A multi-dimensional Smolyak collocation method in curvilinear coordinates for computing vibrational spectra

Gustavo Avila and Tucker Carrington Jr.

Citation: *The Journal of Chemical Physics* **143**, 214108 (2015); doi: 10.1063/1.4936294

View online: <http://dx.doi.org/10.1063/1.4936294>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/jcp/143/21?ver=pdfcov>

Published by the AIP Publishing

Articles you may be interested in

Using multi-dimensional Smolyak interpolation to make a sum-of-products potential

J. Chem. Phys. **143**, 044106 (2015); 10.1063/1.4926651

Hierarchical expansion of the kinetic energy operator in curvilinear coordinates for the vibrational self-consistent field method

J. Chem. Phys. **135**, 124102 (2011); 10.1063/1.3637629

Toward large scale vibrational configuration interaction calculations

J. Chem. Phys. **131**, 124129 (2009); 10.1063/1.3243862

Vibrational energy levels of CH₅ +

J. Chem. Phys. **129**, 234102 (2008); 10.1063/1.3027825

Iterative solutions with energy selected bases for highly excited vibrations of tetra-atomic molecules

J. Chem. Phys. **120**, 4626 (2004); 10.1063/1.1646370

The logo for AIP APL Photonics. It features the letters 'AIP' in a large, white, sans-serif font, followed by a vertical yellow bar and the words 'APL Photonics' in a smaller, white, sans-serif font. The background is a red gradient with a bright yellow sunburst effect.

APL Photonics is pleased to announce
Benjamin Eggleton as its Editor-in-Chief

A multi-dimensional Smolyak collocation method in curvilinear coordinates for computing vibrational spectra

Gustavo Avila^{a)} and Tucker Carrington, Jr.^{b)}

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

(Received 4 September 2015; accepted 10 November 2015; published online 2 December 2015)

In this paper, we improve the collocation method for computing vibrational spectra that was presented in Avila and Carrington, Jr. [J. Chem. Phys. **139**, 134114 (2013)]. Using an iterative eigensolver, energy levels and wavefunctions are determined from values of the potential on a Smolyak grid. The kinetic energy matrix-vector product is evaluated by transforming a vector labelled with (nondirect product) grid indices to a vector labelled by (nondirect product) basis indices. Both the transformation and application of the kinetic energy operator (KEO) scale favorably. Collocation facilitates dealing with complicated KEOs because it obviates the need to calculate integrals of coordinate dependent coefficients of differential operators. The ideas are tested by computing energy levels of HONO using a KEO in bond coordinates. © 2015 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4936294>]

I. INTRODUCTION

This paper is about a new collocation method for solving the Schrödinger equation that describes the motion of nuclei on a potential energy surface (PES). All collocation methods determine solutions by requiring that the equation being solved be satisfied at a set of points. Although collocation was introduced into chemical physics decades ago,^{1,2} it is rarely used. The two most important reasons are: (1) the standard collocation approach requires solving a generalized eigenvalue problem; (2) collocation has not been used with structured basis sets and grids that make it possible to reduce the cost of evaluating the matrix-vector products required to solve the eigenvalue problem with an iterative eigensolver. Owing to these problems, variational methods have thrived.^{3–7} They come in various types, e.g., vibrational self-consistent field, vibrational configuration interaction, multi-configuration time dependent Hartree, etc., but all share the essential idea that wavefunctions are represented as a sum of basis functions, $b_k(x_1, x_2, \dots, x_D)$, and coefficients are obtained from a matrix representation of the Hamiltonian operator in the basis. Such methods require computing integrals $\langle b_k(x_1, x_2, \dots, x_D) | \hat{H} | b_l(x_1, x_2, \dots, x_D) \rangle$. When D , the number of coordinates, is larger than about nine, computing integrals of the potential is costly. To implement this approach, it is common to approximate the potential as a sum of products (SOP) of low dimensional functions.^{8,9} Vibrational coupled cluster calculations can also be used with a SOP potential.¹⁰ Full dimensional quadrature is possible,^{11–13} but remains a significant barrier. Collocation obviates the need for integrals, but one must confront the two problems listed above. In Ref. 14, a collocation method is presented that bypasses both problems. In this paper, we generalize and refine that method.

In a standard collocation calculation, wavefunctions are represented,

$$\Psi_n(x_1, x_2, \dots, x_D) = \sum_i F_i b_i(x_1, x_2, \dots, x_D); \quad (1)$$

the collocation grid has as many points as there are for basis functions, and the eigenvalue problem one must solve is generalized because the matrix obtained by evaluating basis functions at points is full.^{1,2,15} It is also possible to solve a rectangular generalized eigenvalue problem and use more points than basis functions.^{16–19} If the matrices are large, solving a generalized eigenvalue problem is costly. The generalized eigenvalue problem can be avoided by using a direct product discrete variable representation (DVR) (Lagrange-type) basis. Because each 1D DVR function is one at one point and zero at all other points, the matrix obtained by evaluating basis functions at points is an identity matrix. Regrettably, this simple idea, which enables one to use the collocation idea without solving a generalized eigenvalue problem, is useless, because the size of the direct product basis and grid is huge. In Ref. 14, we present a collocation method that uses Lagrange-type functions, to avoid the need to solve a generalized eigenvalue problem; however, rather than combining them to make a direct product basis, they are combined to make interpolants that exactly represent functions that satisfy certain conditions. When D is large, the size of the collocation grid of Ref. 14 is many orders of magnitude smaller than the size of the direct product grid. The eigenvalue problem one derives from the Schrödinger equation (see Section IV) by replacing the wavefunction with the interpolant is simple: $(\mathbf{K} + \mathbf{V})\mathbf{U} = \mathbf{U}\mathbf{E}$, i.e., it is not generalized. The potential matrix is diagonal. $\mathbf{K}$, for the kinetic energy operator (KEO), is *not* simple.

In this paper, we evaluate matrix-vector products for the KEO by using a new idea for writing the interpolant of Ref. 14 as a sum of product basis functions. The sum-of-basis functions form is obtained, using a sequential summation

^{a)}Electronic address: Gustavo_Avila@telefonica.net

^{b)}Electronic address: Tucker.Carrington@queensu.ca. Fax: 613-533-6669.

approach, without solving a linear system whose size is the number of basis functions. The sequential evaluation of sums has been used with direct-product^{5,11,20–25} and nondirect product^{12,26–29} bases. In this paper, in contrast to what was done in Ref. 14, rather than evaluating a matrix-vector product for a term in the KEO by implicitly constructing a grid representation of the operator and applying it to a vector labelled by points, we instead apply the operator to basis functions, evaluate at collocation points, and multiply by a vector labelled by basis indices obtained by transforming the vector labelled by points. Obviating the need to do matrix-vector products with vectors labelled by points reduces the central processing unit cost by orders of magnitude. Transforming the grid vector to a basis vector is not simple because the basis is not a direct product basis (it is pruned) and the grid is not a direct product grid (it is a Smolyak grid). The transformation is the subject of Section V. It is of interest in and of itself. It could be used, for example, to represent a PES as a (pruned) sum of products of 1D basis functions knowing only the value of the PES at a set of points. The points are those in a set that is the union of points on small direct product grids.

Another distinction between this paper and Ref. 14 is the coordinates used. The results presented in Ref. 14 were obtained with normal coordinates, and in this paper, we use curvilinear internal coordinates. A collocation approach should be most advantageous when used with curvilinear internal coordinates. This is due to the fact that coupling will often be less important in curvilinear coordinates³⁰ and that it is therefore possible to choose better basis functions and reduce the number of required points. When using collocation, the error introduced by using non-optimal points decreases as the quality of the basis set improves.³¹ Curvilinear coordinates are better suited for describing large amplitude motion and high-lying states, and it is important to demonstrate that the Smolyak-based collocation method can also be used with these coordinates. Using normal coordinates, it would be very difficult to calculate the HONO vibrational levels reported in Section VII.

Collocation has several advantages: (1) there is no need to compute potential matrix elements by quadrature and therefore no need to choose good quadrature weights; (2) there is no need to either choose coordinates and basis functions so that matrix elements of the KEO can be obtained from simple closed-form equations or use quadrature to compute integrals for terms in complicated KEOs; (3) coordinate-dependent coefficients in the KEO do not complicate calculations; (4) there is great freedom in the choice of basis functions, e.g., they need not be orthogonal; and (5) if the basis is good then one needs fewer collocation points than the number of quadrature points that would be required using a variational method and quadrature.

II. MULTIDIMENSIONAL SMOLYAK INTERPOLATION

In a collocation approach for solving the Schrödinger equation, one replaces wavefunctions with interpolants and determines values of the wavefunctions at interpolation points and the corresponding energies by requiring that the Schrödinger equation be satisfied at the points. The crux

of collocation is interpolation. We therefore begin with a presentation of interpolation methods.

It is, in principle, possible to easily find an interpolant that approximates a function, $F(x_1, x_2, \dots, x_D)$, whose value is known only on a direct product grid, without solving a large linear system. This is done by using a direct product of 1D Lagrange functions. 1D Lagrange functions are reviewed in Ref. 32 and many textbooks. This direct product Lagrange idea is not practical if D is large, but it motivates the Smolyak-based approach, which is practical. Let $a_k(x)$ be a 1D Lagrange function, equal to one at one interpolation point and zero at the other interpolation points. The Lagrange interpolant for a 1D function $f(x)$ is

$$\bar{f}(x) = U^m f(x) = \sum_{k=1}^m a_k(x) f(x_k), \quad (2)$$

where U^m is a 1D operator that performs the interpolation using m points and x_k is one of the m grid points. The $a_k(x)$ might be standard Lagrange functions,³² or weighted Lagrange functions,^{14,33} or combinations of 1D basis functions, $\varphi_n(x)$,

$$a_k(x) = \sum_{n=0}^{n^{max}} B_{k,n} \varphi_n(x), \quad k = 1, 2, \dots, m = n^{max} + 1. \quad (3)$$

In this paper, we shall use Lagrange type functions of the form of Eq. (3). Note that the $\varphi_n(x)$ (and the $a_k(x)$) need not be orthogonal. The $B_{k,n}$ coefficients are elements of the inverse of the matrix whose elements are $\varphi_n(x_k)$.

The most obvious way to use Lagrange interpolation in D dimensions is to use a direct product basis whose functions are products of 1D Lagrange functions. The coefficients are simply values of the function being interpolated,

$$\begin{aligned} \bar{F}(x_1, x_2, \dots, x_D) &= U^{m_1} \otimes U^{m_2} \otimes \dots \otimes U^{m_D} F(x_1, x_2, \dots, x_D) \\ &= \sum_{k_1=1}^{m_1} \sum_{k_2=1}^{m_2} \dots \sum_{k_D=1}^{m_D} a_{k_1}(x_1) a_{k_2}(x_2) \dots a_{k_D}(x_D) \\ &\quad \times F(x_{k_1}, x_{k_2}, \dots, x_{k_D}). \end{aligned} \quad (4)$$

The multi-D interpolant is built from 1D interpolation operators, $U^{m_c}(x_c)$, using 1D basis functions $\varphi_{n_c}(x_c)$, $n_c = 0, \dots, n_c^{max}$ and a set of points x_{k_c} , $k_c = 1, 2, \dots, n_c^{max} + 1$ for each coordinate ($c = 1, 2, \dots, D$). Although it is necessary to solve linear systems with $n_c^{max} + 1$ equations, there is no need to solve a linear system whose size depends on the total number of multi-D Lagrange functions. Nevertheless, the direct product basis is large if D is large, and when $D \sim 12$, it is not possible to store all the function values in memory. To avoid this problem, one must use a nondirect product interpolation method.

A Smolyak-based interpolation is a nondirect product interpolation.^{14,34–39} A multidimensional Smolyak interpolant is defined as

$$\begin{aligned}
I(D, H) &= \sum_{g(i_1, i_2, \dots, i_D) \leq H} \Delta U^{i_1}(x_1) \otimes \Delta U^{i_2}(x_2) \\
&\quad \otimes \dots \otimes \Delta U^{i_D}(x_D) \\
&= \sum_{g(i_1, i_2, \dots, i_D) \leq H} C_{i_1, i_2, \dots, i_D}^{smol} U^{i_1}(x_1) \otimes U^{i_2}(x_2) \\
&\quad \otimes \dots \otimes U^{i_D}(x_D),
\end{aligned} \quad (5)$$

where $U^0(x_c) = 0$ and

$$\Delta U^{i_c}(x_c) = U^{i_c}(x_c) - U^{i_c-1}(x_c). \quad (6)$$

The $C_{i_1, i_2, \dots, i_D}^{smol}$ coefficients are calculated by inserting Eq. (6) into the first line of Eq. (5). Some of the $C_{i_1, i_2, \dots, i_D}^{smol}$ may be zero. i_c , referred to in this article as a level, labels a member of a family of 1D interpolants for every coordinate, $c = 1, 2, \dots, D$. $m_c(i_c) = n_c^{max}(i_c) + 1$ is the number of points and basis functions included in the i_c interpolant for coordinate c ; $m_c(i_c) \geq m_c(i_c - 1)$. The interpolant $I(D, H)F(x_1, x_2, \dots, x_D)$ is equal to $F(x_1, x_2, \dots, x_D)$ if $F(x_1, x_2, \dots, x_D)$ satisfies certain conditions.^{35,40} For more information about the general $g(i_1, i_2, \dots, i_D)$ constraint we use to restrict the terms in Eq. (5), see Refs. 29 and 40. In terms of the Lagrange-type functions,

$$\begin{aligned}
F(x_1, x_2, \dots, x_D) &\approx \bar{F}(x_1, x_2, \dots, x_D) = I(D, H)F(x_1, x_2, \dots, x_D) \\
&= \sum_{g(i_1, i_2, \dots, i_D) \leq H} C_{i_1, i_2, \dots, i_D}^{smol} \sum_{k_1=1}^{m_1(i_1)} \sum_{k_2=1}^{m_2(i_2)} \dots \sum_{k_D=1}^{m_D(i_D)} F(x_{k_1}^{i_1}, x_{k_2}^{i_2}, \dots, x_{k_D}^{i_D}) a_{k_1}^{i_1}(x_1) a_{k_2}^{i_2}(x_2) \dots a_{k_D}^{i_D}(x_D) \quad (7) \\
&= \sum_{g(i_1, i_2, \dots, i_D) \leq H} \sum_{k_1=1}^{m_1(i_1)} \sum_{k_2=1}^{m_2(i_2)} \dots \sum_{k_D=1}^{m_D(i_D)} F(x_{k_1}^{i_1}, x_{k_2}^{i_2}, \dots, x_{k_D}^{i_D}) [a_{k_1}^{i_1}(x_1) - a_{k_1}^{i_1-1}(x_1)] \\
&\quad \times [a_{k_2}^{i_2}(x_2) - a_{k_2}^{i_2-1}(x_2)] \dots [a_{k_D}^{i_D}(x_D) - a_{k_D}^{i_D-1}(x_D)]. \quad (8)
\end{aligned}$$

An $x_{k_c}^{i_c}$ argument of F is the k_c value of x_c for the i_c 1D interpolant; the superscript indicates the set of points and the subscript indicates the point within the set. The points we use in this paper are nested: all points in level i_c are also in level $i_c + 1$. Eq. (8) is simplified by writing it as if $a_{k_c}^{i_c-1}$ were defined when $k_c > m_c(i_c - 1)$. To use it, one must put $a_{k_c}^{i_c-1}(x_c) = 0$ when $k_c > m_c(i_c - 1)$. Everywhere in this paper, we shall set $a_{k_c}^0(x_c) = 0$. $m_c(i_c)$ can be chosen in multiple ways. One option is to choose $m_c(i_c) = i_c$. In this paper, we denote this “interpolation A.” One could also take $m_c(i_c = 1) = 2$, add one point when augmenting the level index from $2i_c - 1$ to

$2i_c$, and add two points when augmenting the level index from $2i_c$ to $2i_c + 1$. In this paper, we denote this “interpolation B.”

It is useful to write a Smolyak interpolant as a sum over constrained n_c indices of products of $\varphi_{n_c}(x_c)$ factors. This is done by replacing the $a_{k_c}^{i_c}(x_c)$ in Eq. (8) using

$$\begin{aligned}
a_{k_c}^{i_c}(x_c) &= \sum_{n_c=0}^{n_c^{max}(i_c)} B_{k_c, n_c}^{i_c} \varphi_{n_c}(x_c), \quad (9) \\
k_c &= 1, \dots, n_c^{max}(i_c) + 1,
\end{aligned}$$

where $\mathbf{B}^{i_c} = (\mathbf{P}^{i_c})^{-1}$ with $(P^{i_c})_{n_c, k_c} = \varphi_{n_c}(x_{k_c})$. One obtains

$$\begin{aligned}
&\sum_{g(i_1, i_2, \dots, i_D) \leq H} \Delta U^{i_1}(x_1) \otimes \Delta U^{i_2}(x_2) \otimes \dots \otimes \Delta U^{i_D}(x_D) F(x_1, x_2, \dots, x_D) \\
&= \sum_{g(i_1, i_2, \dots, i_D) \leq H} \sum_{k_D=1}^{m_D(i_D)} \sum_{n_D=0}^{n_D^{max}(i_D)} [B_{k_D, n_D}^{i_D} - B_{k_D, n_D}^{i_D-1}] \varphi_{n_D}(x_D) \\
&\quad \times \sum_{k_{D-1}=1}^{m_{D-1}(i_{D-1})} \sum_{n_{D-1}=0}^{n_{D-1}^{max}(i_{D-1})} [B_{k_{D-1}, n_{D-1}}^{i_{D-1}} - B_{k_{D-1}, n_{D-1}}^{i_{D-1}-1}] \varphi_{n_{D-1}}(x_{D-1}) \dots \sum_{k_1=1}^{m_1(i_1)} \sum_{n_1=0}^{n_1^{max}(i_1)} [B_{k_1, n_1}^{i_1} - B_{k_1, n_1}^{i_1-1}] \varphi_{n_1}(x_1) F(x_{k_1}^{i_1}, x_{k_2}^{i_2}, \dots, x_{k_D}^{i_D}). \quad (10)
\end{aligned}$$

$B_{k_c, n_c}^{i_c}$ that are not defined, i.e., when $k_c > n_c^{max}(i_c - 1) + 1$ or $n_c > n_c^{max}(i_c - 1)$, are set to zero. Eq. (10) can be re-written as a sum over products of basis functions by moving the sums over $n_1, n_2, \dots$ to the left,

$$\begin{aligned}
&\sum_{g(M^1(n_1+1), M^2(n_2+1), \dots, M^D(n_D+1)) \leq H} \\
&\quad \times C_{n_1, n_2, \dots, n_D} \varphi_{n_1}(x_1) \varphi_{n_2}(x_2) \dots \varphi_{n_D}(x_D). \quad (11)
\end{aligned}$$

The upper limits of the sums over n_c must be constrained because in Eq. (10) they depend on i_c indices which are themselves constrained. For a general $g(i_1, i_2, \dots, i_D)$, a product $\varphi_{n_1}(x_1)\varphi_{n_2}(x_2)\cdots\varphi_{n_D}(x_D)$ will occur in the sum if

$$g(M^1(n_1+1), M^2(n_2+1), \dots, M^D(n_D+1)) \leq H, \quad (12)$$

where $M^i(n_i+1)$ is an integer that is the index of the interpolation rule U^i with the smallest value of i containing the function n_i+1 . For interpolation A (vide supra), $M^c(n_c) = n_c + 1$. For interpolation B (vide supra), $M^c(1) = M^c(2) = 1$, $M^c(3) = 2$, $M^c(4) = M^c(5) = 3$, etc. If $g(i_1, i_2, \dots, i_D) = i_1 + i_2 + \dots + i_D$, then the upper limits one must use to write Eq. (11) with sums over the individual n_c are $n_c^{\max} = m_c[H - M^{c+1}(1) \cdots - M^D(1) - M^1(n_1+1) - M^2(n_2+1) - \dots - M^{c-1}(n_{c-1}+1)] - 1$. In the simplest case, $m_c(i_c) = i_c$,

$$\begin{aligned} n_1^{\max} &= H - D, \quad n_2^{\max} = H - D - n_1, \\ n_3^{\max} &= H - D - n_1 - n_2, \dots, \text{etc.} \end{aligned} \quad (13)$$

It is important that it be possible to efficiently obtain $C_{n_1, n_2, \dots, n_D}$ from $F(x_{k_1}^{i_1}, x_{k_2}^{i_2}, \dots, x_{k_D}^{i_D})$. This transformation is the subject of Section VI. It is somewhat similar to transforming from a vector labelled by DVR points to a vector labelled by variational basis representation (VBR) indices.⁵ However, $F(x_{k_1}^{i_1}, x_{k_2}^{i_2}, \dots, x_{k_D}^{i_D})$ in Eq. (10) is not a coefficient in a direct product DVR and $C_{n_1, n_2, \dots, n_D}$ is not a coefficient in a direct product VBR. The transformation is only efficient if we use nested points, in which case the number of $F(x_{k_1}^{i_1}, x_{k_2}^{i_2}, \dots, x_{k_D}^{i_D})$ values is equal to the number of $C_{n_1, n_2, \dots, n_D}$ values. When the points are nested, some points on the multi-D grids that contribute to Eq. (5) are on many grids. The unique points will be written $(x_{k_1}, x_{k_2}, \dots, x_{k_{D-1}}, x_{k_D})$ and such a point will be referred to as a point on the Smolyak grid. There are no superscripts because the points from all contributing grids are included. If the points are nested, $F(x_{k_1}^{i_1}, x_{k_2}^{i_2}, \dots, x_{k_D}^{i_D})$ in Eq. (10) can be replaced by $F(x_{k_1}, x_{k_2}, \dots, x_{k_{D-1}}, x_{k_D})$. Just as Eq. (12) determines whether a product of $\varphi_{n_c}(x_c)$ functions is included in Eq. (11),

$$g(M^1(k_1), M^2(k_2), \dots, M^D(k_D)) \leq H \quad (14)$$

determines if a point $k_1, k_2, \dots, k_D$ is on the Smolyak grid. $M^c(k_c)$ is the smallest i_c that contains the point k_c .

III. HIERARCHICAL BASIS FUNCTIONS

It is advantageous to replace the $\varphi_{n_c}(x_c)$ used to make the Lagrange type functions in Eq. (9) with linear combinations of the $\varphi_{n_c}(x_c)$, denoted $\tilde{\varphi}_{n_c}(x_c)$, and to replace Eq. (11) with

$$\begin{aligned} &\sum_{g(M^1(n_1+1), M^2(n_2+1), \dots, M^D(n_D+1)) \leq H} \\ &\times \tilde{C}_{n_1, n_2, \dots, n_D} \tilde{\varphi}_{n_1}(x_1) \tilde{\varphi}_{n_2}(x_2) \cdots \tilde{\varphi}_{n_D}(x_D). \end{aligned} \quad (15)$$

Introducing the $\tilde{\varphi}_{n_c}(x_c)$ is helpful because it simplifies obtaining $\tilde{C}_{n_1, n_2, \dots, n_D}$ from $F(x_{k_1}, x_{k_2}, \dots, x_{k_D})$ (see Section VI). Because $\tilde{\varphi}_{n_c}(x_c)$ is a linear combination of $\varphi_{n'_c}(x_c)$, $n'_c = 0, 1, \dots, n_c$, the interpolant $\tilde{U}^{ic}$ built, as in

Eq. (2), but using $\tilde{a}_{k_c}^{ic}(x)$ Lagrange functions constructed as in Eq. (9), but from $\tilde{\varphi}_{n_c}(x_c)$ instead of $\varphi_{n_c}(x_c)$, is equivalent to the interpolant $\tilde{U}^{ic}$. Because the 1D interpolants are equivalent, for all coordinates and levels, Eqs. (15) and (11) are also equivalent. In other words, any function that can be represented as Eq. (11) can also be written as Eq. (15).

The $\tilde{\varphi}_{n_c}(x_c)$ we use are defined as follows:

$$\tilde{\varphi}_0(x_c) = \varphi_0(x_c), \quad (16)$$

and for $n_c > 0$,

$$\tilde{\varphi}_{n_c}(x_c) = \sum_{p_c=0}^{n_c-1} \tilde{A}_{n_c, p_c} \varphi_{p_c}(x_c) + \varphi_{n_c}(x_c). \quad (17)$$

The $\tilde{A}_{n_c, p_c}$ are determined so that

$$\tilde{\varphi}_{n_c}(x_{p_c}) = 0, \text{ if } n_c > 0, p_c = 1, \dots, n_c. \quad (18)$$

When the $\tilde{\varphi}_{n_c}(x_c)$ are organized into levels, as in interpolation A, a function in level i will be zero at the points in levels $1, 2, \dots, i-1$ and, if $n_c^{\max}(i) > n_c^{\max}(i-1) + 1$, $\tilde{\varphi}_{n_c}(x_c)$ in level i will also be zero at points x_{k_c} in level i for which $k_c < n_c$. Note that the $\tilde{\varphi}_{n_c}(x_c)$ depend on the interpolation points and the $\varphi_{n_c}(x_c)$ do not.

Several groups have used similar hierarchical functions.^{36,41-44} They, however, differ from our hierarchical functions in important ways. (1) It is most common to make a hierarchical basis with piecewise linear basis functions.⁴⁵ Smooth basis functions are better for our purposes. Some authors have used hierarchical bases built from Fourier or polynomial basis functions.^{36,41,42,46} (2) Our hierarchical basis functions can be made from any $\varphi_{n_c}(x_c)$. They are general and flexible. The $\varphi_{n_c}(x_c)$ from which our hierarchical functions are made are not polynomials whose degree is linked to n_c . The degree of our $\varphi_{n_c}(x_c)$ may be the same in all levels. (3) We have complete control over the number of functions in level $i_c + 1$ that are not in level i_c . In the standard hierarchical bases made from piecewise linear functions or Fourier or polynomial functions, the number of functions per level increases exponentially with i_c (without “delay”).⁴⁵ We are able to define levels so that the number of functions increases by one or two when the level index is augmented by one. By minimizing the number of functions in level $i_c + 1$ that are not in level i_c , we significantly reduce the number of points in the Smolyak grid. (4) Unlike all previous hierarchical bases, our hierarchical functions in level i_c are zero not only at points in level $i_c - 1$ but, if $n_c^{\max}(i) > n_c^{\max}(i-1) + 1$, also at some points in level i_c that are not in level $i_c - 1$. This makes it possible to make a full set of functions and then divide it up into levels as one wishes (any choice of $m_c(i_c)$ is possible).

When the interpolant is written as the first line of Eq. (5), using the hierarchical basis has the advantage that each term in the sum is a sum of a small number of products of $\tilde{\varphi}_{n_c}(x_c)$ factors. If $m_c(i_c) = m_c(i_c - 1) + 1$, then each term in the sum in Eq. (5) is a single product of $\tilde{\varphi}_{n_c}(x_c)$ factors. To see why

this is true, consider $\Delta \tilde{U}^{i_c}(x_c) = \tilde{U}^{i_c}(x_c) - \tilde{U}^{i_c-1}(x_c)$,

$$\begin{aligned} \Delta \tilde{U}^{i_c}(x_c) f(x_c) &= \tilde{U}^{i_c}(x_c) f(x_c) - \tilde{U}^{i_c-1}(x_c) f(x_c) \\ &= \sum_{n_c=0}^{n_c^{\max}(i_c)} \tilde{D}_{n_c} \tilde{\varphi}_{n_c}(x_c) - \sum_{n_c=0}^{n_c^{\max}(i_c-1)} \tilde{D}_{n_c} \tilde{\varphi}_{n_c}(x_c) \\ &= \sum_{n_c=n_c^{\max}(i_c-1)+1}^{n_c^{\max}(i_c)} \tilde{D}_{n_c} \tilde{\varphi}_{n_c}(x_c), \end{aligned} \quad (19)$$

where

$$\tilde{U}^{i_c}(x_c) f(x_c) = \sum_{n_c=0}^{n_c^{\max}(i_c)} \tilde{D}_{n_c} \tilde{\varphi}_{n_c}(x_c). \quad (20)$$

The $\tilde{D}_{n_c}$ coefficients are determined by resolving a system of $n_c^{\max}(i_c) + 1$ equations

$$f(x_{k_c}) = \sum_{n_c=0}^{n_c^{\max}(i_c)} \tilde{D}_{n_c} \tilde{\varphi}_{n_c}(x_{k_c}), \quad k_c = 1, \dots, n_c^{\max}(i_c) + 1, \quad (21)$$

or in matrix form

$$\mathbf{f} = (\tilde{\mathbf{P}}^{i_c})^T \mathbf{D}, \quad (22)$$

and $f(x_c)$ is (as in Eq. (2)) a 1D function one wishes to interpolate. $\tilde{\mathbf{P}}^{i_c}$ is the matrix whose (n_c, k_c) element is $\tilde{\varphi}_{n_c}(x_{k_c})$, with $n_c = 0, \dots, n_c^{\max}(i_c)$ and $k_c = 1, \dots, m_c(i_c) = n_c^{\max}(i_c) + 1$, and $\mathbf{f}$ and $\mathbf{D}$ are column vectors. The second line of Eq. (19) is equal to the third line because the $\tilde{D}_{n_c}$ are independent of i_c . Why? Organize the elements of $(\tilde{\mathbf{P}}^{i_c})^T$ into blocks so that the upper left block has the points and functions in level $i_c - 1$, and the lower right block has the points and functions in level i_c that are not in level $i_c - 1$. The upper right block of $(\tilde{\mathbf{P}}^{i_c})^T$ is zero (by virtue of Eq. (18)). The upper right block of $(\tilde{\mathbf{P}}^{i_c})^{-T}$ is also zero and therefore the top portion of the vector $\tilde{\mathbf{D}}$ computed with i_c is the same as $\tilde{\mathbf{D}}$ computed with $i_c - 1$.

To see why, as a consequence of Eq. (19), each term in the sum of products of Δ in Eq. (5) is itself a sum of a small number of products of $\tilde{\varphi}_{n_c}(x_c)$, consider an example in 2D and take $m_c = i_c$ and the term $i_1 = i_2 = 3$,

$$\begin{aligned} \Delta \tilde{U}^3(x_1) \otimes \Delta \tilde{U}^3(x_2) F(x_1, x_2) &= [\tilde{U}^3(x_1) - \tilde{U}^2(x_1)] \otimes [\tilde{U}^3(x_2) - \tilde{U}^2(x_2)] F(x_1, x_2) \\ &= \tilde{U}^3(x_1) \otimes \tilde{U}^3(x_2) F(x_1, x_2) - \tilde{U}^3(x_1) \otimes \tilde{U}^2(x_2) F(x_1, x_2) \\ &\quad - \tilde{U}^2(x_1) \otimes \tilde{U}^3(x_2) F(x_1, x_2) + \tilde{U}^2(x_1) \otimes \tilde{U}^2(x_2) F(x_1, x_2). \end{aligned} \quad (23)$$

Each of these terms can be written as a sum of products of $\tilde{\varphi}_{n_c}(x_c)$,

$$\begin{aligned} \tilde{U}^3(x_1) \otimes \tilde{U}^3(x_2) F(x_1, x_2) &= \tilde{D}_{2,2} \tilde{\varphi}_2(x_1) \tilde{\varphi}_2(x_2) + \tilde{D}_{2,1} \tilde{\varphi}_2(x_1) \tilde{\varphi}_1(x_2) \\ &\quad + \tilde{D}_{2,0} \tilde{\varphi}_2(x_1) \tilde{\varphi}_0(x_2) + \tilde{D}_{1,2} \tilde{\varphi}_1(x_1) \tilde{\varphi}_2(x_2) \\ &\quad + \tilde{D}_{1,1} \tilde{\varphi}_1(x_1) \tilde{\varphi}_1(x_2) + \tilde{D}_{1,0} \tilde{\varphi}_1(x_1) \tilde{\varphi}_0(x_2) \\ &\quad + \tilde{D}_{0,2} \tilde{\varphi}_0(x_1) \tilde{\varphi}_2(x_2) + \tilde{D}_{0,1} \tilde{\varphi}_0(x_1) \tilde{\varphi}_1(x_2) \\ &\quad + \tilde{D}_{0,0} \tilde{\varphi}_0(x_1) \tilde{\varphi}_0(x_2); \end{aligned} \quad (24)$$

$$\begin{aligned} \tilde{U}^3(x_1) \otimes \tilde{U}^2(x_2) F(x_1, x_2) &= \tilde{D}_{2,1} \tilde{\varphi}_2(x_1) \tilde{\varphi}_1(x_2) + \tilde{D}_{2,0} \tilde{\varphi}_2(x_1) \tilde{\varphi}_0(x_2) \\ &\quad + \tilde{D}_{1,1} \tilde{\varphi}_1(x_1) \tilde{\varphi}_1(x_2) + \tilde{D}_{1,0} \tilde{\varphi}_1(x_1) \tilde{\varphi}_0(x_2) \\ &\quad + \tilde{D}_{0,1} \tilde{\varphi}_0(x_1) \tilde{\varphi}_1(x_2) + \tilde{D}_{0,0} \tilde{\varphi}_0(x_1) \tilde{\varphi}_0(x_2); \end{aligned} \quad (25)$$

$$\begin{aligned} \tilde{U}^2(x_1) \otimes \tilde{U}^3(x_2) F(x_1, x_2) &= +\tilde{D}_{1,2} \tilde{\varphi}_1(x_1) \tilde{\varphi}_2(x_2) + \tilde{D}_{1,1} \tilde{\varphi}_1(x_1) \tilde{\varphi}_1(x_2) \\ &\quad + \tilde{D}_{1,0} \tilde{\varphi}_1(x_1) \tilde{\varphi}_0(x_2) + \tilde{D}_{0,2} \tilde{\varphi}_0(x_1) \tilde{\varphi}_2(x_2) \\ &\quad + \tilde{D}_{0,1} \tilde{\varphi}_0(x_1) \tilde{\varphi}_1(x_2) + \tilde{D}_{0,0} \tilde{\varphi}_0(x_1) \tilde{\varphi}_0(x_2); \end{aligned} \quad (26)$$

$$\begin{aligned} \tilde{U}^2(x_1) \otimes \tilde{U}^2(x_2) F(x_1, x_2) &= \tilde{D}_{1,1} \tilde{\varphi}_1(x_1) \tilde{\varphi}_1(x_2) + \tilde{D}_{1,0} \tilde{\varphi}_1(x_1) \tilde{\varphi}_0(x_2) \\ &\quad + \tilde{D}_{0,1} \tilde{\varphi}_0(x_1) \tilde{\varphi}_1(x_2) + \tilde{D}_{0,0} \tilde{\varphi}_0(x_1) \tilde{\varphi}_0(x_2). \end{aligned} \quad (27)$$

$\tilde{D}_{n_1, n_2}$ are coefficients in a 2D direct product representation,

$$\begin{aligned} \tilde{F}(x_1, x_2) &= \tilde{U}^{i_1}(x_1) \otimes \tilde{U}^{i_2}(x_2) F(x_1, x_2) \\ &= \sum_{n_1=0}^{n_1^{\max}(i_1)} \sum_{n_2=0}^{n_2^{\max}(i_2)} \tilde{D}_{n_1, n_2} \tilde{\varphi}_{n_1}(x_1) \tilde{\varphi}_{n_2}(x_2). \end{aligned} \quad (28)$$

Upon combining Eqs. (24)–(27), as indicated in Eq. (23), and exploiting the fact that the $\tilde{D}$ coefficients are independent of i_1 and i_2 , one finds

$$\Delta \tilde{U}^3(x_1) \otimes \Delta \tilde{U}^3(x_2) F(x_1, x_2) = \tilde{D}_{2,2} \tilde{\varphi}_2(x_1) \tilde{\varphi}_2(x_2). \quad (29)$$

Thus, the $i_1 = i_2 = 3$ term in Eq. (5) contains just one product of $\tilde{\varphi}_{n_c}(x_c)$ factors. In 3D, with $m_c(i_c) = i_c$,

$$\Delta \tilde{U}^5(x_1) \otimes \Delta \tilde{U}^2(x_2) \otimes \Delta \tilde{U}^{10}(x_3), \quad (30)$$

for example, is proportional to $\tilde{\varphi}_4(x_1) \tilde{\varphi}_1(x_2) \tilde{\varphi}_9(x_3)$. (When the original $\varphi_{n_c}(x_c)$ functions are used (instead of the $\tilde{\varphi}_{n_c}(x_c)$), $\Delta \tilde{U}^5(x_1) \otimes \Delta \tilde{U}^2(x_2) \otimes \Delta \tilde{U}^{10}(x_3)$ is a sum of 100 $\varphi_{n_1}(x_1) \varphi_{n_2}(x_2) \varphi_{n_3}(x_3)$ terms.) When there are D coordinates, the $\tilde{D}_{n_1, n_2, \dots, n_D}$ coefficients could be obtained by inverting a matrix whose size is $(n_1^{\max} + 1)(n_2^{\max} + 1) \dots (n_D^{\max} + 1)$ or alternatively from direct product Lagrange interpolation. Neither is required when we solve the Schrödinger equation, instead we make use of the transformation outlined in Section VI.

IV. DEFINING THE INTERPOLANTS TO USE SMOLYAK-BASED COLLOCATION

To make the Smolyak interpolant of Sec. III, we choose basis functions and points, define 1D interpolation rules, and choose the function $g(i_1, i_2, \dots, i_D)$ to minimize the

number of points (and functions) required to represent wavefunctions.

A. Choosing basis functions

For every coordinate, one must make basis functions $\varphi_n(x_c)$ $n = 0, \dots, N_c^{\max}$. At level i_c , Lagrange type functions like those of Eq. (9), $\tilde{a}_{k_c}^{i_c}(x)$, span the same space as $\tilde{\varphi}_{n_c}(x_c)$, $n_c = 0, 1, \dots, n_c^{\max}(i_c)$, from which they are made. The $\tilde{\varphi}_{n_c}(x_c)$, $n_c = 0, 1, \dots, n_c^{\max}(i_c)$ span the same space as the $\varphi_{n_c}(x_c)$, $n_c = 0, 1, \dots, n_c^{\max}(i_c)$. The $\tilde{a}_{k_c}^{i_c}(x)$ are made from $\tilde{\varphi}_{n_c}(x_c)$, which are made from $\varphi_{n_c}(x_c)$: everything depends on the choice of the $\varphi_{n_c}(x_c)$. We choose $\varphi_{n_c}(x_c)$ that are eigenfunctions of the 1D Schrödinger equation whose potential is a cut through the full PES. They are ordered so that functions with lower 1D energies have smaller n_c indices. It is ideal to choose the $\varphi_{n_c}(x_c)$ so that all wavefunctions can be represented with a small number of products. When coupling is weak, 1D cut eigenfunctions are certain to be good basis functions. It is sometimes useful to choose a minimized potential cut.

B. Choosing which functions are used at each level

$\tilde{U}^{i_c}$ is made from $m_c(i_c)$ $\tilde{a}_{k_c}^{i_c}$ functions, each of which is a linear combination of $m_c(i_c)$ $\tilde{\varphi}_{n_c}(x_c)$ and each of which is a linear combination of $m_c(i_c)$ $\varphi_{n_c}(x_c)$. Which $\tilde{\varphi}_{n_c}(x_c)$ functions are used to build $\tilde{U}^{i_c}$ interpolants? We use the first $m_c(i_c)$ functions in the $\varphi_{n_c}(x_c)$ $n = 0, \dots, N_c^{\max}$ list to make $\tilde{U}^{i_c}$. The functions in the i_c family include the functions in the $i_c - 1$ family and $n_c^{\max}(i_c) - (n_c^{\max}(i_c - 1))$ new functions. One simple way to choose the interpolation rules is to set $m_c(i_c) = i_c$. $\tilde{U}^1(x_c)$ uses $\tilde{\varphi}_0(x_c)$; $\tilde{U}^2(x_c)$ uses $\tilde{\varphi}_0(x_c), \tilde{\varphi}_1(x_c)$; $\tilde{U}^3(x_c)$ uses $\tilde{\varphi}_0(x_c), \tilde{\varphi}_1(x_c), \tilde{\varphi}_2(x_c)$; etc. This is interpolation A. It is also possible to start with two functions, add one function when augmenting the level index from $2i - 1$ to $2i$, and to add two points and functions when augmenting the level index from $2i$ to $2i + 1$. In this case, $\tilde{U}^1(x_c)$ has two functions; $\tilde{U}^2(x_c)$ has three functions; $\tilde{U}^3(x_c)$ has five functions; etc. This is interpolation B. It is also possible to “delay”⁴⁷ the sequence of interpolants by using the same number of functions in $\tilde{U}^i(x_c)$ and $\tilde{U}^{i-1}(x_c)$.

C. Choosing the points used at each level

For every coordinate, and for each i_c , one must choose points $x_{k_c}^{i_c}$. The points of each level are linked to the functions in the same level. When piecewise linear functions are used to interpolate, as is frequently done by mathematicians,⁴⁵ it is easy to link basis functions and points because the basis functions are localized and peaked at a single point.⁴⁵ However, piecewise linear functions are poor basis functions if one wishes to accurately interpolate smooth functions; spectral basis functions are much better. We need a good procedure for choosing points when using spectral basis functions. Szabados argues that when using 1D spectral functions, Gauss points are close to points that minimize the Lebesgue constant

and therefore probably good interpolation points.⁴⁸ Owing to the link between collocation and variational methods,³¹ good quadrature points might also be good collocation points. Sequences of Gauss quadrature points are, nevertheless, not good collocation points for our purpose because we wish to use nested points. When new basis function(s) are added to those of level i_c to obtain $\tilde{U}^{i_c+1}(x_c)$, the same number of points must be added to the points of level i_c to obtain the points of level $i_c + 1$. The question is how are those new points chosen? We use Pseudo-Gauss points.¹² There are certainly other good ways of choosing the new points. Any good set of points will have the property that none of the $\varphi_{n_c}(x_{k_c})$ are very small when evaluated at all the points.

According to the Pseudo-Gauss procedure, the new points are obtained by optimizing the calculation of elements of an overlap matrix S_{n_c, n'_c} with $n_c + n'_c \leq 2m_c(i_c) - 1$ for $\varphi_{n_c}(x_c)$ basis functions. We call these points Pseudo-Gauss because the $m_c(i_c) - m_c(i_c - 1)$ new points are chosen to calculate, as accurately as possible, all overlap integrals between $\varphi_{n_c}(x_c)$ functions, the sum of whose indices is less than or equal to $2m_c(i_c) - 1$, and if the $\varphi_{n_c}(x_c)$ were harmonic oscillator functions, then the overlap integrals for which points are optimized for level i_c would be exact if m_c Gauss points were used.

D. Choosing the $g(i_1, i_2, \dots, i_D)$

The function $g(i_1, i_2, \dots, i_D)$ determines which combinations of the 1D interpolation rules are included in $I(D, H)$ in Eq. (7). It is common to choose

$$g(i_1, i_2, \dots, i_D) = i_1 + i_2 + \dots + i_D. \quad (31)$$

Better choices often exist.^{29,40,49} Which $\varphi_{n_1}(x_1) \varphi_{n_2}(x_2) \dots \varphi_{n_D}(x_D)$ terms occur in Eq. (11) depends not only on $g(i_1, i_2, \dots, i_D)$ but also on the choice of $m_c(i_c)$, and of course also on H . In Ref. 50, to obtain a sum-of-products interpolated potential, we set $m_c(i_c) = i_c$ and the accuracy of the multi-D interpolations was controlled by carefully choosing a good $g(i_1, i_2, \dots, i_D)$. In this paper, to solve the Schrödinger equation, we make the simple choice, $g(i_1, i_2, \dots, i_D) = i_1 + i_2 + \dots + i_D$, and control the quality of the interpolation by adjusting $m_c(i_c)$.

V. SMOLYAK-BASED COLLOCATION TO SOLVE THE SCHRÖDINGER EQUATION

Sections II–IV explain how to make multi-D interpolants from 1D Lagrange functions that themselves are made from hierarchical basis functions. We shall now use those interpolants to devise a Smolyak collocation method to solve the Schrödinger equation by requiring that

$$\begin{aligned} (\hat{K} + V)I(D, H)\Psi_n(x_1, x_2, \dots, x_D) \\ = E_n I(D, H)\Psi_n(x_1, x_2, \dots, x_D) \end{aligned} \quad (32)$$

be satisfied on the Smolyak grid built from the nested points that are also used to make $I(D, H)$. To obtain, from this equation, an eigenvalue problem, we apply $\hat{K}$

and V and then evaluate both sides of the equation at $x_{k'_1}, x_{k'_2}, \dots, x_{k'_D}$. When evaluated at $x_{k'_1}, x_{k'_2}, \dots, x_{k'_D}$, the right hand side (RHS) is simply $E_n \Psi_n(x_{k'_1}, x_{k'_2}, \dots, x_{k'_D})$. This is due to the fact that the Smolyak interpolant is designed to be equal to $\Psi_n(x_{k'_1}, x_{k'_2}, \dots, x_{k'_D})$ at the interpolation point $x_{k'_1}, x_{k'_2}, \dots, x_{k'_D}$. For the same reason, applying V is simple. It is only applying $\hat{K}$ that is difficult. Simplifying the V term and the RHS, one obtains

$$\begin{aligned} & \hat{K}I(D, H)\Psi_n(x_1, x_2, \dots, x_D)|_{[x_1=x_{k'_1}, x_2=x_{k'_2}, \dots, x_D=x_{k'_D}]} \\ & + V(x_{k'_1}, x_{k'_2}, \dots, x_{k'_D})\Psi_n(x_{k'_1}, x_{k'_2}, \dots, x_{k'_D}) \\ & = E_n \Psi_n(x_{k'_1}, x_{k'_2}, \dots, x_{k'_D}). \end{aligned} \quad (33)$$

To obtain a matrix-eigenvalue problem from Eq. (33), we must write $\hat{K}I(D, H)\Psi_n(x_1, x_2, \dots, x_D)|_{[x_1=x_{k'_1}, x_2=x_{k'_2}, \dots, x_D=x_{k'_D}]}$ in terms of $\Psi_n(x_{k_1}, x_{k_2}, \dots, x_{k_D})$. In a previous paper, we presented equations that enable one to do this.¹⁴ In Ref. 14, we implicitly work with a grid representation of the KEO and a grid representation of the wavefunctions.

In this paper, to apply $\hat{K}$, we exploit the fact that the Smolyak interpolant can be written as in Eq. (15),

$$\begin{aligned} I(D, H)\Psi_n(x_1, x_2, \dots, x_D) \\ = \sum_{g(M^1(n_1+1), M^2(n_2+1), \dots, M^D(n_D+1)) \leq H} \tilde{\Gamma}_{n_1, n_2, \dots, n_D}^n \\ \times \tilde{\varphi}_{n_1}(x_1) \tilde{\varphi}_{n_2}(x_2) \cdots \tilde{\varphi}_{n_D}(x_D). \end{aligned} \quad (34)$$

$\tilde{\Gamma}_{n_1, n_2, \dots, n_D}^n$ in Eq. (15) represents the function $F(x_{k_1}, x_{k_2}, \dots, x_{k_D})$. $\tilde{\Gamma}_{n_1, n_2, \dots, n_D}^n$ in Eq. (34) represents the function $\Psi_n(x_1, x_2, \dots, x_D)$. It is straightforward to apply $\hat{K}$ to Eq. (34), but if we substitute the result into Eq. (33), we do not get a matrix eigenvalue problem because the first term does not depend on $\Psi_n(x_{k_1}, x_{k_2}, \dots, x_{k_D})$. To introduce $\Psi_n(x_{k_1}, x_{k_2}, \dots, x_{k_D})$, we replace

$$\begin{aligned} \tilde{\Gamma}_{n_1, n_2, \dots, n_D}^n &= \sum_{g(M^1(k_1)+M^2(k_2)+\dots+M^D(k_D)) \leq H} \\ &\times T_{(n_1, n_2, \dots, n_D), (k_1, k_2, \dots, k_D)} \\ &\times \Psi_n(x_{k_1}, x_{k_2}, \dots, x_{k_D}). \end{aligned} \quad (35)$$

$\mathbf{T}$ transforms a vector labelled by points to a vector labelled by function indices. The number of functions in the basis in Eq. (15) and the number of points on the nested Smolyak grid are equal. It is like a DVR to FBR (finite basis representation) transformation (but neither the $\tilde{\varphi}_{n_1}(x_1)\tilde{\varphi}_{n_2}(x_2)\cdots\tilde{\varphi}_{n_D}(x_D)$ basis nor the $k_1, k_2, \dots, k_D$ grid is a direct product). Now,

$$\begin{aligned} & \hat{K}I(D, H)\Psi_n(x_1, x_2, \dots, x_D)|_{[x_1=x_{k'_1}, x_2=x_{k'_2}, \dots, x_D=x_{k'_D}]} \\ &= \sum_{g(M^1(n_1+1)+M^2(n_2+1)+\dots+M^D(n_D+1)) \leq H} [\hat{K}\varphi_{n_1}(x_1)\varphi_{n_2}(x_2)\cdots\varphi_{n_D}(x_D)]|_{[x_1=x_{k'_1}, x_2=x_{k'_2}, \dots, x_D=x_{k'_D}]} \\ &\times \sum_{g(M^1(k_1)+M^2(k_2)+\dots+M^D(k_D)) \leq H} T_{(n_1, n_2, \dots, n_D), (k_1, k_2, \dots, k_D)} \Psi_n(x_{k_1}, x_{k_2}, \dots, x_{k_D}). \end{aligned} \quad (36)$$

Applying $\hat{K}$ in this fashion, Eq. (33) becomes

$$\begin{aligned} & \sum_{g(M^1(k_1)+M^2(k_2)+\dots+M^D(k_D)) \leq H} \left(\sum_{g(M^1(n_1+1)+M^2(n_2+1)+\dots+M^D(n_D+1)) \leq H} [\hat{K}\varphi_{n_1}(x_1)\varphi_{n_2}(x_2)\cdots\varphi_{n_D}(x_D)]|_{[x_1=x_{k'_1}, x_2=x_{k'_2}, \dots, x_D=x_{k'_D}]} \right. \\ & \left. \times T_{(n_1, n_2, \dots, n_D), (k_1, k_2, \dots, k_D)} + V(x_{k'_1}, x_{k'_2}, \dots, x_{k'_D})\delta_{k_1, k'_1}\delta_{k_2, k'_2}\cdots\delta_{k_D, k'_D} \right) \Psi_n(x_{k_1}, x_{k_2}, \dots, x_{k_D}) = E_n \Psi_n(x_{k'_1}, x_{k'_2}, \dots, x_{k'_D}), \end{aligned}$$

where we have pulled $\Psi_n(x_{k_1}, x_{k_2}, \dots, x_{k_D})$ to the right on the left hand side (LHS). Energies are eigenvalues of the matrix in brackets on the LHS. The eigenvalue problem is

$$[\mathbf{K} + \mathbf{V}]\mathbf{U} = \mathbf{E}\mathbf{U}, \quad (37)$$

where $\Psi_n(x_{k_1}, x_{k_2}, \dots, x_{k_D})$ is a column of $\mathbf{U}$.

Because the Smolyak grid will be large if $D \geq 6$, it is best to use an iterative eigensolver to compute energies. Iterative methods are frequently used because they make it possible to work with large basis sets.^{5,6,21,51–61} Because $\mathbf{K} + \mathbf{V}$ is not symmetric, the eigenvalue problem is solved with an Arnoldi method;⁶² we use ARPACK.⁶³ Matrix-vector products, $\mathbf{w} = (\mathbf{K} + \mathbf{V})\mathbf{z}$, must be evaluated. $\mathbf{w}$ and $\mathbf{z}$ have as many elements as there are points on the Smolyak grid (or basis functions in Eq. (15)). The matrix-vector product with $\mathbf{V}$ is trivial because it is diagonal. For the matrix-vector product with $\mathbf{K}$, we need

$$\begin{aligned} \tilde{\mathbf{C}}_{n_1, n_2, \dots, n_D} &= \sum_{g(M^1(k_1)+M^2(k_2)+\dots+M^D(k_D)) \leq H} \\ &\times T_{(n_1, n_2, \dots, n_D), (k_1, k_2, \dots, k_D)} \tilde{z}_{k_1, k_2, \dots, k_D}. \end{aligned} \quad (38)$$

To get it, we do not build $T_{(n_1, n_2, \dots, n_D), (k_1, k_2, \dots, k_D)}$, but instead transform from the grid to the basis by doing sums sequentially (see Section VI). Once $\tilde{\mathbf{C}}_{n_1, n_2, \dots, n_D}$ has been obtained, each term in the KEO must be applied separately to $\tilde{\varphi}_{n_1}(x_1)\tilde{\varphi}_{n_2}(x_2)\cdots\tilde{\varphi}_{n_D}(x_D)$ (see Eq. (36)). The part of the KEO that depends on differential operators can always be written,

$$\begin{aligned} \hat{K} &= \sum_{c=1}^D \mathcal{K}_1^c(x_1, x_2, \dots, x_D) \frac{\partial}{\partial x_c} \\ &+ \sum_{c=1}^D \sum_{c' \leq c=1}^D \mathcal{K}_2^{c, c'}(x_1, x_2, \dots, x_D) \frac{\partial^2}{\partial x_c \partial x_{c'}}. \end{aligned} \quad (39)$$

For example, for the term $K_{12} = \mathcal{K}_2^{1,2}(x_1, x_2, \dots, x_D)$ $\partial^2/\partial x_1 \partial x_2$, the matrix-vector product is

$$\begin{aligned} w^{1,2}(k'_1, k'_2, \dots, k'_D) &= \mathcal{K}_2^{1,2}(x_{k'_1}, x_{k'_2}, \dots, x_{k'_D}) \\ &\times \sum_{g(M^1(n_1+1), M^2(n_2+1), \dots, M^D(n_D+1)) \leq H} \left. \frac{\partial \tilde{\varphi}_{n_1}(x_1)}{\partial x_1} \right|_{x_1=x_{k'_1}} \\ &\times \left. \frac{\partial \tilde{\varphi}_{n_2}(x_2)}{\partial x_2} \right|_{x_2=x_{k'_2}} \tilde{\varphi}_{n_3}(x_{k'_3}) \cdots \tilde{\varphi}_{n_D}(x_{k'_D}) \tilde{C}_{n_1, n_2, \dots, n_D}. \end{aligned} \quad (40)$$

Derivatives of $\tilde{\varphi}_{n_c}(x_c)$ are straightforwardly obtained from derivatives of $\varphi_{n_c}(x_c)$. It is best to evaluate the sums in Eq. (40) sequentially.^{12,26,64} The vector $\mathbf{w} = (\mathbf{K} + \mathbf{V})\mathbf{z}$ is calculated by summing contributions from the potential and each of the terms in the KEO.

To evaluate the sums in Eq. (40) sequentially, we use ideas similar to those used to evaluate potential matrix-vector products with a variational method.¹² For example, for a 6D problem, one evaluates sequentially (the order is important) sums over $n_6, n_5, \dots, n_1$ to generate six intermediate vectors. The upper limit on all but one of the sums depends on values of other n_c indices. The allowed values of the indices of the intermediate vectors are determined by Eqs. (14) and (12). For the term of Eq. (40), the steps are outlined for a 6D case, in the supplementary material.⁷²

To apply one term in the KEO to $\tilde{C}_{n_1, n_2, n_3, n_4, n_5, n_6}$, one requires $D + 1$ nested loops. If neither the basis nor the grid were pruned the cost of evaluating, a matrix-vector product would be n^{D+1} . If $g(i_1, i_2, \dots, i_D) = i_1 + i_2 + \dots + i_D$ and $m_c(i_c)$, then the true cost, accounting for the pruning, can be estimated by using the fact that the number of functions (points) is equal to $\frac{H!}{D!(H-D)!}$. Of course, if more than $D + 1$ loops were required to obtain $\tilde{C}_{n_1, n_2, n_3, n_4, n_5, n_6}$, then the cost of transforming from a grid vector to a basis vector would dominate (see Section VI). The application of the KEO is parallelized over the terms in Eq. (39). The number of terms is the number of differential operators.

An important advantage of collocation is that the $\mathcal{K}_1^c$ and $\mathcal{K}_2^{c,c'}$ coefficients in Eq. (39) need only be evaluated at points. However, if the coefficients are simple enough that the $\varphi_{n_1}(x_1)\varphi_{n_2}(x_2)\cdots\varphi_{n_D}(x_D)$ representation of the KEO is sparse, then the matrix-vector product with $\mathbf{K}$ can be simplified. In that case, it is better to transform $\tilde{C}_{n_1, n_2, \dots, n_D}$ to $C_{n_1, n_2, \dots, n_D}$, then apply the sparse KEO matrix, and finally evaluate on the Smolyak grid.

VI. TRANSFORMING FROM A VECTOR LABELLED BY POINTS TO A VECTOR LABELLED BY BASIS INDICES

In Section V, we explain that to evaluate a matrix-vector product $\mathbf{Kz}$, it is advantageous to transform $\mathbf{z}$, which is labelled by points to $\tilde{C}_{n_1, n_2, \dots, n_D}$, a vector labelled by basis indices. See Eq. (38). In this section, we present ideas for obtaining $\tilde{C}_{n_1, n_2, \dots, n_D}$. The transformation is useful not just

for evaluating $\mathbf{Kz}$, but whenever one needs to evaluate a Smolyak interpolant at points that are not interpolation points. When the sum-of-products PES generated from a Smolyak interpolation⁴⁰ has so many terms that it is not efficient to exploit the SOP structure, it is better to use quadrature (or collocation) to solve the Schrödinger equation. It would be foolish to use the quadrature (collocation) grid dense enough to represent wavefunctions to interpolate the potential and it is therefore necessary, in this case, to evaluate the PES at points that are not interpolation points.⁶⁵ Although it is possible to evaluate Eq. (7) at any point, it is costly to evaluate it at many points because the summand has both i_c and k_c labels. It is less costly to evaluate Eqs. (11) and (15) than to evaluate Eq. (7). Eqs. (11) and (15) have the important advantage that they require only a sum over basis labels $(n_1, n_2, \dots, n_D)$.

In the two subsections of this section, we present two methods of obtaining $\tilde{C}_{n_1, n_2, \dots, n_D}$. Both rely on using nested points. If transforming from a vector labelled by points to a vector labelled by basis indices were costly, then collocation calculations would be more expensive than variational (quadrature) calculations. However, the scaling of the methods of this section and the scaling of the quadrature matrix-vector product of Ref. 12 are identical. This opens the door to exploiting the advantages of collocation reviewed in the introduction.

For the purpose of applying the KEO, there is no need to obtain $C_{n_1, n_2, \dots, n_D}$; it is enough to have $\tilde{C}_{n_1, n_2, \dots, n_D}$. However, there are situations in which one would like to be able to transform from the Smolyak grid to a pruned basis of one's choice, perhaps one that is not hierarchical. That is, it is sometimes desirable to have $C_{n_1, n_2, \dots, n_D}$ rather than $\tilde{C}_{n_1, n_2, \dots, n_D}$. For example, after solving the Schrödinger equation, we may wish to use the $\varphi_{n_1}(x_1)\varphi_{n_2}(x_2)\cdots\varphi_{n_D}(x_D)$ basis to assign wavefunctions. To do this, we need $C_{n_1, n_2, \dots, n_D}$. The $\tilde{\varphi}_{n_1}(x_1)\tilde{\varphi}_{n_2}(x_2)\cdots\tilde{\varphi}_{n_D}(x_D)$ basis makes it possible to efficiently calculate $\tilde{C}_{n_1, n_2, \dots, n_D}$, but unlike the $\varphi_{n_1}(x_1)\varphi_{n_2}(x_2)\cdots\varphi_{n_D}(x_D)$ basis, its functions do not have physical meaning. Fortunately, $C_{n_1, n_2, \dots, n_D}$ can be obtained easily from $\tilde{C}_{n_1, n_2, \dots, n_D}$ by doing sums sequentially:

$$\begin{aligned} C_{n_1, n_2, \dots, n_D} &= \sum_{n'_1=0}^{n_1^{\max}} \tilde{A}_{n_1, n'_1}^{-1} \sum_{n'_2=0}^{n_2^{\max}} \tilde{A}_{n_2, n'_2}^{-1} \cdots \\ &\times \sum_{n'_D=0}^{n_D^{\max}} \tilde{A}_{n_D, n'_D}^{-1} \tilde{C}_{n_1, n_2, \dots, n_D}, \end{aligned} \quad (41)$$

where $n_c^{\max} = m_c[H - M^{c+1}(1) \cdots - M^D(1) - M^1(n_1 + 1) - M^2(n_2 + 1) - \cdots - M^{c-1}(n_{c-1} + 1)] - 1$ and $\tilde{A}^{-1}$ is the inverse of the $\tilde{A}$ matrix in Eq. (17).

A. Computing $\tilde{C}_{n_1, n_2, \dots, n_D}$ separately

If the points are nested, it is straightforward to obtain an equation for $\tilde{C}_{n_1, n_2, \dots, n_D}$, from Eq. (10), by moving the sums over n_c to the left and comparing it to Eq. (11). $\tilde{C}_{n_1, n_2, \dots, n_D}$ is obtained by replacing $B_{k_c, n_c}^{i_c}$ with $\tilde{B}_{k_c, n_c}^{i_c}$, which is defined using $\tilde{\varphi}_{n_c}(x_c)$. One finds

$$\tilde{C}_{n_1, n_2, \dots, n_D} = \sum_{g(i_1, i_2, \dots, i_D) \leq H} \sum_{k_D=1}^{m_D(i_D)} [\tilde{B}_{k_D, n_D}^{i_D} - \tilde{B}_{k_D, n_D}^{i_D-1}] \times \sum_{k_{D-1}=1}^{m_{D-1}(i_{D-1})} [\tilde{B}_{k_{D-1}, n_{D-1}}^{i_{D-1}} - \tilde{B}_{k_{D-1}, n_{D-1}}^{i_{D-1}-1}] \cdots \sum_{k_2=1}^{m_2(i_2)} [\tilde{B}_{k_2, n_2}^{i_2} - \tilde{B}_{k_2, n_2}^{i_2-1}] \sum_{k_1=1}^{m_1(i_1)} [\tilde{B}_{k_1, n_1}^{i_1} - \tilde{B}_{k_1, n_1}^{i_1-1}] z_{k_1, k_2, \dots, k_D}. \quad (42)$$

$\tilde{\mathbf{B}}^{\mathbf{i}_c}$ in Eq. (42) is $(\tilde{\mathbf{P}}^{\mathbf{i}_c})^{-1}$ (see Eq. (22)). To account for the fact that for a given $\tilde{C}_{n_1, n_2, \dots, n_D}$, not all $i_1, i_2, \dots, i_D$ values that satisfy the constraint in Eq. (42) contribute, one can either set $\tilde{B}_{k_c, n_c}^{i_c}$ equal to zero if $k_c > m_c(i_c - 1)$ or $n_c > n_c^{\max}(i_c - 1)$ or impose in addition the constraint (for all c) $m_c(i_c) \geq n_c + 1$.

It is useful to store,

$$\Delta \tilde{B}_{k_c, n_c}^{i_c} = \tilde{B}_{k_c, n_c}^{i_c} - \tilde{B}_{k_c, n_c}^{i_c-1}. \quad (43)$$

The $\Delta \tilde{B}_{k_c, n_c}^{i_c}$ have two properties that make it easy to calculate the $\tilde{C}_{n_1, n_2, \dots, n_D}$. First,

$$\Delta \tilde{B}_{k_c, n_c}^{i_c} = 0 \quad \text{if } n_c = 0, 1, \dots, n_c^{\max}(i_c - 1) \quad (44)$$

and second,

$$\Delta \tilde{B}_{k_c, n_c}^{i_c} = 0 \quad \text{if } n_c > n_c^{\max}(i_c - 1) \text{ and } k_c > n_c + 1. \quad (45)$$

Where do these properties come from? The top left block of $(\tilde{\mathbf{B}}^{\mathbf{i}_c})^T$ whose size is $m_c(i_c - 1) \times m_c(i_c - 1)$ is identical to $(\tilde{\mathbf{B}}^{\mathbf{i}_c-1})^T$. This is due to the fact that the top right $m_c(i_c - 1) \times (m_c(i_c) - m_c(i_c - 1))$ block of $(\tilde{\mathbf{P}}^{\mathbf{i}_c})^T$ is zero. This establishes Eq. (44). According to Eq. (45), satisfied by the hierarchical functions, we use, but not by those used by other authors, the bottom right corner of $(\tilde{\mathbf{B}}^{\mathbf{i}_c})^T$ is lower triangular. This is due to Eq. (18).

One way to obtain $\tilde{C}_{n_1, \dots, n_D}$ is to start with Eq. (42) and evaluate sums sequentially. As an example, consider a case with $D = 3$ and with $m_c(i_c) = i_c$ and calculate $\tilde{C}_{4,1,9}$. The only non-zero term in the sum over the i_c indices in Eq. (42) has $i_1 = 5, i_2 = 2, i_3 = 10$. $\tilde{C}_{4,1,9}$ is obtained in three steps. First, one transforms $k_1 \rightarrow n_1 = 4$ by calculating the array

$$w^1(4, k_2, k_3) = \sum_{k_1=1}^5 \Delta \tilde{B}_{k_1, 4}^{i_1} z_{k_1, k_2, k_3} \quad (46)$$

with $1 \leq k_2 \leq 2$ and $1 \leq k_3 \leq 10$. Second, one transforms $k_2 \rightarrow n_2 = 1$ by calculating the array

$$w^2(4, 1, k_3) = \sum_{k_2=1}^2 \Delta \tilde{B}_{k_2, 1}^{i_2} w^1(n_1 = 4, k_2, k_3) \quad (47)$$

with $1 \leq k_3 \leq 10$. Third, one transforms $k_3 \rightarrow n_3 = 9$ and calculates

$$\tilde{C}_{4,1,9} = \sum_{k_3=1}^{10} \Delta \tilde{B}_{k_3, 9}^{i_3} w^2(n_1 = 4, n_2 = 1, k_3). \quad (48)$$

Parallelization of this calculation of $\tilde{C}_{n_1, n_2, \dots, n_D}$ is simple. Because only one term in Eq. (42) contributes to a given $\tilde{C}_{n_1, n_2, \dots, n_D}$, different $\tilde{C}_{n_1, n_2, \dots, n_D}$ can be computed in different threads with no communication. When there are many terms in Eq. (11), it is possible to take full advantage of a large number of processors. However, because we parallelize over only 13 threads, in our calculations, the method of Subsection VI B is faster. The first step is the most costly. If $k_c^{\max}$ were n , for all c , then for the D coordinate case, its cost would be n^D . The steps must be repeated for each $\tilde{C}_{n_1, n_2, \dots, n_D}$. In actuality, $k_c^{\max}$ is different for each $\tilde{C}_{n_1, n_2, \dots, n_D}$. In practice, the cost of obtaining $\tilde{C}_{n_1, n_2, \dots, n_D}$, with the method of this subsection, and the cost of evaluating the matrix-vector product for one term in the KEO (Section V) are comparable.

When $m_c(i_c) - m_c(i_c - 1) > 1$, obtaining $\tilde{C}_{n_1, n_2, \dots, n_D}$ is not as easy. Although it is true also in this case that only one term on the RHS of Eq. (42) is nonzero, now one term in the sum over $g(i_1, i_2, \dots, i_D)$ contributes to several $\tilde{C}_{n_1, n_2, \dots, n_D}$. In general, a term with $(i_1, i_2, \dots, i_D)$ will contribute to all $\tilde{C}_{n_1, n_2, \dots, n_D}$ for which $n_c^{\max}(i_c - 1) < n_c \leq n_c^{\max}(i_c)$. Note that two terms do not contribute to the same $\tilde{C}_{n_1, n_2, \dots, n_D}$. For example, if $m_c(i_c) = 2i_c - 1$, then the term on the RHS of the 3D version of Eq. (42) with $i_1 = 5, i_2 = 2$, and $i_3 = 10$ contributes to $\tilde{C}_{n_1, n_2, n_3}$ for which $n_1^{\max}(4) < n_1 \leq n_1^{\max}(5), n_2^{\max}(1) < n_2 \leq n_2^{\max}(2), n_3^{\max}(9) < n_3 \leq n_3^{\max}(10)$, i.e., to $\tilde{C}_{7,1,17}, \tilde{C}_{7,1,18}, \tilde{C}_{7,2,17}, \tilde{C}_{7,2,18}, \tilde{C}_{8,1,17}, \tilde{C}_{8,1,18}, \tilde{C}_{8,2,17}$, and $\tilde{C}_{8,2,18}$. These coefficients are calculated sequentially in three steps as in the example above. Parallelization is efficient also in this case.

B. Computing $\tilde{C}_{n_1, n_2, \dots, n_D}$ simultaneously

We shall now present an alternative method of computing $\tilde{C}_{n_1, n_2, \dots, n_D}$. The method of the Section VI A is simple, but it has the disadvantage that identical intermediate vectors used to obtain different $\tilde{C}_{n_1, n_2, \dots, n_D}$ are calculated more than once. Consider the example of Eqs. (46)–(48) at the end of Section VI A. To compute $\tilde{C}_{4,1,9}$, we need $w^2(4, 1, k_3)$ for all possible k_3 . To compute $\tilde{C}_{4,1,10}$, we need $w^2(4, 1, k_3)$ for some of the same k_3 . The same idea applies to $w^1(4, k_2, k_3)$. One could store intermediate vectors to avoid recomputing them, but this would require communication when the method of Section VI A is parallelized. The method of this subsection obviates the need to recompute elements of vectors.

The ideas are outlined here for the case $g(i_1, i_2, \dots, i_D) = i_1 + i_2 + \dots + i_D \leq H$. Eq. (42) can be re-written by applying the constraint directly to the grid labels,

$$\begin{aligned}
\tilde{C}_{n_1, n_2, \dots, n_{D-1}, n_D} &= \sum_{k_D=1}^{k_D^{\max}} \Delta \tilde{B}_{k_D, n_D}^{M^D(n_D+1)} \sum_{k_{D-1}=1}^{k_{D-1}^{\max}} \Delta \tilde{B}_{k_{D-1}, n_{D-1}}^{M^{D-1}(n_{D-1}+1)} \dots \sum_{k_2=1}^{k_2^{\max}} \Delta \tilde{B}_{k_2, n_2}^{M^2(n_2+1)} \sum_{k_1=1}^{k_1^{\max}} \Delta \tilde{B}_{k_1, n_1}^{M^1(n_1+1)} z_{k_1, k_2, k_3, k_4, k_5, k_6}, \\
k_D^{\max} &= \text{MIN}(m_D[H - M^1(1) - M^2(1) - \dots - M^{D-1}(1)], n_D + 1), \\
k_{D-1}^{\max} &= \text{MIN}(m_{D-1}[H - M^1(1) - M^2(1) - \dots - M^{D-2}(1) - M^D(k_D)], n_{D-1} + 1), \\
&\vdots \\
k_2^{\max} &= \text{MIN}(m_2[H - M^1(1) - M^3(k_3) - \dots - M^{D-1}(k_{D-1}) - M^D(k_D)], n_2 + 1), \\
k_1^{\max} &= \text{MIN}(m_1[H - M^2(k_2) - M^3(k_3) - \dots - M^{D-1}(k_{D-1}) - M^D(k_D)], n_1 + 1).
\end{aligned} \tag{49}$$

There are no sums on i_c because, when n_c is fixed, it is only when $i_c = M^c(n_c + 1)$ that the summand is nonzero. This is due to property 1 (Eq. (44)). If $m_c(i_c + 1) - m_c(i_c) > 1$ and one uses standard hierarchical functions, rather than those defined in Eq. (17), the second argument in the upper limits $k_c^{\max}$ is $m_c(M^c(n_c + 1))$.

In a six dimensional case, Eq. (49) is implemented in 6 steps that generate five intermediate vectors and finally $\tilde{C}_{n_1, n_2, n_3, n_4, n_5, n_6}$. The possible values of the indices labelling the vectors are limited by Eqs. (12) and (14). The upper limit of the sum over k_c is $\text{MIN}(a_c, b_c)$ where $b_c = n_c + 1$ and a_c is determined from Eq. (14) and $m_c(i_c)$. A 6D example is given in the supplementary material.⁷²

Note that in each step, the method of this subsection requires a single sum and every vector has D indices. Each step therefore requires $D + 1$ loops. If neither the basis nor the grid were pruned the cost of evaluating a matrix-vector product would be n^{D+1} . If $g(i_1, i_2, \dots, i_D) = i_1 + i_2 + \dots + i_D$ and $m_c(i_c) = i_c$ then the true cost, accounting for the pruning, can be estimated by using the fact that the number of functions (points) is equal to $\frac{H!}{D!(H-D)!}$. Thus, the cost of this method for obtaining $\tilde{C}_{n_1, n_2, n_3, n_4, n_5, n_6}$ is also comparable to the cost of a matrix-vector product for one term in the KEO (Section IV).

VII. COMPUTING THE SPECTRUM

We have used the ideas of Section V to compute vibrational energy levels of HONO with the PES of Ref. 50. Both methods (Section VI) of obtaining a vector labelled by basis indices from a vector labelled by point indices have been tested in order to apply the KEO. We use bond coordinates. Because curvilinear coordinates are better suited to the description of large amplitude motion, it is important to demonstrate that Smolyak collocation works not only in normal coordinates but also in curvilinear coordinates. An approach that works only with normal coordinates is only applicable to molecules with one minimum and for which vibration is of small enough amplitude that coupling is not too large. A well-known disadvantage of curvilinear coordinates is the associated complicated KEO. When a variational method is used, complicated coordinate dependence of the coefficients of the KEO makes computing matrix elements costly when it is necessary to evaluate integrals for terms in the KEO by quadrature. When collocation is used, it is only necessary to

evaluate the coefficients at points: no integrals are required. In Ref. 14, it has been shown that this advantage of collocation can be exploited, without solving a generalized eigenvalue problem, which was heretofore required to use a collocation method; however, in that paper, normal coordinates were used.

The coordinates we use for HONO are r_2 for the outer NO bond, r_3 for the inner ON bond, r_1 for the OH bond, θ_2 for the ONO angle, θ_1 for the HON angle, and τ for the dihedral angle, $0^\circ \leq \tau \leq 360^\circ$.⁶⁶ These coordinates are less strongly coupled than Jacobi coordinates. Although even better coordinates can certainly be found, the coordinates we use do make it possible to show that our internal coordinate collocation method works. The KEO is written, not in terms of the bending coordinates θ_i , $i = 1, 2$, but instead, as did Ritcher *et al.*, in terms of $\xi_i = \cos(\theta_i)$, $i = 1, 2$ since this simplifies the KEO. The full Hamiltonian has the form,

$$\begin{aligned}
\hat{H} &= \sum_{i=1}^6 \sum_{j \leq i}^6 G^{i,j}(r_1, r_2, r_3, r_4, r_5, r_6) \frac{\partial^2}{\partial r_i \partial r_j} \\
&+ \sum_{i=1}^6 G^i(r_1, r_2, r_3, r_4, r_5, r_6) \frac{\partial}{\partial r_i} \\
&+ V^{ps}(r_1, r_2, r_3, r_4, r_5, r_6) + V(r_1, r_2, r_3, r_4, r_5, r_6)
\end{aligned} \tag{50}$$

with $r_4, r_5, r_6 = \xi_1, \xi_2, \tau$. V^{ps} is the pseudo-potential term obtained by changing the volume element.⁶⁶

The 1D $\varphi_{n_c}(x_c)$ basis functions are eigenfunctions of 1D Hamiltonians, $\hat{H}_c^0$, $c = 1, \dots, 6$. For r_1 , we use

$$\hat{H}_1^0 = -\frac{m_H + m_O}{2m_H m_O} \frac{\partial^2}{\partial r_1^2} + V^1(r_1), \tag{51}$$

where

$$V^1(r_1) = V(r_1, r_2^{eq}, r_3^{eq}, \xi_1^{eq}, \xi_2^{eq}, \tau = 180^\circ). \tag{52}$$

In this section, *eq* refer to *cis* equilibrium values: $r_1^{eq} = 1.823$ bohrs, $r_2^{eq} = 2.213$ bohrs, $r_3^{eq} = 2.697$ bohrs, $\xi_1^{eq} = \cos(110.7^\circ)$, $\xi_2^{eq} = \cos(101.9^\circ)$.⁵⁰ Although the 1D potential we use is evaluated at *cis* values, it enables us to compute *cis*-like and *trans*-like wavefunctions. For r_2 , we use

$$\hat{H}_2^0 = -\frac{m_N + m_O}{2m_N m_O} \frac{\partial^2}{\partial r_2^2} + V^2(r_2), \tag{53}$$

where

$$V^2(r_2) = V(r_1^{eq}, r_2, r_3^{eq}, \xi_1^{eq}, \xi_2^{eq}, \tau = 180^\circ). \tag{54}$$

For r_3 , we use

$$\hat{H}_3^0 = -\frac{m_N + m_O}{2m_N m_O} \frac{\partial^2}{\partial r_3^2} + V^3(r_3), \quad (55)$$

where

$$V^3(r_3) = V(r_1^{eq}, r_2^{eq}, r_3, \xi_1^{eq}, \xi_2^{eq}, \tau = 180^\circ). \quad (56)$$

$\hat{H}_4^0$ for ξ_1 is a sum of a 1D potential and the Hermitian terms in the full KEO with derivatives with respect to ξ_1 ,

$$\begin{aligned} \hat{H}_4^0 = & - \left[\left(\frac{1}{m_H} + \frac{1}{m_O} \right) \frac{1}{2(r_1^{eq})^2} + \left(\frac{1}{m_O} + \frac{1}{m_N} \right) \frac{1}{2(r_3^{eq})^2} \right] \\ & \times \frac{\partial}{\partial \xi_1} (1 - \xi_1^2) \frac{\partial}{\partial \xi_1} \\ & + \frac{1}{m_O r_3^{eq} r_1^{eq}} \frac{\partial}{\partial \xi_1} (1 - \xi_1^2) \xi_1 \frac{\partial}{\partial \xi_1} + V^4(\xi_1), \end{aligned} \quad (57)$$

where

$$V^4(\xi_1) = V(r_1^{eq}, r_2^{eq}, r_3^{eq}, \xi_1, \xi_2^{eq}, \tau = 180^\circ). \quad (58)$$

$\hat{H}_5^0$ is similar,

$$\begin{aligned} \hat{H}_5^0 = & - \left[\left(\frac{1}{m_O} + \frac{1}{m_N} \right) \frac{1}{2(r_2^{eq})^2} + \left(\frac{1}{m_O} + \frac{1}{m_N} \right) \frac{1}{2(r_3^{eq})^2} \right] \\ & \times \frac{\partial}{\partial \xi_2} (1 - \xi_2^2) \frac{\partial}{\partial \xi_2} \\ & + \frac{1}{m_N r_3^{eq} r_2^{eq}} \frac{\partial}{\partial \xi_2} (1 - \xi_2^2) \xi_2 \frac{\partial}{\partial \xi_2} + V^5(\xi_2), \end{aligned} \quad (59)$$

where

$$V^5(\xi_2) = V(r_1^{eq}, r_2^{eq}, r_3^{eq}, \xi_1^{eq}, \xi_2, \tau = 180^\circ). \quad (60)$$

The KEO for $\hat{H}_6^0$ (for τ) is a sum of the Hermitian terms in the full KEO with two derivatives with respect to τ and

$$\begin{aligned} \hat{H}_6^0 = & - \left[\left(\frac{1}{m_H} + \frac{1}{m_O} \right) \frac{1}{2(r_1^{eq})^2 (1 - (\xi_1^{eq})^2)} + \left(\frac{1}{m_O} + \frac{1}{m_N} \right) \frac{(\xi_1^{eq})^2}{2(r_3^{eq})^2 (1 - (\xi_1^{eq})^2)} \right. \\ & - \left(\frac{1}{m_O} \right) \frac{\xi_1^{eq}}{r_3^{eq} r_1^{eq} (1 - (\xi_1^{eq})^2)} \left(\frac{1}{m_O} + \frac{1}{m_N} \right) \frac{1}{2(r_2^{eq})^2 (1 - (\xi_2^{eq})^2)} \\ & + \left(\frac{1}{m_O} + \frac{1}{m_N} \right) \frac{(\xi_2^{eq})^2}{2(r_3^{eq})^2 (1 - (\xi_2^{eq})^2)} + \left(-\frac{1}{m_N} \right) \frac{\xi_2^{eq}}{r_3^{eq} r_2^{eq} (1 - (\xi_2^{eq})^2)} \left. \right] \frac{\partial^2}{\partial \tau^2} \\ & + \left[\left(\frac{1}{m_O} \right) \frac{\xi_2^{eq}}{r_3^{eq} r_1^{eq} \sqrt{1 - \xi_1^{eq}} \sqrt{1 - \xi_2^{eq}}} - \left(-\frac{1}{m_N} \right) \frac{\xi_1^{eq}}{r_3^{eq} r_2^{eq} \sqrt{1 - \xi_1^{eq}} \sqrt{1 - \xi_2^{eq}}} \right. \\ & \left. - \left(\frac{1}{m_O} + \frac{1}{m_N} \right) \frac{\xi_1^{eq} \xi_2^{eq}}{(r_3^{eq})^2 \sqrt{1 - \xi_1^{eq}} \sqrt{1 - \xi_2^{eq}}} \right] \frac{\partial}{\partial \tau} \cos(\tau) \frac{\partial}{\partial \tau} + V^6(\tau). \end{aligned} \quad (61)$$

$V^6(\tau)$ is the potential obtained by finding values of the other coordinates that minimize the potential for each value of τ from $\tau = 0^\circ$ to $\tau = 180^\circ$. Minimized potentials of this type would probably be better for the other coordinates as well, but holes in the PES made using such minimized cut potentials impossible.

The 1D eigenfunctions are computed by diagonalizing large matrix representations of the 1D Hamiltonians. Matrix elements are calculated with Gauss quadrature. For the r_1 , r_2 , and r_3 coordinates, tridiagonal Morse basis sets^{67–69} and large (1D) Gauss quadratures were used. For the ξ_1 and ξ_2 coordinates, Jacobi functions^{70,71} with a and b parameters larger than unity were used, and for τ , a Fourier basis was used. The 1D eigenfunctions are sorted by energy and used to make $\tilde{\varphi}_{n_c}(x_c)$ and Lagrange type functions $\tilde{a}_k(x_c)$.

As explained earlier, we use, $g(i_1, i_2, \dots, i_D) = i_1 + i_2 + \dots + i_D$, and control the quality of the interpolation by

adjusting $m_c(i_c)$ and H . With $g(i_1, i_2, \dots, i_D) = i_1 + i_2 + \dots + i_D$, we can use Eq. (49) without modification. The point to basis index transformation can also be done for other $g(i_1, i_2, \dots, i_D)$, but the equation that replaces Eq. (49) is more complicated. The coordinates r_1 and r_2 are not strongly coupled to the other coordinates and have large zeroth-order energies and therefore require rather few basis functions. The coordinates r_3 , ξ_1 , and ξ_2 are strongly coupled among themselves and therefore we need to use many basis functions for these three coordinates. The 1D τ energies are closely spaced and low. The low-lying 1D torsional states are localized in the *trans* or the *cis* well, whereas the high-lying excited torsional vibrational states are de-localized. Therefore, a large number of 1D τ functions are required. We have done calculations with $H = 29, 28, 27, 26$ (and maximum levels, $i_c^{max} = 24, 23, 22, 21$) and $m_1(i) = m_2(i) = i$ for r_1 and r_2 , and

$$\begin{aligned} m_c(1) &= 1, m_c(2) = 4, m_c(3) = 7, m_c(4) = 10, m_c(5) = 12, m_c(6) = 14, m_c(7) = 16, m_c(8) = 18, \\ m_c(9) &= 20, m_c(10) = 21, m_c(11) = 22, m_c(12) = 23, m_c(13) = 24, m_c(14) = 25, m_c(15) = 26, \\ m_c(16) &= 27, m_c(17) = 28, m_c(18) = 29, m_c(19) = 30, m_c(20) = 31, m_c(n_c \geq 21) = 32, \end{aligned} \quad (62)$$

TABLE I. Vibrational levels of HONO for which normal mode assignments are possible. The 20 lowest and the 20 highest assignable levels for each well are shown. A full set of levels is given in the supplementary material.⁷² The corresponding wavefunctions are mostly localized in either the *cis* or the *trans* well. $\Delta E^{H=28} = E^{H=29} - E^{H=28}$, $\Delta E^{H=27} = E^{H=29} - E^{H=27}$, $\Delta E^{H=26} = E^{H=29} - E^{H=26}$.

Well	Assignment	$E^{H=29} - ZPE^{H=29}$	$E^{H=29}$	$\Delta E^{H=28}$	$\Delta E^{H=27}$	$\Delta E^{H=26}$
<i>trans</i>	(0, 0, 0, 0, 0, 0)	0.00	4366.89	0.00	0.00	0.01
<i>trans</i>	(0, 0, 0, 0, 0, 1)	537.72	4904.61	0.00	0.00	0.01
<i>trans</i>	(0, 0, 0, 0, 1, 0)	600.77	4967.67	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 1, 0, 0)	795.72	5162.61	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 0, 0, 2)	1055.35	5422.25	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 0, 1, 1)	1131.32	5498.21	0.00	0.01	0.00
<i>trans</i>	(0, 0, 0, 0, 2, 0)	1187.87	5554.76	0.00	0.01	0.01
<i>trans</i>	(0, 0, 1, 0, 0, 0)	1267.27	5634.16	0.00	0.00	0.01
<i>trans</i>	(0, 0, 0, 1, 0, 1)	1323.07	5689.96	0.00	0.01	0.00
<i>trans</i>	(0, 0, 0, 1, 1, 0)	1384.86	5751.76	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 0, 0, 3)	1551.60	5918.49	0.00	0.01	0.00
<i>trans</i>	(0, 0, 0, 2, 0, 0)	1574.26	5941.15	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 0, 1, 2)	1640.75	6007.64	0.00	0.01	-0.00
<i>trans</i>	(0, 1, 0, 0, 0, 0)	1689.56	6056.45	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 0, 2, 1)	1710.20	6077.10	0.00	0.01	0.00
<i>trans</i>	(0, 0, 0, 0, 3, 0)	1761.27	6128.16	0.00	0.01	0.01
<i>trans</i>	(0, 0, 1, 0, 0, 1)	1797.63	6164.52	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 1, 0, 2)	1828.72	6195.61	0.00	0.00	0.00
<i>trans</i>	(0, 0, 1, 0, 1, 0)	1857.86	6224.76	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 1, 1, 1)	1904.56	6271.45	0.00	0.01	0.00
<i>trans</i>	(1, 0, 0, 0, 0, 0)	3589.49	7956.39	-0.00	-0.01	-0.02
<i>trans</i>	(0, 0, 2, 1, 1, 0)	3849.71	8216.61	0.00	0.01	0.01
<i>trans</i>	(0, 0, 0, 2, 0, 5)	3856.01	8222.91	-0.06	0.03	-0.14
<i>trans</i>	(0, 2, 0, 0, 0, 1)	3872.02	8238.92	0.02	0.01	0.02
<i>trans</i>	(0, 0, 1, 2, 1, 1)	3879.58	8246.48	-0.00	-0.00	-0.03
<i>trans</i>	(0, 1, 0, 0, 1, 4)	3881.61	8248.50	-0.01	0.00	-0.07
<i>trans</i>	(0, 0, 0, 3, 1, 2)	3882.58	8249.47	-0.03	-0.03	-0.10
<i>trans</i>	(0, 1, 0, 0, 2, 2)	3892.55	8259.44	-0.08	-0.04	-0.27
<i>trans</i>	(0, 0, 1, 0, 2, 3)	3897.02	8263.92	-0.01	-0.02	-0.02
<i>trans</i>	(0, 0, 0, 1, 1, 9)	3919.94	8286.83	-0.03	-0.05	-0.17
<i>trans</i>	(0, 0, 0, 0, 7, 0)	3922.75	8289.64	0.05	0.03	-0.13
<i>trans</i>	(0, 0, 0, 1, 2, 6)	3929.64	8296.53	-0.01	-0.08	-0.11
<i>trans</i>	(0, 0, 1, 2, 2, 0)	3935.31	8302.21	0.01	0.02	0.01
<i>trans</i>	(0, 1, 1, 0, 0, 2)	3949.23	8316.12	-0.00	-0.01	-0.03
<i>trans</i>	(0, 2, 0, 0, 1, 0)	3949.50	8316.39	0.01	0.01	0.01
<i>trans</i>	(0, 0, 0, 2, 3, 1)	3959.23	8326.12	-0.01	-0.02	-0.01
<i>trans</i>	(0, 1, 0, 0, 3, 1)	3960.38	8327.27	0.01	0.01	-0.00
<i>trans</i>	(0, 1, 0, 1, 0, 3)	3973.27	8340.16	0.00	0.00	-0.01
<i>trans</i>	(0, 1, 1, 0, 0, 2)	3976.77	8343.67	0.00	-0.02	-0.02
<i>trans</i>	(0, 0, 0, 0, 2, 6)	3990.61	8357.50	-0.13	-0.04	-0.30
<i>trans</i>	(0, 0, 2, 0, 0, 3)	4005.14	8372.04	0.01	0.00	0.00
<i>cis</i>	(0, 0, 0, 0, 0, 0)	0.00	4460.75	-0.00	-0.01	-0.01
<i>cis</i>	(0, 0, 0, 0, 1, 0)	616.57	5077.32	0.00	0.00	-0.09
<i>cis</i>	(0, 0, 0, 0, 0, 1)	631.80	5092.55	-0.00	-0.01	-0.01
<i>cis</i>	(0, 0, 0, 1, 0, 0)	849.84	5310.59	-0.01	-0.02	-0.06
<i>cis</i>	(0, 0, 0, 0, 0, 2)	1212.54	5673.29	-0.00	-0.01	-0.08
<i>cis</i>	(0, 0, 0, 0, 2, 0)	1218.60	5679.35	0.00	-0.03	-0.10
<i>cis</i>	(0, 0, 0, 0, 1, 1)	1238.04	5698.79	0.00	0.01	-0.08
<i>cis</i>	(0, 0, 1, 0, 0, 0)	1311.20	5771.95	-0.00	-0.00	-0.08
<i>cis</i>	(0, 0, 0, 1, 1, 0)	1453.08	5913.83	-0.01	-0.03	-0.10
<i>cis</i>	(0, 0, 0, 1, 0, 1)	1460.98	5921.73	-0.00	-0.02	-0.07
<i>cis</i>	(0, 1, 0, 0, 0, 0)	1631.56	6092.31	-0.01	-0.01	-0.08
<i>cis</i>	(0, 0, 0, 2, 0, 0)	1684.81	6145.56	-0.01	-0.03	-0.08
<i>cis</i>	(0, 0, 0, 0, 0, 3)	1767.17	6227.92	-0.00	-0.00	-0.08
<i>cis</i>	(0, 0, 0, 0, 3, 0)	1803.34	6264.09	0.00	-0.10	-0.09
<i>cis</i>	(0, 0, 0, 0, 1, 2)	1808.70	6269.45	-0.01	-0.08	-0.08
<i>cis</i>	(0, 0, 0, 0, 2, 1)	1827.41	6288.16	0.00	-0.03	-0.09

TABLE I. (Continued.)

Well	Assignment	$E^{H=29} - ZPE^{H=29}$	$E^{H=29}$	$\Delta E^{H=28}$	$\Delta E^{H=27}$	$\Delta E^{H=26}$
<i>cis</i>	(0, 0, 1, 0, 1, 0)	1907.90	6368.65	0.00	−0.07	−0.08
<i>cis</i>	(0, 0, 1, 0, 0, 1)	1944.00	6404.75	0.00	0.00	−0.08
<i>cis</i>	(0, 0, 0, 1, 0, 2)	2025.69	6486.44	−0.01	−0.06	−0.08
<i>cis</i>	(0, 0, 0, 1, 2, 0)	2041.84	6502.58	−0.00	−0.09	−0.09
<i>cis</i>	(1, 0, 0, 0, 0, 0)	3435.02	7895.77	0.00	0.02	0.03
<i>cis</i>	(0, 0, 1, 1, 2, 3)	3872.27	8333.01	−0.03	−0.05	−0.12
<i>cis</i>	(0, 0, 3, 0, 0, 0)	3878.38	8339.13	−0.01	−0.05	−0.06
<i>cis</i>	(0, 0, 0, 0, 2, 5)	3887.16	8347.91	−0.09	−0.18	−0.30
<i>cis</i>	(0, 1, 0, 0, 0, 4)	3896.46	8357.21	−0.04	−0.11	−0.06
<i>cis</i>	(0, 1, 0, 2, 0, 1)	3901.15	8361.90	−0.02	−0.09	−0.12
<i>cis</i>	(0, 0, 1, 1, 1, 2)	3902.59	8363.34	−0.07	−0.17	−0.17
<i>cis</i>	(0, 0, 0, 2, 0, 4)	3915.61	8376.36	−0.07	−0.34	−0.36
<i>cis</i>	(0, 0, 0, 1, 2, 5)	3923.59	8384.34	−0.11	−0.10	−0.38
<i>cis</i>	(0, 0, 0, 2, 1, 4)	3924.19	8384.94	−0.03	−0.09	−0.10
<i>cis</i>	(0, 0, 2, 1, 1, 0)	3979.95	8440.70	−0.00	−0.07	−0.09
<i>cis</i>	(0, 1, 0, 0, 4, 0)	4000.15	8460.90	−0.12	−0.04	−0.09
<i>cis</i>	(0, 1, 0, 0, 2, 2)	4004.73	8465.48	−0.06	−0.02	−0.13
<i>cis</i>	(0, 0, 2, 1, 0, 1)	4017.18	8477.92	−0.07	−0.48	−0.65
<i>cis</i>	(0, 1, 0, 0, 4, 1)	4021.11	8481.86	−0.09	−0.11	−0.09
<i>cis</i>	(0, 0, 1, 1, 0, 5)	4031.60	8492.35	−0.25	−0.32	−0.65
<i>cis</i>	(0, 0, 1, 0, 1, 4)	4045.45	8506.20	−0.25	−0.18	−0.22
<i>cis</i>	(1, 0, 0, 0, 1, 0)	4046.04	8506.79	−0.14	−0.07	−0.41
<i>cis</i>	(0, 1, 0, 3, 0, 0)	4064.19	8524.94	−0.05	−0.07	−0.10
<i>cis</i>	(0, 0, 1, 2, 2, 0)	4084.47	8545.22	−0.07	−0.05	−0.12
<i>cis</i>	(0, 0, 0, 5, 0, 0)	4151.14	8611.89	−0.11	−0.19	−0.35

for r_3 , ξ_1 and ξ_2 , and

$$\begin{aligned} m_6(1) = 1, m_6(2) = 5, m_6(3) = 9, m_6(4) = 13, m_6(5) = 17, m_6(6) = 21, m_6(7) = 25, m_6(8) = 27, \\ m_6(9) = 29, m_6(10) = 31, m_6(11) = 33, m_6(12) = 35, m_6(13) = 37, m_6(14) = 39, m_6(15) = 41, \\ m_6(16) = 43, m_6(17) = 45, m_6(18) = 47, m_6(19) = 49, m_6(20) = 51, m_6(i_c \geq 21) = 53, \end{aligned}$$

(63)

for τ .

For τ , the pseudo-Gauss procedure is used to determine pairs of new points. When $m_6(i + 1) - m_6(i) = 4$, the points added to those of level i are two pairs. Each pair is composed of a point τ and its partner $2\pi - \tau$. Let N be the number of points in the set being extended (when $m_6(i) - m_6(i - 1) = 2$, then $N = m_6(i - 1)$). The size of the overlap matrix used to determine (see Ref. 12) the two new points is $N + 2$. One of the new functions is symmetric and the other is antisymmetric with respect to reflection with respect to $\tau = 180^\circ$. Only one point position is optimized. The partner point is obtained by symmetry. When one or two pairs of points are added to the points of level i to obtain the points of level $i + 1$, $\tilde{A}$ in Eq. (17) is sometimes nearly singular. This occurs when rows of $\tilde{A}$ are linearly dependent because when $\phi_k(\tau)$ have amplitude in both wells it sometimes happens that $\phi_k(\tau) = \phi_k(2\pi - \tau)$. To avoid the singularity, we switch the pseudo-Gauss point responsible for the singularity with one in an interpolation rule $\tilde{U}^{i'}$ for which $i' > i + 1$. Owing to the switch, the points we use at each level are not the pseudo-Gauss points we think would be best; however, we have verified that the switch affects energy levels very little.

At least 400 levels are converged to a fraction of a cm^{-1} . The first 20 and last 20 assigned levels, for each well,

are reported in Table I. The complete set is given in the supplementary material.⁷² In the tables, we compare levels computed with $H = 29$ (total number of basis functions is 4 229 195) with those computed with $H = 28$ and $H = 27$ (basis functions 3 434 211 and 2 761 114). The average error is $2.5 \times 10^{-2} \text{ cm}^{-1}$ and the maximum error is 0.39 cm^{-1} . Average and maximum errors are defined with respect to the $H = 29$ calculation. Note that energy levels do not necessarily decrease when the basis and grid size are increased. Our energies agree rather well with those of Ref. 50, but our levels are systematically lower.

As a function of H energy levels oscillate slightly. Small amplitude oscillations persist even for large H , due to holes in the PES. If it is possible to identify regions in which holes occur, the PES can be modified so that they do not cause problems.

The assignments in Table I are in terms of rectilinear normal coordinate labels. The normal coordinates for *trans*-HONO and *cis*-HONO are: v_1 is OH stretching, v_2 is NO stretching, v_3 is NOH bending, v_4 is ONO bending, v_5 is ON stretching, and v_6 is torsional mode. Only assigned levels are included in the table; we have calculated more. Assignment becomes meaningless at higher energies. Levels are assigned

by fitting each of the wavefunctions with a sum of products of 1D harmonic oscillator functions in normal coordinates. For a given wavefunction, between 100 and 1000 coefficients are chosen to nearly reproduce the largest wavefunction values evaluated at between 10 000 and 50 000 points on the Smolyak grid used in collocation. This is done by solving the normal least squares equations. This requires transforming the internal coordinates to normal coordinates.

VIII. CONCLUSION

The variational method, in which wavefunctions are represented as linear combinations of basis functions,

$$\Psi_n(x_1, x_2, \dots, x_D) = \sum_k F_k b_k(x_1, x_2, \dots, x_D) \quad (64)$$

with coefficients F_k determined by substituting Eq. (64) into the Schrödinger equation, multiplying on the left with $b_k(x_1, x_2, \dots, x_D)$, integrating, and solving an eigenvalue problem, has made it possible, over the last 35 years to calculate numerically exact (ro-)vibrational energy levels and intensities of polyatomic molecules.⁶ A numerical approach of this kind is most useful, e.g., for van der Waals molecules or high-lying states of semi-rigid molecules, when the effect of coupling is important.^{25,73,74} It is necessary, when one wishes to refine a PES. In this paper, we develop a competitive collocation method.

Collocation was introduced into chemical physics many years ago.^{1,2,15} Heretofore, it has usually been used by representing wavefunctions as in Eq. (64) and requiring that the Schrödinger equation be satisfied at a set of points. This is most straightforward if the number of points equals the number of basis functions, but rectangular collocation methods do exist.¹⁶ The standard collocation approach is disfavored for two reasons: (1) it requires solving a generalized eigenvalue problem; (2) the matrix-vector products required to use an iterative eigensolver are costly. In a previous paper, we deal with (1) by combining a Smolyak-type interpolation idea and Lagrange-type basis functions in a way that makes it possible to use collocation without solving a generalized eigenvalue problem.¹⁴ This is crucial because when matrices are too large to store in memory, iterative eigensolvers are imperative and, regrettably, iterative eigensolvers for the generalized eigenvalue problem are costly. For each coordinate, the Lagrange-type basis we use spans the space spanned by a set of eigenfunctions of a 1D cut Hamiltonian. The basis is certain to be good if the coupling is weak. In this paper, we focus on (2) and reduce the cost of the collocation matrix-vector product. Using hierarchical basis functions of Section III, the cost of the most expensive matrix-vector product required to use the collocation method is similar to the cost of the potential matrix-vector product of the variational approach.¹² For a molecule with four atoms, the cost is about two orders of magnitude less than in Ref. 14.

Iterative eigensolvers are now commonly used to solve the time-independent Schrödinger equation.^{5,21,22,52,75–82} Because they enable one to compute (only) the desired eigenvalues without storing matrices, it might seem that they should always be eigensolvers of first resort. Why limit oneself to matrices small enough that they can be stored in memory?

However, in practice, iterative eigensolvers are only efficient if matrix-vector products are cheap. To reduce the cost of matrix-vector products, engineers often exploit sparsity. If the PES can be massaged into sum-of-products form,^{40,83,84} then sparsity can be exploited. In chemical physics, mostly because product bases are often used, it is more common to take advantage of the structure of the Hamiltonian matrix.^{52,79,80,85} When one wishes to use a general potential, it is necessary that both the basis and the grid, used either to do quadrature or as a collocation grid, have structure. In conjunction with quadrature, it has long been understood how to exploit structure by doing sums sequentially when both the basis and the grid are a direct product. Provided that the quadrature grid is a direct product, the same idea works when the basis is a nondirect product (spherical harmonic type basis) with shared indices.⁸⁶ It also efficacious when basis indices are constrained.^{12,26–29,40,64}

When using collocation, it is also possible to use an iterative eigensolver and to impose and exploit compatible structure on both the basis and the grid. Because the matrix whose eigenvalues are computed is non-symmetric, we cannot use the standard Lanczos eigensolver. In this paper, we use ARPACK. It should be possible to use a generalized two-sided Lanczos method.⁸⁷ In the collocation case, it is imperative that one uses an iterative method and takes advantage of structure because matrices are not sparse: exploiting sparsity is not an option. As in the quadrature case, one needs to use two different representations and to be able to efficiently transform between them. One representation is in terms of basis functions, cf. Eq. (11), and the other is a representation on a grid, cf. Eq. (10). These two representations are comparable to the well-known “VBR” and “DVR.”⁵ The principle difference between the VBR/DVR and the representations we use is that ours are not direct products. Of course, it is crucial to obviate the need to use direct products owing to the fact that the size of a direct product basis/grid scales as n^D , where n is a representation number of functions/points in a single coordinate. To use an iterative eigensolver with collocation, one must evaluate matrix-vector products. It is obviously trivial to apply the potential in the grid representation. In this paper, we have applied the KEO by first transforming from the grid representation to the basis representation. The transformation is efficient because the grid and the basis have compatible structures. The structure of the basis is determined by $g(M^1(n_1 + 1), M^2(n_2 + 1), \dots, M^D(n_D + 1)) \leq H$ and the structure of the grid by $g(M^1(k_1), M^2(k_2), \dots, M^D(k_D)) \leq H$, and both structures are inherited from $g(i_1, i_2, \dots, i_D) \leq H$. When the KEO matrix-vector product is evaluated as in this paper, the scaling of the KEO matrix-vector product required to do a collocation calculation and the scaling of the potential (quadrature) matrix-vector product required to do a variational calculation¹² are identical. Note, however, that all Smolyak-based methods will work best when it is possible to find a $g(i_1, i_2, \dots, i_D) = g(i_1) + g(i_2) + \dots + g(i_D) \leq H$ condition that retains all of the important basis functions. When KEO and/or PES terms simultaneously couple many coordinates, Smolyak ideas will work less well.

Why bother with collocation if the scaling is not better? One important advantage of collocation is that it simplifies and

reduces the cost of KEO matrix-vector products. Sometimes the variational KEO matrix-vector product is easy and inexpensive because coordinates and basis functions are chosen so that the variational KEO matrix is simple and sparse. Although these advantages can always be achieved, it is generally preferable to choose coordinates and basis functions to reduce the size of the basis and this is often not compatible with sparsity of the KEO matrix. In terms of the best coordinates, the KEO has the general form of Eq. (39). In general, to compute its matrix elements, one must calculate D dimensional integrals. If many of the $\mathcal{K}_1^c(x_1, x_2, \dots, x_D)$ and $\mathcal{K}_2^{c,c'}(x_1, x_2, \dots, x_D)$ coefficients are complicated functions, then the quadrature problem is severe not only for the potential but also for the KEO. When using collocation, it is only necessary to evaluate $\mathcal{K}_1^c(x_1, x_2, \dots, x_D)$ and $\mathcal{K}_2^{c,c'}(x_1, x_2, \dots, x_D)$ at collocation points.

It is not difficult to devise good sequences of quadrature rules when using 1D polynomial basis sets.^{12,27,28,88,89} This, however, is not possible if the 1D basis functions are more general, e.g., 1D cut eigenfunctions. The inability to use general 1D basis functions is a disadvantage of the Smolyak quadrature variational method and another reason to use collocation. Another advantage of collocation is that one often needs fewer collocation points than quadrature points. Wavefunctions that can be represented by the basis can be computed using only as many points as basis functions; there is no need to use a grid large enough to compute overlap and potential integrals accurately. Now that ideas are in place for using collocation without solving a generalized eigenvalue problem and without costly kinetic matrix-vector products, it will be possible to apply it to problems for which good coordinates and basis functions yield a costly and full variational KEO matrix.

ACKNOWLEDGMENTS

The research described in this paper was supported by the Canadian Natural Sciences and Engineering Research Council. It was completed in Heidelberg during a visit funded in part by the Ontario-Baden-Wuerttemberg exchange program and the German Academic Exchange Service (DAAD). We thank H.-D. Meyer for comments on a preliminary version of the manuscript.

¹W. Yang and A. C. Peet, "The collocation method for bound solutions of the Schrödinger equation," *Chem. Phys. Lett.* **153**(1), 98-104 (1988).

²W. Yang and A. C. Peet, "A method for calculating vibrational bound states: Iterative solution of the collocation equations constructed from localized basis sets," *J. Chem. Phys.* **92**(1), 522-526 (1990).

³S. Carter and N. C. Handy, "The variational method for the calculation of ro-vibrational energy levels," *Comput. Phys. Rep.* **5**(3), 117-171 (1986).

⁴J. Tennyson, "The calculation of the vibration-rotation energies of triatomic molecules using scattering coordinates," *Comput. Phys. Rep.* **4**(1), 1-36 (1986).

⁵J. C. Light and T. Carrington, Jr., "Discrete-variable representations and their utilization," *Adv. Chem. Phys.* **114**, 263-310 (2000).

⁶J. M. Bowman, T. Carrington, Jr., and H.-D. Meyer, "Variational quantum approaches for computing vibrational energies of polyatomic molecules," *Mol. Phys.* **106**(16-18), 2145-2182 (2008).

⁷M. H. Beck, A. Jäckle, G. A. Worth, and H. D. Meyer, "The multiconfiguration time-dependent Hartree (MCTDH) method: A highly efficient algorithm for propagating wavepackets," *Phys. Rep.* **324**(1), 1-105 (2000).

⁸S. Carter, S. J. Culik, and J. M. Bowman, "Vibrational self-consistent field method for many-mode systems: A new approach and application to the vibrations of CO adsorbed on Cu (100)," *J. Chem. Phys.* **107**(24), 10458-10469 (1997).

⁹O. Vendrell, F. Gatti, and H.-D. Meyer, "Full dimensional (15 dimensional) quantum-dynamical simulation of the protonated water-dimer IV: Isotope effects in the infrared spectra of $D(D_2O)_2^+$, $H(D_2O)_2^+$, and $D(H_2O)_2^+$ isotopologues," *J. Chem. Phys.* **131**(3), 034308 (2009).

¹⁰A. Zocante, P. Seidler, M. Bo Hansen, and O. Christiansen, *J. Chem. Phys.* **136**, 204118 (2012).

¹¹X.-G. Wang and T. Carrington, Jr., "A contracted basis-Lanczos calculation of vibrational levels of methane: Solving the Schrödinger equation in nine dimensions," *J. Chem. Phys.* **119**(1), 101-117 (2003).

¹²G. Avila and T. Carrington, Jr., "Nonproduct quadrature grids for solving the vibrational Schrödinger equation," *J. Chem. Phys.* **131**(17), 174103 (2009).

¹³D. Lauvergnat and A. Nauts, "Quantum dynamics with sparse grids: A combination of Smolyak scheme and cubature. Application to methanol in full dimensionality," *Spectrochim. Acta, Part A* **119**, 18-25 (2014).

¹⁴G. Avila and T. Carrington, Jr., *J. Chem. Phys.* **139**, 134114 (2013).

¹⁵A. C. Peet, W. Yang, A. C. Peet, and W. Yang, "The collocation method for calculating vibrational bound states of molecular systems with application to ArHCl," *J. Chem. Phys.* **90**(3), 1746-1751 (1989).

¹⁶S. Manzhos and T. Carrington, Jr., "An improved neural network method for solving the Schrödinger equation 1," *Can. J. Chem.* **87**(7), 864-871 (2009).

¹⁷S. Manzhos, K. Yamashita, and T. Carrington, Jr., "On the advantages of a rectangular matrix collocation equation for computing vibrational spectra from small basis sets," *Chem. Phys. Lett.* **511**(4), 434-439 (2011).

¹⁸M. Chan, T. Carrington, Jr., and S. Manzhos, "Anharmonic vibrations of the carboxyl group in acetic acid on TiO₂: Implications for adsorption mode assignment in dye-sensitized solar cells," *Phys. Chem. Chem. Phys.* **15**(25), 10028-10034 (2013).

¹⁹S. Manzhos, M. Chan, and T. Carrington, Jr., "Communication: Favorable dimensionality scaling of rectangular collocation with adaptable basis functions up to 7 dimensions," *J. Chem. Phys.* **139**(5), 051101 (2013).

²⁰U. Manthe and H. Koeppel, *J. Chem. Phys.* **93**, 345 (1990).

²¹M. J. Bramley and T. Carrington, Jr., "A general discrete variable method to calculate vibrational energy levels of three and four atom molecules," *J. Chem. Phys.* **99**(11), 8519-8541 (1993).

²²J. C. Tremblay and T. Carrington, Jr., "Calculating vibrational energies and wave functions of vinylidene using a contracted basis with a locally reorthogonalized coupled two-term Lanczos eigensolver," *J. Chem. Phys.* **125**(9), 094311 (2006).

²³P. Sarkar, N. Poulin, and T. Carrington, Jr., "Calculating rovibrational energy levels of a triatomic molecule with a simple Lanczos method," *J. Chem. Phys.* **110**(21), 10269-10274 (1999).

²⁴X.-G. Wang and T. Carrington, Jr., "Vibrational energy levels of CH_3^+ ," *J. Chem. Phys.* **129**(23), 234102 (2008).

²⁵R. Dawes, X. G. Wang, A. W. Jasper, and T. Carrington, Jr., "Nitrous oxide dimer: A new potential energy surface and rovibrational spectrum of the nonpolar isomer," *J. Chem. Phys.* **133**(13), 134304 (2010).

²⁶X.-G. Wang and T. Carrington, Jr., "The utility of constraining basis function indices when using the Lanczos algorithm to calculate vibrational energy levels," *J. Phys. Chem. A* **105**(12), 2575-2581 (2001).

²⁷G. Avila and T. Carrington, Jr., "Using nonproduct quadrature grids to solve the vibrational Schrödinger equation in 12D," *J. Chem. Phys.* **134**(5), 054126 (2011).

²⁸G. Avila and T. Carrington, Jr., "Using a pruned basis, a non-product quadrature grid, and the exact Watson normal-coordinate kinetic energy operator to solve the vibrational Schrödinger equation for C_2H_4 ," *J. Chem. Phys.* **135**(6), 064101 (2011).

²⁹G. Avila and T. Carrington, Jr., "Solving the vibrational Schrödinger equation using bases pruned to include strongly coupled functions and compatible quadratures," *J. Chem. Phys.* **137**(17), 174108 (2012).

³⁰E. L. Sibert, J. T. Hynes, and W. P. Reinhardt, *J. Chem. Phys.* **77**, 3595-3604 (1982).

³¹S. F. Boys, "Some bilinear convergence characteristics of the solutions of dissymmetric secular equations," *Proc. R. Soc. London, Ser. A* **309**, 195-208 (1969).

³²J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation," *SIAM Rev.* **46**(3), 501-517 (2004).

³³G. Mastroianni and D. Occorsio, "Lagrange interpolation at Laguerre zeros in some weighted uniform spaces," *Acta Math. Hung.* **91**(1-2), 27-52 (2001).

- ³⁴H.-J. Bungartz and M. Griebel, "Sparse grids," *Acta Numer.* **13**, 147-269 (2004).
- ³⁵V. Barthelmann, E. Novak, and K. Ritter, "High dimensional polynomial interpolation on sparse grids," *Adv. Comput. Math.* **12**(4), 273-288 (2000).
- ³⁶J. Shen and Y. Haijun, "Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems," *SIAM J. Sci. Comput.* **32**(6), 3228-3250 (2010).
- ³⁷B. A. Malin, D. Krueger, and F. Kubler, "Solving the multi-country real business cycle model using a Smolyak-collocation method," *J. Econ. Dyn. Control* **35**(2), 229-239 (2011).
- ³⁸B. Ganapathysubramanian and N. Zabarar, "Sparse grid collocation schemes for stochastic natural convection problems," *J. Comput. Phys.* **225**(1), 652-685 (2007).
- ³⁹F. Nobile, R. Tempone, and C. G. Webster, "A sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM J. Numer. Anal.* **46**(5), 2309-2345 (2008).
- ⁴⁰G. Avila and T. Carrington, Jr., "Using multi-dimensional Smolyak interpolation to make a sum-of-products potential," *J. Chem. Phys.* **143**(4), 044106 (2015).
- ⁴¹K. Hallatschek, "Fourier-transform on sparse grids with hierarchical bases," *Numer. Math.* **63**(1), 83-97 (1992).
- ⁴²V. Gradinaru, "Fourier transform on sparse grids: Code design and the time dependent Schrödinger equation," *Computing* **80**(1), 1-22 (2007).
- ⁴³X. Ma and N. Zabarar, "An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations," *J. Comput. Phys.* **228**(8), 3084-3113 (2009).
- ⁴⁴*Sparse Grids and Applications*, edited by J. Garcke and M. Griebel (Springer Science Business Media, 2012), Vol. 88.
- ⁴⁵A. Klimke and B. Wohlmuth, "Algorithm 847: Spinterp. Piecewise multilinear hierarchical sparse grid interpolation in MATLAB," *ACM Trans. Math. Software* **31**(4), 561-579 (2005).
- ⁴⁶M. Griebel and J. Hamaekers, "Fast discrete Fourier transform on generalized sparse grids," in *Sparse Grids and Applications* (Springer International Publishing, 2014), pp. 75-107.
- ⁴⁷K. Petras, "Smolyak cubature of given polynomial degree with few nodes for increasing dimension," *Numer. Math.* **93**(4), 729-753 (2003).
- ⁴⁸J. Szabados, "Weighted Lagrange and Hermite-Fejér interpolation on the real line," *J. Inequalities Appl.* **1997**, 481267, pdf available at <http://www.emis.ams.org/journals/HOA/JIA/1/299.pdf>.
- ⁴⁹M. Griebel and S. Knapek, "Optimized tensor-product approximation spaces," *Constr. Approximation* **16**(4), 525-540 (2000).
- ⁵⁰F. Richter, M. Hochlaf, P. Rosmus, F. Gatti, and H.-D. Meyer, "A study of the mode-selective transcis isomerization in HONO using *ab initio* methodology," *J. Chem. Phys.* **120**(3), 1306-1317 (2004).
- ⁵¹A. McNichols and T. Carrington, Jr., "Vibrational energy levels of formaldehyde calculated from an internal coordinate Hamiltonian using the Lanczos algorithm," *Chem. Phys. Lett.* **202**(6), 464-470 (1993).
- ⁵²H.-G. Yu and J. T. Muckerman, *J. Mol. Spectrosc.* **214**, 11-20 (2002).
- ⁵³D. Neuhauser, "Bound state eigenfunctions from wave packets: Time energy resolution," *J. Chem. Phys.* **93**(4), 2611-2616 (1990).
- ⁵⁴R. B. Lehoucq, S. K. Gray, D.-H. Zhang, and J. C. Light, "Vibrational eigenstates of four-atom molecules: A parallel strategy employing the implicitly restarted Lanczos method," *Comput. Phys. Commun.* **109**(1), 15-25 (1998).
- ⁵⁵F. Ribeiro, C. Iung, and C. Leforestier, "A Jacobi-Wilson description coupled to a block-Davidson algorithm: An efficient scheme to calculate highly excited vibrational levels," *J. Chem. Phys.* **123**(5), 054106 (2005).
- ⁵⁶C. Iung and C. Leforestier, "Accurate determination of a potential energy surface for CD₃H," *J. Chem. Phys.* **90**(6), 3198-3203 (1989).
- ⁵⁷S.-W. Huang and T. Carrington, Jr., "A comparison of filter diagonalisation methods with the Lanczos method for calculating vibrational energy levels," *Chem. Phys. Lett.* **312**(2), 311-318 (1999).
- ⁵⁸G. Balint-Kurti and P. Pulay, "A new grid-based method for the direct computation of excited molecular vibrational states: Test application to formaldehyde," *J. Mol. Struct.: THEOCHEM* **341**(1), 1-11 (1995).
- ⁵⁹D. Xu, R. Chen, and H. Guo, "Probing highly excited vibrational eigenfunctions using a modified single Lanczos propagation method: Application to acetylene (HCCH)," *J. Chem. Phys.* **118**(16), 7273-7282 (2003).
- ⁶⁰R. Chen, G. Ma, and H. Guo, "Six-dimensional quantum calculations of highly excited vibrational energy levels of hydrogen peroxide and its deuterated isotopomers," *J. Chem. Phys.* **114**(11), 4763-4774 (2001).
- ⁶¹H. G. Yu, "An efficient grid calculation of vibrational states for H₃ with geometric phase in hyperspherical coordinates," *Chem. Phys. Lett.* **281**(4), 312-318 (1997).
- ⁶²G. H. Golub and C. F. Van Loan, *Matrix Computations* (JHU Press, 2012), Vol. 3.
- ⁶³R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods* (SIAM, Philadelphia, PA, 1998), available at <http://www.caam.rice.edu/software/ARPACK>.
- ⁶⁴G. Avila and T. Carrington, Jr., "Non-product quadrature grids: Solving the vibrational Schroedinger equation in 12D," in *Quantum Dynamic Imaging: Theoretical and Numerical Methods*, edited by A. D. Bandrauk and M. Y. Ivanov (Springer, 2010).
- ⁶⁵B. Brumm, "A fast matrix-free algorithm for spectral approximations to the Schrödinger equation," *J. Sci. Comput.* **37**, A2003-A2025 (2015).
- ⁶⁶N. C. Handy, "The derivation of vibration-rotation kinetic energy operators, in internal coordinates," *Mol. Phys.* **61**(1), 207-223 (1987).
- ⁶⁷E.-W. Knapp and D. J. Diestler, "Quantum-mechanical treatment of collision induced dissociation," *J. Chem. Phys.* **67**(11), 4969-4975 (1977).
- ⁶⁸J. Tennyson and B. T. Sutcliffe, "The *ab initio* calculation of the vibrational rotational spectrum of triatomic systems in the close coupling approach, with KCN and H₂Ne as examples," *J. Chem. Phys.* **77**(8), 4061-4072 (1982).
- ⁶⁹H. Wei and T. Carrington, Jr., "The discrete variable representation for a triatomic Hamiltonian in bond length-bond angle coordinates," *J. Chem. Phys.* **97**, 3029-3037 (1992).
- ⁷⁰H. Wei and T. Carrington, Jr., "Discrete variable representations of complicated kinetic energy operators," *J. Chem. Phys.* **101**, 1343-1360 (1994).
- ⁷¹D. W. Schwenke, "On the computation of ro-vibrational energy levels of triatomic molecules," *Comput. Phys. Commun.* **70**(1), 1-14 (1992).
- ⁷²See supplementary material at <http://dx.doi.org/10.1063/1.4936294> for a table of all the assignable HONO vibrational levels.
- ⁷³J. M. Hutson, "Intermolecular forces from the spectroscopy of van der Waals complexes," *Ann. Rev. Phys. Chem.* **41**, 123-154 (1990).
- ⁷⁴E. L. Sibert, *Int. Rev. Phys. Chem.* **9**, 1 (1990).
- ⁷⁵H. Koeppel, W. Domcke, and L. S. Cederbaum, *Adv. Chem. Phys.* **57**, 59 (1984).
- ⁷⁶C. Leforestier, K. Szalewicz, and A. van der Avoird, "Spectra of water dimer from a new *ab initio* potential with flexible monomers," *J. Chem. Phys.* **137**, 014305 (2012).
- ⁷⁷H. Y. Mussa and J. Tennyson, "Calculation of the rotation-vibration states of water up to dissociation," *J. Chem. Phys.* **109**, 10885-10892 (1998).
- ⁷⁸V. A. Mandelshtam and H. S. Taylor, "Spectral projection approach to the quantum scattering calculations," *J. Chem. Phys.* **102**, 7390-7399 (1995).
- ⁷⁹H.-G. Yu and S. C. Smith, "The calculation of vibrational eigenstates by MINRES filter diagonalization," *Ber. Bunsenges. Phys. Chem.* **101**, 400-406 (1997).
- ⁸⁰R. Chen and H. Guo, "Discrete energy representation and generalized propagation of physical systems," *J. Chem. Phys.* **108**, 6068-6077 (1998).
- ⁸¹P.-N. Roy and T. Carrington, Jr., "An evaluation of methods designed to calculate energy levels in a selected range and application to a (one-dimensional) Morse oscillator and (three-dimensional) HCN/HNC," *J. Chem. Phys.* **103**, 5600-5612 (1995).
- ⁸²D. Xu, G. Li, D. Xie, and H. Guo, "Full dimensional quantum calculations of vibrational energy levels of acetylene (HCCH) up to 13000 cm⁻¹," *Chem. Phys. Lett.* **365**, 480-486 (2002).
- ⁸³A. Jaekle and H.-D. Meyer, "Product representation of potential energy surfaces," *J. Chem. Phys.* **104**(20), 7974-7984 (1996).
- ⁸⁴S. Manzhos and T. Carrington, Jr., "Using redundant coordinates to represent potential energy surfaces with lower-dimensional functions," *J. Chem. Phys.* **127**(1), 014103 (2007).
- ⁸⁵H.-G. Yu and J. T. Muckerman, "A general variational algorithm to calculate vibrational energy levels of tetraatomic molecules," *J. Mol. Spectrosc.* **214**, 11 (2002).
- ⁸⁶M. J. Bramley, J. W. Tromp, T. Carrington, Jr., and C. G. Corey, "Efficient calculation of highly excited vibrational energy levels of floppy molecules: The band origins of H₃⁺ up to 35 000 cm⁻¹," *J. Chem. Phys.* **100**(9), 6175-6194 (1994).
- ⁸⁷J. C. Tremblay and T. Carrington, Jr., "A refined unsymmetric Lanczos eigensolver for computing accurate eigentriples of a real unsymmetric matrix," *Electron. Trans. Numer. Anal.* **28**, 95-113 (2007), pdf available at <http://www.emis.ams.org/journals/ETNA/vol.28.2007/pp95-113.dir/pp95-113.pdf>.
- ⁸⁸F. Heiss and V. Wünschel, "Likelihood approximation by numerical integration on sparse grids," *J. Econ.* **144**(1), 62-80 (2008).
- ⁸⁹R. Bourquin, "Exhaustive search for higher-order Kronrod-Patterson extensions" (2015), pdf available at <http://www.sam.math.ethz.ch/~raoulb/research/preprints/kes.pdf>.