

RESEARCH ARTICLE | AUGUST 19 2025

Taylor-mode automatic differentiation for constructing molecular rovibrational Hamiltonian operators

Andrey Yachmenev  ; Emil Vogt  ; Álvaro Fernández Corral  ; Yahya Saleh 

 Check for updates

J. Chem. Phys. 163, 072501 (2025)
<https://doi.org/10.1063/5.0287347>

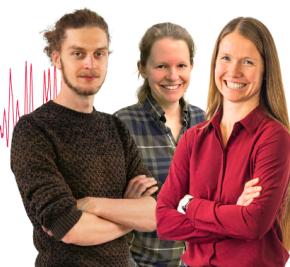


Webinar From Noise to Knowledge

May 13th – Register now



Universität
Konstanz



Taylor-mode automatic differentiation for constructing molecular rovibrational Hamiltonian operators

Cite as: J. Chem. Phys. 163, 072501 (2025); doi: 10.1063/5.0287347

Submitted: 24 June 2025 • Accepted: 24 July 2025 •

Published Online: 19 August 2025



View Online



Export Citation



CrossMark

Andrey Yachmenev,^{a)} Emil Vogt, Álvaro Fernández Corral, and Yahya Saleh

AFFILIATIONS

Center for Free-Electron Laser Science CFEL, Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg, Germany

^{a)}Author to whom correspondence should be addressed: andrey.yachmenev@robochimps.com

URL: <https://github.com/robochimps>

ABSTRACT

We present an automated framework for constructing Taylor series expansions of rovibrational kinetic and potential energy operators for arbitrary molecules, internal coordinate systems, and molecular frame embedding conditions. Expressing operators in a sum-of-products form allows for computationally efficient evaluations of matrix elements in product basis sets. Our approach uses automatic differentiation tools from the Python machine learning ecosystem, particularly the JAX library, to efficiently and accurately generate high-order Taylor expansions of rovibrational operators. The implementation is available at <https://github.com/robochimps/vibrojet>.

© 2025 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0287347>

I. INTRODUCTION

Accurate calculations of molecular rovibrational spectra, especially for floppy molecules, weakly bound complexes, and highly excited states in general, are often computationally demanding. These calculations benefit significantly from the use of tailored coordinates that effectively capture the essential rovibrational motions specific to each molecule.^{1–4} While several advanced generalized methodologies exist to perform such calculations,^{5–7} applying them to a new molecule typically requires more than simply providing a potential energy surface (PES). It also involves defining an appropriate kinetic energy operator (KEO) based on the chosen internal coordinates and molecular frame embedding conditions. This setup process demands considerable expertise in both the theoretical foundations and practical aspects of implementation.

The challenge of choosing optimal internal coordinates and frame embeddings has been the subject of ongoing research, with most recent advances focused on variational optimization of coordinate systems.^{3,4,8} However, even after suitable coordinates are defined, constructing the PES and KEO in those coordinates and efficiently evaluating their matrix elements in a chosen basis set remains nontrivial. The matrix elements are typically computed using

multidimensional numerical integration methods, such as Gaussian quadrature, often combined with Smolyak sparse grids to mitigate the curse of dimensionality.⁹ Related discrete variable representation (DVR) methods also build on Gaussian quadrature to define an orthonormal basis that simplifies the numerical integration.^{6,7,10,11} An alternative approach, which is the focus of this study, involves expressing the rovibrational operators as sums of products of univariate functions, which significantly reduces the complexity of multidimensional integration. For example, when a direct product basis of univariate functions is used, multidimensional integrals reduce to a sum of products of one-dimensional integrals. However, in DVR methods, a sum-of-products form of the KEO is not required.

Molecular PESs and dipole moment surfaces are commonly represented in a sum-of-products form by fitting analytic functions to electronic structure data computed on a grid of molecular geometries. For the KEO, one established approach involves using symbolic algebra to derive an analytical expression, followed by manual restructuring into a sum-of-products form and hand-coded implementation.¹² This process is tedious and highly system specific; it must be repeated for every new molecule type, coordinate system, and frame embedding condition. Moreover, the embedding

must be analytically expressible, which often excludes commonly used schemes such as the Eckart frame due to their algebraic complexity. The choice of embedding is particularly important when rotational motion is considered, as it affects the separability of vibrational and rotational degrees of freedom. It influences not only the basis set convergence of rovibrational energies but also the accuracy of vibrational transition intensities, even when rotational degrees of freedom are not explicitly included in the model.

Certain choices of vibrational coordinates and embedding conditions, such as polyspherical coordinates, enable the derivation of the rovibrational KEO directly in the sum-of-products form.^{13–16} This avoids complex symbolic algebra and facilitates the construction of the KEO from molecular fragments.

A more general and flexible alternative is to construct the rovibrational operators in a sum-of-products form via Taylor series expansions^{5,17} or least-squares fitting techniques.^{18,19} In the Taylor expansion approach, derivatives can be computed using finite-difference methods²⁰ or automatic differentiation (AD), particularly via forward-mode propagation of Taylor series.¹⁷ These methods are well-suited for handling complex coordinate transformations and frame embeddings, including those defined implicitly by nonlinear equations (e.g., Eckart conditions), enabling fully automated construction of KEOs without manual symbolic manipulation.

In this work, we introduce *Vibrojet*, a general Python-based framework for constructing Taylor series expansions of molecular rovibrational operators using modern AD tools. In particular, we employ the JAX library and its *jet* module, which supports efficient Taylor-mode differentiation. We further employ an efficient approach for computing high-order partial derivatives by propagating families of univariate Taylor series.^{21,22} Compared to our previous Fortran-based implementation,¹⁷ the new Python framework offers several advantages, including easier coding for setting up new molecules and coordinate systems, integration with a broader scientific computing ecosystem, and more accessible platform-specific optimization and parallelization.

This article is organized as follows: In Sec. II, we present a general numerical approach for constructing the KEOs of molecules. The Taylor-mode AD approach is described in Sec. III and details of its implementation in Sec. IV. Finally, Sec. V provides illustrative examples of Taylor series expansion for the PES and KEO of selected molecules and demonstrates their use in variational calculations of vibrational energy levels.

II. NUMERICAL PROCEDURE FOR KEO

The general form of the rovibrational KEO is given by

$$\hat{T} = \sum_{\lambda, \mu=1}^{M+6} p_\lambda^\dagger G_{\lambda\mu}(\xi) p_\mu + U(\xi),$$

where $p_{\lambda=1..M+6} = \{-i\hbar\partial/\partial\xi_1, \dots, -i\hbar\partial/\partial\xi_M, \hat{J}_x, \hat{J}_y, \hat{J}_z, -i\hbar\partial/\partial X_{\text{cm}}, -i\hbar\partial/\partial Y_{\text{cm}}, \text{ and } -i\hbar\partial/\partial Z_{\text{cm}}\}$ are the generalized momentum operators conjugate to the M vibrational coordinates $\xi = \xi_1, \xi_2, \dots, \xi_M$, three Cartesian components of the rotational angular momentum operator \mathbf{J} , and three Cartesian momentum operators of the overall translation. The mass-weighted contravariant metric tensor $G_{\lambda\mu}$ and pseudopotential U can be computed as^{6,23}

$$G_{\lambda\mu} = [\mathbf{g}^{-1}]_{\lambda\mu},$$

$$g_{\lambda\mu} = \sum_{i=1}^N \sum_{\alpha=x,y,z} m_i t_{i\alpha, \lambda} t_{i\alpha, \mu},$$

$$U = \frac{\hbar^2}{32} \sum_{\lambda, \mu=1}^M \left[\frac{G_{\lambda\mu}}{\tilde{g}^2} \frac{\partial \tilde{g}}{\partial \xi_\lambda} \frac{\partial \tilde{g}}{\partial \xi_\mu} + 4 \frac{\partial}{\partial \xi_\lambda} \left(\frac{G_{\lambda\mu}}{\tilde{g}} \frac{\partial \tilde{g}}{\partial \xi_\mu} \right) \right],$$

where $\tilde{g} = \det(g_{\lambda\mu})$. The $t_{i\alpha, \lambda}$ vectors are defined in terms of the Cartesian coordinates $r_{i\alpha}$ ($\alpha = x, y, z$) of the $i = 1 \dots N$ atoms in the molecule, as

$$t_{i\alpha, \lambda} = \frac{\partial r_{i\alpha}}{\partial \xi_\lambda} \quad (\lambda = 1..M),$$

$$t_{i\alpha, M+\beta} = \sum_{y=x,y,z} \epsilon_{\alpha\beta y} r_{iy} \quad (\beta = x, y, z),$$

$$t_{i\alpha, M+3+\beta} = \delta_{\alpha\lambda} \quad (\beta = X_{\text{cm}}, Y_{\text{cm}}, Z_{\text{cm}})$$

for vibrational, rotational, and translational coordinates, respectively, where $\epsilon_{\alpha\beta y}$ is the three-dimensional Levi–Civita symbol ($\epsilon = +1$ for even permutations of α , β , and y , and $\epsilon = -1$ for odd permutations). The pseudopotential U is a scalar operator that can be combined with the PES to avoid the need for evaluating and storing additional integrals.

From the above equations, it follows that the KEO can be constructed automatically, provided a coordinate transformation function is defined that maps the internal vibrational coordinates ξ_λ ($\lambda = 1 \dots M$) to the Cartesian coordinates $r_{i\alpha}$ of the atoms,

$$\mathbf{r}_i \equiv (x_i, y_i, z_i) = \mathbf{f}_i(\xi_1, \xi_2, \dots, \xi_M). \quad (1)$$

Defining such a transformation establishes, either implicitly or explicitly, an orientation of the Cartesian x, y, z axes relative to the molecule. This orientation may itself vary with changes of the internal coordinates.

High-order Taylor expansions of the coordinate mappings in (1), computed using AD tools, have been successfully employed in computational chemistry to accelerate geometry optimization, reaction path search, and classical trajectory integration.²⁴

Popular choices for the orientation of the molecular frame x, y, z axes include the principal axes system (PAS) and the Eckart frame. The PAS choice minimizes the off-diagonal elements in the pure rotational block of the G -matrix, whereas the Eckart frame is designed to minimize the off-diagonal elements in the rovibrational (Coriolis) part of the G -matrix. The orientation of the molecular frame can be conveniently expressed using a rotation matrix $\mathbf{d}(\xi)$, which maps reference Cartesian coordinates of atoms (user-defined or derived from a Z-matrix) into the Eckart or PAS frame,

$$\mathbf{r}_i = \mathbf{d} \cdot \bar{\mathbf{r}}_i \equiv \mathbf{d} \cdot \mathbf{f}_i(\xi_1, \dots, \xi_M). \quad (2)$$

The conditions defining the Eckart or PAS frames can be formulated in terms of the rotation matrix \mathbf{d} as follows:

$$\mathbf{u}\mathbf{d}^T - \mathbf{d}\mathbf{u}^T = \mathbf{0} \quad (\text{Eckart}), \quad (3)$$

$$[\mathbf{d}\bar{\mathbf{u}}\mathbf{d}^T]_{\alpha\beta} = 0, \quad \alpha \neq \beta \quad (\text{PAS}). \quad (4)$$

Here, the matrices \mathbf{u} and $\bar{\mathbf{u}}$ are defined as

$$u_{\alpha\beta} = \sum_{i=1}^N m_i r_{i\alpha}^{(\text{ref})} \bar{r}_{i\beta},$$

$$\bar{u}_{\alpha\beta} = \sum_{i=1}^N m_i \bar{r}_{i\alpha} \bar{r}_{i\beta}.$$

The Eckart conditions are defined with respect to a reference geometry, typically chosen as the equilibrium geometry $r_{i\alpha}^{(\text{ref})} = r_{i\alpha}^{(\text{eq})}$, around which the rovibrational coupling elements of the G -matrix are minimized.

The frame embedding conditions can be incorporated directly into the coordinate transformation function \mathbf{f}_i in (2) or handled externally through generalized routines. We briefly outline a numerical solution of the Eckart equations, originally presented in Ref. 17. Several alternative approaches have been proposed based on eigenvalue problems,^{25,26} similar to the diagonalization of the moment of inertia tensor used in constructing the PAS frame. However, we found that methods based on eigenvalue decomposition can encounter difficulties when computing derivatives required for constructing the Taylor series expansions of the KEO (see Sec. III). These issues are particularly pronounced in highly symmetric molecules such as methane, where degenerate eigenvalues arise²⁷—not to be confused with eigenvalues of the Hamiltonian. In such cases, the derivatives of eigenvectors are not well defined, making differentiation of the eigenvalue-based solutions fundamentally problematic.

The present solution is based on the parameterization of the rotation matrix \mathbf{d} in (2) using a 3×3 skew-symmetric matrix $\boldsymbol{\kappa}$, such that

$$\mathbf{d} = e^{-\boldsymbol{\kappa}},$$

$$\boldsymbol{\kappa}^T = -\boldsymbol{\kappa}.$$
(5)

The exponential parameterization ensures that the orthogonality of \mathbf{d} is preserved for all values of the three independent elements κ_{xy} , κ_{xz} , and κ_{yz} . These elements are determined by solving the Eckart equations in (3) rewritten in terms of $\boldsymbol{\kappa}$. Substituting (5) into the Eckart conditions and rearranging terms, we obtain the following linear system for $\boldsymbol{\kappa}$:

$$\begin{pmatrix} u_{xx} + u_{yy} & u_{yz} & -u_{xz} \\ u_{zy} & u_{xx} + u_{zz} & u_{xy} \\ -u_{zx} & u_{yx} & u_{yy} + u_{zz} \end{pmatrix} \cdot \begin{pmatrix} \kappa_{xy} \\ \kappa_{xz} \\ \kappa_{yz} \end{pmatrix} = \sum_{\alpha=x,y,z} \begin{pmatrix} \lambda_{x\alpha} u_{ya} - \lambda_{ya} u_{x\alpha} \\ \lambda_{x\alpha} u_{za} - \lambda_{za} u_{x\alpha} \\ \lambda_{ya} u_{za} - \lambda_{za} u_{ya} \end{pmatrix},$$
(6)

where

$$\boldsymbol{\lambda} = e^{-\boldsymbol{\kappa}} + \boldsymbol{\kappa}.$$

This system is solved iteratively, starting with the initial guess $\boldsymbol{\lambda} = \mathbf{I}$. At each iteration, the elements of $\boldsymbol{\kappa}$ are updated, and the exponential matrix $e^{-\boldsymbol{\kappa}}$ is computed using a Taylor series expansion, Rodrigues' formula, or the Padé approximation.²⁸

III. TAYLOR POLYNOMIALS

Standard AD relies on the fact that a function to be differentiated can be expressed as a composition of smooth elementary functions. For example, if a function is defined as $f(x) = g \circ h(x)$, then by the chain rule, its derivative is $f'(x) = g' \circ h(x)h'(x)$. For higher-order derivatives of such compositions, the Faà di Bruno formula generalizes the chain rule and enables recursive computation of higher-order derivatives from lower-order ones. If the functions involved in the composition admit converging Taylor series expansions, a convenient method for computing higher-order derivatives is to use truncated Taylor polynomials. Let the function $h(x)$ be represented by the Taylor expansion in x around $x_0 = 0$,

$$h(x) = h_0 + h_1 x + \frac{1}{2} h_2 x^2 + \frac{1}{3!} h_3 x^3 + \cdots + \frac{1}{n!} h_n x^n.$$

Assuming both h and g admit converging Taylor expansions, the composite function $f(x) = g \circ h(x)$ can also be expressed as a Taylor series around $x_0 = 0$,

$$f(x) = f_0 + f_1 x + \frac{1}{2} f_2 x^2 + \frac{1}{3!} f_3 x^3 + \cdots + \frac{1}{n!} f_n x^n,$$

with coefficients given by

$$\begin{aligned} f_0 &= g(h_0), \\ f_1 &= g'(h_0)h_1, \\ f_2 &= g'(h_0)h_2 + g''(h_0)h_1 h_1, \\ f_3 &= g'(h_0)h_3 + 3g''(h_0)h_1 h_2 + g'''(h_0)h_1 h_1 h_1, \\ &\vdots \end{aligned}$$

Each coefficient f_k corresponds to the k -th derivative of the composition $g \circ h(x)$ with respect to x , as described by the Faà di Bruno formula. The terms $g^{(k)}(h_0)$ represent the coefficients of the Taylor series expansion of g around h_0 .

To extend AD to higher-order derivatives, it is necessary to reformulate elementary arithmetic operations, intrinsic functions (e.g., exp, sin), and linear algebra routines (e.g., det, eigh) to operate on truncated Taylor polynomials. This can be accomplished by systematically applying the Faà di Bruno rule for function composition and the Leibniz rule for product differentiation.

Several prior works have implemented efficient polynomial arithmetic for truncated Taylor expansions. Notable contributions include the foundational work of Griewank and Walther^{21,22,29} and their C/C++ implementation,³⁰ the Julia implementation by Benet and Sanders,³¹ and the Python-based approach by Bettencourt, Johnson, and Duvenaud³² implemented in the JAX library.³³ In our own earlier work, Taylor polynomial arithmetic in Fortran 90 was implemented as part of the rovibrational code TROVE.¹⁷

In this work, we use the `jax.experimental.jax` module of the JAX library.³³ Although `jax` provides high-order derivative rules for many functions and linear algebra operations, at the time of writing, it lacks support for several essential primitives. These include eigenvalue and LU decompositions, matrix determinant, matrix inverse, and matrix exponential. Furthermore, certain arithmetic operations specific to the KEO construction, such as solutions of the molecular frame embedding equations in (3) and (4), can be more

efficiently implemented by expressing them directly within the Taylor polynomial algebra framework. To that end, our code *Vibrojet* extends JAX's Taylor polynomial functionality by adding support for missing linear algebra primitives required for the KEO construction. It also introduces new operations tailored to the Eckart and PAS frame embedding equations. For eigenvalue decomposition, we utilized formulas derived in Ref. 34.

As an example, we describe the computation of high-order derivatives of the frame rotation matrix in (5), which satisfies the Eckart embedding conditions. Differentiating (3) l times, we obtain

$$\sum_{m=0}^l \binom{l}{m} (\mathbf{u}_m \mathbf{d}_{l-m}^T - \mathbf{d}_{l-m} \mathbf{u}_m^T) = 0,$$

where the subscript l denotes the l -th derivative. Rearranging terms yields a more convenient form for the recursive computation of derivatives \mathbf{d}_l ,

$$\mathbf{u}\boldsymbol{\kappa}_l + \boldsymbol{\kappa}_l \mathbf{u}^T = \lambda_l \mathbf{u}^T - \mathbf{u} \lambda_l^T + \sum_{m=1}^l \binom{l}{m} (\mathbf{d}_{l-m} \mathbf{u}_m^T - \mathbf{u}_m \mathbf{d}_{l-m}^T), \quad (7)$$

with

$$\lambda_l = \mathbf{d}_l + \boldsymbol{\kappa}_l, \quad (8)$$

and matrix exponential

$$\mathbf{d}_l = [e^{-\boldsymbol{\kappa}}]_l = \sum_n \frac{(-1)^n}{n!} [\boldsymbol{\kappa}^n]_l \quad (9)$$

This system of equations is solved iteratively for each derivative $\boldsymbol{\kappa}_l$, starting with the initial value $\lambda_l = 0$ for $l > 0$ and $\lambda_0 = \mathbf{I}$. At each iteration, the three independent elements, $\kappa_l^{(xy)}$, $\kappa_l^{(xz)}$, and $\kappa_l^{(yz)}$, are determined by solving a linear system. The corresponding linear-system matrix remains constant across all derivative orders [same as in (6)], allowing its inverse to be computed once and reused. The right-hand side of (7) depends on λ_l [also as in (6)], which is updated recursively using (8) and (9). It also includes a sum over low-order derivatives \mathbf{d}_k ($k = 0 \dots l-1$), which are computed in the earlier steps of the recursion.

Derivatives of matrix powers $[\boldsymbol{\kappa}^n]_l$ are computed using the Leibniz product rule,

$$[\boldsymbol{\kappa}^n]_l = \sum_{m=0}^l \binom{l}{m} [\boldsymbol{\kappa}^{n-1}]_m [\boldsymbol{\kappa}]_{l-m}.$$

To ensure convergence of the exponential series in (9), we use the scaling and squaring technique.²⁸ In practice, truncating the expansion after the 6th-order term with two to four scaling steps yields accurate results. The solution of the Eckart equations typically converges within 6–10 iterations. All operations are formulated as products of 3×3 matrices, allowing efficient execution in Python/JAX using just-in-time compilation.

Finally, a note on multivariate partial derivatives. The coefficients of multivariate Taylor polynomials can be efficiently computed by propagating univariate directional derivatives using the interpolation approach of Griewank, Utke, and Walther²¹ (see also

Chap. 13.3 in Ref. 29). This method is implemented in *Vibrojet* and is especially effective for Taylor expansions of rovibrational operators in large molecules, where an N -mode truncation scheme is commonly employed to improve convergence.³⁵ The interpolation of directional derivatives enables computation of partial derivatives along selected directions without constructing the full high-dimensional derivative tensor, resulting in significant computational savings.

IV. IMPLEMENTATION

The mapping from internal to Cartesian coordinates, as described in (2), is implemented via a user-defined function with the following interface:

`x = internal_to_cartesian(q).`

Here, q is an array of length $3N - 6$ containing internal coordinate values, and x is an array of shape $(N, 3)$, containing the Cartesian coordinates of the N atoms in the molecule. Users can implement frame embeddings either directly within the `internal_to_cartesian` function or by applying predefined function decorators for common embeddings. For example, to enforce the Eckart frame, the `eckart` decorator can be applied, which rotates the coordinate system accordingly,

`eckart(q0, m)(internal_to_cartesian).`

Here, q_0 is an array of reference (e.g., equilibrium) internal coordinates, and m is an array of atomic masses. Similarly, to shift coordinates to the center of mass, users can apply the `com(m)(internal_to_cartesian)` decorator.

Below is an example of a coordinate mapping for a triatomic H₂O molecule using valence internal coordinates and the Eckart frame:

```

1  from jax import numpy as jnp
2  from vibrojet.eckart import eckart
3
4  # Masses of O, H, H atoms
5  masses = [15.9994, 1.00782505, 1.00782505]
6
7  # Equilibrium values of valence coordinates
8  r1, r2, alpha = 0.958, 0.958, 1.824
9  q0 = [r1, r2, alpha]
10
11 @eckart(q0, masses)
12 def valence_to_cartesian(q):
13     r1, r2, a = q
14     return jnp.array([
15         [0.0, 0.0, 0.0],
16         [r1*jnp.sin(a/2), 0.0, r1*jnp.cos(a/2)],
17         [-r2*jnp.sin(a/2), 0.0, r2*jnp.cos(a/2)],
18     ])
19
20 xyz0 = valence_to_cartesian(q0)

```

Given a coordinate mapping function, the kinetic energy G -matrix and pseudopotential U can be evaluated at specific internal

coordinate values or over a grid using functions from the keo module: `Gmat`, `batch_Gmat`, `pseudo`, and `batch_pseudo`. To compute the G -matrix at a single point,

```
g = Gmat(q, m, internal_to_cartesian).
```

Here, g is an array of shape $(3N, 3N)$ representing the full G -matrix. The first $3N - 6$ rows and columns correspond to vibrational coordinates, followed by three rotational and three translational coordinates. All values are given in units of cm^{-1} , assuming input bond distances are in \AA and angles in radians. For evaluating over a batch of points,

```
g = batch_Gmat(q, m, internal_to_cartesian).
```

In this case, q is a 2D array of shape $(D, 3N - 6)$, where D is the number of grid points. The output g has shape $(D, 3N, 3N)$. The functions `pseudo` and `batch_pseudo` follow the same interface and unit convention.

Below is an example of computing the G -matrix and U at a single point and over a grid using the coordinate mapping for the H_2O molecule defined above:

```
1 from vibrojet.keo import Gmat, batch_Gmat,
2   pseudo, batch_pseudo
3
4 masses = [15.9994, 1.00782505, 1.00782505]
5 r1, r2, alpha = 0.958, 0.958, 1.824
6
7 # compute G and U at single point
8 q0 = [r1, r2, alpha]
9 g = Gmat(q0, masses, valence_to_cartesian)
10 u = pseudo(q0, masses, valence_to_cartesian)
11
12 # compute G and U at grid points
13 q = jnp.array([
14     [r1, r2, alpha],
15     [r1 + 0.1, r2, alpha],
16     [r1, r2 + 0.1, alpha],
17     [r1, r2, alpha + 0.1],
18 ])
19
20 g = batch_Gmat(q, masses, valence_to_cartesian)
21 u = batch_pseudo(q, masses, valence_to_cartesian)
```

The Taylor series coefficients of a function can be computed using the `deriv_list` function from the `taylor` module. It takes the function to be differentiated, an expansion point, and a list of multi-indices specifying the desired derivative terms,

```
coeffs = deriv_list(func, x0, deriv_ind, if_taylor).
```

The boolean parameter `if_taylor` determines the output: if set to `True`, the function returns Taylor series coefficients; if set to `False`, it returns the corresponding partial derivatives. For example, to compute the Taylor expansion of the G -matrix for the H_2O molecule up to fourth order:

```
1 from vibrojet.taylor import deriv_list
2 import itertools
3
4 # Generate the list of multi-indices specifying
5 # the integer exponents for each coordinate
6 # in the Taylor series expansion
7 max_order = 4 # max total expansion order
8 deriv_ind = [
9     elem
10    for elem in itertools.product(*[range(0,
11        max_order + 1) for _ in range(len(q0))])
12    if sum(elem) <= max_order
13 ]
14
15 # Function for computing kinetic G-matrix
16 # for the given masses of atoms and coordinates
17 func = lambda x: Gmat(x, masses,
18     valence_to_cartesian)
19
20 g_coeffs = deriv_list(func, deriv_ind, q0,
21     if_taylor=True)
```

The output `g_coeffs` is an array where the first dimension indexes the corresponding derivative multi-indices in `deriv_ind`, and the second and third dimensions correspond to the rows and columns of the G -matrix. Each entry in `g_coeffs` gives the Taylor series coefficient of a G -matrix element associated with a specific multi-index, evaluated at the expansion point q_0 .

The `deriv_list` function is not specific to the KEO and can be employed to compute Taylor expansion coefficients for any user-defined multivariate function. For example, it can be used to expand PESs or dipole moment surfaces in the desired coordinate system.

V. EXAMPLES

We present variational calculations of the vibrational energies of formaldehyde (H_2CO) and ammonia (NH_3), using the KEO and PES represented by Taylor series expansions around a single reference configuration. These expansions are cast into an N -mode representation, and the convergence of the computed vibrational energies is examined with respect to both the Taylor expansion order and the N -mode truncation order of both the KEO and PES. Complete example calculations are available in the `examples` folder of the `Vibrojet` repository.

For molecules such as NH_3 , which exhibit multiple minima in the PES along one or more large-amplitude coordinates, the standard approach is the Hougen–Bunker–Johns method.³⁶ This method treats large-amplitude coordinates on a grid while constructing Taylor expansions for the remaining quasi-rigid coordinates at each grid point. However, it has been shown that reasonably accurate expansions of the KEO and PES for NH_3 can also be obtained using rectilinear coordinates by expanding around a reference configuration corresponding to the planar equilibrium geometry.³⁷ In our NH_3 example, we employ this alternative approach, treating the inversion coordinate within a Taylor expansion. We show that the PES, kinetic G -matrix, and pseudopotential can be efficiently expanded in terms of functions of individual internal

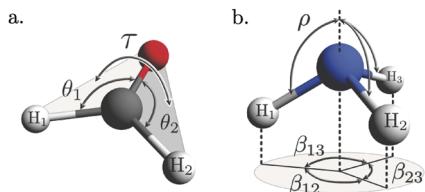


FIG. 1. Definition of the internal angular coordinates used in vibrational energy calculations for the (a) H_2CO and (b) NH_3 molecules.

curvilinear coordinates, such as Morse functions for stretching coordinates and trigonometric functions for bending coordinates, resulting in improved convergence of the Taylor series even for the inversion vibrational motion.

For H_2CO , the vibrational problem is solved using valence bond coordinates: r_1 , r_2 , and r_3 representing the C–O, C–H₁, and C–H₂ bond lengths, respectively; two bond angles, θ_1 and θ_2 ; and one dihedral angle, τ , as defined in Fig. 1(a). The G-matrix, pseudopotential, and PES, originally published in Ref. 38 and reimplemented in Python, are expanded in these coordinates around the equilibrium configuration: $r_1 = 1.2034 \text{ \AA}$, $r_2 = r_3 = 1.1038 \text{ \AA}$, $\theta_1 = \theta_2 = 121.84^\circ$, and $\tau = 180^\circ$.

For NH_3 , the vibrational problem is solved using internal coordinates: r_1 , r_2 , and r_3 represent the N–H₁, N–H₂, and N–H₃ bond lengths, respectively; $s_4 = (2\beta_{23} - \beta_{13} - \beta_{12})/\sqrt{6}$ and $s_5 = (\beta_{13} - \beta_{12})/\sqrt{2}$, which are symmetry-adapted combinations of bond angles; and ρ is the umbrella inversion coordinate, as defined in Fig. 1(b). Unlike H_2CO , we employ two sets of transformed coordinates, $y^{(G)}$ and $y^{(V)}$, for expanding the KEO (G-matrix and pseudopotential) and PES, respectively,

$$\begin{aligned} y_1^{(G)} &= r_1 - r_1^{(0)}, & y_1^{(V)} &= 1 - e^{-a_m(r_1 - r_1^{(0)})}, \\ y_2^{(G)} &= r_2 - r_2^{(0)}, & y_2^{(V)} &= 1 - e^{-a_m(r_2 - r_2^{(0)})}, \\ y_3^{(G)} &= r_3 - r_3^{(0)}, & y_3^{(V)} &= 1 - e^{-a_m(r_3 - r_3^{(0)})}, \\ y_4^{(G)} &= s_4, & y_4^{(V)} &= s_4, \\ y_5^{(G)} &= s_5, & y_5^{(V)} &= s_5, \\ y_6^{(G)} &= \cos \rho, & y_6^{(V)} &= \sin \rho. \end{aligned} \quad (10)$$

The KEO expansions are constructed using the $y^{(G)}$ coordinates around the reference configuration: $r_1^{(0)} = r_2^{(0)} = r_3^{(0)} = 1.0116 \text{ \AA}$, $s_4 = s_5 = 0$, and $\rho = 90^\circ$ (planar configuration). The PES, adapted from Ref. 39, is re-expanded in the $y^{(V)}$ coordinates around the same reference configuration, but with $\rho = 112.1^\circ$, corresponding to one of the two equivalent minima of the PES, and using the Morse exponent $a_m = 2.0 \text{ \AA}^{-1}$.

To demonstrate the efficiency of the Taylor expansion in the transformed $y^{(G)}$ and $y^{(V)}$ coordinates, we present in Figs. 2 and 3 one-dimensional slices of selected elements of the G-matrix and PES for NH_3 along the umbrella coordinate ρ . These slices, calculated exactly on a grid, are compared with fourth-order Taylor expansions in both the transformed coordinates and the original internal coordinates. The results show that expansions in $y^{(G)}$ and $y^{(V)}$ coordinates yield significantly higher accuracy than those in

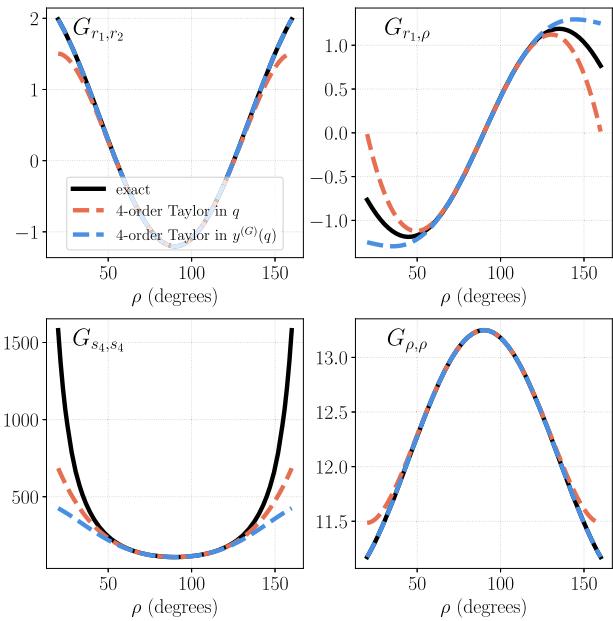


FIG. 2. One-dimensional slices of selected elements of the G-matrix (in cm^{-1}) for NH_3 , shown along the umbrella coordinate ρ . Exact values are compared with fourth-order Taylor series expansions around $\rho = 90^\circ$, expressed in both original internal coordinates $q = \{r_1, r_2, r_3, s_4, s_5, \rho\}$ and transformed coordinates $y^{(G)}(q)$ in (10).

the original internal coordinates at the same truncation order. This improvement is especially notable for the PES, where the expansion in internal coordinates around one of the minima fails to reproduce the characteristic double-well feature.

To solve the vibrational problem, we implemented a basis set contraction procedure that involves several steps of reduced-mode variational calculations (see, e.g., Ref. 40). In the first step, we defined a primitive one-dimensional basis set for each coordinate and computed matrix elements of the coordinate powers and momentum operators (i.e., x^t and $x^t \partial/\partial x$) appearing in the Taylor expansions

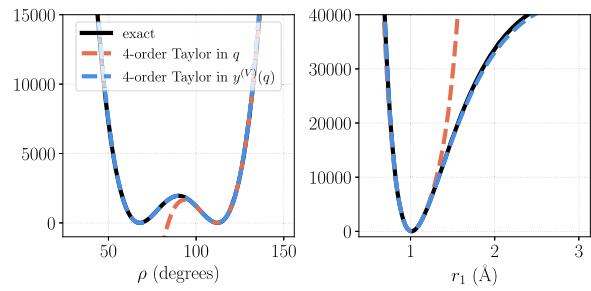


FIG. 3. One-dimensional slices of the PES (in cm^{-1}) for NH_3 , shown along the umbrella coordinate ρ and stretching coordinate r_1 . Exact values are compared with fourth-order Taylor series expansions around equilibrium $\rho = 112.1^\circ$ and $r_1 = 1.0116 \text{ \AA}$, expressed in both internal coordinates $q = \{r_1, r_2, r_3, s_4, s_5, \rho\}$ and transformed coordinates $y^{(V)}(q)$ in (10).

of the KEO and PES. We used Hermite functions as a primitive basis for all coordinates of both molecules. In the second step, one-dimensional (1D) reduced-mode Schrödinger equations were solved independently for each coordinate. Each equation used the primitive basis for the target coordinate with the Hamiltonian averaged over the remaining coordinates using the zero-order primitive basis functions. The resulting eigenfunctions were then truncated using an energy threshold (typically 40 000–60 000 cm^{-1}), producing six sets of 1D contracted basis functions, one for each coordinate.

Next, we constructed a combined basis for the equivalent stretching coordinates: r_2 and r_3 for H_2CO and r_1 , r_2 , and r_3 for NH_3 . This was done by solving reduced-mode Schrödinger equations using the previously obtained 1D contracted basis functions for the stretching coordinates. As in the 1D case, the Hamiltonian was averaged over the remaining coordinates using the ground-state contracted basis functions. The same procedure was applied in parallel to the bending coordinates, i.e., θ_1 and θ_2 for H_2CO and s_4 and s_5 for NH_3 . The resulting eigenfunctions were again truncated using an energy threshold ($30 000 \text{ cm}^{-1}$) and used to form new contracted basis sets for stretching and bending vibrations.

In the final step, the full vibrational basis was constructed as a direct product of contracted basis functions for all coordinates. To reduce the basis set size, we included only the product functions whose total energy falls below a specified threshold. The total energy is estimated as a sum of the individual energies of the contracted basis functions. The resulting full-dimensional vibrational Schrödinger equation was then solved to obtain the vibrational energies and wavefunctions. Since our goal is to demonstrate the convergence behavior of the KEO and PES Taylor expansions, we employed relatively modest basis sets by setting the energy threshold for the product basis at $12 000 \text{ cm}^{-1}$ for both molecules.

In Figs. 4 and 5, we show the convergence of the first 100 vibrational energy levels of H_2CO and NH_3 with respect to the truncation order of the N -mode expansion for the kinetic G -matrix and pseudopotential (a) and the PES (b). The results are presented as absolute

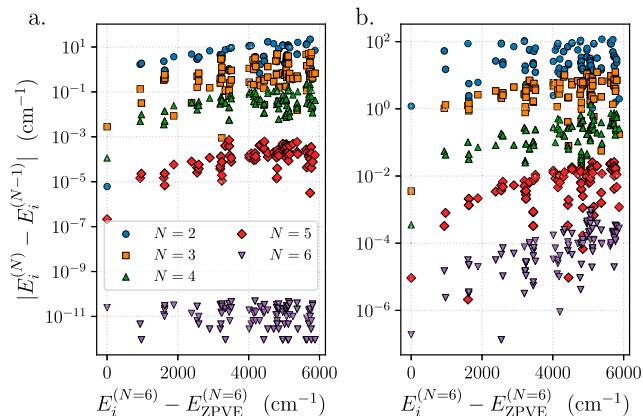


FIG. 5. Convergence of the KEO and PES N -mode expansions for the first 100 vibrational energy levels of NH_3 . Plotted are the absolute differences between vibrational energies E_i ($i = 1 \dots 100$) computed using N and $N - 1$ expansion orders, for $N = 2 \dots 6$, shown as functions of the corresponding vibrational energies relative to the zero-point energy. Panel (a) shows convergence for the G -matrix and pseudopotential, and panel (b) shows convergence for the PES.

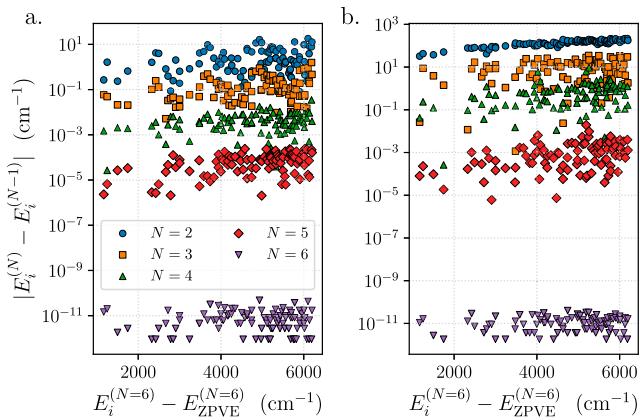


FIG. 4. Convergence of the KEO and PES N -mode expansions for the first 100 vibrational energy levels of H_2CO . Plotted are the absolute differences between vibrational energies E_i ($i = 1 \dots 100$) computed using N and $N - 1$ expansion orders, for $N = 2 \dots 6$, shown as functions of the corresponding vibrational energies relative to the zero-point energy. Panel (a) shows convergence for the G -matrix and pseudopotential, and panel (b) shows convergence for the PES.

energy differences between calculations performed with N and $N - 1$ truncation levels, plotted against the energy values relative to the zero-point energy. In all calculations, the maximum order of the Taylor series expansions for both the KEO and PES was fixed at 8th order. When testing the N -mode convergence of the G -matrix and pseudopotential, the PES expansion was held fixed at eighth order, and vice versa. The results for both molecules demonstrate fast convergence of the N -mode expansion for the KEO operator and a bit slower convergence for the PES. Overall, a truncation at $N = 4$ is sufficient to achieve sub-wavenumber accuracy for the lowest 100 vibrational states.

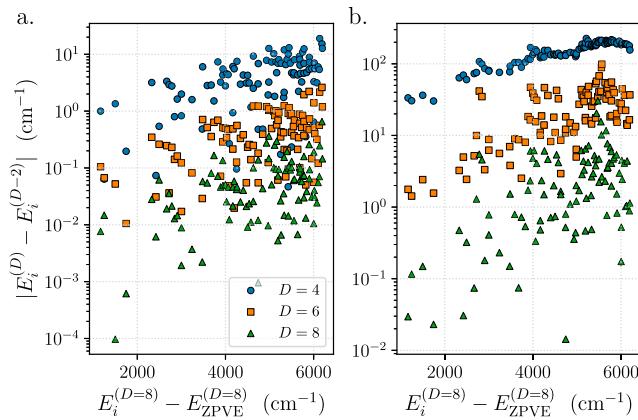


FIG. 6. Convergence of the KEO and PES Taylor series expansions for the first 100 vibrational energy levels of H_2CO . Plotted are the absolute differences between vibrational energies E_i ($i = 1 \dots 100$) computed using Taylor truncation orders D and $D - 2$, for $D = 4, 6, 8$, shown as functions of the corresponding vibrational energies relative to the zero-point energy. Panel (a) shows convergence for the G -matrix and pseudopotential, and panel (b) shows convergence for the PES.

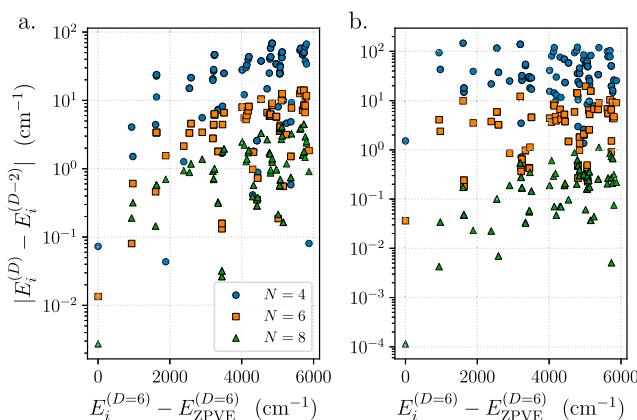


FIG. 7. Convergence of the KEO and PES Taylor series expansions for the first 100 vibrational energy levels of NH_3 . Plotted are the absolute differences between vibrational energies E_i ($i = 1 \dots 100$) computed using Taylor truncation orders D and $D - 2$, for $D = 4, 6, 8$, shown as functions of the corresponding vibrational energies relative to the zero-point energy. Panel (a) shows convergence for the G-matrix and pseudopotential, and panel (b) shows convergence for the PES.

The convergence of vibrational energies for H_2CO and NH_3 with respect to the truncation order of the Taylor series expansion is shown in Figs. 6 and 7, respectively. The plots present the absolute energy differences between calculations using Taylor expansions of orders D and $D - 2$, plotted against the corresponding energy values relative to the zero-point energy. For H_2CO , convergence of the KEO is achieved within 1 cm^{-1} at the sixth expansion order, with the difference between $D = 8$ and $D = 6$ being less than 1 cm^{-1} . For NH_3 , the KEO converges more slowly, which we attribute to the choice of bending coordinates β_{ij} [see Fig. 1(b)], where higher-order terms are necessary for accurate expansions. The PES expansion for NH_3 converges faster than for H_2CO due to the use of transformed coordinates $y^{(V)}$ in (10), particularly the use of Morse functions for stretching coordinates.

VI. CONCLUSIONS

We presented Vibrojet, a Python implementation of a general framework for constructing rovibrational KEOs and PESs for arbitrary molecules, using user-defined internal coordinate systems and frame embedding conditions. The framework supports efficient evaluation of rovibrational operators either on grids of internal coordinates or as truncated Taylor series expansions.

The implementation leverages Taylor-mode automatic differentiation capabilities from the JAX library, specifically its jet module, and includes extensions tailored to the specific needs associated with rovibrational operator construction. These include, for example, an efficient Taylor expansion of the KEO in the Eckart frame embedding.

We demonstrated the utility of the framework through variational calculations of molecular vibrational energies based on Taylor (N -mode) expansions of the KEO and PES. The results confirm good convergence behavior of the computed energies with respect to the expansion order.

The present approach, based on Taylor series expansion, can be combined with traditional least-squares fitting methods to balance the accuracy and the computational cost of high-order expansions. For example, Taylor expansions can be performed at multiple reference points, and fitting approaches can then be used to construct a unified representation by interpolating the corresponding derivatives.

ACKNOWLEDGMENTS

We thank Benoit Richard for proofreading this manuscript and for his helpful suggestions. The work of E.V. was supported by the European Union's Horizon Europe research and innovation program under Marie Skłodowska-Curie Grant Agreement No. 101155136. The work of A.F.C. was supported by the Data Science in Hamburg HELMHOLTZ Graduate School for the Structure of Matter (DASHH, HIDSS-0002).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Andrey Yachmenev: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Project administration (lead); Resources (lead); Software (lead); Supervision (lead); Validation (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (lead). **Emil Vogt:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Resources (equal); Validation (equal); Visualization (equal); Writing – review & editing (equal). **Álvaro Fernández Corral:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Resources (equal); Validation (equal); Visualization (equal); Writing – review & editing (equal). **Yahya Saleh:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Resources (equal); Validation (equal); Visualization (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available within the article and through the following repository: <https://github.com/robochimps/vibrojet>.

REFERENCES

- 1 K. Oenen, D. F. Dinu, and K. R. Liedl, "Determining internal coordinate sets for optimal representation of molecular vibration," *J. Chem. Phys.* **160**, 014104 (2024).
- 2 M. Schneider and G. Rauhut, "Comparison of curvilinear coordinates within vibrational structure calculations based on automatically generated potential energy surfaces," *J. Chem. Phys.* **161**, 094102 (2024).

³Y. Saleh, Á. Fernández Corral, E. Vogt, A. Iske, J. Küpper, and A. Yachmenev, “Computing excited states of molecules using normalizing flows,” *J. Chem. Theory Comput.* **21**, 5221 (2025).

⁴E. Vogt, Á. F. Corral, Y. Saleh, and A. Yachmenev, “Transferability and interpretability of vibrational normalizing-flow coordinates,” [arXiv:2502.15750](https://arxiv.org/abs/2502.15750) [physics] (2025).

⁵S. N. Yurchenko, W. Thiel, and P. Jensen, “Theoretical ROVibrational Energies (TROVE): A robust numerical approach to the calculation of rovibrational energies for polyatomic molecules,” *J. Mol. Spectrosc.* **245**, 126 (2007).

⁶E. Mátyus, G. Czakó, and A. G. Császár, “Toward black-box-type full- and reduced-dimensional variational (ro)vibrational computations,” *J. Chem. Phys.* **130**, 134112 (2009).

⁷X.-G. Wang and T. Carrington, “A discrete variable representation method for studying the rovibrational quantum dynamics of molecules with more than three atoms,” *J. Chem. Phys.* **130**, 094101 (2009).

⁸Q. Zhang, R.-S. Wang, and L. Wang, “Neural canonical transformations for vibrational spectra of molecules,” *J. Chem. Phys.* **161**, 024103 (2024).

⁹G. Avila and T. Carrington, “Solving the Schrödinger equation using Smolyak interpolants,” *J. Chem. Phys.* **139**, 134114 (2013).

¹⁰J. C. Light and T. Carrington, “Discrete-variable representations and their utilization,” *Adv. Chem. Phys.* **114**, 263 (2000).

¹¹J. Tennyson, M. A. Kostin, P. Barletta, G. J. Harris, O. L. Polyansky, J. Ramanlal, and N. F. Zobov, “DVR3D: A program suite for the calculation of rotation-vibration spectra of triatomic molecules,” *Comput. Phys. Commun.* **163**, 85 (2004).

¹²S. N. Yurchenko and T. M. Mellor, “Treating linear molecules in calculations of rotation-vibration spectra,” *J. Chem. Phys.* **153**, 154106 (2020).

¹³M. Mladenović, “Rovibrational Hamiltonians for general polyatomic molecules in spherical polar parametrization. I. Orthogonal representations,” *J. Chem. Phys.* **112**, 1070 (2000).

¹⁴X.-G. Wang and T. Carrington, “A simple method for deriving kinetic energy operators,” *J. Chem. Phys.* **113**, 7097 (2000).

¹⁵F. Gatti, C. Muñoz, and C. Iung, “A general expression of the exact kinetic energy operator in polyspherical coordinates,” *J. Chem. Phys.* **114**, 8275 (2001).

¹⁶D. W. Schwenke, “New rovibrational kinetic energy operators using polyspherical coordinates for polyatomic molecules,” *J. Chem. Phys.* **118**, 10431 (2003).

¹⁷A. Yachmenev and S. N. Yurchenko, “Automatic differentiation method for numerical construction of the rotational-vibrational Hamiltonian as a power series in the curvilinear internal coordinates using the Eckart frame,” *J. Chem. Phys.* **143**, 014105 (2015).

¹⁸D. Peláez and H.-D. Meyer, “The multigrid POTFIT (MGPF) method: Grid representations of potentials for quantum dynamics of large systems,” *J. Chem. Phys.* **138**, 014108 (2013).

¹⁹B. Ziegler and G. Rauhut, “Efficient generation of sum-of-products representations of high-dimensional potential energy surfaces based on multimode expansions,” *J. Chem. Phys.* **144**, 114114 (2016).

²⁰R. I. Ovsyannikov, P. Jensen, M. Y. Tretyakov, and S. N. Yurchenko, “On the use of the finite difference method in a calculation of vibration-rotation energies,” *Opt. Spectrosc.* **107**, 221–227 (2009).

²¹A. Griewank, J. Utke, and A. Walther, “Evaluating higher derivative tensors by forward propagation of univariate Taylor series,” *Math. Comput.* **69**, 1117 (2000).

²²S. Walter, “Structured higher-order algorithmic differentiation in the forward and reverse mode with application in optimum experimental design,” Ph.D. thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2012.

²³G. O. Sørensen, “A new approach to the Hamiltonian of nonrigid molecules,” in *Large Amplitude Motion in Molecules II* (Springer, 1979), pp. 97–175.

²⁴V. V. Rybkin, U. Ekström, and T. Helgaker, “Internal-to-Cartesian back transformation of molecular geometry steps using high-order geometric derivatives,” *J. Comput. Chem.* **34**, 1842 (2013).

²⁵A. Y. Dymarsky and K. N. Kudin, “Computation of the pseudorotation matrix to satisfy the Eckart axis conditions,” *J. Chem. Phys.* **122**, 124103 (2005).

²⁶S. V. Krasnoshchekov, E. V. Isayeva, and N. F. Stepanov, “Determination of the Eckart molecule-fixed frame by use of the apparatus of quaternion algebra,” *J. Chem. Phys.* **140**, 154104 (2014).

²⁷A. Yachmenev (2025). “Problem with Eckart frame solution for methane using quaternion method,” *GitHub*. <https://github.com/robochimps/vibrojet/issues/4>

²⁸C. Moler and C. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,” *SIAM Rev.* **45**, 3 (2003).

²⁹A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed. (Society for Industrial and Applied Mathematics, 2008).

³⁰A. Griewank, D. Juedes, and J. Utke, “Algorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++,” *ACM Trans. Math. Software* **22**, 131 (1996).

³¹L. Benet and D. Sanders, “TaylorSeries.jl: Taylor expansions in one and several variables in Julia,” *J. Open Source Software* **4**, 1043 (2019).

³²J. Bettencourt, M. J. Johnson, and D. Duvenaud, “Taylor-mode automatic differentiation for higher-order derivatives in JAX,” in Program Transformations for ML Workshop at NeurIPS 2019, 2019.

³³J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang (2018). “JAX: Composable transformations of Python + NumPy programs,” *GitHub*. <https://github.com/jax-ml/jax>

³⁴T. Mach and M. A. Freitag, “Solving the parametric eigenvalue problem by Taylor series and Chebyshev expansion,” *SIAM J. Matrix Anal. Appl.* **46**, 957 (2025).

³⁵J. M. Bowman, T. Carrington, and H.-D. Meyer, “Variational quantum approaches for computing vibrational energies of polyatomic molecules,” *Mol. Phys.* **106**, 2145 (2008).

³⁶J. T. Hougen, P. R. Bunker, and J. W. C. Johns, “The vibration-rotation problem in triatomic molecules allowing for a large-amplitude bending vibration,” *J. Mol. Spectrosc.* **34**, 136 (1970).

³⁷M. Neff and G. Rauhut, “Towards black-box calculations of tunneling splittings obtained from vibrational structure methods based on normal coordinates,” *Spectrochim. Acta, Part A* **119**, 100 (2014).

³⁸A. F. Al-Refaie, A. Yachmenev, J. Tennyson, and S. N. Yurchenko, “ExoMol line lists–VIII. A variationally computed line list for hot formaldehyde,” *Mon. Not. R. Astron. Soc.* **448**, 1704 (2015).

³⁹O. L. Polyansky, R. I. Ovsyannikov, A. A. Kyuberis, L. Lodi, J. Tennyson, A. Yachmenev, S. N. Yurchenko, and N. F. Zobov, “Calculation of rotation-vibration energy levels of the ammonia molecule based on an *ab initio* potential energy surface,” *J. Mol. Spectrosc.* **327**, 21 (2016).

⁴⁰S. N. Yurchenko, R. J. Barber, A. Yachmenev, W. Thiel, P. Jensen, and J. Tennyson, “A variationally computed $T = 300$ K line list for NH_3 ,” *J. Phys. Chem. A* **113**, 11845 (2009).