2D Vibron Model Program Suite

Curro Perez-Bernal*[1], Jamil Khalouf Rivera[1], and Miguel Carvajal Zaera[1]

[1]Depto. CC. Integradas y Centro de Estudios Avanzados en Física, Matemáticas y Computación, Fac. CC. Experimentales, Universidad de Huelva

September 12, 2018

Contents

*curropb@uhu.es

1 Abstract

This is the documentation of the `2DVM_U3` program suite that is composed of mainly `Fortran 90` but also `Perl` and `Python` scripts to perform calculations using the 2D limit of the Vibron model (2DVM).

2 Theory Outline

The original paper where the 2DVM was presented is Ref.~[**Iachello1996**]. We also make extensive use of the detailed description of the model provided in Ref.~[**PBernal2008**]. Other references deal mainly with the application of the 2DVM to molecular bending vibrations [**Ishikawa2002**, **Iachello2003**, **PBernal2005**], quantum phase transitions [**Caprio2008**, **PBernal2010**, **PFernandez2011**, **PRA86**, **Santos2015**, **santos2016**, **Castanos2016**, **PB_Santos2016**], or both [**Larese2011**, **algmonod1**]. New developments also consider the coupling of several $U(3)$ dynamical algebras for the modeling of systems with several degrees of freedom [**mp_u3xu3**, **pla_376**, **cocoyoc**, **Larese2014**, **Calixto2014**].

We proceed to review the main theory results used in this program suite.

The 2DVM has an $U(3)$ dynamical algebra whose nine generators are built as bilinear products of one scalar $\sigma^\dagger$ and two circular $\tau_\pm^\dagger$ bosonic creation operators with the corresponding annihilation operators, $\sigma, \tau_\pm$ [**PBernal2008**]. The nine generators are recasted to provide a deeper physical insight as [**Iachello1996**]

$$\begin{array}{lcl} \hat{n} = \tau_+^\dagger\tau_+ + \tau_-^\dagger\tau_- & , & \hat{n}_s = \sigma^\dagger\sigma \\ \hat{\ell} = \tau_+^\dagger\tau_+ - \tau_-^\dagger\tau_- & , & \hat{Q}_\pm = \sqrt{2}\,\tau_\pm^\dagger\tau_\mp \\ \hat{D}_\pm = \sqrt{2}(\pm\tau_\pm^\dagger\sigma \mp \sigma^\dagger\tau_\mp) & , & \hat{R}_\pm = \sqrt{2}(\tau_\pm^\dagger\sigma + \sigma^\dagger\tau_\mp) \ . \end{array} \tag{1}$$

There are two possible subalgebra chains starting from the dynamical algebra $U(3)$ and ending in the symmetry algebra $SO(2)$. The symmetry algebra has the 2D angular momentum $\hat{\ell}$ as generator, which imply that this is a conserved quantity:

$$\begin{tikzcd} & U(2) \arrow[dr] & & Chain(I) \\ U(3) \arrow[ur] \arrow[dr] & & SO(2) & \\ & SO(3) \arrow[ur] & & Chain(II) \end{tikzcd} \tag{2}$$

The corresponding subalgebras are composed by the following elements

$$\begin{array}{cc} U(2) & \{\hat{n}, \hat{\ell}, \hat{Q}_+, \hat{Q}_-\} \\ SO(3) & \{\hat{l}, \hat{D}_+, \hat{D}_-\} \\ SO(2) & \{\hat{l}\} \end{array} \tag{3}$$

where the $SO(3)$ elements satisfy the usual angular momentum commutation relations. Because of an automorphism of the Lie algebra $U(3)$ constructed with $\tau_\pm$, σ there is an alternative $SO(3)$ subalgebra of $U(3)$, called $\overline{SO(3)}$, with elements

$$\overline{SO(3)} \quad \{\hat{l}, \hat{R}_+, \hat{R}_-\} \ . \tag{4}$$

Another ingredient of the algebraic approach are the Casimir (or invariant) operators associated to each subalgebra chain [**frank**, **bookalg**]. The first and second order Casimir operators for the subalgebras in

Eqs.\ (3) and (4) are

$$\begin{array}{l} \hat{C}_1[U(2)] = \hat{n} \quad , \quad \hat{C}_2[U(2)] = \hat{n}(\hat{n}+1) \\ \hat{C}_2[SO(3)] = \hat{W}^2 = (\hat{D}_+\hat{D}_- + \hat{D}_-\hat{D}_+)/2 + \hat{l}^2 \\ \hat{C}_1[SO(2)] = \hat{l} \quad , \quad \hat{C}_2[SO(2)] = \hat{l}^2 \end{array} \tag{5}$$

and

$$\hat{C}_2[\overline{SO(3)}] = \hat{\overline{W}}^2 = (\hat{R}_+\hat{R}_- + \hat{R}_-\hat{R}_+)/2 + \hat{\ell}^2 \ . \tag{6}$$

We proceed to state the basis quantum numbers and branching rules for each dynamical symmetry.

2.1 $U(2)$ or Cylindrical Oscillator Dynamical Symmetry

In this chain states are labeled by the quantum numbers n and ℓ

$$\left| \begin{array}{ccccc} U(3) & \supset & U(2) & \supset & SO(2) \\ [N] & & n & & \ell \end{array} \right\rangle , \tag{7}$$

and the basis is $|[N];n,\ell\rangle$. An alternative notation, often used in molecular physics, is $|[N];n^\ell\rangle$. The quantum number N labels the totally symmetric representation of $U(3)$, $[N]$, and it is related to the total number of bound states of the system, n is the vibrational quantum number, and ℓ is the vibrational angular momentum. The branching rules are

$$\begin{array}{l} n = N, N-1, N-2, \ldots, 0 \\ l = \pm n, \pm(n-2), \ldots, \pm 1 \text{ or } 0 \ , \ \ (n = \text{odd or even}) \ . \end{array} \tag{8}$$

This is the natural basis to fit experimental bending vibration data in the case of linear and quasilinear molecules.

2.2 $SO(3)$ or Displaced Oscillator Dynamical Symmetry

States in this chain are characterized by the quantum numbers

$$\left| \begin{array}{ccccc} U(3) & \supset & SO(3) & \supset & SO(2) \\ [N] & & \omega & & \ell \end{array} \right\rangle . \tag{9}$$

and will be labelled as $|[N];\omega,\ell\rangle$. The branching rules are

$$\begin{array}{l} \omega = N, N-2, N-4, \ldots, 1 \text{ or } 0 \ , \ \ (N = \text{odd or even}), \\ l = \pm\omega, \pm(\omega-1), \ldots, 0 \ . \end{array} \tag{10}$$

In this case, it is convenient to introduce a vibrational quantum number v, which can be identified with the number of quanta of excitation in the displaced oscillator:

$$v = \frac{N-\omega}{2} \ . \tag{11}$$

The branching rules in this case are

$$\begin{array}{rcl} v & = & 0, 1, \ldots, \frac{N-1}{2} \text{ or } \frac{N}{2} \ , \ \ (N = \text{odd or even}), \\ l & = & 0, \pm 1, \pm 2, \ldots, \pm(N-2v) \ . \end{array} \tag{12}$$

This is the natural basis to fit experimental bending vibration data in the case of quasi-rigid and rigidly-bent molecules.

2.3 2DVM Hamiltonians included in the program suite

We present the three different Hamiltonians that are considered in the present programs. The simplest Hamiltonian, used mainly in the study of QPTs and ESQPTs, is the so called model Hamiltonian, that includes the $\hat{n} = \tau_+^\dagger \tau_+ + \tau_-^\dagger \tau_-$ operator from the dynamical symmetry (I) (DS-I) and the) and the Pairing operator $\hat{P}$ which is built adding a diagonal contribution to the $SO(3)$ second order Casimir operator $\hat{W}^2 = (\hat{D}_+\hat{D}_- + \hat{D}_-\hat{D}_+)/2 + \hat{\ell}^2$. The Pairing operator is $\hat{P} = N(N+1) - \hat{W}^2$. the $SO(3)$ algebra).

The model Hamiltonian $\hat{\mathscr{H}}$ is then

$$\hat{\mathscr{H}} = \varepsilon \left[(1-\xi)\hat{n} + \frac{\xi}{N-1}\hat{P} \right] \quad . \tag{13}$$

The Hamiltonian has two parameters: a global energy scale ε and a control parameter ξ such that for $\xi = 0.0$ the system is in the DS-I and for $\xi = 1.0$ the system is in the DS-I. This is specially adequate to characterize the ground state and excited state QPTs as there is a second order ground state QPT for the critical value $\xi_c = 0.2$ [**PBernal2008**].

A second Hamiltonian is $\hat{H}_{2b}$, the most general one- and two-body Hamiltonian of the 2DVM defined in [**Iachello1996**].

$$\hat{H}_{2b} = E_0 + \varepsilon\hat{n} + \alpha\hat{n}(\hat{n}+1) + \beta\hat{\ell}^2 + A\hat{P} \quad , \tag{14}$$

where the operators $\hat{n}$ and $\hat{n}(\hat{n}+1)$ belong to the DS-I and the pairing operator $\hat{P}$ to the DS-II while the vibrational angular momentum, $\hat{\ell}$, is common to both dynamical chains. In fact in all the cases considered the angular momentum is a constant of the motion, ℓ is a good quantum number, and the Hamiltonian are block diagonal in ℓ, a fact that simplifies the numerical calculations reducing the dimension of the considered matrices. In addition to this, as in the absence of external fields, the first order angular momentum operator $\hat{\ell}$ is not included there is a degeneracy between positive and negative ℓ values for $\ell \neq 0$ and only $\ell \geq 0$ are considered. The formula for the dimension of a subspace of the totally symmetric representation N with angular momentum ℓ is $(N - | \ll |)/2 + 1$, if N and ℓ have the same parity, and $(N - |\ell| + 1)/2$ if they have different parity. Both cases are considered simultaneously in the programs as

$$dim(N,\ell) = \frac{N - |\ell| + mod(N - |\ell|, 2) + 1}{2} \quad . \tag{15}$$

The third and last Hamiltonian considered is $\hat{H}_{4b}$, the most general 1-, 2-, 3-, and 4-body Hamiltonian, that can be expressed in terms of the Casimir operators $\hat{n}$, $\hat{n}^2$, $\hat{\ell}^2$, $\hat{W}^2$, and $\hat{\overline{W}}^2$ can be expressed as follows

$$\begin{aligned}
\hat{H}_{4b} =& P_{11}\hat{n} \\
& + P_{21}\hat{n}^2 + P_{22}\hat{\ell}^2 + P_{23}\hat{W}^2 \\
& + P_{31}\hat{n}^3 + P_{32}\hat{n}\hat{\ell}^2 + P_{33}(\hat{n}\hat{W}^2 + \hat{W}^2\hat{n}) \\
& + P_{41}\hat{n}^4 + P_{42}\hat{n}^2\hat{\ell}^2 + P_{43}\hat{\ell}^4 + P_{44}\hat{\ell}^2\hat{W}^2 \\
& + P_{45}(\hat{n}^2\hat{W}^2 + \hat{W}^2\hat{n}^2) + P_{46}\hat{W}^4 + P_{47}(\hat{W}^2\hat{\overline{W}}^2 + \hat{\overline{W}}^2\hat{W}^2)/2 \; .
\end{aligned} \tag{16}$$

The Hamiltonian has fourteen spectroscopic constants P_{ij}, where the subindeces indicate that this parameter corresponds to the j-th operator of the i-body operators.

From the matrix elements of the creation and annihilation sigma and tau bosons in the two dynamical symmetries published in [**PBernal2008**], the ensuing operator matrix elements can be derived.

2.3.1 Operator Matrix Elements in the Dynamical Symmetry (I)

The diagonal operators in this dynamical symmetry are

Operator $\hat{n}^p$ $\langle [N]; n^\ell | \hat{n}^p | [N]; n^\ell \rangle = n^p$ for $p = 1,2,3,4$.

Operator $\hat{\ell}^{2q}$ $\langle [N]; n^\ell | \hat{\ell}^{2q} | [N]; n^\ell \rangle = \ell^{2q}$ for $q = 1,2$.

Operator $\hat{n}^p \hat{\ell}^{2q}$ $\langle [N]; n^\ell | \hat{n}^p \ell^{2q} | [N]; n^\ell \rangle = n^p \ell^{2q}$ for $p = 1,2$ and $q = 1$.

The non-diagonal matrix elements in this basis are

$SO(3)$ **Casimir Operator $\hat{W}^2$**

$$\begin{aligned}\langle [N]; n_2^l | \hat{W}^2 | [N]; n_1^l \rangle = & \left[(N-n_1)(n_1+2) + (N-n_1+1)n_1 + l^2 \right] \delta_{n_2,n_1} \\ & - \sqrt{(N-n_1+2)(N-n_1+1)(n_1+l)(n_1-l)}\, \delta_{n_2,n_1-2} \\ & - \sqrt{(N-n_1)(N-n_1-1)(n_1+l+2)(n_1-l+2)}\, \delta_{n_2,n_1+2} \quad .\end{aligned} \tag{17}$$

Note that this is the main nondiagonal operator in this case and it is a band matrix as the non-zero matrix elements are located in the main and first diagonals only.

Operator $\hat{n}\hat{W}^2 + \hat{W}^2\hat{n}$ As the operator $\hat{n}$ is diagonal the matrix is also band diagonal with matrix elements

$$\begin{aligned}\langle [N]; n_2^l | \hat{n}\hat{W}^2 + \hat{W}^2\hat{n} | [N]; n_1^l \rangle = & 2n_1 \left[(N-n_1)(n_1+2) + (N-n_1+1)n_1 + l^2 \right] \delta_{n_2,n_1} \\ & - (2n_1-2) \sqrt{(N-n_1+2)(N-n_1+1)(n_1+l)(n_1-l)}\, \delta_{n_2,n_1-2} \\ & - (2n_1+2) \sqrt{(N-n_1)(N-n_1-1)(n_1+l+2)(n_1-l+2)}\, \delta_{n_2,n_1+2} \quad .\end{aligned} \tag{18}$$

Operator $\hat{\ell}^2\hat{W}^2$ This operator is computed for $\ell \neq 0$ multiplying the $\hat{W}^2$ operator matrix times ℓ^2.

Operator $\hat{n}^2\hat{W}^2 + \hat{W}^2\hat{n}^2$ This is computed as the $\hat{n}\hat{W}^2 + \hat{W}^2\hat{n}$ operator.

$$\begin{aligned}\langle [N]; n_2^l | \hat{n}^2\hat{W}^2 + \hat{W}^2\hat{n}^2 | [N]; n_1^l \rangle = & 2n_1^2 \left[(N-n_1)(n_1+2) + (N-n_1+1)n_1 + l^2 \right] \delta_{n_2,n_1} \\ & - [n_1^2 + (n_1-2)^2] \sqrt{(N-n_1+2)(N-n_1+1)(n_1+l)(n_1-l)}\, \delta_{n_2,n_1-2} \\ & - [n_1^2 + (n_1+2)^2] \sqrt{(N-n_1)(N-n_1-1)(n_1+l+2)(n_1-l+2)}\, \delta_{n_2,n_1+2} \quad .\end{aligned} \tag{19}$$

Operator $\hat{W}^4$ This operator is computed as the matrix product of the $\hat{W}^2$ operator matrix times itself.

Operator $\hat{W}^2\hat{\overline{W}}^2 + \hat{\overline{W}}^2\hat{W}^2$ In this basis the only difference between the matrix elements of the $\hat{W}^2$ and $\hat{\overline{W}}^2$ operators is the sign of the non-diagonal contribution, which is positive in this case. The full operator is computed via matrix multiplication.

2.3.2 Operator Matrix Elements in the Dynamical Symmetry (II)

The diagonal operators in this dynamical symmetry are

$SO(3)$ **Casimir Operator** $\hat{W}^2$ $\langle [N];\omega\ell|\hat{W}^2|[N];\omega\ell\rangle = \omega(\omega+1)$.

Operator $\hat{\ell}^{2q}$ $\langle [N];\omega\ell|\hat{\ell}^{2q}|[N];\omega\ell\rangle = \ell^{2q}$ for $q = 1,2$.

Operator $\hat{\ell}^2\hat{W}^2$ $\langle [N];\omega\ell|\hat{\ell}^2\hat{W}^2|[N];\omega\ell\rangle = \ell^2\omega(\omega+1$.

Operator $\hat{W}^4$ $\langle [N];\omega\ell|\hat{W}^2|[N];\omega\ell\rangle = \omega^2(\omega+1)^2$.

The non-diagonal matrix elements in this basis are

Operator $\hat{n}$

$$\begin{aligned}\langle [N];w_2^l|\hat{n}|[N];w_1^l\rangle = &\left\{\frac{(N-w_1)\left[(w_1-l+2)(w_1-l+1)+(w_1+l+2)(w_1+l+1)\right]}{2(2w_1+1)(2w_1+3)}\right.\\ &+\left.\frac{(N+w_1+1)\left[(w_1+l)(w_1+l-1)+(w_1-l)(w_1-l-1)\right]}{2(2w_1+1)(2w_1-1)}\right\}\delta_{w_2,w_1} \qquad (20)\\ &+\sqrt{\frac{(N-w_1)(N+w_1+3)(w_1-l+2)(w_1-l+1)(w_1+l+2)(w_1+l+1)}{(2w_1+1)(2w_1+3)^2(2w_1+5)}}\delta_{w_2,w_1+2}\\ &+\sqrt{\frac{(N-w_1+2)(N+w_1+1)(w_1-l)(w_1-l-1)(w_1+l)(w_1+l-1)}{(2w_1-3)(2w_1-1)^2(2w_1+1)}}\delta_{w_2,w_1-2}\end{aligned}$$

(21)

Note that this is the main nondiagonal operator in this case and it is again a band matrix with non-zero matrix elements are located in the main and first diagonals only. The $\hat{n}$ matrix element in this basis are taken from [**PBernal2008**] with a typo that has been corrected.

Operators $\hat{n}^2$, $\hat{n}^3$, and $\hat{n}^4$ These three operators are computed by matrix multiplication of the basic operator (20)

Operator $\hat{n}\hat{\ell}^2$ This operator is computed for $\ell \neq 0$ multiplying the $\hat{n}$ operator matrix times ℓ^2.

Operator $\hat{n}\hat{W}^2 + \hat{W}^2\hat{n}$ As the operator $\hat{n}$ is band diagonal $\hat{W}^2$ is diagonal this operator matrix is also band diagonal with matrix elements

$$\begin{aligned}\langle [N]; w_2\ell | \hat{n}\hat{W}^2 + \hat{W}^2\hat{n} | [N]; w_1\ell\rangle = 2\omega_1(\omega_1+1) & \left\{ \frac{(N-w_1)\left[(w_1-l+2)(w_1-l+1)+(w_1+l+2)(w_1+l+1)\right]}{2(2w_1+1)(2w_1+3)} \right. \\ & \left. + \frac{(N+w_1+1)\left[(w_1+l)(w_1+l-1)+(w_1-l)(w_1-l-1)\right]}{2(2w_1+1)(2w_1-1)} \right\} \delta_{w_2,w_1} \qquad (22) \\ & + \left[\omega_1(\omega_1+1)+(\omega_1+2)(\omega_1+3)\right] \\ & \times \sqrt{\frac{(N-w_1)(N+w_1+3)(w_1-l+2)(w_1-l+1)(w_1+l+2)(w_1+l+1)}{(2w_1+1)(2w_1+3)^2(2w_1+5)}}\,\delta_{w_2,w_1+} \\ & + \left[(\omega_1-2)(\omega_1-1)+\omega_1(\omega_1+1)\right] \\ & \times \sqrt{\frac{(N-w_1+2)(N+w_1+1)(w_1-l)(w_1-l-1)(w_1+l)(w_1+l-1)}{(2w_1-3)(2w_1-1)^2(2w_1+1)}}\,\delta_{w_2,w_1-2}\end{aligned}$$

Operator $\hat{n}^2\hat{W}^2 + \hat{W}^2\hat{n}^2$ This is computed in the same way that the $\hat{n}\hat{W}^2 + \hat{W}^2\hat{n}$ operator but taking into account that the $\hat{n}^2$ operator is double banded. Therefore the operator matrix elements can be expressed as follow

$$\begin{aligned}\langle [N]; w_2\ell | \hat{n}^2\hat{W}^2 + \hat{W}^2\hat{n}^2 | [N]; w_1\ell\rangle = & \ 2\omega_1(\omega_1+1)[\hat{n}^2]_{w_1,w_1}\,\delta_{w_2,w_1} \\ & + \left[\omega_1(\omega_1+1)+(\omega_1+2)(\omega_1+3)\right][\hat{n}^2]_{w_1,w_1+2}\delta_{w_2,w_1+2} \\ & + \left[(\omega_1-2)(\omega_1-1)+\omega_1(\omega_1+1)\right][\hat{n}^2]_{w_1,w_1-2}\delta_{w_2,w_1-2} \\ & + \left[\omega_1(\omega_1+1)+(\omega_1+4)(\omega_1+5)\right][\hat{n}^2]_{w_1,w_1+4}\delta_{w_2,w_1+4} \\ & + \left[(\omega_1-4)(\omega_1-3)+\omega_1(\omega_1+1)\right][\hat{n}^2]_{w_1,w_1-4}\delta_{w_2,w_1-4}\ ,\end{aligned} \qquad (23)$$

where $[\hat{n}^2]_{w_i,w_j}$ are the $\hat{n}^2$ operator matrix elements.

Operator $\hat{W}^2\hat{\overline{W}}^2 + \hat{\overline{W}}^2\hat{W}^2$ In this basis we need first to compute the matrix elements of the $\hat{\overline{W}}^2$ making use of Eqs.\ (37) and (38) of Ref.\ [**PBernal2008**].

$$\langle [N]; w_2\ell_2 | \hat{R}_- | [N]; w_1\ell_1\rangle = A_{w_1,\ell_1}\delta_{w_2,w_1}\delta_{\ell_2,\ell_1-1} + B_{w_1,\ell_1}\delta_{w_2,w_1-2}\delta_{\ell_2,\ell_1-1} + C_{w_1,\ell_1}\delta_{w_2,w_1+2}\delta_{\ell_2,\ell_1-1}\ , \qquad (24)$$

where

$$\begin{aligned}A_{w,\ell} = & \frac{(2N+3)(2\ell+1)}{(2w-1)(2w+3)}\sqrt{(w+\ell)(w-\ell+1)/2} \\ B_{w,\ell} = & -\sqrt{\frac{2(N+w+1)(N-w+2)(w+\ell)(w-\ell)(w+\ell-1)(w+\ell-2)}{(2w+1)(2w-1)^2(2w-3)}} \\ C_{w,\ell} = & \sqrt{\frac{2(N+w+3)(N-w)(w+\ell+1)(w-\ell+1)(w-\ell+2)(w-\ell+3)}{(2w+1)(2w+3)^2(2w+5)}}\ .\end{aligned}$$

The previous result can be used for the obtention of an expression for the $\hat{R}_+$ operator matrix elements

$$\begin{aligned}\langle [N]; w_2\ell_2 | \hat{R}_- | [N]; w_1\ell_1\rangle^\dagger = & \langle [N]; w_1\ell_1 | \hat{R}_+ | [N]; w_2\ell_2\rangle \qquad (25) \\ = & A_{w_2,\ell_2+1}\delta_{w_1,w_2}\delta_{\ell_1,\ell_2+1} + B_{w_2+2,\ell_2+1}\delta_{w_1,w_2-2}\delta_{\ell_1,\ell_2+1} + C_{w_2-2,\ell_2+1}\delta_{w_1,w_2+2}\delta_{\ell_1,\ell_2+1}\ .\end{aligned}$$

The upper diagonal matrix elements of the Casimir operator $\hat{\hat{W}}^2 = \hat{R}_+\hat{R}_- + \hat{\ell}^2$ can then be expressed as

$$\begin{aligned}\langle [N]; w_2\ell|\hat{\hat{W}}^2|[N]; w_1\ell\rangle = & A^2_{w_1,\ell_1} + B^2_{w_1,\ell_1} + C^2_{w_1,\ell_1}\,\delta_{w_2,w_1} \\ & + (A_{w_1,\ell}B_{w_1+2,\ell} + C_{w_1,\ell}A_{w_1+2,\ell})\delta_{w_2,w_1+2} \\ & + C_{w_1,\ell}B_{w_1+4,\ell})\delta_{w_2,w_1+4}\ ,\end{aligned} \tag{26}$$

and the lower diagonal matrix elements can be computed from this equation.

The $\hat{W}^2\hat{\hat{W}}^2 + \hat{\hat{W}}^2\hat{W}^2$ operator is then computed as in Eq. (23)

$$\begin{aligned}\langle [N]; w_2\ell|\hat{W}^2\hat{\hat{W}}^2 + \hat{\hat{W}}^2\hat{W}^2|[N]; w_1\ell\rangle = & 2\omega_1(\omega_1+1)[\hat{\hat{W}}^2]_{w_1,w_1}\,\delta_{w_2,w_1} \\ & + [\omega_1(\omega_1+1) + (\omega_1+2)(\omega_1+3)]\,[\hat{\hat{W}}^2]_{w_1,w_1+2}\delta_{w_2,w_1+2} \\ & + [(\omega_1-2)(\omega_1-1) + \omega_1(\omega_1+1)]\,[\hat{\hat{W}}^2]_{w_1,w_1-2}\delta_{w_2,w_1-2} \\ & + [\omega_1(\omega_1+1) + (\omega_1+4)(\omega_1+5)]\,[\hat{\hat{W}}^2]_{w_1,w_1+4}\delta_{w_2,w_1+4} \\ & + [(\omega_1-4)(\omega_1-3) + \omega_1(\omega_1+1)]\,[\hat{\hat{W}}^2]_{w_1,w_1-4}\delta_{w_2,w_1-4}\ ,\end{aligned} \tag{27}$$

where $[\hat{n}^2]_{w_i,w_j}$ are the $\hat{n}^2$ operator matrix elements.

3 Programs included in the package

We proceed to describe the different programs provided. Compiled `Fortran` programs have names with the suffix `_gfortran` or `_ifort` depending on the compiler used.

The subroutine used for matrix diagonalization is `LA_SYEVR` from the `Lapack95` [**LAPACK95**], the `Fortran 95` interface to `Lapack` [**laug**].

3.1 `Fortran` programs to compute eigenvectors, eigenvalues, and other quantities

3.1.1 Programs `avec_U3_U2_mh` and `avec_U3_SO3_mh`

The programs `avec_U3_XX_mh` compute, and eventually save, eigenvalues and eigenvectors of the model Hamiltonian (13) using the `XX = U2` -Eq. (7)- or `SO3` -Eq. (9)- basis. The input file for this program is of `namelist` type with the following syntax

```
#
#       INPUT FILE FOR 2DVM U(3) Calculations
#       Model Hamiltonian
&par_aux Iprint = 1, Eigenvec_Log = .T., Excitation_Log = .F., Save_avec_Log = .T.  /
&par_0   N_val = 20, L_val = 0 /
&par_1   epsilon = 1.0,   xi = 0.4 /
```

And the different variables are:

- `Iprint` *Integer variable*. Control program verbosity.
- `Eigenvec_Log` *Logical variable* If the variable is `.True.` eigenvalues and eigenvectors are computed.
- `Excitation_Log` *Logical variable* If the variable is `.True.` excitation energies wrt the ground state energy are reported.
- `Save_avec_Log` *Logical variable* If the variable is `.True.` eigenstates are saved in a file with the name `eigvec_XX_NYY_LZZ.dat` where `XX = u2/so3`, `YY = N_val`, and `ZZ = L_val`.
- `N_val` *Integer variable* Totally symmetric irrep of $U(3)$. Determines the Hilbert space size.
- `L_val` *Integer variable* Vibrational (2D) angular momentum ℓ value.
- `epsilon` *Real (DP) variable* Model Hamiltonian global energy scale ε in Eq.(13).
- `xi` *Real (DP) variable* Model Hamiltonian control parameter, $0 \leq \xi \leq 1$ defined in Eq. (13).

The result of running the program using the `U2` basis with the previous input file is

```
$ ../bin/avec_U3_U2_mh_gfortran < u3_modham.inp
 L_val =            0
           0   4.7178381890544925
  -6.6913473582377064E-002 |           0 >
  0.26757490446002280      |           2 >
 -0.50918508872650692      |           4 >
  0.59782591356876802      |           6 >
 -0.47489555175329112      |           8 >
  0.26470245150057914      |          10 >
 -0.10411984650838124      |          12 >
   2.8307962444698111E-002 |          14 >
  -5.0262214812812631E-003 |          16 >
   5.1057145041144211E-004 |          18 >
  -1.9360623933001603E-005 |          20 >
.
.
.
          20   20.560646361706880
   4.8846592744607648E-008 |           0 >
   7.4751184587909931E-007 |           2 >
   6.4855302773322584E-006 |           4 >
   4.1763856737063440E-005 |           6 >
   2.2180851797258004E-004 |           8 >
   1.0276743631749022E-003 |          10 >
   4.3044673503380383E-003 |          12 >
   1.6740682624946063E-002 |          14 >
   6.2024723264416194E-002 |          16 >
  0.22776900283209650      |          18 >
  0.97158341333025411      |          20 >
```

A file named `eigvec_u2_N20_L0.dat` is created. This file includes the energies, the Hamiltonian matrix diagonal[1] and the eigenstates in column-major order and it has the following structure, where `XX = u2` or `so3`, depending on the basis selected.

```
# N =           N_val ; L =            L_val  ; XX basis
# Eigenvalues
  E_1       E_2    ....... E_dim
# Hamiltonian Diagonal
  H_11      H_22    ...... H_dimdim
  Psi_1_1   Psi_2_1 ...... Psi_dim_1
  ..................................
  Psi_dim_1 Psi_dim_2 .... Psi_dim_dim
```

3.1.2 Programs `avec_U3_U2` and `avec_U3_SO3`

The Programs `avec_U3_XX` compute the 2D Vibron Model energies and eigenstates using the XX chain for fixed parameters of -Eq. (14). The input file is similar to the one in the last section:

```
#
#       INPUT FILE FOR U(3) 2 Body Hamiltonian Calculations
#
&par_aux Iprint = 0, Eigenvec_Log = .T., Excitation_Log = .T., Save_avec_Log = .F.  /
&par_0   N_val = 10, L_val = 0 /
&par_1   epsilon_p=0.8, alpha_p=-0.000667779632721202, beta_p=0.0, A_p=0.0003339 /
```

The new elements in this input file are ε, α, β and A of the -Eq. (14). To run this program using an `UNIX` terminal:

```
$ [...]/bin/avec_U3_XX_gfortran < u3_ham.inp
```

where `[...]` is the path of the program's `bin` folder. The output of running it with the given input is:

```
 0   0.0000000000000000
 2   1.5859642569051831
 4   3.1719656273638952
 6   4.7579771193315414
 8   6.3439717420554338
10   7.9299225026296547
```

3.1.3 Programs `ipr_U3_U2_mh` and `ipr_U3_SO3_mh`

The programs `ipr_U3_XX_mh` compute the energies and the Inverse Participation Ratio (IPR) in the XX chain for -Eq.(13) Hamiltonian. The input looks like the last one:

```
#
#       INPUT FILE FOR U(3) Model Hamiltonian Calculations
```

[1] In case `Excitation_Log = .True.` the ground state energy is also substracted from the Hamiltonian matrix diagonal.

```
#
&par_aux Iprint = 2, Eigenvec_Log = .T., Excitation_Log = .T. /
&par_0   N_val = 15, L_val = 0 /
&par_1   epsilon = 1.0,   xi = 0.75 /
```

Introducing

```
$[...]/bin/ipr_U3_U2_mh_gfortran < ipr_u3_modham.inp
```

in the command line, it is given back the following output:

```
Reading  N_val, L_val
Reading  epsilon, xi
 Iprint =  2; Eigenvec_LOG =  T; Excitation_Log =  T
N_val =     15; L_val =      0
epsilon =  1.0000000E+00; xi =  7.5000000E-01
Operator number           1
Operator number           2
GS_energy =    1.7321844551756294
L_val =            0
          1   0.0000000000000000        3.5165700354623559
          2   3.0929130257955548        4.4667740770364324
          3   5.7488037073509926        4.8502861663079422
          4   7.9586934507799825        5.5371533436156843
          5   9.6925122152480512        4.1265672238584008
          6   10.908945553417363        4.9730348558314166
          7   12.086226648623684        5.5191888106052103
          8   13.654429757379329        2.7538257245146016
```

In this case, the verbosity has been increased using IPRINT=2.

3.1.4 Program ipr_Husimi_U3_U2_mh

3.1.5 Programs ipr_U3_U2 and ipr_U3_SO3

The ipr_U3_XX program is similar to ipr_U3_XX_mh. The main difference is the input file, because of the Hamiltonian used is -Eq. (14) instead of -Eq.(13). In this case, an example of the input file could be:

```
#
#       INPUT FILE FOR U(3) Model Hamiltonian Calculations
#
&par_aux Iprint = 2, Eigenvec_Log = .T., Excitation_Log = .T., Save_avec_Log = .F./
&par_0   N_val = 30, L_val = 0 /
&par_1   epsilon_p = 1.0, alpha_p = 0.0, beta_p = 0.0, A_p = 0.0 /
```

Running it in the $U(2)$ chain, we can analyze the output to test the program:

```
$ [...]/bin/ipr_U3_U2_gfortran < ipr_u3_ham.inp

 Reading  N_val, L_val
 Reading  Hamiltonian Paramenters
  Iprint =  2; Eigenvec_LOG =  T; Excitation_Log =  T; Save_Avec_Log =  F
 N_val =     30; L_val =      0
 epsilon =  1.0000000E+00; alpha =  0.0000000E+00 ; beta =  0.0000000E+00; A =  0.0000000
 Operator number           1
 Operator number           2
 Operator number           3
 Operator number           4
 GS_energy =    0.0000000000000000
 L_val =            0
           1   0.0000000000000000        1.0000000000000000
           2   2.0000000000000000        1.0000000000000000
           3   4.0000000000000000        1.0000000000000000
           4   6.0000000000000000        1.0000000000000000
           5   8.0000000000000000        1.0000000000000000
           6   10.000000000000000        1.0000000000000000
           7   12.000000000000000        1.0000000000000000
           8   14.000000000000000        1.0000000000000000
           9   16.000000000000000        1.0000000000000000
          10   18.000000000000000        1.0000000000000000
          11   20.000000000000000        1.0000000000000000
          12   22.000000000000000        1.0000000000000000
          13   24.000000000000000        1.0000000000000000
          14   26.000000000000000        1.0000000000000000
          15   28.000000000000000        1.0000000000000000
          16   30.000000000000000        1.0000000000000000
```

The output consist of three columns: an integer one, the eigenvalues of $\hat{n}$ and the `IPR`. The last column was expected because $\hat{n}$ eigenstates are well located in the $U(2)$ basis (they match basis' components). Now, if we compute it using $SO(3)$ basis,

```
$ [...]/bin/ipr_U3_SO3_gfortran < ipr_u3_ham.inp

 Reading  N_val, L_val
 Reading  Hamiltonian Paramenters
  Iprint =  2; Eigenvec_LOG =  T; Excitation_Log =  T; Save_Avec_Log =  F
 N_val =     30; L_val =      0
 epsilon =  1.0000000E+00; alpha =  0.0000000E+00 ; beta =  0.0000000E+00; A =  0.0000000
 Operator number           1
 Operator number           2
 Operator number           3
 Operator number           4
 GS_energy =   -7.1054273576010019E-015
 L_val =            0
           1   0.0000000000000000        6.2158795395844608
           2   2.0000000000000142        7.8800707358483129
           3   4.0000000000000107        8.8145249866016897
           4   6.0000000000000213        9.3394577175011282
           5   8.0000000000000284         9.5645594435444661
           6   10.000000000000000         9.5200527843596721
           7   12.000000000000014         9.1667631466188073
           8   14.000000000000014         8.2391939350606123
           9   16.000000000000014         8.2352287186286670
          10   18.000000000000007         9.1475889006614128
          11   20.000000000000021         9.4784447604423558
          12   22.000000000000011         9.4915382955106420
          13   24.000000000000014         9.2204419120130368
          14   26.000000000000011         8.6203298376166941
          15   28.000000000000004         7.5323034205731147
          16   30.000000000000007         5.3098659835781525
```

we obtain the same result excluding the IPR column. In the $SO(3)$ basis, the $\hat{n}$ eingenstates are not well located.

3.1.6 Program `ipr_4b_U3_U2` and `ipr_4b_U3_SO3`

If the reader needs more precision, `ipr_4b_U3_XX` program is available to compute the energies, eigenvectors and IPR in both chains considering higher interactions of the Hamiltonian. Here, the difference is the same than in the last case: the input file.

```
 &INP1 N_val=10, L_val=1, IPRINT=1, Eigenvec_log = .TRUE., Excitation_Log = .True., Save_
# Hamiltonian Parameters
 &INP1b P11=2378.0D0 /
 &INP2b P21=-37.597D0, P22=19.312D0, P23=-9.6422D0 /
 &INP3b P31=0.0D0, P32=0.0D0, P33=0.0D0 /
 &INP4b P41=0.0D0, P42=0.0D0, P43=0.0D0, P44=0.0D0, P45=0.0D0, P46=0.0D0, P47=0.0D0 /
```

P_{ij} elements correspond to the parameters of the Hamiltonian in the -Eq. (16). To run this program in the $SO(3)$ chain, for example, write the following commands in the terminal:

```
$ echo ipr_4b.inp | [...]/bin/ipr_4b_U3_SO3_gfortran
```

The output in this case is

```
GS_energy =   -200.60192731579809
L_val =            0
          1   0.0000000000000000         3.7732656477725937
          2   4311.9024599974764         4.1827037058586312
          3   8488.9872790600057         3.5354457958151770
          4   12523.812560924473         3.8730517993482945
          5   16407.548813126232         3.9800419071796886
          6   20129.470450786630         3.1403615181153479
```

3.2 Programs to fit experimental vibrational bending spectra

3.2.1 Program `chi2_U3`

`2D Vibron Model Program Suite` includes a `chi2_U3` program to compute the χ^2 and use it in different minimization methods. This package provides two methods: one of them is implemented in `Python` and the other has been written in `FORTRAN 90`.

1. Minimization using the `Python-script`
2. Minimization using `Minuit-CERN` package [**minuit**]

   In this section we are going to explain how to use this minimization method, where `Minuit-CERN` package has been used. First of all we need to collect the experimental data into a file. In this example we use the `hydrogen isocyanide` (HNC) molecule.

   ```
   19
   462.72    1.0   1    1
   926.50    1.0   2    0
   936.05    1.0   2    2
   1398.56   1.0   3    1
   1419.97   1.0   3    3
   1867.05   1.0   4    0
   1878.72   1.0   4    2
   1913.87   1.0   4    4
   2341.84   1.0   5    1
   2366.83   1.0   5    3
   2417.57   1.0   5    5
   2809.29   1.0   6    0
   2822.75   1.0   6    2
   2863.11   1.0   6    4
   2930.90   1.0   6    6
   3281.50   1.0   7    1
   3309.78   1.0   7    3
   3367.37   1.0   7    5
   3453.78   1.0   7    7
   ```

   The first line must be the total number of experimental data. After that, the file is structured in 4 columns. The first column corresponds with the energy, the next one to the experimental error of data and, the last two are the quantum numbers. If you want to include any data in the prediction but not at the fitting time, you can set the experiemntal error to 0.0 . The energy's label must be in accordance with the nature of the molecule: linear, $U(2)$ basis $|n\ell\rangle$, or bent, $SO(3)$ $\left|\nu_b{}^{\ell}\right\rangle$.

   Once finished, the next step is to construct the input file:

   ```
    &INP0 BENT=.F., exp_data_file='exp_HNC.dat', output_file_0="output_HNC" /
    &INP1 N_val=40, LMAX=7, VMAX=7, EMINL=.F. /
   #
   # PARAMETERS THAT TUNE THE OUTPUT
   #
    &INP2 IPRINT=0, DIS_RES = .F. /
    &INP1b P11=740.0D0 /
   ```

```
&INP2b P21=-3.597D0, P22=1.5D0, P23=-7.2D0 /
&INP3b P31=0.0D0, P32=0.002D0, P33=0.0D0 /
&INP4b P41=0.0D0, P42=0.0D0, P43=0.0D0, P44=0.0D0, P45=0.0D0, P46=0.0D0, P47=0.0D0
&fix_par fixed_par = ’fix  5 7 8 9 10 11 12 13 14’ /

# P11 ---> 1
# P21 ---> 2
# P22 ---> 3
# P23 ---> 4
# P31 ---> 5
# P32 ---> 6
# P33 ---> 7
# P41 ---> 8
# P42 ---> 9
# P43 ---> 10
# P44 ---> 11
# P45 ---> 12
# P46 ---> 13
# P47 ---> 14
```

Most of definitions have been explained in previous sections, the others mean:

- `BENT`: `.T.` (`.F.`) if the molecule is bent (linear).
- `exp_data_file`: The name of the experimental energies’ file.
- `output_file_0`: To choose the name of the file where the fit report will be saved.
- `LMAX`: The highest angular momentum number ℓ in `exp_data_file`.
- `VMAX`: The highest harmonic number n for the linear case or the bending number v_b when `BENT` is true.
- `DIS_RES`: Match as true (.T.) to display residuals energies.
- P_{ij}: Initial value of the parameter P_{ij}
- `fixed_par`: `Minuit` order. This string of characters consists of different integers that are associated with the P_{ij} parameters. If one of them appears, `Minuit` will keep it fixed.

If the program `Min_minuit_gfortran` is executed as mentioned:

```
$ echo fit.inp | ../bin/Min_minuit_gfortran

Note: The following floating-point exceptions are signalling: IEEE_INVALID_FLAG IEEE

$ tail -n 43 output_HNC

  EXT PARAMETER                   APPROXIMATE        STEP         FIRST
  NO.   NAME        VALUE            ERROR          SIZE       DERIVATIVE
   1      P11        1413.8        0.37063       0.67417E-03  -0.14902
   2      P21       -29.832        0.15233E-01   0.14225E-04   -1.8667
   3      P22        15.807        0.10429       0.75373E-05  -0.14985
   4      P23       -8.0528        0.28648E-02   0.38399E-05   -13.647
   5      P31        0.0000          fixed
   6      P32       0.48757E-01    0.95612E-02   0.65872E-06   -1.3627
   7      P33         0.0000         fixed
   8      P41         0.0000         fixed
   9      P42         0.0000         fixed
  10      P43         0.0000         fixed
  11      P44         0.0000         fixed
  12      P45         0.0000         fixed
  13      P46         0.0000         fixed
  14      P47         0.0000         fixed

 EXTERNAL ERROR MATRIX.    NDIM=  50    NPAR=  5    ERR DEF=  1.00
  0.137E+00-0.432E-02-0.926E-03-0.915E-03 0.200E-03
 -0.432E-02 0.232E-03 0.563E-03 0.154E-04-0.537E-04
 -0.926E-03 0.563E-03 0.109E-01-0.862E-04-0.978E-03
 -0.915E-03 0.154E-04-0.862E-04 0.821E-05 0.628E-05
  0.200E-03-0.537E-04-0.978E-03 0.628E-05 0.914E-04
 ERR MATRIX NOT POS-DEF

 PARAMETER  CORRELATION COEFFICIENTS
       NO.  GLOBAL      1      2      3      4      6
        1  0.99666  1.000-0.766-0.024-0.861 0.056
        2  0.98707 -0.766 1.000 0.355 0.353-0.369
        3  0.99201 -0.024 0.355 1.000-0.288-0.980
        4  0.99374 -0.861 0.353-0.288 1.000 0.229
        6  0.99086  0.056-0.369-0.980 0.229 1.000
 ERR MATRIX NOT POS-DEF
minuit>  **********
 **    4 **CALL    3.000
 **********
  -----------------------------------
 | rms =    7.9576384169698269E-002  |
 |___________________________________|
minuit>  **********
 **    5 **EXIT
 **********
 ..........MINUIT TERMINATED BY MINUIT COMMAND: EXIT

$
```

3.3 Perl scripts