SpectrumSDT: a program for parallel calculation of coupled rotational-vibrational energies and lifetimes of bound states and scattering resonances in triatomic systems

Igor Gayday,[*] Alexander Teplukhin[†], Jonathan Moussa[‡] and Dmitri Babikov[**]

[*]*Department of Chemistry, Wehr Chemistry Building, Marquette University, Milwaukee, Wisconsin 53201-1881, USA*

[†]*Theoretical Division (T-1, MS B221), Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

[‡]*Molecular Sciences Software Institute, Blacksburg, VA 24060, USA*

Abstract

We present SpectrumSDT – a program for calculations of energies and lifetimes of bound rotational-vibrational states below and scattering resonances above the dissociation threshold on a global potential energy surface of a triatomic system, which may include stable molecules, weekly-bound van-der-Waals complexes, and unbound atom + diatom scattering systems. Large-amplitude vibrational motion is treated explicitly using hyper-spherical coordinates. Three options for the rotational-vibrational interaction are supported: uncoupled (symmetric top rotor), partially coupled (to include interaction between several nearest states only) and full-coupled (vibrating asymmetric-top rotor). In addition to energies and lifetimes, SpectrumSDT is able to integrate ro-vibrational wave functions over the user-defined regions of potential energy surface, which helps to classify these states. In this release of the code, SpectrumSDT is limited to ABA-type molecules with wave functions that do not extend into the regions near Eckart singularities.

[*] Author to whom all correspondence should be addressed; electronic mail: dmitri.babikov@mu.edu

Program summary

Program title: SpectrumSDT

Developer's repository link: https://github.com/IgorGayday/SpectrumSDT

Licensing provisions: GNU General Public License 3 (GPL)

Programming language: Fortran

Nature of problem: Calculations of energies and lifetimes of bound rotational-vibrational states below and scattering resonances above the dissociation threshold on a global potential energy surface of a triatomic system, which may include stable molecules, weekly-bound van-der-Waals complexes, and unbound atom + diatom scattering systems.

Solution method: A Hamiltonian matrix is built in APH coordinates using an optimal 2D basis set, adjusted for a given problem. A complex absorbing potential (CAP) is added to define a boundary condition for calculation of scattering resonances above the dissociation threshold. The eigenstates of the Hamiltonain matrix are found using a state-of-the-art iterative eigensolver.

Restrictions: The present version is restricted to ABA-molecules and wave functions that do not extend into linear and equilateral triangle configurations.

Unusual features: Probabilities and lifetimes of the wave functions can be calculated in user-defined regions on the PES, which allows to analyze their localization properties (i.e. automatic isotopomer assignment) and obtain channel-specific lifetimes for scattering resonances.

I. THEORY

This section contains a brief overview of the most relevant theoretical aspects of SpectrumSDT for a lay user. For a more complete description the readers are encouraged to refer to other recent publications, where all theory aspects have been covered in detail.[1–4]

A. Adiabatically adjusting Principal axis Hyperspherical (APH) coordinates

SpectrumSDT operates in APH coordinates,[5] which allow to treat all dissociation channels on equal footing. Molecular geometry in the APH coordinates is described with hyper-radius ρ and two hyper-angles θ and φ. Qualitatively ρ serves as a measure of the overall "size" of the bounding triangle for the system (the "breathing" motion); θ changes in the range from 0 to $\pi/2$ and corresponds to translation from equilateral triangle at $\theta = 0$ to a linear molecule at $\theta = \pi/2$ (the "bending" motion); φ changes in the range from 0 to 2π and is responsible for isomerization, such that $\varphi = \pi/3$, π and $5\pi/3$ correspond to AAB, ABA and BAA arrangements respectively, as shown in Figure 1. See also Ref. 6 for an interactive visualization tool of the motion along φ and Ref. 7 for 3D representation of APH coordinates on the example of ozone.

B. Basis sets and Sequential Diagonalization Truncation (SDT) procedure

On the lowest level, SpectrumSDT uses discrete variable representation (DVR) for ρ and θ coordinates (basically, a grid of points) and variational basis representation (VBR) of cosines or sines functions for φ, which enforces either symmetric or antisymmetric property of wave functions with respect to $\varphi = 0$ and $\varphi = \pi$, required for ABA-molecules in the APH coordinates. The two symmetries are independent, so the calculations are split into two separate runs for each symmetry.

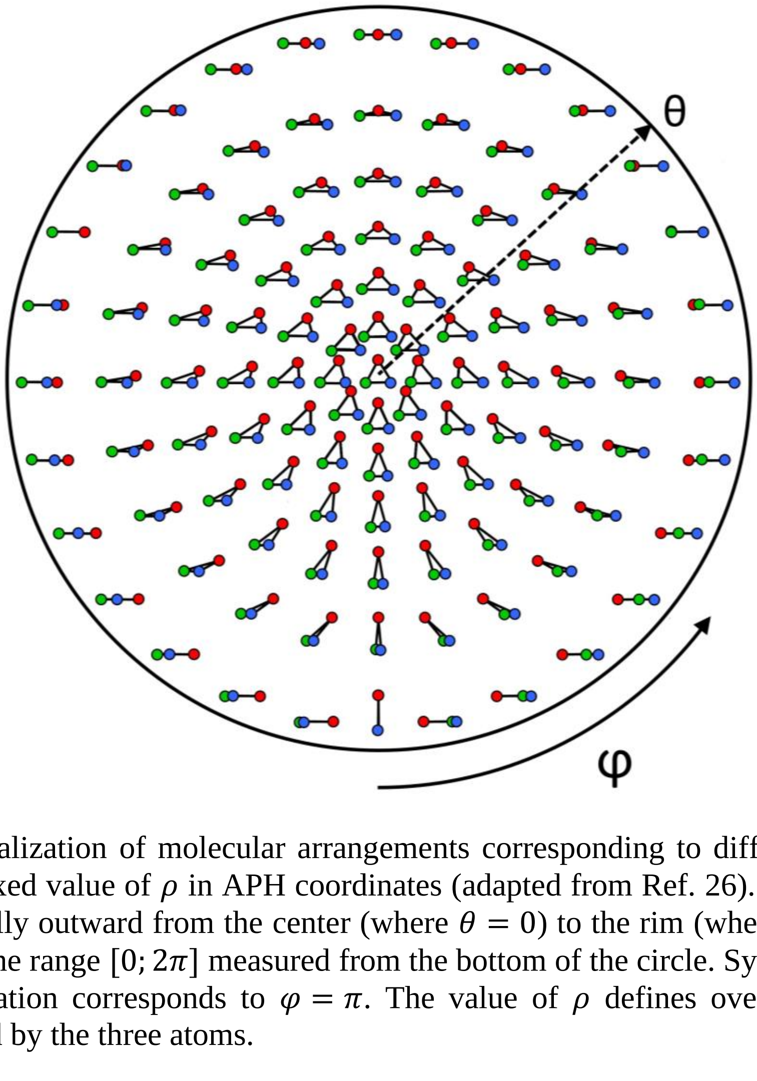

Figure 1: Visualization of molecular arrangements corresponding to different values of (θ, φ) and a fixed value of ρ in APH coordinates (adapted from Ref. 26). The value of θ increases radially outward from the center (where $\theta = 0$) to the rim (where $\theta = \pi/2$). φ is an angle in the range $[0; 2\pi]$ measured from the bottom of the circle. Symmetric obtuse ABA configuration corresponds to $\varphi = \pi$. The value of ρ defines overall size of the triangle formed by the three atoms.

SpectrumSDT constructs a hierarchy of progressively more optimal basis sets through the procedure known as Sequential Diagonalization Truncation (SDT).[8,9] In short, SDT starts with solving a one-dimensional Schrodinger equation, where all couplings with other coordinates are not included, in each of the 1D-slices along φ at every combination of grid points in ρ and θ. The obtained 1D-solutions are used as a locally optimal basis set to solve 2D problem in φ and θ, where the additional couplings are introduced, and the basis set is further optimized (sequential diagonalization). Finally, the 2D solutions are used as a basis set for the overall 3D problem, with all couplings taken into account. Of course keeping all the solutions from 1D and 2D levels would result in unnecessarily large basis, so only the solutions with energies less than a specified convergence parameter E_{cut} are retained in the basis set (truncation).

C. Hamiltonian operator in APH coordinates and Eckart singularities

The Hamiltonian operator in APH coordinates can be written as:[3]

$$\hat{H} = \hat{T}_\rho + \hat{T}_\theta + \hat{T}_\varphi + V_{\text{pes}} + V_{\text{ext}} + \hat{T}_{\text{sym}} + \hat{T}_{\text{asym}} + \hat{T}_{\text{cor}} \,, \tag{1}$$

where the individual terms are given by:

$$\hat{T}_\rho = -\frac{\hbar^2}{2\mu}\frac{\partial^2}{\partial\rho^2} \,, \tag{2}$$

$$\hat{T}_\theta = -\frac{2\hbar^2}{\mu\rho^2}\frac{\partial^2}{\partial\theta^2} \,, \tag{3}$$

$$\hat{T}_\varphi = -\frac{2\hbar^2}{\mu\rho^2 \sin^2\theta}\frac{\partial^2}{\partial\varphi^2} \,, \tag{4}$$

$$V_{\text{ext}} = -\frac{\hbar^2}{2\mu\rho^2}\left(\frac{1}{4} + \frac{4}{\sin^2 2\theta}\right) , \tag{5}$$

$$\hat{T}_{\text{sym}} = \frac{A+B}{2}\hat{J}^2 + \left(C - \frac{A+B}{2}\right)\hat{J}_z^2 \,, \tag{6}$$

$$\hat{T}_{\text{asym}} = \frac{A-B}{2}\left(\hat{J}_x^2 - \hat{J}_y^2\right) , \tag{7}$$

$$\hat{T}_{\mathrm{cor}} = 4B\,cos\,\theta\left(i\hbar\frac{\partial}{\partial\varphi}\right)\hat{J}_y\,, \tag{8}$$

and V_{pes} is the potential energy term.

The rotational constants A, B and C in Eqs. (6-8) are given by:

$$A^{-1} = \mu\rho^2(1+\sin\theta)\,, \tag{9}$$

$$B^{-1} = 2\mu\rho^2\sin^2\theta\,, \tag{10}$$

$$C^{-1} = \mu\rho^2(1-\sin\theta)\,, \tag{11}$$

and μ is the reduced mass of the triatomic system, defined as:

$$\mu = \sqrt{\frac{m_1m_2m_3}{m_1+m_2+m_3}}\,, \tag{12}$$

One can see that $\hat{T}_\varphi \to -\infty$ as $\theta \to 0$ in Eq. (4) and $V_{\mathrm{ext}} \to -\infty$ as $\theta \to 0$ or $\pi/2$ in Eq. (5). These are Eckart singularities corresponding to the equilateral configuration at $\theta = 0$ and the linear configuration at $\theta = \pi/2$. While this problem is in principle solvable within existing framework by using a special basis set,[10] the current release of SpectrumSDT does not implement this feature yet, therefore if a wave function for a particular molecule is allowed to have a significant probability near $\theta = 0$ or $\pi/2$, the solution is not reliable. This issue will be addressed in future releases.

Fortunately, in many cases this issue is naturally avoided. First, many covalently bound triatomic molecules with *sp*^{2} hybridization, such as O_3, naturally have a highly repulsive PES in the regions of Eckart singularities, which prevents the vibrational wave function from reaching there. Second, the rotational potential term $\hat{T}_{\mathrm{sym}}$, given by Eq. (6), approaches $+\infty$ in the same regions due to the B and C terms in Eqs. (10) and (11). This extra potential "shields" Eckart singularities at sufficiently high values of rotational excitation (J, Λ), even if the PES does not. For both of these cases our code is able to give accurate results.

D. Hamiltonian matrix structure

The rotational state of the basis functions used to build the Hamiltonian matrix can be either *fixed* or *adiabatic*.[4] In the *adiabatic* case, a separate basis set is used for each value of the symmetric top quantum number Λ (projection of total angular momentum onto *z*-axis) that is going to be included in the Hamiltonian matrix for a given value of total angular momentum quantum number *J*. This creates a better suited basis set, since it experiences the same rotational potential as a target wave function. This approach is relatively standard and often found in the literature,[11–13] but it requires computing many different basis sets and, most importantly, the corresponding overlaps for all values of *J* and Λ, which can be numerically demanding.

In the alternative *fixed* option, the basis set with one (representative) combination of *J* and Λ is used to describe wave functions with arbitrary values of *J* and Λ (a similar idea has been utilized in the two-step approach of Ref. 14). This reduces the computational cost of both *basis* and *overlaps* calculation stages, but creates a less efficient basis set, therefore a higher value of the *cut-off energy* (E_{cut}) may be needed. Recently we demonstrated that this approach is more efficient.[4] However, one has to be careful when handling a molecule with wave functions reaching the ends of θ-grid. In this case, using a basis set with high values of *J* and Λ relative to a target wave function, will exclude basis functions near the ends of the θ-grid (due to the effect of rotational potential), which may result in less accurate representation of the target wave function in that area.

The rotational terms in Eqs. (6-8) translate to the Hamiltonian matrix structure shown in Figure 2. In the cases when couplings between different Λ blocks can be neglected, the overall matrix can be split into the diagonal Λ blocks, which significantly simplifies the process of solving the eigenvalue problem. This is known as symmetric-top rotor approximation. As a compromise between speed and accuracy, when the effects of ro-vibrational coupling cannot be neglected, it is also possible to partially take it into account, by including only a certain number of adjacent blocks for each value of Λ. For example, including up to five blocks, within the $\Lambda \pm 2$ range, permits to capture most of the rotational-vibrational interaction.

E. Optimized ρ-grid (optional)

Suppose we have a 1D potential $V(\rho)$ and we want to place points along ρ-coordinate in an optimal way to represent wave functions. The idea is to make grid spacing in ρ proportional to the local de Broglie wavelength $\lambda(V(\rho))$. This way one can accurately capture wave function

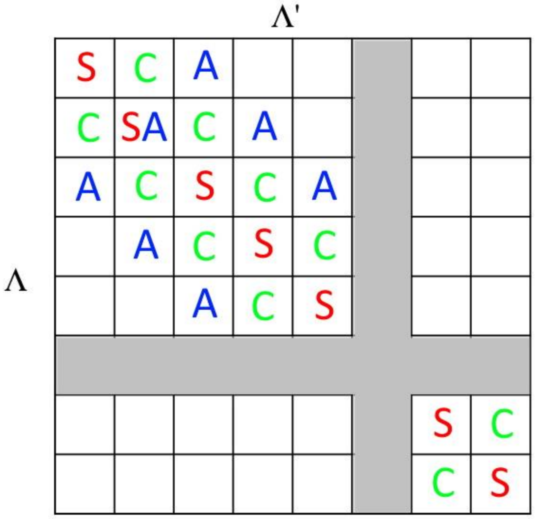

Figure 2: Rotational block structure of the Hamiltonian matrix. Letters S, A and C indicate contributions from symmetric-top rotor, asymmetric-top rotor and Coriolis coupling terms, given in Eqs. (6-8) respectively. Other blocks of the matrix are zero. The blocks are labelled by Λ and Λ', the value of projection of total angular momentum J onto z-axis.

oscillations, without placing too many points in the regions where wave periods are large. To achieve this, one can define an auxiliary grid x, where the distance is unitless and a given point $x = t$ corresponds to t oscillations of a wave with total energy E_{max} in the original space (ρ), as measured from a starting point ρ_0. The points in x can be mapped to ρ and vice versa, therefore if one has an equidistant grid in x, its mapping onto ρ would be an optimal grid with a fixed number of points per oscillation of the wave. Assuming the step size in x is $\alpha/2$, one can derive the following differential equation that relates the two grids:[1,15]

$$\frac{d\rho}{dx} = \frac{\alpha}{2}\lambda\big(V(\rho)\big) = \frac{\alpha}{2}\frac{h}{\sqrt{2\mu(E_{max} - V(\rho))}} \tag{13}$$

where factor of 2 in the denominator is chosen to make $\alpha = 1$ correspond to 2 points per period – the smallest meaningful number. This way the values of α range from 0 to 1 and control the density of the resulting optimal grid (smaller values of α generate denser grids). The exact value of α for a given problem is a convergence parameter. The derivative $d\rho/dx$ in Eq. (13) is further referred to as Jacobian of the grid ρ.

In general, any function smaller than the actual potential can be used as $V(\rho)$ in Eq. (13). Using such potential "envelopes" can be helpful to increase density of points in selected regions or improve integration accuracy if the envelope function is analytical.

II. PRACTICAL CONSIDERATIONS

A. General

The workflow of the program is separated into the following stages (runs): `grids`, `potential`, `basis`, `overlaps`, `eigensolve` and `properties`. Using multiple stages allows to utilize processors efficiently by separating computationally expensive sections with different degree of parallelizability. The stages have to be launched sequentially one after another from their respective folders.

The folder structure for each calculation may consist of up to 4 levels (see Figure 3). On the first level, there are subfolders for each value of Λ (referred to as more traditional *"K"* in the code, parameters, file names and further in the text to avoid confusion), as well as separate folders for each range of *K* that a user wants to consider for calculation of eigenpairs. The range that includes all values of *K* for a given *J* is referred to with a keyword `all`. The next two levels in a *K* range folder identify the values of parity and wave function symmetry corresponding to the calculation within. The folders corresponding to a scalar value of *K* skip the level corresponding to parity since parity only has the effect on calculations involving more than one value of *K*.[3] The last level is divided into folders corresponding to stages from `basis` to `properties`. `Grids` and `potential` stages do not have a folder within a particular calculation since they can be

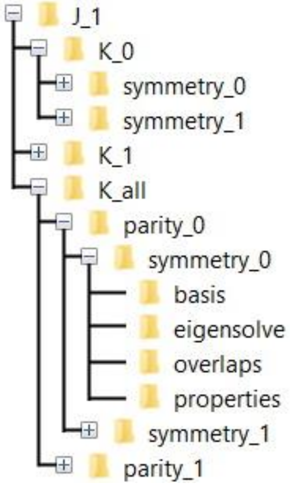

Figure 3: An example of root folder structure for *J* = 1. Inner structure of the folders marked by "+" is not shown for clarity. All folders on the same level of hierarchy have identical inner structures, except *K*-folders corresponding to a singular value of *K* skip the *parity* level.

shared across multiple calculations. The output files from each stage are stored either in the stage folder itself, or in `out_stagename` subfolder within the stage folder.

The stage folders all need to have a copy of a configuration file with the values of K, parity, symmetry, and stage corresponding to a given subfolder. A python script `init_spectrumsdt_folders.py` in `scripts` folder is available to generate the correct folder structure, copy a given template configuration file to all target subfolders and fill out placeholder values corresponding to each folder for a given value of *K*. Run `init_spectrumsdt_folders.py --help` for more details. To keep the parameters consistent between stages, we do not recommend manually changing the configuration files generated by `init_spectrumsdt_folders.py` in the stage subfolders.

The configuration file has to be named `spectrumsdt.config` and has to be present in every stage folder. Each parameter is assigned in the format *key* = *value*. Both keys and values are case-sensitive, and each assignment has to start on a new line and finish on the same line, except for parameter groups. The order of parameters, number of spaces, empty lines, or characters after the comment mark "!" do not matter. In the case of parameter groups, the open parenthesis "(" has to be present on the same line after the assignment character "=", followed by assignment of individual parameters of the group on the following lines and ending with the close parenthesis ")" on a separate line. Some examples of input files can be found in `config_examples` folder.

All stages except `grids` support parallel execution with an arbitrary number of processes, implemented via message passing interface (MPI).

B. Grids

The purpose of this first stage is to generate grids for all APH coordinates. Even though the basis set for φ is not a DVR, a φ-grid is still generated for numerical integration of the basis functions in φ. The grids for ρ and θ are described with the parameter groups grid_rho and grid_theta, which include at least the following parameters (determined by convergence studies):

- from – the left border of the grid,
- to – the right border of the grid,
- step – grid step,
- num_points – the total number of points in the specified interval.

Specifying step automatically defines num_points and vice versa, so these parameters cannot be specified together. All parameters with units are specified in Bohr for grid_rho and in degrees for grid_theta. The φ-grid is always defined in the range from 0 to 2π, so only the number of points is specified via num_points_phi key.

After this stage is completed, the three grid files corresponding to each coordinate are generated. The grid files are written in the following format: the first line specifies the values of from, to, step and num_points that were used to generate this grid. The next num_points lines specify the coordinate of each point and, in case of ρ-grid, Jacobian value at that point. Jacobians are meaningful only for optimized grids and are always equal to 1 for equidistant grids. All grid points are placed in the centers of their respective intervals. All values with units in the grid files are specified in Bohr for ρ-grid and radians for θ- and φ-grids.

Advanced parameters.

In the calculations of scattering resonances one typically needs to set up a long-range grid in ρ to describe the dissociation region. In these cases, using an equidistant grid leads to unnecessarily dense points in the dissociative region. One can reduce the cost of such calculations by optimizing the grid step as described in Section I-E.

Enabling optimization procedure for ρ-grid requires specifying grid parameter `optimal = 1`, alongside with the necessary information to define Eq. (13), which includes the following grid parameters:

- `envelope_path` – the path to the file defining the envelope function $V(\rho)$. The function is defined by specifying its values at the grid points. Each row contains 2 numbers: ρ in Bohr and $V(\rho)$ in Hartree. The grid on which $V(\rho)$ is defined does not have to be the same as the ρ-grid for the eigenvalue problem, which allows to re-use the same envelope function for any ρ-grid. As a rule of thumb, it is recommended to use minimum energy path (MEP) along ρ in a given PES as the envelope function $V(\rho)$.

A python script `extract_MEP.py` is available in the `scripts` folder to automatically generate minimum energy path along ρ. The script takes no arguments and has to be invoked in the folder with calculated `pes.out` and grid files. The extracted MEP is written to a file named `MEP_rho.dat`. Note that the MEP does not have to be calculated precisely, since precision of MEP does not affect the precision of subsequent calculations. A rough MEP may generate slightly suboptimal grid spacing, but the final grid density is a convergence parameter, controlled by `step`.

In the program, the points specified in the MEP file are used to build a cubic spline for the right branch ($\rho > \rho_0$) of $V(\rho)$, and fit analytical Eckart potential for the left branch ($\rho < \rho_0$). Possible oscillations of the cubic spline can easily be fixed by changing MEP grid density, which can be done for free since it does not affect complexity of the overall problem. Moreover, as outlined in the previous paragraph, even if some oscillations are present, they will not alter accuracy of calculations.

- max_energy – the value of E_{max} in wave numbers. This specifies the maximum total energy considered for a particle moving in the specified envelope potential $V(\rho)$. E_{max} has to be larger than $V(\rho)$ for any value of ρ. Local density of grid points is proportional to $E_{max} - V(\rho)$, therefore E_{max} can be considered as a tool to control balance of points between the covalent well and the dissociation region. In the limiting case when E_{max} is equal to the asymptotic value of $V(\rho)$ in the dissociation region, all the grid points would be placed in the covalent well region and little or none in the dissociation region. For larger values of E_{max}, the difference between $E_{max} - V(\rho)$ in the well and dissociation regions becomes less pronounced and points are generated in a more uniform fashion.
- solver_steps – optional, specifies number of steps in Runge-Kutta algorithm used to solve Eq. (13). Typically, does not need to be set explicitly. The default value of 2048 is expected to work well for majority of applications.

Note that when optimized grid is requested, step grid parameter controls the step size in the auxiliary equidistant x-grid, i.e. step sets the value of α in Eq. (13).

In addition to these parameters, the following config parameter has to be specified to calculate the reduced particle mass μ:

- mass – the description of masses of atoms in the system under consideration. The masses have to be listed as 3 comma-separated numbers corresponding to masses given in atomic unit of mass (*i.e.*, in masses of an electron, m_e) of atoms in ABA order. Instead of providing an explicit mass value, it is possible to use a shortcut consisting of an element symbol and its mass number. For example, H1 will be replaced with the mass of hydrogen-1 and O18 will be replaced with the mass of oxygen-18. These shortcuts exist for all stable isotopes of the elements in the first three periods. The exact values of isotopic masses are taken from Ref. 16 and can be found in `src/base/constants.f90` file.

Note that `mass` is a config parameter, and not a grid parameter, therefore it is not a part of a grid setting group and should be specified as a top-level parameter. An example of a config file that generates optimized grid for ozone can be found in `config_examples` folder.

C. Potential

Majority of PESs do not operate in APH coordinates directly, therefore a conversion from APH to a more common coordinate system is likely required. SpectrumSDT offers two ways to alleviate this procedure. One way is to use `output_coordinate_system` key to carry out coordinate conversion at `grids` stage. If used with valid values other than `aph`, `pes.in` file will be created in addition to the grid files, where each row after the header row specifies molecular geometry converted to the chosen coordinate system. In response user needs to provide a file named `pes.out`, with the values of the PES in Hartree at geometries in `pes.in`. Providing `pes.out` file is how SpectrumSDT obtains information about user-specified potential, i.e. there is no need to modify the code to define a potential subroutine. Note that `pes.out` is not expected to have a header line, therefore row numbering is shifted by 1 relative to `pes.in`.

The following values of `output_coordinate_system` are supported:

- `aph` – the default value. `pes.in` file is not generated since it is trivial to simply iterate the values in the grid files.
- `mass jacobi` – mass-scaled version of Jacobi coordinates. See Eqs. (52a)-(52c) in Ref. 5.

- `jacobi` – regular Jacobi coordinates, where r (Bohr) is the distance between the terminal A-atoms (in ABA notation); R (Bohr) is the distance from the center of mass between the two A atoms (midpoint) to the B atom; and Θ (in radians) is the angle between these two vectors.
- `cartesian` – molecular geometry is described with x- and y- coordinates of atom B and x-coordinate of an atom A. The remaining 6 Cartesian coordinates are fixed to 0. All values are given in Bohr.
- `all bonds` – three pair-wise distances in Bohr between the atoms. Indexes 1 and 3 are assigned to the A atoms and index 2 is assigned to B.
- `internal` – two A-B bond lengths in Bohr and A-B-A angle in radians.

Note that `mass` has to be specified if a value other than `aph` or `mass jacobi` is given to `output_coordinate_system`, since coordinate conversion is mass-dependent. This makes it necessary to re-calculate potential values for different isotopomers even if underlying PES program is the same.

Another way is to directly use `src/base/coordinate_conversion.f90` module in user's PES program to convert coordinates dynamically. In this case the user can set `output_coordinate_system` to `aph` to avoid generating `pes.in` and read the APH-grids directly. The combinations of APH points should be iterated in order ρ, θ, φ (*i.e.*, ρ changes least frequently, φ changes most frequently). An example of a Fortran program that reads the grids and writes ozone potential of Dawes et al.[17] in this way can be found in `PES_examples/ozone/ozone_pes.f90`.

D. Basis

This stage uses the grids and potential from the previous two stages to solve 1D and 2D problems in the given potential (see Section I-B). The following parameters have to be specified:

- `J` – total angular momentum quantum number for basis functions.
- `K` – projection of total angular momentum onto *z*-axis quantum number for basis functions.
- `basis` – parameter group that includes the following parameters:
    - `num_functions_phi` – number of VBR basis functions to include in the primitive basis set for 1D problem (convergence parameter).
    - `symmetry` – “0” for symmetric basis functions (cosines); “1” for antisymmetric functions (sines).
    - `cutoff_energy` – maximum energy of a solution in 1D or 2D problem (convergence parameter, in wavenumbers). Solutions with energies above this are not included in the basis set.
    - `min_solutions` – minimum number of solutions retained in each 1D and 2D problem, even when energies exceed `cutoff_energy`, typically does not need to be set. The default value of 3 is expected to work well for majority of applications.
- `grid_path` – full path to a folder with APH-grids and `pes.out` files.
- `root_path` – full path to the top-level folder of the current calculation (from where `init_spectrumsdt_folders.py` was executed).

In addition, `stage` has to be set to `basis` and `mass` has to be specified. The number of 2D solutions kept in basis for each value of ρ is written to `num_vectors_2d.dat` file. The binary files storing 1D and 2D basis functions are written to `out_basis` folder.

Note that multiple basis calculations with different values of K and symmetry are required for a single problem in cases when the Hamiltonian matrix consists of multiple *K*-blocks and the *adiabatic* basis set is employed.

The basis set is calculated independently for each value of ρ, which are distributed among the available processors. Therefore the number of points in ρ-grid limits the maximum number of processors that can do useful work at this stage.

E. Overlaps

This stage uses basis functions from the previous stage (basis) to calculate the Hamiltonian matrix elements for the row of blocks with the given value of K. In addition to the parameters specified at basis stage, the following parameters are required:

- fixed_basis – at this stage specifying this parameter group instructs the program to compute additional overlaps necessary for the *fixed* basis mode (see Section I-D).
- use_rovib_coupling – "0" is for uncoupled symmetric top; specifying "1" instructs the program to compute additional overlaps for the off-diagonal *K*-blocks (see Figure 2) necessary to build the Hamiltonian matrix with rotational-vibrational coupling terms.

All output files from this stage are binary and are stored in the out_overlaps folder. Calculation of individual matrix elements is independent and distributed among available processors, therefore this stage is expected to scale efficiently with the number of processors.

F. Eigensolve

This stage uses the matrix elements computed at the previous stage (`overlaps`) to build the Hamiltonian matrix and find its eigenpairs (energies and wave functions). In addition to parameters specified at the `overlaps` stage, the following parameter are required:

- `J` – total angular momentum quantum number for the current problem. Note that this can be different from the value of `J` at the `basis` stage if *fixed* basis mode is employed.

- `parity` – the value of inversion parity of the wave functions, “0” or “1”.[3] Only matters if rotational-vibrational coupling is enabled (`use_rovib_coupling = 1`), $J > 0$ and $K = 0$ or 1 block is included in the Hamiltonian matrix.

- `symmetry` – the symmetry of the basis for $K = 0$ block, “0” for symmetric or “1” for antisymmetric basis. The subsequent values of K have to have alternating symmetry, therefore this choice defines symmetry for all values of K included in the Hamiltonian. Note that the symmetry is always defined with respect to $K = 0$ even in cases when there is no block corresponding to $K = 0$ in the Hamiltonian matrix.

- `K` – the values of K to include in the Hamiltonian matrix. Can either be a scalar number for a symmetric top rotor calculation; “all” for a fully coupled rotational-vibrational calculation with all valid values of K for given J and *parity*, from mod(J+*parity*, 2) to J; or a custom K-range in the form $K_1..K_2$ to include only values of $K_1 \leq K \leq K_2$ for a partially coupled calculation. In the case of *adiabatic* basis, the `basis` and `overlaps` stages of appropriate symmetry have to be finished and available for all values of K specified in this parameter.

- `eigensolve` – parameter group with the options for an eigenvalue solver. The options include the following:

- num_states – number of eigenpairs to converge.
- ncv – largest dimension of the working subspace, optional, typically does not need to be set explicitly, set mpd instead. Determined automatically by SLEPc if left unset.
- mpd – maximum projected dimension, optional, used to further restrict the size of projected eigenproblem on certain steps of the algorithm. Determined automatically by SLEPc if left unset. Refer to SLEPc manual for more details on ncv and mpd parameters.[18] For small number of eigenvalues (< 300) we recommend leaving mpd unset. For larger number of eigenvalues one can improve performance by setting mpd manually. Too small values of mpd lead to quickly increasing solution time, while too large values require too much memory and also worsen running time, although not as quickly as too small values. The optimal value of mpd is problem-dependent, but in our experience the values around 0.2*num_states worked reasonably well.
- max_iterations – maximum number of iterations the solver is allowed to perform, optional, typically does not need to be set explicitly. The default value of 10000 is expected to work well for majority of applications.

For calculations in the *fixed* basis mode, one has to describe what rotational state should be used for the basis set. This is done through the fixed_basis parameter group, which includes the following parameters:

- J – total angular momentum of the basis set functions.
- K – projection of the total angular momentum onto *z*-axis of the basis set functions.
- root_path – full path to the top-level folder with the basis and overlaps calculations for the values of J and K specified in this parameter group.

The lifetimes of the states are computed by adding a complex absorbing potential (CAP) in the form suggested by Manolopoulos.[19] By default CAP is not added. One can choose to add it by specifying parameter group `cap`, which consists of only one parameter:

- `min_absorbed_energy` – minimum energy at which absorption is required, specified in wavenumbers relative to energy of the dissociated molecule in the lowest dissociation channel (i.e. electronic dissociation energy plus zero-point energy). Note that enabling CAP is only meaningful for calculation of scattering resonances above the dissociation threshold.

If `cap` is specified, `cap.txt` file will be created. The value on i-th line of this file is the value of CAP at ρ_i given in cm^{-1}. The CAP is always positioned relative to the far end of the ρ-grid (largest value of ρ), where it has the largest value and then gradually decreases for smaller values of ρ.

The computed energies and widths of resonances are printed to file `states.fwc`. The wavefunctions are saved in binary form in `out_eigensolve` folder. This stage is parallelizable to some extent, but parallelization efficiency is decreasing with the number of processors, therefore a relatively low number of processors (around 32) is advised for efficient utilization, but higher number of processors can also be used if efficiency is not a concern.

G. Properties (optional)

At this stage one can characterize the wave functions calculated at the previous stage by analyzing their properties, such as probability distributions over specified *sections* of the overall space and splitting of resonance width between the existing dissociation channels. The boundaries of these sections typically correspond to the edges of covalent or Van der Waals wells, or other objects of interest, which are PES-dependent, therefore it is up to the user to define them for a particular problem. Each section is defined as a new subgroup of a parameter group `wf_sections`, specified by an arbitrary unique key and may include the following parameters:

- `name` – specifies section name for the header row in the output file. Optional, the value of the key, specifying this section, will be used as `name` if this parameter is not explicitly given.
- `K` – specifies the range of the values of *K* for the current section, optional.
- `rho` – specifies the range of ρ-values for the current section (in Bohr), optional.
- `theta` – specifies the range of θ-values for the current section (in degrees), optional.
- `phi` – specifies the range of φ-values for the current section (in degrees), optional.
- `stat` – determines which statistic is to be calculated for the current section, optional. The possible values are `probability` (default) or `gamma` (resonance width). Specifying `gamma` instructs the program to integrate the wave function probability in a given section, multiplied by CAP, to get a portion of the overall width associated with this section. This can be useful to calculate channel-specific decay rates of the metastable states.

All ranges are specified in the form *A*..*B* to include the values from *A* to *B* inclusively. The final shape of the section in the 4D (K,ρ,θ,φ) space is the intersection of the ranges specified for each individual coordinate. Omitting all range parameters defines a trivial section that spans the whole space and has to evaluate to 1 if `stat = probability`, or total resonance width if `stat = gamma`.

The method of section definition implemented here limits their shapes to hyper-rectangles, but hyper-rectangles in the APH coordinates are often what one is looking for and more complicated shapes can be approximated by adding up multiple hyper-rectangles.

The results are written to `states.fwc` file, where the first two columns duplicate the results of the previous stage, and the next *n* (number of sections) columns add the statistics corresponding to the specified sections. Each processor computes its fraction of the wave functions independently from the other processors (except for the final merge of data for printing to a file), therefore high parallelization efficiency is expected for this stage.

LIBRARIES CREDIT

SpectrumSDT uses the following libraries:

- LAPACK for diagonalization of 1D and 2D Hamiltonians.[20]
- fdict for an implementation of dictionaries in Fortran.[21]
- SLEPc for an implementation of an iterative eigensolver.[22–24]
- PETSc as an underlying library for SLEPc.[25]

ACKNOWLEDGMENTS

This research was supported by the NSF AGS program Grant No. AGS-1920523. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231. Igor Gayday acknowledges the support of MolSSI Investment Fellowship, funded by NSF ACI-1547580.

REFERENCES

(1) Teplukhin, A.; Babikov, D. Efficient Method for Calculations of Ro-Vibrational States in Triatomic Molecules near Dissociation Threshold: Application to Ozone. *J. Chem. Phys.* **2016**, *145*, 114106.

(2) Gayday, I.; Teplukhin, A.; Kendrick, B. K.; Babikov, D. Theoretical Treatment of the Coriolis Effect Using Hyperspherical Coordinates, with Application to the Ro-Vibrational Spectrum of Ozone. *J. Phys. Chem. A* **2020**, *124*, 2808–2819.

(3) Gayday, I.; Teplukhin, A.; Kendrick, B. K.; Babikov, D. The Role of Rotation–Vibration Coupling in Symmetric and Asymmetric Isotopomers of Ozone. *J. Chem. Phys.* **2020**, *152*, 144104.

(4) Gayday, I.; Grushnikova, E.; Babikov, D. Influence of the Coriolis Effect on the Properties of Scattering Resonances in Symmetric and Asymmetric Isotopomers of Ozone. *Phys. Chem. Chem. Phys.* **2020**, *22*, 27560–27571.

(5) Pack, R. T.; Parker, G. A. Quantum Reactive Scattering in Three Dimensions Using Hyperspherical (APH) Coordinates. Theory. *J. Chem. Phys.* **1987**, *87*, 3888–3921.

(6) Teplukhin, A.; Babikov, D. Interactive Tool for Visualization of Adiabatic Adjustment in APH Coordinates for Computational Studies of Vibrational Motion and Chemical Reactions. *Chem. Phys. Lett.* **2014**, *614*, 99–103.

(7) Teplukhin, A.; Babikov, D. Visualization of Potential Energy Function Using an Isoenergy Approach and 3D Prototyping. *J. Chem. Educ.* **2015**, *92*, 305–309.

(8) Light, J. C.; Carrington, T. Discrete-Variable Representations and Their Utilization. In *Advances in Chemical Physics*; Prigogine, I., Rice, S. A., Eds.; John Wiley & Sons, 2000; Vol. 114, pp 263–310.

(9) Bačić, Z.; Whitnell, R. M.; Brown, D.; Light, J. C. Localized Representations for Large Amplitude Molecular Vibrations. *Comput. Phys. Commun.* **1988**, *51*, 35–47.

(10) Kendrick, B. K.; Pack, R. T.; Walker, R. B.; Hayes, E. F. Hyperspherical Surface Functions for Nonzero Total Angular Momentum. I. Eckart Singularities. *J. Chem. Phys.* **1999**, *110*, 6673–6693.

(11) Teplukhin, A.; Babikov, D. Several Levels of Theory for Description of Isotope Effects in Ozone: Symmetry Effect and Mass Effect. *J. Phys. Chem. A* **2018**, *122*, 9177–9190.

(12) Teplukhin, A.; Gayday, I.; Babikov, D. Several Levels of Theory for Description of Isotope Effects in Ozone: Effect of Resonance Lifetimes and Channel Couplings. *J. Chem. Phys.* **2018**, *149*, 164302.

(13) Teplukhin, A.; Babikov, D. Properties of Feshbach and "Shape"-Resonances in Ozone and Their Role in Recombination Reactions and Anomalous Isotope Effects. *Faraday Discuss.* **2018**, *212*, 259–280.

(14) Tennyson, J.; Miller, S.; Le Sueur, C. R. Triatom: Programs for the Calculation of Ro-Vibrational Spectra of Triatomic Molecules. *Comput. Phys. Commun.* **1993**, *75*, 339–364.

(15) Kokoouline, V.; Dulieu, O.; Kosloff, R.; Masnou-Seeuws, F. Mapped Fourier Methods for Long-Range Molecules: Application to Perturbations in the $Rb_2(0_u^+)$ Photoassociation Spectrum. *J. Chem. Phys.* **1999**, *110*, 9865–9876.

(16) Bruno, T. J.; Svoronos, P. D. *CRC Handbook of Chemistry and Physics*, 93rd ed.; 2012.

(17) Dawes, R.; Lolur, P.; Li, A.; Jiang, B.; Guo, H. Communication: An Accurate Global Potential Energy Surface for the Ground Electronic State of Ozone. *J. Chem. Phys.* **2013**, *139*, 201103.

(18) Roman, J. E.; Campos, C.; Romero, E.; Tomas, A. *SLEPc Users Manual*; 2020.

(19) Manolopoulos, D. E. Derivation and Reflection Properties of a Transmission-Free Absorbing Potential. *J. Chem. Phys.* **2002**, *117*, 9552–9559.

(20) Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Demmel, J.; Dongarra, J.; Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A.; et al. *LAPACK Users' Guide*, 3rd ed.; Society for Industrial and Applied Mathematics: Philadelphia, 1999.

(21) fdict https://github.com/zerothi/fdict.

(22) Hernandez, V.; Roman, J. E.; Vidal, V. SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems. *ACM Trans. Math. Softw.* **2005**, *31*, 351–362.

(23) Hernandez, V.; Roman, J. E.; Vidal, V. SLEPc: Scalable Library for Eigenvalue Problem Computations. In *Lecture Notes in Computer Science*; Springer, 2003; pp 377–391.

(24) Hernandez, V.; Roman, J. E.; Tomas, A. Parallel Arnoldi Eigensolvers with Enhanced Scalability via Global Communications Rearrangement. *Parallel Comput.* **2007**, *33*, 521–540.

(25) PETSc https://www.mcs.anl.gov/petsc.

(26) Kokoouline, V.; Greene, C. H. Unified Theoretical Treatment of Dissociative Recombination of D_{3h} Triatomic Ions: Application to H_3^+ and D_3^+. *Phys. Rev. A* **2003**, *68*, 012703.