &= H - (a_3 - 1) - \ldots - (a_D - 1) + 1 \\
a_{1}^\text{max} &= H - (a_2 - 1) - \ldots - (a_D - 1) + 1 .
\end{align*}
\]

Note that \( a_k^\text{max} \) \((k = 1, 2, \ldots, (D - 1))\) is obtained from \( a_{k+1}, a_{k+2}, \ldots, a_D \) in the same way that \( i_k \) \((k = 1, 2, \ldots, (D - 1))\) is obtained from \( i_{k+1}, i_{k+2}, \ldots, i_D \). Eq. (22) is correct when \( m_k(\ell_k) = \ell_k \). If more than one point is added per level, then it is also possible to evaluate sums sequentially, but it is necessary to use a function \( M(a_k) \) that returns the smallest level that contains point \( a_k \).

Knowing how to loop over points on the grid and indices of the pruned basis, we are able to evaluate MVPs with \( B \). For example, for a 9-D problem, the MVP with \( B \) is,

\[
\begin{align*}
v^{(\text{out})}(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9) &= \sum_{i_9=1}^{i_{9}^\text{max}} B_{a_9,i_9}^{(9)} \sum_{i_8=1}^{i_{8}^\text{max}} B_{a_8,i_8}^{(8)} \sum_{i_7=1}^{i_{7}^\text{max}} B_{a_7,i_7}^{(7)} \\
&\sum_{i_6=1}^{i_{6}^\text{max}} B_{a_6,i_6}^{(6)} \sum_{i_5=1}^{i_{5}^\text{max}} B_{a_5,i_5}^{(5)} \sum_{i_4=1}^{i_{4}^\text{max}} B_{a_4,i_4}^{(4)} \sum_{i_3=1}^{i_{3}^\text{max}} B_{a_3,i_3}^{(3)} \sum_{i_2=1}^{i_{2}^\text{max}} B_{a_2,i_2}^{(2)} \sum_{i_1=1}^{i_{1}^\text{max}} B_{a_1,i_1}^{(1)} \\
v^{(\text{in})}(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9) \end{align*}
\]

The \( i_k \) are constrained among themselves and \( i_k^\text{max} \) is computed from

\[
(i_k^\text{max} - 1) + \sum_{k' = k+1}^{9} (i_{k'}^\text{max} - 1) = H .
\]
Eq. (23) is similar to the MVP in Eq. 16 of Ref. 83, but in this paper the input vector is a basis vector and the output vector is a grid vector. In general, the intermediate vectors are larger than the input and output vectors. In the MVP of Ref. 83, $B^{(k)}$ is replaced by a matrix $\tilde{T}^{(k)}$, made by taking the difference of two interpolants. Using the hierarchical functions of Ref. 83, with the ZAPPL property, $\tilde{T}^{(k)}$ is lower triangular which makes it possible to intertwine the indices when determining the upper limits on the sums. We can do the same in this paper because $B^{(k)}$ is also lower triangular. The intermediate vector obtained by doing the sum over $i_k$ is constrained by

$$(i_g - 1) + \ldots + (i_{k+1} - 1) + (a_k - 1) + \ldots + (a_1 - 1) \leq H$$  \hspace{1cm} (25)

and $i_k^{\max}$ is given by

$$i_k^{\max} = a_k.$$  \hspace{1cm} (26)

To implement Eq. (23) on a computer, one option is to store vectors labelled by 9 indices in arrays with a single index by using a mapping. When evaluating the sums, it is then necessary to convert multi-indices $(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9)$ into a mapping index. In Ref. 36, we instead use multi-index lists and we use the same idea in this paper. We need only a single multi-index list because the number of $a_k$ values is equal to the number of $i_k$ values, for all $k$. The lower triangularity of the $B_{a_k,i_k}^{(k)}$ matrices is encoded in the multi-index list. The multi-index list is used to determine the maximum index of each of the three loops of Eq. (18) of Ref. 36.

It is also necessary to compute MVPs with $K$. They can always be done term by term. Because each term in the normal-coordinate KEO we use has only one differential operator, a MVP for one of the terms in $K$ could be done the same way as the $B$ MVP, by first doing the sum for the coordinate being differentiated, see Ref. 83 for details. However, this would complicated the code and instead we use multi-index lists and the general scheme of Ref. 36. This means that some intermediate vectors have more elements than we have pruned basis functions. When it is necessary to avoid larger intermediate vectors, the approach of Ref. 83 can be used.
D. Computing matrix-vector products with $B^{-1}$

It is imperative that it be possible to efficiently evaluate MVPs with $B^{-1}$. When both the basis and the collocation point set are tensor products, then one can exploit the fact that $B = \bigotimes_{k=1}^D B^{(k)}$, a tensor product of small matrices ($B^{(k)}$), by replacing $B^{-1}$ with a tensor product of inverses of $B^{(k)}$ matrices. It is then easy to evaluate MVPs by doing sums sequentially. When the basis is pruned, $B$ is not a tensor product of small matrices and it is not obvious that it is possible to evaluate MVPs with $B^{-1}$ by doing sums sequentially.

In previous papers,\textsuperscript{72,83,84} the matrix $\tilde{T}$ transformed a vector labelled by points to a vector labelled by function indices. In this paper, the matrix that transforms a vector labelled by points to a vector labelled by basis indices is $B^{-1}$, and we present here equations for multiplying it by a vector that are simpler than those of Refs. 72,83. To understand the approach of this paper, it is not necessary to think in terms of Lagrange-type functions and a Smolyak interpolant written as a sum of tensor products of differences of interpolants.\textsuperscript{72,83} Both the equations of Ref. 72 for applying $\tilde{T}$ to a vector, and Eq. (38) of this paper can be thought of as prescriptions for transforming from values of a function at points on a Smolyak grid to coefficients representing the function in a pruned basis.

It appears that is not possible to use the ideas of subsection III C to evaluate a MVP with $B^{-1}$,

$$\sum_{g(a_1,\ldots,a_D) \leq H} \left( B^{-1} \right)_{a_1,\ldots,a_D}^{a_{i_1},\ldots,a_{i_D}} \zeta_{a_1,\ldots,a_D}^{i_1,\ldots,i_D},$$

because $B^{-1}$ is not a tensor product of lower triangular matrices. $B^{-1}$ is the inverse of a chopped tensor product matrix

$$B = C^T B^{TP} C,$$

where

$$B^{TP} = B^{(1)} \otimes \ldots \otimes B^{(D)},$$

and each $B^{(k)}$ is $n \times n$. $C$ is a rectangular chopping matrix. The number of columns is the number of functions in the pruned basis (and the number of points on the union grid). The number of rows is $n^D$. It is an identity matrix from which the columns that correspond to $g(i_1,\ldots,i_D) > H$ have been deleted.
We shall now show that although $B^{-1}$ is the inverse of a chopped tensor product matrix, it is nonetheless possible, again for a 9-D problem, to evaluate a $B^{-1}$ MVP using
\[
u^{(\text{out})}(i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9) = \sum_{a_9=1}^{a_9^{\max}} \left((B^{(9)})^{-1}\right)_{i_9,a_9} \sum_{a_8=1}^{a_8^{\max}} \left((B^{(8)})^{-1}\right)_{i_8,a_8} \sum_{a_7=1}^{a_7^{\max}} \left((B^{(7)})^{-1}\right)_{i_7,a_7}
\]
\[
\sum_{a_{6}=1}^{a_{6}^{\max}} \left((B^{(6)})^{-1}\right)_{i_6,a_6} \sum_{a_{5}=1}^{a_{5}^{\max}} \left((B^{(5)})^{-1}\right)_{i_5,a_5} \sum_{a_{4}=1}^{a_{4}^{\max}} \left((B^{(4)})^{-1}\right)_{i_4,a_4} \sum_{a_{3}=1}^{a_{3}^{\max}} \left((B^{(3)})^{-1}\right)_{i_3,a_3} \sum_{a_{2}=1}^{a_{2}^{\max}} \left((B^{(2)})^{-1}\right)_{i_2,a_2}
\]
\[
\sum_{a_{1}=1}^{a_{1}^{\max}} \left((B^{(1)})^{-1}\right)_{i_1,a_1} \nu^{(\text{in})}(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9),
\] (30)

where
\[
a_k^{\max} = i_k, \quad (a_9 - 1) + \ldots + (a_{k+1} - 1) + (i_k - 1) + \ldots + (i_1 - 1) \leq H, \quad (31, 32)
\]
in correspondence to Eqs. (25) and (26). We can do this by demonstrating that
\[
[C^T B^{\text{TP}} C]^{-1} = C^T (B^{\text{TP}})^{-1} C,
\] (33)

where $I$ is an identity matrix whose size is the size of the pruned basis. It is not hard to confirm that $P = CC^T$ is an $n^D \times n^D$ projection matrix that is diagonal and has ones and zeros on the diagonal. Using $P$, the condition, Eq. (33), that must be satisfied can be recast as
\[
[C^T B^{\text{TP}} C] C^T (B^{\text{TP}})^{-1} C = I,
\] (34)

In words, Eq. (34) requires that the block of $M = B^{\text{TP}} P [B^{\text{TP}}]^{-1}$ whose rows and columns are labelled by points on the union grid be an identity matrix. Thus, we examine
\[
M_{a_1',\ldots,a_D'}^{a_1,\ldots,a_D} = \sum_{k_1,\ldots,k_D} \sum_{\ell_1,\ldots,\ell_D} (B^{\text{TP}})_{a_1',\ldots,a_D'}^{K_1',\ldots,K_D} P_{k_1,\ldots,k_D}^{\ell_1,\ldots,\ell_D} \left((B^{\text{TP}})^{-1}\right)_{\ell_1,\ldots,\ell_D}^{a_1,\ldots,a_D}
\]
\[
\forall g(a_1',\ldots,a_D') \leq H \quad \text{and} \quad g(a_1,\ldots,a_D) \leq H.
\] (35)

Because $P$ is diagonal, $\sum_{\ell_p}$ can be removed and $\ell_p$ replaced by $k_p$. Because only the diagonal elements of $P$ that correspond to functions in the pruned basis are non-zero, only terms in $\sum_{k_1,\ldots,k_D}$ with $g(k_1,\ldots,k_D) \leq H$ are non-zero. Therefore
\[
M_{a_1',\ldots,a_D'}^{a_1,\ldots,a_D} = \sum_{g(k_1,\ldots,k_D) \leq H} (B^{\text{TP}})_{a_1',\ldots,a_D'}^{k_1,\ldots,k_D} \left((B^{\text{TP}})^{-1}\right)_{k_1,\ldots,k_D}^{a_1,\ldots,a_D}.
\] (36)
Owing to the fact that we sum over only some of the columns of $\mathbf{B}^{TP}$ and some of the rows of $[\mathbf{B}^{TP}]^{-1}$, the block of $\mathbf{M}$ in Eq. (36) is, in general, not an identity matrix. However, when the $\mathbf{B}^{(k)}$’s are lower triangular, then the terms omitted from the sum are zero and

$$M_{a'_1,\ldots,a'_D}^{a_1,\ldots,a_D} = \delta_a^{a'_1} \delta_{a_1} \delta_{a_2} \ldots \delta_{a'_k a_k}$$

$$\forall g(a'_1,\ldots,a'_D) \leq H \text{ and } g(a_1,\ldots,a_D) \leq H.$$  (37)

Because the block of $\mathbf{M}$ whose rows and columns are labelled by points on the union grid is an identity matrix, Eq. (34) is satisfied, $[\mathbf{C}^T \mathbf{B}^{TP} \mathbf{C}]^{-1} = \mathbf{C}^T \left( \mathbf{B}^{TP} \right)^{-1} \mathbf{C}$, and consequently the MVP in Eq. (27) can be re-written

$$\sum_{g(a_1,\ldots,a_D)\leq H} \left( \mathbf{C}^T \left( \mathbf{B}^{TP} \right)^{-1} \mathbf{C} \right)_{i_1,\ldots,i_D}^{a_1,\ldots,a_D} z_{i_1,\ldots,i_D}^{a_1,\ldots,a_D}.$$  (38)

Eq. (23) is simply an efficient way to compute a MVP with a chopped tensor product of lower triangular matrices and can therefore be used to evaluate Eq. (38). The resulting MVP is written explicitly in Eq. (30).

Evaluating the MVPs in Eq. (30) and Eq. (23) requires executing $D + 1$ nested loops. If the range of the indices were not restricted, then the cost would scale as $Dn^{D+1}$. In the Heidelberg SOP MCTDH package, $n^{D+1}$ is the cost of the MVP required to evaluate the MVP for a single term in the SOP PES. Because in the collocation calculations, the range of the indices is restricted, the MVPs with $\mathbf{B}^{-1}$ and $\mathbf{B}$ are less expensive than the cost of the MVP for a single term in the SOP case. If there are $n$ basis functions in each 1-D basis, then the cost of evaluating Eq. (23) and Eq. (30) scales as

$$\mathcal{O} \left( D \left[ \frac{n-1}{D+1} + 1 \right] N_{\text{prune}} \right),$$  (39)

where $N_{\text{prune}} = \binom{D+n-1}{D}$.

E. Pruned equations for the SPFs

The equation for the SPFs is obtained from Eq. (7) by restricting the sums in the definitions of the mean-field and density matrices. In Eq. (10), for example, allowed values of the grid and basis indices must satisfy the pruning condition. To compute the mean field matrices, it is necessary to evaluate MVPs with $\mathbf{B}^{(\sim k)}$ and $\left( \mathbf{B}^{(\sim k)} \right)^{-1}$, which are $\mathbf{B}$ and $\left( \mathbf{B} \right)^{-1}$ matrices from which the $k$-th factor has been removed. All the required restricted sums can be done sequentially.
F. Leja points

As explained in subsection III A, our collocation points are the points of the union Smolyak grid. It is built from nested sequences of 1-D grids for each coordinate. In this paper, the first 1-D interpolation rule (level $\ell = 1$) has one point, the second (level $\ell = 2$) has two points, the third (level $\ell = 3$) has three points, etc. We need a general scheme for making nested sets of points. One option is the Pseudo-Gauss quadratures of Ref. 23. Leja points also work well and are easier to determine.\textsuperscript{83–85} Leja points are designed to be good interpolation points.\textsuperscript{86–89} The most common Leja points are good when used with a basis of monomials.\textsuperscript{89–92} In Refs. 83,85, the 1-D points are Leja points that are good when a harmonic oscillator basis is used.\textsuperscript{87,88} Our basis functions are SPFs. In this paper, we use the corresponding Leja points. This is the first time that Leja points are used with a basis whose $k$-th function is not the square root of a weight function times a $(k - 1)$-th degree polynomial. For a particular coordinate, the SPF Leja points we use are obtained from the general recipe:\textsuperscript{93,94} 1) choose the first point $x_1$ to be the point at which $\varphi_1 (x)$ has its maximum value; 2) choose the $(k + 1)$-th point so that

$$x_{k+1} = \arg \max_{x \in (-\infty, +\infty)} \left| \varphi_{k+1} (x) - U^k (x_1, x_2, \ldots, x_k) \left[ \varphi_{k+1} (x) \right] \right|,$$

(40)

where $U^k (x_1, x_2, \ldots, x_k) \left[ \varphi_{k+1} (x) \right]$ is the interpolant for $\varphi_{k+1} (x)$ made using the first $k$ basis functions and points. $U^k (x_1, x_2, \ldots, x_k)$ is determined by choosing coefficients $c_j$ so that

$$\tilde{f} (x) = U^k f (x) = \sum_{j=1}^{k} \varphi_j (x) c_j ,$$

(41)

an interpolant for any $f (x)$, is exact at $x_1, \ldots, x_k$, i.e., $\tilde{f} (x_a) = f (x_a)$. The same points are obtained from the SPFs and the hierarchical functions made from them. From Eq. (40) it follows that the global maximum of a hierarchical function $\varphi_{k+1} (x)$ is at $x_{k+1}$.

As in Ref. 65, we shall do two different types of calculations. One calculation uses time-dependent collocation points. In this calculation, the SPFs evolve and with them the Leja points. If $\varphi_{k+1} (x)$ on the right side of Eq. (40) is time-dependent, then $x_{k+1}$ on the left side is also time dependent. Another set of calculations uses time-independent collocation points. In these calculations, the SPFs evolve, but Leja points do not: they are the Leja points obtained from the SPFs before the propagation begins (at $\beta = 0$, see Sec. IV A ).
G. Integrating the PC-MCTDH differential equations

As mentioned in Sec. II A, specifically adapted CMF integrators have been introduced for MCTDH calculations. The basic idea is to decouple the propagation of the $A$ coefficients and the SPFs during a CMF time step by evaluating the Hamiltonian matrix elements, the mean field matrices, and the density matrices at a particular time.\textsuperscript{70,95} Because the PC-MCTDH and MCTDH equations have the same structure, CMF integrators can also be used to solve the PC-MCTDH equations. We use the CMF2 integrator of Ref. 70. In a calculation with time-dependent points, each CMF integration step (i.e., $\Delta t$ in Ref. 70) converts an input basis into an output basis and consists of four stages: 1) transform the input basis into a natural orbital (NO) basis, analogous to what is done in P-MCTDH; 2) calculate new 1-D collocation (Leja) points; 3) transform the input basis to a hierarchical basis; 4) integrate the equations with the CMF procedure. In a calculation with time-independent points, stage 2) is omitted. One step has several intermediate sub-steps.\textsuperscript{70,95} During the intermediate sub-steps, several sets of SPFs and $A$ coefficients are calculated. Some of them are used to compute density matrices, mean field matrices, and a Hamiltonian matrix. To compute the mean field matrices and to use the Hamiltonian matrix to propagate the $A$ coefficients, one must evaluate MVPs with $B$ and with $B^{-1}$. These MVPs can be done by doing sums sequentially only, as explained in Secs. III C and III D, if $B^{(k)}$ is lower triangular. It is therefore imperative that $B^{(k)}$, for every set of SPFs calculated during the CMF integration, be lower triangular. This lower triangularity could be achieved by finding new Leja points and new hierarchical functions after each intermediate step of the CMF integrator. Instead, after completing an intermediate step, we define new hierarchical bases and transform the $A$ coefficients, but we use the same Leja points. This maintains the triangularity of the $B^{(k)}$s because

$$\frac{d}{dt} \varphi_{ik} (r(t), t) = \frac{\partial \varphi_{ik} (r(t), t)}{\partial r} \dot{r} + \frac{\partial \varphi_{ik} (r(t), t)}{\partial t} = 0 .$$

(42)

The second term is zero because the C-MCTDH equations we use are derived with the gauge

$$\dot{\varphi}_{ik}^{(k)} (r^{(k)} _{ak} (t), t) = 0 ,$$

(43)

and therefore the values of the SPFs at the collocation points do not change during the integration. The first term is also zero because the collocation points are fixed during each integration step, even for a calculation with time-dependent points.
IV. VIBRATIONAL ENERGY LEVELS OF CH$_2$NH

A. Obtaining energy eigenpairs in PC-MCTDH

We test the new PC-MCTDH approach by calculating the first 50 vibrational eigenpairs of CH$_2$NH. To this end, we solve the time-independent Schrödinger equation. As first presented in Ref. 51, this can be done by combining the state-averaged MCTDH idea with the block power method. Because MCTDH is designed to solve the time-dependent Schrödinger equation, the power method is used by applying $e^{-\beta \hat{H}}$ to the $M$ functions in the block. This simply requires the substitution $t = -i\beta$, $\beta \in \mathbb{R}_{>0}$. In the state-averaged framework, multiple functions $\Psi_s(x_1, \ldots, x_D)$, $s = 1 \ldots M$, are expanded in the same SPF basis. In our calculation, the tensors $A_{i_1,\ldots,i_D,s}$ therefore have an additional index $s$ over which a further sum must be done when calculating the density and mean-field matrices. Otherwise, there is no change in the working equations.

The block power method consists of iterative applications of $e^{-\beta \hat{H}}$ to $\Psi_s(x_1, \ldots, x_D)$ intertwined with Gram-Schmidt orthogonalisations of the resulting vectors. We use $\beta = 100$ hartree$^{-1}$ in this paper. After a sufficient number of steps, diagonalisation of the $M \times M$ matrix

$$\langle \Psi_s | \bigotimes_{k=1}^{D} \hat{\rho}^{(k)} \hat{H} | \Psi_{s'} \rangle$$

$$= \sum_{i_1,\ldots,i_D} \sum_{j_1,\ldots,j_D} \sum_{a_1,\ldots,a_D} \sum_{\ell_1,\ldots,\ell_D} A_{i_1,\ldots,i_D,s}^{(1)} S^{(1)}_{i_1,j_1} \ldots S^{(D)}_{i_D,j_D} B^{(1)}_{j_1,a_1} \ldots B^{(D)}_{j_D,a_D} H_{\ell_1,\ldots,\ell_D} A_{\ell_1,\ldots,\ell_D,s'}$$

yields accurate eigenenergies and eigenstates, given that the SPF basis is large enough. The C-MCTDH projector (Eq. (7)) is inserted in Eq. (44) because it makes it possible to apply $\hat{H}$ to $\Psi_{s'}$. Note that in contrast to Eq. (23) of Ref. 65, we now have overlap matrices, $S_{i_k,j_k}^{(k)} = \langle \phi_{i_k}^{(k)} | \phi_{j_k}^{(k)} \rangle$, on the right side of Eq. (44). This is necessary because to apply $H$ and $B^{-1}$ to $A$, it is necessary to transform the SPFs obtained at the end of the final integration step to a hierarchical basis which is necessarily non-orthogonal. In Ref. 65 it was possible to orthogonalise the final SPFs because lower triangularity was not required to evaluate the MVPs in Eq. (44).
B. Numerical details

We use 9 dimensionless (frequency-scaled) normal coordinates $q_k$ and the simple kinetic energy operator

$$
\hat{T} = -\frac{1}{2} \sum_{k=1}^{9} \omega_k \frac{\partial^2}{\partial q_k^2}
$$

with the harmonic wavenumbers 3491.5, 3198.3, 3086.8, 1685.1, 1510.1, 1397.2, 1089.4, 1186.4, and 1101.9 cm$^{-1}$. We employ a quartic force-field potential with the force constants reported in Refs. 97 The results we present are for the SPF basis sizes, in the same order as the harmonic wavenumbers, (5, 5, 5, 5, 5, 7, 7, 7). We use 20 harmonic oscillator basis functions to represent each of the SPFs. No attempt is made to optimise the primitive basis sizes because they have little effect on the cost of the calculation.

C. Results

The errors of the first 50 energies of CH$_2$NH (with respect to the reference values of Ref. 26) computed with time-dependent points are displayed in the form of a histogram in Fig. 1. We used two nondirect product basis sets, one with about 11,000 basis functions and one with about 23,000 functions. The corresponding direct product basis would have 5,359,375 functions. Clearly, roughly doubling the basis set size significantly improves the quality of the results. The maximum error drops from 3.78 to 0.95 cm$^{-1}$ and about half of the energies have errors < 0.1 cm$^{-1}$ when the larger basis is employed. Errors of the higher levels are noticeably larger with the smaller basis.

Although collocation is easy to use, energies computed with it are not the most accurate energies that can be obtained from a given basis. The most accurate energies are obtained by using a variational method and exact integrals (for the potential, overlap, and KEO matrix elements). The difference between an energy computed with collocation and the corresponding energy computed with a variational method and exact integrals decreases as the size of basis and point sets increase and is zero for an energy whose corresponding wavefunction can be exactly represented in the basis. To achieve similar errors, it is therefore expected that collocation will require a somewhat larger basis set. This is indeed what we observe. Energies calculated with the same pruning condition and P-MCTDH are more
FIG. 1: Errors of the first 50 vibrational eigenenergies of CH$_2$NH obtained with time-dependent points and PC-MCTDH and two different nondirect product SPF bases.

accurate than those we obtain with PC-MCTDH.$^{34}$ In the P-MCTDH calculation, all matrix elements are evaluated exactly. For the force-field PES used for CH$_2$NH, this is easy to do. In general, exactly evaluating potential matrix elements is not an option and either collocation or CDVR is necessary. From Fig. 1 it is clear that increasing the basis size decreases errors and it is therefore possible to compensate for the error introduced by using collocation.

The same energy levels were also calculated with time-independent points. The time-independent points are chosen as in Ref. 65. To determine the required sizes of the SPF bases, we do a series of calculations with increasing SPF basis sizes which we denote $B_1$, $B_2$, etc. The $B_{k+1}$ time-independent points we use are Leja points associated with the $B_{k+1}$ SPF bases at $\beta = 0$, some of whose functions are the $B_k$ SPFs at $\beta = \beta_{final}$. For details see Ref. 65. Energies computed with time-independent points are about as accurate as those computed with time-dependent points. In Fig. 2, we show differences of the energy
FIG. 2: Differences of energy eigenvalues obtained with time-dependent and time-independent collocation points, respectively

eigenvalues obtained using time-dependent points and time-independent points for the basis with about 11,000 functions.

Because we are using a SOP PES, it would also possible to compute vibrational energy levels of CH$_2$NH with the Heidelberg MCTDH package and to compare the cost of the collocation calculation with the cost of the SOP calculation. We have not done this, but it seems likely that the pruned collocation calculation would be cheaper. The pruned basis we use is about two orders of magnitude smaller than the direct product basis made from the same 1-D SPF bases. In the SOP case, applying the potential requires evaluating as many MVPs are there are terms in the SOP PES, for CH$_2$NH there are 159 terms. In the collocation case, applying the potential requires a single MVP. It is less costly than the MVP for one term in the SOP calculation (because the range of the sums is restricted). One might also wonder how the cost of our calculation compares to the cost of a calculation with
the CDVR. Since the CDVR uses a direct product basis and we use a nondirect product (pruned) basis, this comparison is not simple. Nevertheless, it is known that in CDVR calculations, roughly 90% of the CPU time\textsuperscript{60} is due to the need to repeatedly evaluate the PES, during the propagation, and when using collocation the PES must only be evaluated once.

V. CONCLUSION

MCTDH is widely recognised as being an excellent tool for solving the Schrödinger equation to compute cross sections, rate constants, and spectra.\textsuperscript{29} The key advantage of MCTDH is its optimised basis set. Due to the optimisation, although it uses a direct product basis, it is possible to compute converged results for Hamiltonians with about 10 degrees of freedom. MCTDH is best when a short propagation is sufficient and when coupling is not too strong. The vast majority of MCTDH calculations are done with the Heidelberg package which requires a SOP PES.\textsuperscript{38} As advantageous as it is, MCTDH is limited by the size of its direct product basis and its reliance on SOP PESs. The PC-MCTDH of this paper uses MCTDH ideas, but a nondirect product, pruned basis and does not require a SOP PES.

In PC-MCTDH, the working equations closely resemble those of unpruned C-MCTDH.\textsuperscript{65} The key difference is that all sums over basis and grid indices are restricted by a pruning condition. Crucial to the success of the PC-MCTDH method is a nondirect product collocation grid. The grid is built from sequences of 1-D grids using ideas of Smolyak. To evaluate matrix-vector products efficiently, it is imperative that the sequences be nested. To use the working equations, we must apply to a vector the inverse of a large matrix whose elements are functions in the nondirect product basis evaluated at points on the Smolyak grid. We show in this paper that this can be done by using nested sequences, so-called hierarchical basis functions, and sequentially summing over indices that label functions and points for a single coordinate. This matrix-vector product essentially transforms a nondirect product grid vector to a nondirect product basis vector. It should be useful also in other fields.

Pruning is one of three strategies that have been implemented to obviate the need for a direct product MCTDH basis.\textsuperscript{34–37} The other two are mode combination\textsuperscript{30} and multi-layer MCTDH\textsuperscript{31–33}. Of these three, pruning is the first to be combined with ideas that make it possible to use non-SOP PESs in a systematically improvable fashion. Collocation is one of
two approaches that have been implemented to obviate the need for a SOP PES. The other is Manthe’s CDVR. CDVRs have the disadvantage that they require repeatedly evaluating the PES during the propagation. It is sometimes not possible to systematically increase the accuracy of a CDVR calculation by increasing the number of CDVR points since additional SPFs are poorly defined. Because this problem develops as the SPF basis becomes complete, it may not affect collocation calculations. Recently, however, Manthe has developed a generalisation of his CDVR that could be used with ML-MCTDH to do numerically exact calculations. The pruning/collocation combination should also make it possible to use non-SOP PESs with bases, such as those one has when using mode combination or ML-MCTDH, whose functions are products of multi-variate factors.

Acknowledgements

The financial support of the Natural Sciences and Engineering Research Council is gratefully acknowledged. We thank Uwe Manthe and Hans-Dieter Meyer for commenting on a preliminary version of the paper.
U. Manthe, private communication.


