

# Schrödinger API - web service for solving multidimensional time-independent Schrödinger equation using Hermite DVR approach

Bojana Koteska and Anastas Mishev  
 Ss. Cyril and Methodius University,  
 Faculty of Computer Science and Engineering,  
 Skopje, North Macedonia  
 Email: bojana.koteska@finki.ukim.mk  
 anastas.mishev@finki.ukim.mk

Ljupco Pejov  
 Ss. Cyril and Methodius University,  
 Faculty of Natural Science and Mathematics,  
 Skopje, North Macedonia  
 Email: ljupcop@pmf.ukim.mk

**Abstract**—Numerous phenomena taking place in the nano-world are inherently quantum in nature. Their description and, more important, quantitative treatment, therefore, requires usage of the apparatus of quantum mechanics. The basic paradigm of today’s “mainstream” quantum mechanics is the Schrödinger equation. The effort required to solve the Schrödinger equation is heavily dependent on the dimensionality and complexity of the problem itself. Here, we present the Schrödinger API - RESTful web service that provides methods for solution of one-dimensional, two-dimensional and three-dimensional time-independent Schrödinger equation based on the the Gauss-Hermite Discrete Variable Representation (DVR) approach. Simple use is ensured thanks to the REST architectural style that uses HTTP requests to access and use data through the website <https://schrodinger.chem-api.finki.ukim.mk>. The application of the web service is shown by a computational physics use case. Specialists, but also nonexperts in computational chemistry or physics can use this service to support their research.

## I. INTRODUCTION

In many subdisciplines of computational molecular sciences, computational physics, chemistry, biology, materials science, exact treatment and analysis of a wide variety of phenomena has to rely on rigorous quantum description of the underlying processes.

Numerous phenomena in the nano-world are quantum in nature which requires usage of the quantum mechanics. The basic paradigm of today’s “mainstream” quantum mechanics is the Schrödinger equation, which is considered as a “quantum analog” to the famous Newton’s second law equation in classical physics.

The effort required to solve the Schrödinger equation is heavily dependent on the dimensionality and complexity of the problem itself (e.g. the exact form of the Hamiltonian, number of the relevant degrees of freedom of the studied system etc.). Numerous methods have been proposed in the literature to achieve the mentioned aim. However, the available codes are most often user-hostile, the procedures for computation and generation of relevant data are non-standardized, and there is a clear lack of in-depth, thorough comparison of performances

of various methods for solving the Schrödinger equation for various purposes.

Figueiras et al. [1] presented a simple Python-based open source software library for the numerical simulation and solution of the linear or nonlinear time-dependent one and two dimensional Schrödinger equation. Srncic et al. [2] used the Python programming environment and the three-point finite-difference numerical method to find the solutions of the Schrödinger equation and plot the results for a particle in an infinite, finite, double finite, harmonic, Morse, or Kronig–Penney finite potential energy well. Web-Schrödinger 3.2 is a program for the interactive solution of the stationary (time independent) and time dependent two dimensional (2D) Schrödinger equation. The program itself runs on our server and can be used through the Internet with a simple Web browser. The user can load, run, and modify ready-made example files, or prepare her/his own configuration(s), which can be saved on her/his own computer for later use [3]. WavePacket [4] is an open-source program package in Matlab for the numerical simulation of quantummechanical dynamics used to solve time-independent or time-dependent linear Schrödinger equation in one or more dimensions. Similar software solutions can be found for Python, such as PyCav [5] - a physics simulation library that provides method for solution of time-dependent Schrödinger equation and QuTiP [6] - an open-source framework that provides method for solving the unitary Schrödinger equation.

Schrödinger API provides friendly and simplified computational platform for solution of one-dimensional, two-dimensional and three-dimensional time-independent Schrödinger equation based on the the Gauss-Hermite Discrete for several model potentials: Morse potential; Simple Harmonic Oscillator (SHO) potential; Sombrero potential (Mexican hat) and Woods-Saxon potential.

The advantage of using API becomes particularly obvious when it comes to automation of a more complex workflow. To show this, we present a computational physics use case which uses the Schrödinger API.

## II. SCIENTIFIC BACKGROUND - DVR TECHNIQUES FOR SOLVING THE SCHRÖDINGER EQUATION

The basic paradigm of today's "mainstream" quantum mechanics is the Schrödinger equation, which is considered as a "quantum analog" to the famous Newton's second law equation in classical physics. When a stationary state of the considered system is of interest, the Schrödinger equation has the form:

$$\hat{H}\psi = E\psi \quad (1)$$

where  $\hat{H}$  denotes the Hamiltonian of the system,  $E$  is the energy, while  $\Psi$  is the wavefunction. The Hamiltonian in (1) consists of the kinetic energy operator ( $\hat{K}(q)$ ), which is a second-order differential operator in coordinate ( $q$ ) representation, and the potential energy operator ( $\hat{V}(q)$ ):

$$\hat{H}(q) = \hat{K}(q) + \hat{V}(q) \quad (2)$$

If one is interested in either molecular bound states or scattering phenomena, the solutions of (1),  $\Psi(q)$ , form either a discrete set of square-integrable eigenfunctions (in  $L^2$ ) which are localized, or (as in the scattering problems), which are not square-integrable, but the asymptotic forms of which are known, and therefore appropriate boundary conditions are applicable. As the analytical solution of (1) is known only for the simplest systems, it is necessary to have an efficient numerical method to solve it. What is often sought is a numerical solution of (1) within some finite range (e.g. finite coordinate range), which is subsequently matched to the asymptotic form. The statement of finite-range solutions has an important consequence, namely that the solution may be sought within a finite basis set composed of  $n$  functions (e.g. of the relevant coordinates -  $\varphi_i(q)$ ), as in the variational approach:

$$\psi(q) = \sum_{i=1}^n c_i \cdot \varphi_i(q) \quad (3)$$

Assuming an orthonormal set of basis functions, *i.e.*:

$$\langle \varphi_i | \varphi_j \rangle = \int \varphi_i^*(q) \cdot \varphi_j(q) dq = \delta_{ij} \quad (4)$$

Equation (3) is a spectral representation of the wavefunction  $\langle q | \psi \rangle$  in a truncated bases within the Hilbert space  $L^2\{\langle q | \varphi_i \rangle; i = 1, 2, \dots, n\}$ , within the Dirac notation written as:

$$\langle q | \psi \rangle = \sum_{i=1}^n c_i \cdot \langle q | \varphi_i \rangle \quad (5)$$

inserting (3) in (1) leads to the conclusion that the expansion coefficients  $c_i$  and the set of eigenenergies can be found by solving the matrix eigenvalue problem (6):

$$\mathbf{Hc} = E\mathbf{c} \quad (6)$$

In (6),  $\mathbf{c}$  is the column  $n$ -vector  $[c_1, c_2, \dots, c_n]^T$ , while  $\mathbf{H}$  is  $n \times n$  square matrix containing the matrix elements of the Hamiltonian  $H_{ij}$ , given by:

$$H_{ij} = \langle \varphi_i | \hat{H} | \varphi_j \rangle = \int \varphi_i^*(q) \hat{H} \varphi_j(q) dq \quad (7)$$

Such matrix-representation is often referred to as variational basis representation (VBR). As a consequence of the orthonormality of the basis set functions, it can be straightforwardly shown that the expansion coefficients in (3) and (5) are given by:

$$c_k = \langle \varphi_k | \psi \rangle = \int \langle \varphi_k | q \rangle \langle q | \psi \rangle dq \quad (8)$$

The first convenient numerical method for approximation of the integrals of the form (7), *i.e.* their analogues with different components of the total  $H$ , dates back to the works of Harris, Engerholm and Gwinn [7]. It has been further on shown that any basis set that consists of orthogonal polynomials (such as the one constituted by harmonic oscillator eigenfunctions) can be approximated by the so-called Gaussian quadratures technique. Approximating, however, the matrix elements by any quadrature technique, leads to the consequence that the results need not be the upper limits of the exact (true) eigenvalues. Such representation of the potential energy operator is called finite basis representation (FBR), to distinguish from the previously mentioned variational basis representation (VBR).

The quadrature approximation is a straightforward method to compute the potential energy matrix elements. We define an  $n$ -point quadrature such that it consists of the abscissa set of values (in a one-dimensional case)  $\{x_1, x_2, \dots, x_n\}$  and the corresponding weights  $\{w_1, w_2, \dots, w_n\}$ . Such  $n$ -point quadrature leads to:

$$V_{ij} \approx V_{ij}^{FBR} = \sum_{k=1}^n w_k \psi_i^*(x_k) V(x_k) \psi_j(x_k) \quad (9)$$

Defining:

$$A_{ki} \equiv w_k^{1/2} \psi_i(x_k) \quad (10)$$

along with the diagonal matrix:

$$V_{kl}^{diag} = \delta_{kl} V(x_k) \quad (11)$$

we can easily arrive at:

$$V_{ij}^{FBR} = \sum_{k=1}^n A_{ki}^* V_{kk}^{diag} A_{kj} \quad (12)$$

In matrix notation (12) can be written as:

$$V^{FBR} = A^\dagger V^{diag} A \quad (13)$$

The method of Harris, Engerholm and Gwinn for computation of potential energy matrix elements is actually equivalent to using the Gaussian quadrature approach in a basis set of orthogonal polynomials [8]. In the FBR, the potential energy

matrix is written as a unitary transformation of a diagonal matrix (eq. (13)).

The fundamental idea of the discrete variable representation (DVR) technique [9] is to apply such unitary transformation that enables the potential energy matrix to be diagonal, *i.e.*:

$$\phi_k(x) = \sum_{i=1}^n A_{ki} \psi_i(x) \quad (14)$$

Evaluating the DVR functions at the quadrature points, one arrives at:

$$\phi_k(x_l) = \sum_{i=1}^n A_{ki} \psi_i(x_l) = \sum_{i=1}^n A_{ki} A_{li} w_l^{-\frac{1}{2}} = \delta_{kl} w_l^{-\frac{1}{2}} \quad (15)$$

Thus, it can be easily verified that any operator which is multiplicative in coordinate representation is diagonal in this basis, provided that the quadrature approximation has been used:

$$\langle \phi_i | v | \phi_j \rangle = \sum_{k=1}^n w_k \phi_i(x_k) V(x_k) \phi_j(x_k) = \sum_{k=1}^n \delta_{ik} V(x_k) \delta_{jk} = V(x_k) \delta_{ij} \quad (16)$$

As the kinetic energy matrix is not diagonal in DVR (since the kinetic energy operator is not multiplicative in coordinate representation), in the case of one-dimensional problems, the advantage of DVR over FBR is questionable. However, in multidimensional problems, the advantage of DVR becomes immediately evident [10].

### III. WEB SERVICE DESCRIPTION AND EXAMPLES

This RESTful web service (SchrodingerAPI) provides a method for solution of one-dimensional, two-dimensional and three-dimensional time-independent Schrödinger equation based on the Gauss-Hermite Discrete Variable Representation (DVR) approach. Web service source code is based on the python module for solving 1D Schrödinger equation [11], with several code adaptations and modifications. Methods for solving 2D and 3D Schrödinger equations are developed from scratch.

The solution of one-dimensional Schrödinger equation is illustrated in the case of following model potentials: **Morse potential; Simple Harmonic Oscillator (SHO) potential; Sombbrero potential (Mexican hat); Woods-Saxon potential.**

Solutions of two-dimensional and three-dimensional Schrödinger equations are illustrated for the following two model potentials: multidimensional Morse potential and multidimensional SHO potential.

All service methods are GET methods and they can be tested by entering the parameters directly on the SchrodingerAPI home page (supported by Swagger) at <https://schrodinger.chem-api.finki.ukim.mk>. Other possibilities to test the service methods include using browser URL bar, consuming the

service methods in your source code or by using specific API testing tools such as Postman [12].

For example, if user wants to test the 1dHermiteSho method directly at <https://schrodinger.chem-api.finki.ukim.mk>, the first step is to click the 1dHermiteSho method from the list shown in Fig. 1.

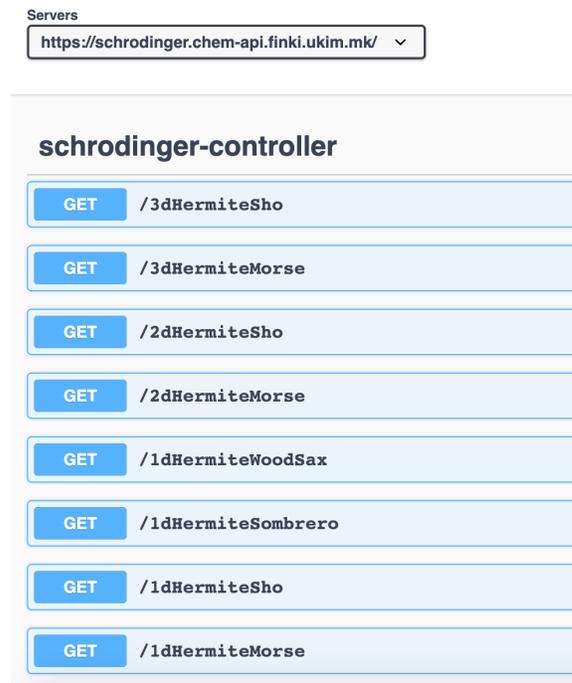


Fig. 1. List of Schrödinger API methods

1D harmonic oscillator potential is defined as:

$$V(x) = \frac{1}{2} k \cdot (x - x_0)^2 \quad (17)$$

Next, user should enter the parameters or use the default ones as shown in Fig. 2. The following parameters are required: **npts** - number of points; **k** - wavenumber of the SHO potential; **x0** - displacement from origin and **prec** - precision.

The results (eigenvalues) will be shown in the Response Body form below (Fig 3). User can download or copy them.

Other way to use these REST API methods is to access it directly from the browser address bar.

If no parameters are provided, the default parameter values will be taken into consideration. For example, for the 1dHermiteSho method the link should be <https://schrodinger.chem-api.finki.ukim.mk/1dHermiteSho>.

If user wants to change only the default parameters npts and k and to provide his/hers, the URL should look like this <https://schrodinger.chem-api.finki.ukim.mk/1dHermiteSho?npts=10&k=1>.

If user prefers to change other parameters, they can be added with the &PARAMETER=VALUE.

Another option is to consume the method in a program source code. An example in Python is provided in Listing 1.

Name	Description
npts	integer (\$int32) (query) 5
k	number (\$double) (query) 1
x0	number (\$double) (query) 0
prec	integer (\$int32) (query) 8

Fig. 2. Parameters for 1dHermiteSho method

Code	Details
200	Response body [0.5 1.5 2.5 3.5 4.5]

Fig. 3. Results from 1dHermiteSho method

```

1 import requests
2 response = requests.get('https://schrodinger.chem-
  api.finki.ukim.mk/1dHermiteSho?npts=20&k=1.0&x0
  =0.0&prec=19')
3 if response.status_code == 200:
4     print(response.content.decode('utf-8'))
5 else:
6     print("None")

```

Listing 1. Python example

#### IV. A COMPUTATIONAL PHYSICS USE CASE USING THE SCHRÖDINGER API

##### A. Problem description and algorithm steps

The present use-case illustration is a part of a bigger project aiming to provide in-depth theoretical understanding of the tagging of protonated glycine with H<sub>2</sub> and He in gas phase. To avoid any ambiguities in the data interpretation, it is rather useful to consider the tagging of protonated glycine's analogue in which all amine group hydrogen atoms have been replaced with methyl groups, known as betaine [13]. Specifically, in the present study we have focused on the process of betaine(+) (Fig. 4) tagging with molecular hydrogen in gas phase.

To compute the anharmonic vibrational potential of the O-H stretching mode, which is essentially localized and decoupled from all other intra- and intermolecular modes, we have generated a series of configurations in which the positions of all atoms and the center of mass of the O-H oscillator were

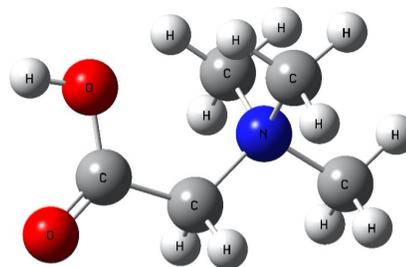


Fig. 4. The structure of betaine(+)

kept fixed, while only the O and H atoms were moved in opposite directions.

Further, we have carried out a series of single-point energy calculations for each of the generated configurations in the case of all considered O-H oscillators.

Finally, the vibrational Schrödinger equations were solved by the discrete variable representation using the 1D Morse potential:

$$V(x) = D \cdot \{1 - \exp[-a \cdot (x - x_0)]\}^2 - D$$

For the purpose of our study, we have used the API function for 1D Schrödinger equation in case of Morse potential – **1dHermiteMorse**.

As we have computed the vibrational potential energy curves in a pointwise manner, and subsequently fitted the potential to the Morse model function, for each oscillator we have different values for parameters  $D$ ,  $a$  and  $x_0$ . The number of points for DVR computations (**npts**), as well as the precision (**prec**) remains the same for all cases, to keep the computational consistency throughout the study and to enable comparability of the results.

The advantage of using Schrödinger API becomes particularly obvious when it comes to automation of the process of analyzing the results from the molecular dynamics trajectories. For each trajectory, a total of 100 snapshots have been extracted and the OH stretching vibrational potentials have subsequently been computed and fitted to a Morse function. One has, therefore, to solve a total of 100 vibrational Schrödinger equations per trajectory. Using the Schrödinger API simplifies the computational efforts to a great extent, and also allows for an automation of the process.

The obtained eigenvalues were further on used to calculate the anharmonic vibrational frequencies (i.e. wavenumbers) of the fundamental  $|0\rangle \rightarrow |1\rangle$  vibrational transitions corresponding to the O-H stretching modes, from the energy differences between the ground and the first excited vibrational states.

##### B. Results and discussion

Two minima have been located so far on all studied PESs of betaine(+) – H<sub>2</sub> noncovalently bonded gas-phase dimer. The first one corresponds to the tagging of the charged group of betaine, while the second one corresponds to the O-H ... H<sub>2</sub> contact (tagging of the hydroxyl group). As we are

especially interested in the influence of tagging on the O-H stretching vibrational frequencies, we pay a special attention to this minimum.

Fig. 5 shows typical vibrational energy curves for free betaine(+) as well as for the betaine(+)OH...H<sub>2</sub> dimer where tagging is realized on the O-H oscillator side.

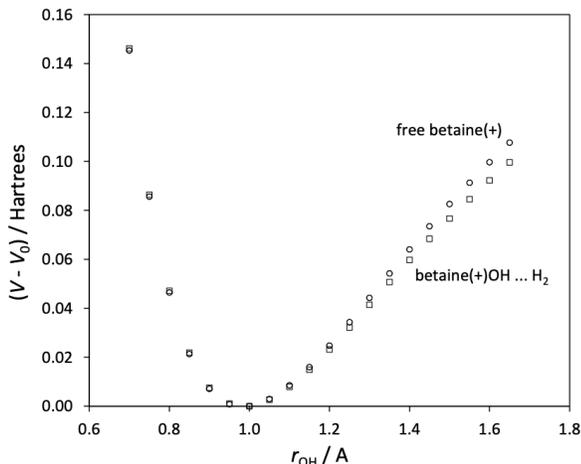


Fig. 5. Vibrational energy curves for free betaine(+) as well as for the betaine(+)...H<sub>2</sub> dimer where tagging is realized on the O-H oscillator side

One can readily observe a “downshift”, *i.e.* a shift of the potential energy curve to lower energies at higher O-H distances, implying increase of the anharmonicity and a red shift of the O-H stretching wavenumber in the case of betaine(+)OH...H<sub>2</sub> dimer, as compared to free betaine(+) cation. A typical fit of the O-H stretching potential energy function with Morse function is shown in Fig. 6. All the Morse function fits were excellent, judging from all statistical figures of merit (e.g. adjusted R<sup>2</sup> was always higher than 0.999).

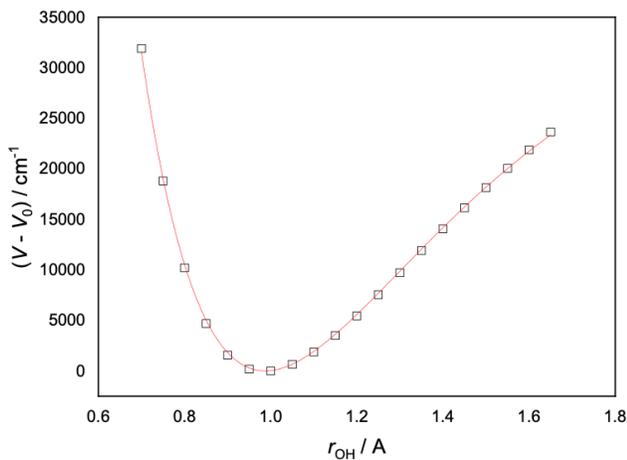


Fig. 6. A typical fit of the O-H stretching potential energy function with Morse function

The frequencies (*i.e.* wavenumbers) of the fundamental

$|0\rangle \rightarrow |1\rangle$  vibrational transitions corresponding to the O-H stretching modes in the case of free betaine(+) and betaine(+)...H<sub>2</sub>, with the H<sub>2</sub> tagging realized on the CH<sub>3</sub> and OH sides, and the corresponding frequency shifts calculated according to the described procedure are summarized in Table 1, where the available experimental IRMPD data [13] are also shown. All values are in cm<sup>-1</sup>.

TABLE I  
THE WAVENUMBERS OF THE FUNDAMENTAL  $|0\rangle \rightarrow |1\rangle$  VIBRATIONAL TRANSITIONS CORRESPONDING TO THE O-H STRETCHING MODE IN THE CASE OF FREE BETAINE(+) AND BETAINE(+)...H<sub>2</sub>, WITH THE H<sub>2</sub> TAGGING REALIZED ON THE CH<sub>3</sub> AND OH SIDES

	B3LYP	CAM-B3LYP	DFTB(A)	Exp.
<b>Betaine(+)</b>	3534.6	3575.6	3486.7	3554
<b>Betaine(+)-OH...H<sub>2</sub></b>	3447.7	3483.0	3412.7	3514
$\Delta v$	-86.9	-92.6	-74.0	-40
<b>Betaine(+)-CH<sub>3</sub>...H<sub>2</sub></b>	3551.4	3593.1	3486.9	3558
$\Delta v$	16.8	17.5	0.2	4

As can be seen from Table 1, the noncovalent interaction of betaine(+) with molecular hydrogen “tag” from the O-H oscillator side leads to substantial downshift (red-shift) of the frequency of the O-H stretching vibration. The implemented rigorous theoretical approaches so far, clearly overestimate the frequency shift of this mode (computed with reference to free betaine(+)). Including the long-range corrections into the B3LYP combination of functionals (*i.e.* using the CAM-B3LYP functional) leads to even a more pronounced disagreement with the experiment. The semiempirical DFTB-A method, however, seems to outperform the other two.

Tagging the betaine(+) molecule with molecular hydrogen from the charged group side, on the other hand, leads to a very small blue-shift of the O-H stretching frequency of the betaine(+) intramolecular O-H oscillator. While all theoretical approaches implemented in the present study predict correctly the sign of this shift, the quantitative agreement between the two advanced DFT methods (B3LYP and CAM-B3LYP) and the experiment is much poorer than that of semiempirical DFTB-A.

The previously outlined results clearly indicate that tagging with molecular hydrogen in photodissociation techniques can lead to notable perturbation of the intramolecular vibrational force field of the tagged molecular system. This is particularly pronounced in the case of typical vibrational chromophores, such as the O-H group.

## V. CONCLUSION

The RESTful web service Schrödinger API enables solution of one-dimensional, two-dimensional and three-dimensional time-independent Schrödinger equation based on the DVR approach. It is free open-access API which encompasses several different model potentials: Morse potential, Simple Harmonic Oscillator (SHO) potential, Sombrero potential (Mexican hat) and Woods-Saxon potential. Schrödinger API provides friendly, simplified and login-free computational platform that enables easy input and efficient computation through Swagger tooling ecosystem for developing APIs [14] or directly

from a browser which does not require any prior knowledge in programming. The advantage of using Schrödinger API becomes particularly obvious when it comes to automation of the process of a more complex workflow. In such case, the API functions can be consumed in the source code. As a result, Schrödinger API has been designed to support the entire community, both specialists and nonexperts in their scientific research.

#### ACKNOWLEDGMENT

The work presented in this paper is supported by the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, North Macedonia.

#### REFERENCES

- [1] E. Figueiras, D. Olivieri, A. Paredes, and H. Michinel, "An open source virtual laboratory for the schrödinger equation," *European Journal of Physics*, vol. 39, no. 5, p. 055802, 2018.
- [2] M. N. Srncic, S. Upadhyay, and J. D. Madura, "A python program for solving schrödinger's equation in undergraduate physical chemistry," 2017.
- [3] G. I. Márk, "Web-Schrödinger: Program for the interactive solution of the time dependent and stationary two dimensional (2D) Schrödinger equation," *arXiv preprint arXiv:2004.10046*, 2020.
- [4] B. Schmidt and U. Lorenz, "Wavepacket: A matlab package for numerical quantum dynamics. i: Closed quantum systems and discrete variable representations," *Computer Physics Communications*, vol. 213, pp. 223–234, 2017.
- [5] P. Team, "Pycav," <https://pycav.readthedocs.io/en/latest/>, 2016, (Accessed on 03/23/2021).
- [6] J. R. Johansson, P. D. Nation, and F. Nori, "Qutip: An open-source python framework for the dynamics of open quantum systems," *Computer Physics Communications*, vol. 183, no. 8, pp. 1760–1772, 2012.
- [7] D. O. Harris, G. G. Engerholm, and W. D. Gwinn, "Calculation of matrix elements for one-dimensional quantum-mechanical problems and the application to anharmonic oscillators," *The Journal of Chemical Physics*, vol. 43, no. 5, pp. 1515–1517, 1965.
- [8] J. Stoer and R. Bulirsch, "Finding zeros and minimum points by iterative methods," in *Introduction to Numerical Analysis*. Springer, 1980, pp. 244–313.
- [9] J. Lill, "Ga parker and jc light," *Chem. Phys. Lett.*, vol. 89, p. 483, 1982.
- [10] P. Wormer, G. Groenenboom, and A. van der Avoird, "Ab initio prediction of the vibration-rotation-tunneling spectrum of hcl-(h<sub>2</sub>o)<sub>2</sub>," *The Journal of Chemical Physics*, vol. 115, no. 8, pp. 3604–3613, 2001.
- [11] A. Richie-Halford, "dvr py- a python module that solves one-dimensional potentials using a discrete variable representation method." [https://github.com/richford/dvr\\_py/](https://github.com/richford/dvr_py/) (Accessed on 03/10/2021).
- [12] I. Postman, "Api documentation tool — postman," <https://www.postman.com/api-documentation-tool/> (Accessed on 03/10/2021).
- [13] A. Masson, E. R. Williams, and T. R. Rizzo, "Molecular hydrogen messengers can lead to structural infidelity: A cautionary tale of protonated glycine," *The Journal of chemical physics*, vol. 143, no. 10, p. 104313, 2015.
- [14] S. Software, "Api documentation & design tools for teams — swagger," <https://swagger.io/> (Accessed on 03/25/2021).