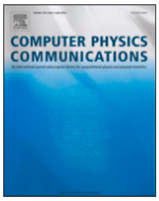
Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



A general code for fitting global potential energy surfaces via CHIPR method: Direct-Fit Diatomic and tetratomic molecules[☆],[☆☆]

C.M.R. Rocha[a,1], A.J.C. Varandas[a,b,*,1]

[a] Department of Chemistry and Coimbra Chemistry Centre, University of Coimbra 3004-535 Coimbra, Portugal
[b] School of Physics and Physical Engineering, Qufu Normal University, 273165, PR China

ARTICLE INFO

Article history:
Received 29 January 2020
Received in revised form 16 July 2020
Accepted 3 August 2020
Available online 26 August 2020

Keywords:
Potential energy surfaces
Global potential functions
CHIPR

ABSTRACT

A general program to fit global adiabatic potential energy surfaces of up to tetratomic molecules to *ab initio* points and available spectroscopic data for simple diatomics is reported. It is based on the Combined-Hyperbolic-Inverse-Power-Representation (CHIPR) method. The final form describes all dissociating fragments and long-range/valence interactions, while obeying the system permutational symmetry. The code yields as output a Fortran 90 subroutine that readily evaluates the potential and gradient at any arbitrary geometry.

Program summary

Program title: CHIPR-4.0
CPC Library link to program files: http://dx.doi.org/10.17632/8wdv87gt5x.2
Licensing provisions: GPLv3
Programming language: Fortran 90
Journal reference of previous version: C. M. R. Rocha, A. J. C. Varandas, Comput. Phys. Commun. 247 (2020) 106913
Does the new version supersede the previous version?: Yes
Reasons for the new version: CHIPR-4.0 is a major extension of the previous CHIPR-3.0 code [1] which, besides diatomics and triatomics, also includes the possibility of fitting global adiabatic potential energy surfaces (PESs) of A_4-, AB_3-, A_2B_2-, ABC_2- and $ABCD$-type tetratomic molecules [3]. Additionally, a new feature is that it allows the user to fine-tune ab initio diatomic curves using available spectroscopic data.

Summary of revisions:

- Implementation of CHIPR's polynomial form and permutation operators for tetratomic molecules
- Implementation of a direct-fit module to fine-tune ab initio potential energy curves using spectroscopic data
- Automatic global minimum search on the diatomic, triatomic or tetratomic final PESs
- Automatic harmonic vibrational analysis at the optimum molecular geometries

Nature of problem: This version of the CHIPR code, CHIPR-4.0, provides a set of subroutines to fit global adiabatic potential energy surfaces of up to tetratomic molecules using *ab initio* and (for diatomics) optional fine-tuning experimental data.
Solution method: CHIPR-4.0 uses the Combined-Hyperbolic-Inverse-Power-Representation (CHIPR) [2] method to interpolate and extrapolate *ab initio* data points.
Additional comments including restrictions and unusual features: For triatomic and tetratomic fits, the user must supply subroutines containing the analytic forms defining the sum of two- and two-plus-three-bodies, respectively. These are utilized internally to calculate (from *ab initio* data) the corresponding three- and four-body energies to be fitted. The current version of this code is strictly

[☆] The review of this paper was arranged by Prof. Stephan Fritzsche.

[☆☆] This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

[*] Corresponding author at: Department of Chemistry and Coimbra Chemistry Centre, University of Coimbra 3004-535 Coimbra, Portugal.
E-mail addresses: carlosmurilorocha@gmail.com (C.M.R. Rocha), varandas@uc.pt (A.J.C. Varandas).

[1] For due purposes, the authors declare to have contributed equally.

https://doi.org/10.1016/j.cpc.2020.107556

based on single-sheeted analytical PESs. High-accuracy in diatomic refinements is currently limited to the $^1\Sigma$ single-curve case.

References

[1] C. M. R. Rocha, A. J. C. Varandas, Comput. Phys. Commun. 247 (2020) 106913 DOI: 10.1016/j.cpc.2019.106913

[2] A. J. C. Varandas, J. Chem. Phys. 138 (2013) 054120 DOI: 10.1063/1.4788912

[3] C. M. R. Rocha, A. J. C. Varandas, Phys. Chem. Chem. Phys. 21 (2019) 24406 DOI: 10.1039/C9CP04890A

1. Introduction

In previous work [1], we reported a general code, referred to as CHIPR-3.0, to fit global adiabatic potential energy surfaces (PESs) [2] of any diatomic and triatomic molecule using the Combined-Hyperbolic-Inverse-Power-Representation (CHIPR) method [3–8] and *ab initio* data points as the only calibrating set. It is our plan in this paper to extend CHIPR-3.0 [1] by providing a generalization capable of representing global adiabatic PESs of up to tetratomic species. Besides dealing with molecules of up to four atoms, a new feature has been added in CHIPR-4.0 to allow users to obtain experimentally-derived potential energy curves by performing a direct-fit to available spectroscopic data on simple diatomics [7,9–12]. All such new features are further complemented with improved modules and novel routines that perform automatic global minima search [13] and harmonic vibrational analyses [14] on the global minima so found, in addition to provide ready-to-use subroutines containing the fitted n-body terms. The present implementation of CHIPR, likewise the previous code [1], is strictly based on analytical single-sheeted PES forms thus leaving to the user the proper description of the non-analiticities associated with curve crossings and conical intersections; the outcome of our current pursuits in this direction [15–18] will be the subject of future implementations.

We should recall at this point that the term “global” refers throughout to a PES that describes from all united-atom limits up to all dissociation channels, including the complete molecular fragmentation $(A + B + C + D\ldots)$. Although other approaches exist in the literature that claim such a capability for up to tetratomics, and even larger systems (see list of references elsewhere [1]), only a few [1,19–22] appear to fulfill those requirements. The present work summarizes our effort to achieve such a desideratum with CHIPR.

2. The CHIPR method: tetratomic molecules

In the CHIPR method [3–8], the PES of a tetratomic molecule assumes the following many-body expansion (MBE) [2] form

$$V(\mathbf{R}) = \mathcal{V}_{\mathcal{S}}^{(1)} + \mathcal{V}_{\mathcal{S}}^{(2)}(\mathbf{R}) + \mathcal{V}_{\mathcal{S}}^{(3)}(\mathbf{R}) + V_{\mathcal{S}}^{(4)}(\mathbf{R}), \tag{1}$$

where $\mathcal{V}_{\mathcal{S}}^{(1)}$ is the sum of (pseudo-) one-body fragments,

$$\mathcal{V}_{\mathcal{S}}^{(2)}(\mathbf{R}) = \sum_{i=1}^{6} V_i^{(2)}(R_i) \tag{2}$$

is the sum of two-bodies,

$$\mathcal{V}_{\mathcal{S}}^{(3)}(\mathbf{R}) = \sum_{\{R_i,R_j,R_k\}\subset\mathbf{R}}^{4} V_{ijk}^{(3)}(R_i, R_j, R_k) \tag{3}$$

is the sum of three-body potentials and $V_{\mathcal{S}}^{(4)}(\mathbf{R})$ accounts for the four-body interaction; $\mathbf{R}=\{R_1, \ldots, R_6\}$ is the set of all interatomic separations. Note that $\mathcal{V}_{\mathcal{S}}^{(1)}$ is the sum of atom energies in the state produced by adiabatically removing them from the cluster. If the zero of energy is chosen with all atoms in their ground electronic states, $\mathcal{V}_{\mathcal{S}}^{(1)}$ is only non-zero when, on dissociation, one (or some) is (are) left in an excited state [23–26]. In this case, an adequate switching-function formalism [23–26] must be ensured to transform the multi-sheeted PES into a single-sheeted one: the only case here considered, although a generalization to multi-sheeted PESs should be straightforward if all terms in the potential matrix assume the CHIPR form [18].

As in MBE [2] and double many-body expansion (DMBE) [19] theories, $V_{\mathcal{S}}^{(n)}(\mathbf{R})$ must be suitably written to reflect the appropriate permutational symmetry of the full molecular aggregate or supermolecule (which is implied by the subindex $\mathcal{S}$). For the two-body case, this condition is naturally warranted whenever all diatomic potentials are introduced into the summation $\mathcal{V}_{\mathcal{S}}^{(2)}(\mathbf{R})$. Yet, for higher-order terms ($n=3, 4$), permutational invariance is generally ensured by built-in construction via use of symmetrized sums of monomials (see below) [5,27]. Note that, similarly to the MBE/DMBE approaches, all n-body terms in Eq. (1) should vanish as long as an atom moves away from the n-atom aggregate [5].

For a N-atom system, the various CHIPR n-body terms assume the general form [3–5]

$$V_{\mathcal{S}}^{(n)}(\mathbf{R}^n) = \sum_{\substack{i_1=0,\\ \ldots,i_\tau=0}}^{L} C_{i_1,\ldots,i_\tau}\,\hat{\mathcal{P}}\left[\prod_{p=1}^{\tau} y_p^{i_p}\right], \tag{4}$$

where $C_{i_1,\ldots,i_\tau}$ are expansion coefficients of a Lth-order polynomial, $\hat{\mathcal{P}}$ is the operator that ensures permutational symmetry, and the y_p's are the set of $p = 1, 2, \ldots \tau$ coordinates relative to some reference geometry; τ is the total number of $n(n-1)/2$ internal (independent) degrees of freedom on which the subset $\mathbf{R}^n$ depends. As in electronic structure theory [28], every y_p in Eq. (4) is in turn expanded in terms of a distributed-origin contracted basis set [1,3–5]

$$y_p = \sum_{\alpha=1}^{M} c_\alpha\,\phi_{p,\alpha}, \tag{5}$$

where c_α are contraction coefficients, with α defining the index of the each primitive function $\phi_{p,\alpha}$. The latter assume one of the two following convenient forms [1,3–5]:

(i) if long-range terms are ignored, the basis functions are set up as

$$\phi_{p,\alpha} = \operatorname{sech}^{\eta_\alpha}\left(\gamma_{p,\alpha}\rho_{p,\alpha}\right). \tag{6}$$

(ii) if long-range energies are envisaged, an advantageous form is

$$\phi_{p,\alpha} = \left[\frac{\tanh\left(\beta_\alpha R_p\right)}{R_p}\right]^{\sigma_\alpha} \operatorname{sech}^{\eta_\alpha}\left(\gamma_{p,\alpha}\rho_{p,\alpha}\right). \tag{7}$$

In the above equations, $\rho_{p,\alpha}=R_p-R_{p,\alpha}^{\mathrm{ref}}$ (the deviation of the coordinate R_p from the primitive origin $R_{p,\alpha}^{\mathrm{ref}}$) and $\gamma_{p,\alpha}$ are non-linear parameters; $\eta_\alpha \equiv \eta = 1$, $\sigma_\alpha \equiv \sigma = 6$ and $\beta_\alpha \equiv \beta = 1/5$ are constants [3–5]. Likewise CHIPR-3.0 [1], both primitives in Eqs. (6) and (7) are utilized for basis set contraction in the present implementation of the program, with the latter appearing only once as the last term in the expansion of Eq. (5); see, *e.g.*, the `BASIS_SET.f90` module inside the "CHIPR-4.0_SOURCE_CODE/" directory.

Finally, to reduce linear dependencies, the distributed origins ($R^{\text{ref}}_{p,\alpha}$) of the various basis functions defined in Eq. (5) are assumed to be related by

$$R^{\text{ref}}_{p,\alpha} = \zeta (R^{\text{ref}}_{p})^{\alpha-1}, \tag{8}$$

where ζ and R^{ref}_{p} are two judiciously chosen constants. As noted elsewhere [3], the above two-parameter relation resembles the even-tempered algorithm for one-electron basis sets [29–31] in electronic structure theory.

3. CHIPR-4.0: program implementation up to tetratomics

In the following, we specialize in Eq. (1) and provide an overview of CHIPR-4.0's implementation of Eq. (4) for the case n=1, 2, 3, 4. Note that, although the discussion is mainly devoted to tetratomics (`NATOM=4`) – the major novelty of this version – , we summarize for completeness the major features of CHIPR-4.0 as done in the previous code [1] for bare diatomic (`NATOM=2`) and triatomic (`NATOM=3`) molecules. Of course, for the former, a new feature is the implementation in CHIPR-4.0 of the possibility of obtaining experimentally fine-tuned potential energy curves.

3.1. One-body terms

At the current program version, $\mathcal{V}^{(1)}_S$ in Eq. (1) is automatically set to zero. If this is not the case (see, *e.g.*, Refs. [23–26]), the user must perform proper adaptations.

3.2. Two-body terms (diatomic molecules)

The modeling of diatomic curves (`NATOM=2`) is here done using Eq. (4) once the required Coulomb interaction is included [3]

$$V^{(2)}(R) = \frac{Z_\text{A} Z_\text{B}}{R} \sum_{k=1}^{L} C_k\, y^k, \tag{9}$$

where Z_A and Z_B denote the nuclear charges of atoms A and B. The implementation of the above equation can be found in the `CHIPR_DIAT.f90` module. Note that, for the two-body case, it is generally a good choice to perform simultaneously the optimization of both basis set [Eqs. (5)–(7)] and polynomial [Eq. (9)] by setting directly `JOBTYP=FITPOL`. In this case, the target function to be minimized is

$$\chi^2 = \sum_{I=1}^{N} \left\{ W_I \left[V^{(2)}(R_I; C_k, c_\alpha, R^{\text{ref}}, \zeta) - E(R_I) \right] \right\}^2, \tag{10}$$

where W_I is the least-squares weight of the Ith calculated point at R_I, and $E(R_I)$ the corresponding *ab initio* energy given with respect to the reference `EZERO`.

As noted above, for the `NATOM=2` case, a novel capability of CHIPR-4.0 is to allow fine-tuning a purely *ab initio* curve through a direct-fit procedure of spectroscopic data [7,9,10]. This is done with the option `JOBTYP=DIRECTFIT`; see self-explanatory illustration in Fig. 2 for the singlet ground electronic state of dicarbon, $\text{C}_2(X\,^1\Sigma_g^+)$. Besides the "`coeffs_guess.txt`" (for `JOBTYP=OPTBASIS` and `FITPOL`), "`abinitio_data.txt`" and "`mol_gm_guess.txt`" (for `JOBTYP=FITPOL`) input data files, an additional file "`spec_data.txt`" containing all relevant spectroscopic (experimental) information is required; see Section 9 for its full description. The computed parameters of the CHIPR diatomic form are then obtained via minimization of the following objective function [7,9,10]

$$\begin{aligned}\chi^2 = {} & \sum_{I=1}^{N_{abinitio}} \left\{ W_I \left[V^{(2)}(R_I; C_k, c_\alpha, R^{\text{ref}}, \zeta) - E(R_I) \right] \right\}^2 \\ & + \sum_{J=1}^{N_{(ro)vib}} \left[W_J \left(\nu_J^{\text{calc}} - \nu_J^{\text{exp}} \right) \right]^2 + \\ & \left[W \left(w_e^{\text{calc}} - w_e^{\text{exp}} \right) \right]^2 + \left[W \left(D_e^{\text{calc}} - D_e^{\text{exp}} \right) \right]^2 \\ & + \left[W \left. \frac{dV^{(2)}(R; C_k, c_\alpha, R^{\text{ref}}, \zeta)}{dR} \right|_{R=R_e^{\text{exp}}} \right]^2, \end{aligned} \tag{11}$$

where the first term runs over all *ab initio* data [*i.e.*, it reduces to Eq. (10)] while the remaining terms ensure in an obvious correspondence that the final curve fits also the experimental (ro)vibrational levels (ν_J^{exp}), harmonic frequency (w_e^{exp}) and dissociation energy (D_e^{exp}) at the equilibrium geometry (R_e^{exp}) of the target species; the associated W's are specified by the user in "`spec_data.txt`". Note that, in this current version, CHIPR-4.0 uses an adaption of the LEVEL program [32] to perform all bound-state calculations. This procedure is then repeated, together with force constant, D_e and $dV^{(2)}/dR|_{R=R_e^{\text{exp}}}$ evaluations, for each step in the Levenberg–Marquardt algorithm [9,33]; see, *e.g.*, `TOFIT.f90` module.

A word of caution is in order regarding the selection of cutoff values for $N_{(ro)vib}$ in Eq. (11). Besides the availability of experimental data itself, some system-dependent aspects must be considered *a priori* by the users like the presence of avoided or even true crossings, as is the case of $\text{C}_2(X\,^1\Sigma_g^+)$ [7]. This is so inasmuch as nonadiabatic effects are here mostly left to the users care.

To avoid any spurious feature in the final spectroscopic curve and make the direct-fit approach as fast as possible, the users are advised to first perform a polynomial optimization (`JOBTYP=FITPOL`) using the *ab initio* data points only and then employ the optimum coefficients extracted from there as initial guesses for the subsequent `JOBTYP=DIRECTFIT` run. Additionally, if there are poorly or sparsely covered regions by the *ab initio* data (as often is the case at repulsive and long-range distances), the users may find it useful to add some grid points taken from the analytic (purely) *ab initio* curves (with, obviously, lower weights) into the "`abinitio_data.txt`" data file as the rule of thumb to avoid unexpected features there.

Note that, although the inverse problem in CHIPR-4.0 [Eq. (11)] is general, it may benefit from data coming from both the inverse perturbation analysis (IPA) [36,37] and semi-classical Rydberg–Klein–Rees (RKR) [38–41] pointwise schemes, while being strictly dependent on input quantities (ν_J^{exp}, R_e^{exp}, w_e^{exp} and D_e^{exp}) typically derived from experiment. This contrasts with dPotFit [11] wherein multiple sets of measured experimental data are directly employed for such a task [11]. Additionally, by relying on LEVEL [32], high-accuracy in CHIPR-4.0 diatomic refinements is currently limited to the $^1\Sigma$ single-curve case; the user is addressed to, *e.g.*, dPotFit [11] and Duo [12] software suites when dealing with other cases.

3.3. Three-body terms (triatomic molecules)

Likewise CHIPR-3.0 [1], this version of the program determines three-body energies, $\epsilon^{(3)}$, internally as (see `READING.f90`)

$$\epsilon^{(3)}(\mathbf{R}) = E(\mathbf{R}) - \sum_{i=1}^{3} V_i^{(2)}(R_i), \tag{12}$$

Table 1
General constraints in the CHIPR three-body terms of Eq. (13) for A_3-, AB_2- and ABC-type molecules. The permutation elements g for each subgroup G of the symmetric group $\mathcal{S}_3$ are also shown. The particle numbering follows the convention adopted in Fig. 1.

Species	First summation of Eq. (13)	Second summation of Eq. (13)	
	Constraints	Element g[a]	Subgroup G of $\mathcal{S}_3$[a]
A_3	$i+j+k \le L$ $i+j+k \neq i \neq j \neq k$ $i \le j \le k$	(), (1,2), (1,3), (2,3), (1,2,3), (1,3,2)	{(), (1,2), (1,3), (2,3), (1,2,3), (1,3,2)}
AB_2	$i+j+k \le L$ $i+j+k \neq i \neq j \neq k$ $j \le k$	(), (1,2)	{(), (1,2)}
ABC	$i+j+k \le L$ $i+j+k \neq i \neq j \neq k$	()	{()}

[a] All permutations are given in cycle notation [42].

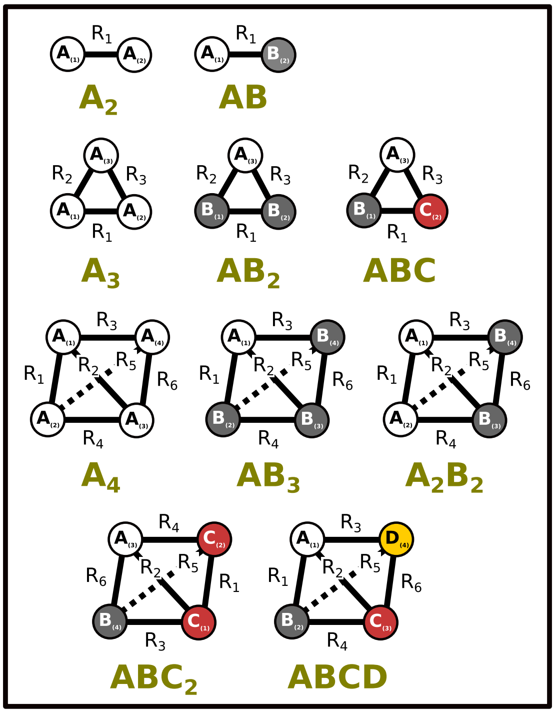


Fig. 1. Interparticle coordinate systems employed in CHIPR-4.0 for diatomic, triatomic and tetratomic molecules.

where E is the *ab initio* interaction energy (with respect to EZERO) for a triatomic molecule (or fragment) at $\mathbf{R}=\{R_1, R_2, R_3\}$, with the last term running over all (three) two-body potentials. If only three-body terms are envisaged (NATOM=3), the user must supply the appropriate analytic forms for the DIAT1(R1), DIAT2(R2), and DIAT3(R3) functions which are defined inside the SUM2BD.f90 module. This allows the program to calculate automatically the $\sum_{i=1}^{3} V_i^{(2)}(R_i)$ term [see, *e.g.*, the SUM2BD (R1,R2,R3,SUMV2) subroutine in SUM2BD.f90], and hence $\epsilon^{(3)}(\mathbf{R})$ in Eq. (12). Alternatively, instead of having to edit files inside "CHIPR-4.0_SOURCE_CODE/", a new functionality in CHIPR-4.0 allows users to include their own subroutine defining the sum of two-bodies in a corresponding Makefile and/or even alter SUM2BD.f90 locally (*i.e.*, in the user's working directory); this is described latter in connection with the compilation process and the use of the "config.csh" script. Note that, for the case of AB_2-type molecules, the coordinate R_1 (and y_1) must define the B-B diatomic; see Fig. 1 to assess all interparticle coordinate systems here considered.

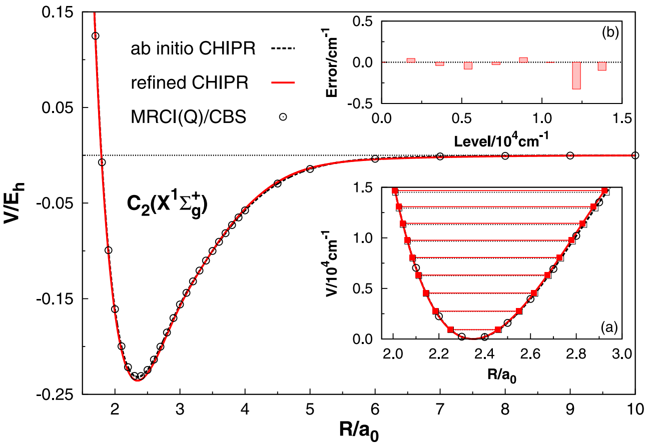


Fig. 2. Example plot of a CHIPR diatomic curve for $C_2(X\,^1\Sigma_g^+)$ calibrated solely with *ab initio* MRCI(Q)/CBS(D, T) energies from Ref. [7] via JOBTYP=FITPOL. Also shown is the corresponding experimentally-derived curve obtained [7] via direct-fit procedure in Eq. (11), JOBTYP=DIRECTFIT. This includes both *ab initio* and experimental data from Refs. [34] and [35]. Inset (a) shows the vibrational levels and associated classical turning points (open and filled squares) calculated [32] with both analytic forms. Inset (b) depicts the deviations between the predicted and observed levels for the refined C_2 curve. Both jobs are provided as test runs in "CHIPR-4.0.zip".

For a general triatomic species, the three-body CHIPR function of Eq. (4) assumes the form [3–5]

$$V^{(3)}(\mathbf{R}) = \sum_{i,j,k=0}^{L} C_{i,j,k} \left\{ \sum^{g\in G} \mathcal{P}_g^{(i,j,k)} \left[y_1^i y_2^j y_3^k \right] \right\}, \qquad (13)$$

where $C_{i,j,k}$ are expansion coefficients and $y_p\,(p = 1, 2, 3)$ the coordinates in Eq. (5). Following Ref. [4], all $C_{i,j,k}$ terms up to a given L are considered, subject to the general constraints: $i{+}j{+}k \neq i \neq j \neq k$ and $i+j+k \le L$. The first is especially chosen to ensure that at least two modes are excited, so as to leave out any two-body energies [3–5]. In Eq. (13), the second summation runs over all possible (permutation) elements $g \in G$, where G is a subgroup of the $\mathcal{S}_3$ symmetric group [42]. Thus, $\mathcal{P}_g^{(i,j,k)}$ are the corresponding operators that reflect the action of the atom permutation g on y_p's (or R_p's), and hence on the (exponent) set $\{i, j, k\}$. This generates therefore the required symmetrized sum of monomials, all invariant with respect to permutations of like atoms. Table 1 summarizes all such constraints of the CHIPR three-body term in Eq. (13) for all types of triatomic molecules; for more details, see Ref. [1].

In CHIPR, it is assumed that the poles due to the Coulomb interaction between the nuclei are self-contained in the two-body terms (note that point-charges interact pair-additively) so that the effects of ($n > 2$)-bodies at high repulsive regions of the PES (close to the united-atom limit) should be of little relevance for the fit. Thus, to damp the three-body terms there and to avoid unphysical holes, we multiply Eq. (13) by [3,5,7]

$$\mathcal{D}^{(3)}(\mathbf{R}) = \left[\prod_{i=1}^{3} h(R_i)\right]^{\xi}, \tag{14}$$

where $h(R_i) = \frac{1}{2}\{1+\tanh[\kappa(R_i - R_0)]\}$. This warrants that the three-body term vanishes for large negative displacements from a chosen reference $R_0 = 0.5\,\mathrm{a}_0$, while becoming unit for large positive ones; $\kappa = 100\,\mathrm{a}_0^{-1}$ and $\xi = 10$. All the above information, including Eq. (13) itself, are programmed in the `CHIPR_TRIAT.f90` module.

Differently from the two-body case, the calibration of Eq. (13) must be made in a two-step manner. First, one needs to set up the contracted basis sets of Eq. (5) by running `JOBTYP=OPTBASIS` (an accompanying "`coeffs_guess.txt`" input file is required for this option). This is done by making 1*D* fits along each symmetry unrelated degree of freedom, `DEG`. For A_3-, AB_2- and ABC-type molecules, `DEG` assumes the values of 1, 2, and 3, respectively. At this stage, the objective function to be minimized is (see `MAIN.f90`)

$$\chi_p^2 = \sum_{I=1}^{N_{basis}} \left\{ W_I \left[y_p(R_p; c_\alpha, R_p^{\mathrm{ref}}, \zeta) - \epsilon_I^{(3)}(\mathbf{R}) \right] \right\}^2, \tag{15}$$

where $p = \{1\}$, $\{1, 2\}$, and $\{1, 2, 3\}$ for A_3, AB_2 and ABC species, respectively. The summation runs therefore over all *ab initio* three-body energies included *per* coordinate dimension, N_{basis}.

With the optimum basis set parameters at hand, the determination of the polynomial coefficients in Eq. (13) can be done by setting `JOBTYP=FITPOL` (this requires the "`coeffs_guess.txt`", "`abinitio_data.txt`" and "`mol_gm_guess.txt`" input files). A good choice is to first set `OPTTYP=FIXBAS`, *i.e.*, keep these parameters frozen at the previously optimized values and complete the input row-vector for the $C_{i,j,k}$'s with unit values. The values so obtained will then be utilized as initial guesses for a subsequent optimization step in which all parameters including the c_α's can be freely varied (`OPTTYP`≠`FIXBAS`). The target function to be minimized is now

$$\chi^2 = \sum_{I=1}^{N_{pol}} \left\{ W_I \left[V^{(3)}(\mathbf{R}; C_{i,j,k}, c_\alpha, R_p^{\mathrm{ref}}, \zeta) \times \mathcal{D}^{(3)}(\mathbf{R}) - \epsilon_I^{(3)}(\mathbf{R}) \right] \right\}^2. \tag{16}$$

Note at this stage, and differently from `JOBTYP=OPTBASIS` runs, that the user can conveniently modify the W_I's by setting the proper form for the `WEIGHT` function in the `WEIGHT_FUNC.f90` module; see details in Sections 6 and 9. An example of a triatomic fit for $C_3(^3A')$ is shown [7] in Fig. 3.

3.4. Four-body terms (tetratomic molecules)

As noted above, a major asset of the present CHIPR-4.0 code is to allow PES fits of tetratomic molecules by setting `NATOM=4`. In this case, the program calculates the required four-body energies [$\epsilon^{(4)}$] internally as (see `READING.f90`)

$$\begin{aligned}\epsilon^{(4)}(\mathbf{R}) &= E(\mathbf{R}) - \left[\sum_{i=1}^{6} V_i^{(2)}(R_i) + \sum_{\{R_i,R_j,R_k\}\subset\mathbf{R}}^{4} V_{ijk}^{(3)}(R_i, R_j, R_k) \right] \\ &= E(\mathbf{R}) - V^{(2+3)}(\mathbf{R})\end{aligned} \tag{17}$$

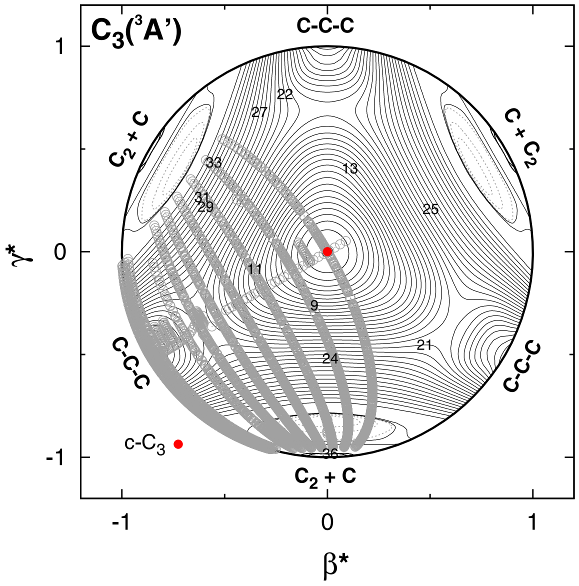


Fig. 3. Sample relaxed plot in hyperspherical coordinates [43] of the CHIPR triatomic PES of $C_3(^3A')$, where $\beta^* = 3^{1/2}(R_2^2 - R_3^2)/Q$, $\gamma^* = (2R_1^2 - R_2^2 - R_3^2)/Q$ and $Q = R_1^2 + R_2^2 + R_3^2$; $R_i (i = 1, 2, 3)$ are interparticle coordinates (Fig. 1). The PES is calibrated using *ab initio* MRCI(Q)/CBS(D, T) energies (gray circles) of Ref. [7] and `JOBTYP=FITPOL`. Also shown is the location of the global minimum, c-C_3, with D_{3h} symmetry; see Ref. [7] for other stationary points on the PES. Black solid and gray dashed lines are equally spaced by $0.005\,\mathrm{E_h}$ and $0.00015\,\mathrm{E_h}$, starting at $-0.5\,\mathrm{E_h}$ and $-0.23155\,\mathrm{E_h}$. This job is provided as test run in "CHIPR-4.0.zip".

where E is an *ab initio* interaction energy (with respect to `EZERO`) for the tetratomic molecule at $\mathbf{R} = \{R_1, R_2, R_3, R_4, R_5, R_6\}$, and the last term accounts for the sum of all (six) two- and (four) three-body potentials. Similarly to the triatomic case, if `NATOM=4`, the user must supply the latter truncated MBE form. Since the analytic function for $V^{(2+3)}(\mathbf{R})$ is generally much more complicated in form than the sum of two-bodies alone, a new feature has been added in CHIPR-4.0 that allows users to provide it as an external program (*e.g.*, `HC3_V23_DMBE.f`, `C2H2_V23.f`, or hypothetically, `POT23.f90`). This set of subroutines given by the user should contain all information needed for the function `SUM23BD(R1,R2,R3,R4,R5,R6,SUMV23)` (in module `SUM23BD.f90`) to evaluate $V^{(2+3)}$ at $\mathbf{R}$ so that $\epsilon^{(4)}(\mathbf{R})$ in Eq. (17) can be determined by the program. As explained later, this is done by placing `POT23.f90` and `SUM23BD.f90` (properly updated) in the user's directory (FOUR_BODY_FIT_TEST, say, assumed outside CHIPR-4.0_SOURCE_CODE/), and run there the "`config.csh`" script to obtain a proper `Makefile`.

For a general tetratomic molecule, the four-body CHIPR function of Eq. (4) assumes the form [8]

$$V^{(4)}(\mathbf{R}) = \sum_{i,j,k,l,m,n=0}^{L} C_{i,j,k,l,m,n} \left\{ \sum^{g\in G} \mathcal{P}_g^{(i,j,k,l,m,n)} \left[y_1^i y_2^j y_3^k y_4^l y_5^m y_6^n \right] \right\}, \tag{18}$$

where only the $C_{i,j,k,l,m,n}$'s that satisfy $i{+}j{+}k{+}l{+}m{+}n \leq L$ and excite at least four modes are included in the first summation. Other specific constraints follow depending on the type of tetratomic considered, being all listed for completeness in Table 2. As in the three-body case, $\mathcal{P}_g^{(i,j,k,l,m,n)}$ in Eq. (18) are permutation operators that reflect the action of the atom permutation $g \in G$ (G is now a subgroup of $\mathcal{S}_4$ [42]) on y_p's (or R_p's), and hence on the (exponent) set $\{i, j, k, l, m, n\}$. Again, these warrant permutation invariance to the polynomial form (18) by generating all required symmetrized sums of monomials [8]. In Table 2, we also summarize the possible permutation elements g and subgroups structure G of $\mathcal{S}_4$

Table 2
General constraints in CHIPR four-body terms of Eq. (18) for A_4-, AB_3-, A_2B_2-, ABC_2- and ABCD-type molecules. The permutation elements g for each subgroup G of the symmetric group $\mathcal{S}_4$ are also shown. The particle numbering follows the convention adopted in Fig. 1.

Species	First summation of Eq. (18)	Second summation of Eq. (18)	
	Constraints	Element g[b]	Subgroup G of $\mathcal{S}_4$[b]
A_4	$i+j+k+l+m+n \leq L$ $i+j+k+l+m+n \neq i \neq j \neq k \neq l \neq m \neq n$ $i+j+k+l+m+n \neq \forall(x+y+z)$[a] $i \leq j \leq k \leq l \leq m \leq n$	(), (3,4), (2,3), (2,3,4), (2,4,3), (2,4), (1,2), (1,2)(3,4), (1,2,3), (1,2,3,4), (1,2,4,3), (1,2,4), (1,3,2), (1,3,4,2), (1,3), (1,3,4), (1,3)(2,4), (1,3,2,4), (1,4,3,2), (1,4,2), (1,4,3), (1,4), (1,4,2,3), (1,4)(2,3)	{(), (3,4), (2,3), (2,3,4), (2,4,3), (2,4), (1,2), (1,2)(3,4), (1,2,3), (1,2,3,4), (1,2,4,3), (1,2,4), (1,3,2), (1,3,4,2), (1,3), (1,3,4), (1,3)(2,4), (1,3,2,4), (1,4,3,2), (1,4,2), (1,4,3), (1,4), (1,4,2,3), (1,4)(2,3)}
AB_3	$i+j+k+l+m+n \leq L$ $i+j+k+l+m+n \neq i \neq j \neq k \neq l \neq m \neq n$ $i+j+k+l+m+n \neq \forall(x+y+z)$[a] $i \leq j \leq k$ and $l \leq m \leq n$	(), (3,4), (2,3), (2,4), (2,3,4), (2,4,3)	{(), (3,4), (2,3), (2,4), (2,3,4), (2,4,3)}
A_2B_2	$i+j+k+l+m+n \leq L$ $i+j+k+l+m+n \neq i \neq j \neq k \neq l \neq m \neq n$ $i+j+k+l+m+n \neq \forall(x+y+z)$[a] $j \leq k \leq l \leq m$	(), (1,2), (3,4)	{(), (1,2), (3,4)}
ABC_2	$i+j+k+l+m+n \leq L$ $i+j+k+l+m+n \neq i \neq j \neq k \neq l \neq m \neq n$ $i+j+k+l+m+n \neq \forall(x+y+z)$[a] $j \leq l$ and $k \leq m$	(), (1,2)	{(), (1,2)}
ABCD	$i+j+k+l+m+n \leq L$ $i+j+k+l+m+n \neq i \neq j \neq k \neq l \neq m \neq n$ $i+j+k+l+m+n \neq \forall(x+y+z)$[a]	()	{()}

[a] The sum of all indices should not be equal to any of their possible three-term sum, *i.e.*, $i+j+k+l+m+n \neq i+j+k \neq i+j+l \neq i+j+m \neq \ldots \neq l+m+n$.
[b] All permutations are given in cycle notation [42].

permitted for each molecule type. To damp the four-body energy term at high repulsive regions, Eq. (18) is then multiplied by [3,8]

$$\mathcal{D}^{(4)}(\mathbf{R}) = \left[\prod_{i=1}^{6} h(R_i)\right]^{\xi}, \tag{19}$$

where all quantities assume a similar meaning as in Eq. (14). This form, together with Eq. (18) itself and all information contained in Table 2, are programmed in the `CHIPR_TETRA.f90` module.

The calibration of Eq. (18), likewise the triatomic case, must also be performed in a two-step manner by first running `JOBTYP=OPTBASIS` and then using the optimized basis set parameters as a first guess for the subsequent `JOBTYP=FITPOL` job. Again, the determination of all 6 contracted basis [Eq. (5)] should be preceded by 1*D ab initio* cuts along each symmetry unique degree of freedom, `DEG`. These calculations are generally carried out at a (strategic) reference geometry, *e.g.*, at the global minimum on the PES, and the resulting grid of energies *per* coordinate dimension are input into "`coeffs_guess.txt`". Note that, for A_4, AB_3, A_2B_2, ABC_2 and ABCD type molecules, `DEG` assumes the values of 1, 2, 3, 4 and 6, respectively. Thus, for `JOBTYP=OPTBASIS`, the coordinates y_p's are calibrated via minimization of the following objective function (see `MAIN.f90`)

$$\chi_p^2 = \sum_{I=1}^{N_{basis}} \left\{ W_I \left[y_p(R_p; c_\alpha, R_p^{\rm ref}, \zeta) - \epsilon_I^{(4)}(\mathbf{R}) \right] \right\}^2, \tag{20}$$

where $p = \{1\}$, $\{1,4\}$, $\{1,3,6\}$, $\{1,2,3,6\}$ and $\{1,2,3,4,5,6\}$, respectively for A_4, AB_3, A_2B_2, ABC_2 and ABCD species. As in Eq. (15), the above summation runs over all *ab initio* four-body energies included *per* coordinate dimension, N_{basis}. Note that the correlations between the y_p and interparticle (R_p's) coordinates in Eqs. (18) and (20) follow a one-to-one correspondence and the strict definitions given in Fig. 1. As such, it is imperative that users deal with them accordingly.

With the set of y_p coordinates and their optimum parameters at hand, one can then proceed further and obtain the coefficients of the polynomial (18) by setting `JOBTYP=FITPOL` (once the proper "`abinitio_data.txt`" and "`mol_gm_guess.txt`" additional input files have been edited). Similarly to triatomics, this could be pursued by first running `OPTTYP=FIXBAS` and then `OPTTYP≠FIXBAS` *a posteriori*. In any case, the target function to be minimized is

$$\chi^2 = \sum_{I=1}^{N_{pol}} \left\{ W_I \left[V^{(4)}(\mathbf{R}; C_{i,j,k,l,m,n}, c_\alpha, R_p^{\rm ref}, \zeta) \times \mathcal{D}^{(4)}(\mathbf{R}) - \epsilon_I^{(4)}(\mathbf{R}) \right] \right\}^2, \tag{21}$$

where, differently from `JOBTYP=OPTBASIS` runs, users have the option to define the W_I's automatically via `WEIGHT` function (see `WEIGHT_FUNC.f90` module, and explanation given in Sections 6 and 9). Examples of tetratomic fits for C_3H [8] and C_2H_2 (preliminary work) are shown in Figs. 4 and 5, respectively.

4. Units and conversion factors

All distances and energies must be provided in a.u. (a_0 and E_h, respectively). For `JOBTYP=DIRECTFIT` runs, spectroscopic attributes are read in standard cm^{-1} units. Output quantities are also given in a.u., with exception of deviations and root mean square deviations (RMSDs) that are in cm^{-1} (1 E_h = 219474.6313702 cm^{-1}).

5. Source code and test runs

CHIPR-4.0 is available as a zip archive file "CHIPR-4.0.zip" where its source code can be assessed; see "CHIPR-4.0_SOURCE_CODE/" directory. In "CHIPR-4.0.zip", the user finds also test runs for $C_2(a\,^3\Pi_u)$ [7], $N_2(X\,^1\Sigma_g^+)$, $C_3(^3A')$ [7], ground-state

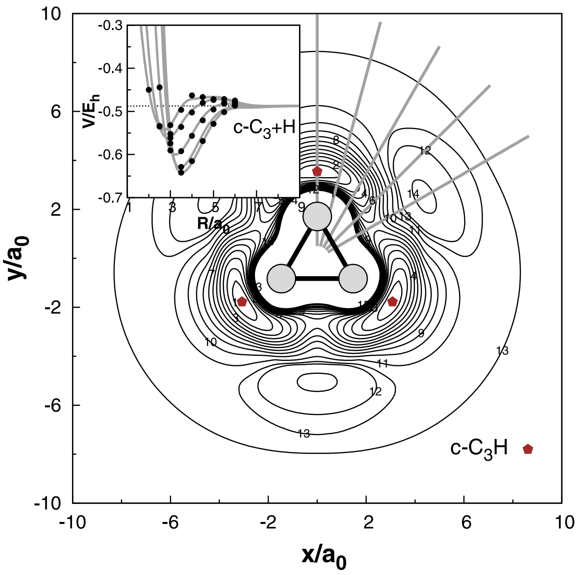


Fig. 4. Sample plot of a CHIPR tetratomic PES of ground-state (doublet) C_3H. The fit used calibrating *ab initio* CCSD(T)/CBS(D, T) energies of Ref. [8] and `JOBTYP=FITPOL`. The plot shows an H atom moving around a partially relaxed *c*-C_3 triatom with center-of-mass fixed at the origin. Also indicated is the location of the global minimum, *c*-C_3H, with C_{2v} symmetry; see Ref. [8] for other stationary points on the PES. Contours are equally spaced by 0.0135 E_h starting at $-0.65\,E_h$. This job is provided as test run in “CHIPR-4.0.zip”.

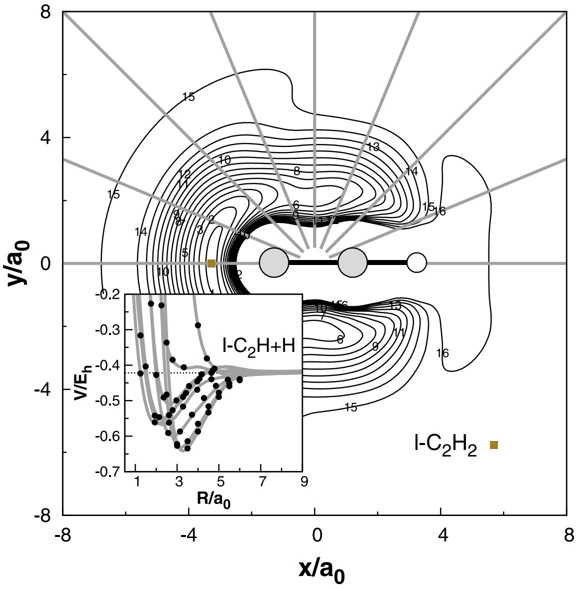


Fig. 5. Sample plot of an exploratory CHIPR tetratomic PES of ground-state (singlet) C_2H_2 calibrated using *ab initio* energies at CCSD(T)/CBS(T, Q) level [8] and `JOBTYP=FITPOL`. The plot shows an H atom moving around a partially relaxed ℓ-C_2H triatom which lies along the x axis with its center-of-mass fixed at the origin. Also shown is the location of the global minimum, *l*-C_2H_2, with $D_{\infty h}$ symmetry. Contours are equally spaced by 0.015 E_h starting at $-0.65\,E_h$. This job is provided as test run in “CHIPR-4.0.zip”.

C_3H [8], and for an exploratory ground-state PES of C_2H_2 employing all job types currently supported. For the `JOBTYP=FITPOL` cases, RMSDs of 2.6, 33.3, 227.1, 673.9 and 442.1 cm^{-1}, respectively, are obtained in these model systems. In the example `JOBTYP=DIRECTFIT` runs for $C_2(a\,^3\Pi_u)$ [7] and $C_2(X\,^1\Sigma_g^+)$ [7], the RMSDs are 0.003 and 0.05 cm^{-1}, respectively, when considering the set of experimental spectroscopic data [34,35] only; the total RMSDs (*ab initio* plus experimental data) are 241.9 and 390.3 cm^{-1}. The reader is directed to all such examples to follow the guide step-by-step.

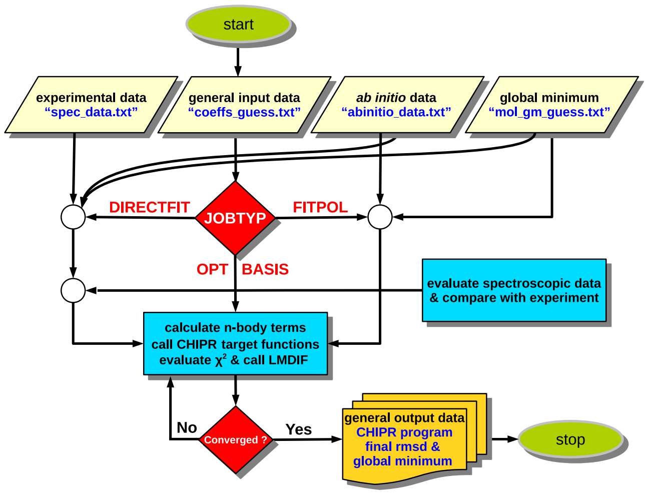


Fig. 6. General flow chart scheme for CHIPR-4.0.

6. Program structure

Fig. 6 shows the flow chart of CHIPR-4.0. The current structure has three main actions: basis set calibration [`JOBTYP=OPTBASIS`; Eqs. (15) and (20)], polynomial optimization [`JOBTYP=FITPOL`; Eqs. (10), (16) and (21)] and polynomial optimization via direct-fit to experimental data [`JOBTYP=DIRECTFIT`; Eq. (11)]. Such options, together with the choice of molecule type, determine the type of data (and files) to be read and govern the interconnections between the main program and its distinct modules. The role of each of these items is scrutinized below. They are inside the “CHIPR-4.0_SOURCE_CODE/” directory.

- **MODULE_COMMON_VAR.f90**: Module containing all common variables.
- **MAIN.f90**: Main program.
- **READING.f90**: Routine that reads all input information.
- **BASIS_SET.f90**: Routine for setting up basis sets and coordinates y_p’s [Eqs. (5)–(8)].
- **CHIPR_DIAT.f90**: CHIPR diatomic model potential [Eq. (9)].
- **CHIPR_TRIAT.f90**: CHIPR three-body model potential [Eq. (13)].
- **CHIPR_TETRA.f90**: CHIPR four-body model potential [Eq. (18)].
- **WEIGHT_FUNC.f90**: Module that specifies all relevant weights (`WW`s). The user must supply a `WEIGHT` function (for `JOBTYP=FITPOL` or `DIRECTFIT`) which at the calling stage assumes the form `WEIGHT=WW`. Both for `JOBTYP=FITPOL` and `DIRECTFIT`, the points for which `WW`$\neq$1, `WEIGHT` is set equal to `WW`. Because there are relatively few data for `JOBTYP=OPTBASIS`, the `WW`s may be set one by one, and hence `WEIGHT` is not operative. *To avoid editing inside “CHIPR-4.0_SOURCE_CODE/”, WEIGHT_FUNC.f90 may be copied into the user’s working directory, where proper adaptations are made; see Section 8.*
- **SUM2BD.f90**: User supplied module to calculate sum of two-body energies for a triatomic [$\sum_{i=1}^{3} V_i^{(2)}(R_i)$ term in Eq. (12)]. The user must define `DIAT1(R1)`, `DIAT2(R2)`, and `DIAT3(R3)` or, alternatively, the subroutine `SUM2BD(R1, R2,R3,SUMV2)`, according to target triatomic. For AB_2-type ones, R_1 must be assigned to the B_2 diatomic. *For editing purposes, follow above recommendation; see Section 8.*

- **SUM23BD.f90**: User supplied module to calculate sum of two-plus-three body energies for a tetratomic [$V^{(2+3)}(\mathbf{R})$ in Eq. (17)]. The user must supply subroutine `SUM23BD(R1,R2,R3,R4,R5,R6,SUMV23)` according to the target tetratomic. The definitions of the interparticle coordinates should follow Fig. 1. *For editing purposes, see above and Section* 8.
- **DIRECT_FIT.f90**: Set of subroutines to evaluate analytic first derivatives, force constants and other diatomic attributes with `JOBTYP=DIRECTFIT`.
- **LEVEL.f**: Modified Le Roy's LEVEL program [32] to perform bound-state calculations in CHIPR diatomic functions; operative only for `JOBTYP=DIRECTFIT`.
- **TOFIT.f90**: Defines χ^2 objective function for minimization [Eqs. (10), (11), (15), (16), (20), (21)].
- **LSFIT.f90**: Calls MINPAC `LMDIF1` routine to minimize χ^2. Its accuracy is controlled by the convergence parameter `TOL`. This is set to the square root of the machine precision ($\approx 10^{-8}$) [33].
- **LMDIF.f**: Non-linear Levenberg–Marquardt set of programs [33] for weighted least-squares fit.
- **FINAL_ANALYSIS.f90**: Performs statistical analyses of fitted forms, *i.e.*, prints RMSD and calls finalizing subroutines.
- **PROP_DIAT.f90, PROP_TRIAT.f90 and PROP_TETRA.f90**: Prints final properties of all fitted functions.
- **HARM_VIB_SA.f90**: Attempts to find a global minimum on the final fitted PES using a simulated annealing optimization algorithm [13]. At the optimal structures, harmonic vibrational analysis [14] are also performed.
- **SA.f**: Simulated annealing code as described by Corana et al. [13].
- **STRAT_RMSD.f90**: Calculates final stratified RMSDs. The energy strata are defined relative to minimum interaction energy.
- **SCRIPT_CHIPR_DIAT.csh, SCRIPT_CHIPR_TRIAT.csh and SCRIPT_CHIPR_TETRA.csh**: C-shell scripts that automatically generate CHIPR diatomic (`CHIPR_DIAT_FUNC.f90`), three-body (`CHIPR_TRIAT_FUNC.f90`) and four-body (`CHIPR_TETRA_FUNC.f90`) ready-to-use routines with the optimized coefficients.

7. Program dependencies

As in previous CHIPR-3.0 code [1], the user must ensure that the following utilities are available:

- C-shell (tcsh or csh) command-line interpreter, version 6.18.01 or an upgrade.
- GNU Fortran compiler (gfortran), version 4.8.3 or an upgrade.
- Gnuplot [44] command-line driven graphing utility, version 4.6 or an upgrade.

8. Compilation and execution

The compilation of CHIPR-4.0 differs slightly from the previous case [1]. In this version, the source code in "CHIPR-4.0_SOURCE_CODE/" is expected to remain untouched. Any modifications should be done once the corresponding module is copied to a separate user's working directory (see below). This is assumed doable only for a limited number of system-specific modules: `WEIGHT_FUNC.f90`, `SUM2BD.f90`, and `SUM23BD.f90`. Gnuplot files (distinctly from CHIPR-3.0) do not accompany the source code but are rather system-specific and hence should be placed in the same working directory. Note that this "one-directory-for-each-job" structure is here devised so that a CHIPR PES data bank may be easily managed in the future.

To incorporate the above feature, a "`config.csh`" C-shell script has been designed which, once executed in the working directory, looks after CHIPR-4.0's modules (Section 6) and writes their paths (either in "../CHIPR-4.0_SOURCE_CODE/" or "./") to a `Makefile`. Modules inside the working directory will always have priority over any other (under the same name) in "CHIPR-4.0_SOURCE_CODE/". A more detailed explanation of this and overall compilation process is given below.

The first step consists of decompressing the original zip file and enter the main "CHIPR-4.0" directory. For this, open a command line and use:

```
unzip CHIPR-4.0.zip
cd CHIPR-4.0/
```

Inside "CHIPR-4.0", users find all example runs discussed in Section 5, the directory source code "CHIPR-4.0_SOURCE_CODE/", and the "`config.csh`" script. For compilation, one then first requires to make the latter executable by entering:

```
chmod +x config.csh
```

To create a new job, the user must now make a new directory (*e.g.*, "FIT_TEST/") inside "CHIPR-4.0" but outside "CHIPR-4.0_SOURCE_CODE/":

```
mkdir FIT_TEST
cd FIT_TEST
```

Similarly to the examples provided, this directory will harbor all input files (described in Section 9), system-specific routines, and Gnuplot files needed for the target system. For polynomial (or direct-) fits, the specific routines to be edited can be copied to the user's directory as

```
% For diatomics
cp ../CHIPR-4.0_SOURCE_CODE/WEIGHT_FUNC.f90 .

% For triatomics
cp ../CHIPR-4.0_SOURCE_CODE/WEIGHT_FUNC.f90 .
cp ../CHIPR-4.0_SOURCE_CODE/SUM2BD.f90 .

% For tetratomics
cp ../CHIPR-4.0_SOURCE_CODE/WEIGHT_FUNC.f90 .
cp ../CHIPR-4.0_SOURCE_CODE/SUM23BD.f90 .
```

Examples on how to adapt the above modules according to the user's target system are in the test runs. For the basis set calibration, only `SUM2BD.f90` and `SUM23BD.f90` routines must be copied and edited in triatomic and tetratomic fits, respectively. For diatomics, none of these actions are required. Gnuplot files to be placed inside "`FIT_TEST`" assume well defined (and self-explanatory) labels and must be named according to the molecule and job type selected as

- **PLOT_BASIS_OPT_CHIPR_DIATOM.gnu**
- **PLOT_BASIS_OPT_CHIPR_TRIATOM.gnu**
- **PLOT_BASIS_OPT_CHIPR_TETRA.gnu**
- **PLOT_POL_OPT_CHIPR_DIATOM.gnu**
- **PLOT_POL_OPT_CHIPR_TRIATOM.gnu**
- **PLOT_POL_OPT_CHIPR_TETRA.gnu**

Inside test runs, users will also find examples.

Assuming that all the above files are properly edited and placed inside "`FIT_TEST`", a `Makefile` can be generated by running inside "FIT_TEST/" the command:

```
../config.csh
```

In so doing, the user will be prompted to answer the following questions:

```
% Question #1
Do you want a diatomic (diat) fit, a triatomic
     (triat) or a tetratomic (tetra)?
```

Please, answer according to system type: `diat` for diatomic, `triat` for triatomic or `tetra` for tetratomic.

```
% Question #2
gfortran is used by default, do you agree? [y/
    n]
```

If a different compiler (other than gfortran) is to be used, answer `n`. If this is the case, remember to edit the final `Makefile` with the preferred compiler options.

```
% Question #3
Is a basis set calibration (bas) planned, a
    polynomial fit (pol) or a direct fit (
    dirfit)?
```

Please answer according to job type: `bas` for basis set contraction (`JOBTYP=OPTBASIS`), `pol` for polynomial optimization (`JOBTYP=FITPOL`) or `dirfit` for direct-fit to diatomic experimental data (`JOBTYP=DIRECTFIT`).

```
% Question #4
Are additional subroutines to be included into
     the program? [y/n]
```

Answer `n` if no extra subroutine is needed. This option is suitable for simple jobs like basis sets calibrations and polynomial (or direct-fit) optimization in diatomics. For triatomics and tetratomics, it will also be the case if users prefer to edit directly the `SUM2BD.f90` and `SUM23BD.f90`, by copying all subroutines required to evaluate the sum of two- and two-plus-three-body potentials to such modules; see example in C_3 test run. Alternatively, if such potentials are provided separately, query #4 must be answered `y`. This is the case in C_3H and C_2H_2 test runs.

```
% Question #5
If yes, specify its name, e.g., XXX.f or XXX.
    f90
```

If the answer to query #4 is yes, the user is asked to provide name of the module to be compiled. Note that this must also exist in "`FIT_TEST`". In C_3H and C_2H_2 examples, they are `HC3_V23_DMBE.f` and `C2H2_V23.f`, respectively.

Execution of script "`config.csh`" will terminate with warning messages. They are intended to make the user aware of the modules present in the working directory (the ones not in there have their paths set to "../CHIPR-4.0_SOURCE_CODE/") and also of which input files are supposed to be in "`FIT_TEST`", depending on the job and molecule type.

The above process will generate a `Makefile`, with the program compiled using:

```
make clean
make
make clean
```

This will generate an executable file `CHIPR.x`, with the program now run using

```
./CHIPR.x
```

To make users familiar with the code compilation, it is perhaps wise to run "`config.csh`" inside any of the example directories and compare the `Makefile` so generated with the instructions contained in "`gfortran_original_Makefile`".

9. Input files for CHIPR-4.0

Besides system-specific routines and Gnuplot files discussed in previous section, the following inputs are needed:

- **coeffs_guess.txt**: file required for all jobs. It provides general information on molecule and job type, basis set calibration, order of polynomial, and guess coefficients.
- **abinitio_data.txt**: file required for `JOBTYP=FITPOL` and `DIRECTFIT`. It contains the *ab initio* data set used to calibrate the PES: interparticle coordinates, *ab initio* energies and corresponding least-squares weights.
- **mol_gm_guess.txt**: file required for `JOBTYP=FITPOL` and `DIRECTFIT`. It provides guess Cartesian coordinates for a global minimum search on the final fitted PES.
- **spec_data.txt**: file required for `JOBTYP=DIRECTFIT`. It contains spectroscopic (experimental) data for a given diatomic molecule.

As noted above, all files should be placed inside the user's working directory. Their details are specified below.

9.1. "*coeffs_guess.txt*" (*FILE #1*)

Line #1. `READ(1,*) NATOM`

Number of atoms (the current version is restricted to `NATOM=2`, 3 or 4).

Line #2.
```
DO I=1,NATOM
   READ(1,*) ATOM(I),Z(I),AMU(I)
ENDDO
```

Specifies atom type, atomic number, and atomic mass number in a.u. As in "`mol_gm_guess.txt`" (see later), the order of the input atoms matters here: it must follow the particle numbering shown in Fig. 1.

Line #3. `READ(1,*) EZERO`

Reference energy in E_h: it defines the zero of the potential energy, often assumed as the energy of the infinitely separated `NATOM`s.

Line #4.
```
IF (NATOM=3 .OR. NATOM=4) THEN
   READ(1,*) MOLTYP
ELSE GO TO Line #5.
ENDIF
```

Specifies molecule type. If `NATOM=3`: `MOLTYP=A3`, `AB2`, or `ABC`. If `NATOM=4`, `MOLTYP=A4`, `AB3`, `A2B2`, `ABC2` or `ABCD`.

Line #5. `READ(1,*) JOBTYP`

Job type: `JOBTYP=OPTBASIS` for basis sets optimization, `FITPOL` for polynomial fit or `DIRECTFIT` for direct-fit to experimental data. Recall that this latter option is only available for diatomics, `NATOM=2`.

Line #6.
```
READ(1,*) OPTTYP
IF (JOBTYP=OPTBASIS) THEN
```

The following options are available for `OPTTYP`:

- `FIXNONLIN`: optimize only linear contraction coefficients c_α in Eq. (5). $\gamma_{p,\alpha}$, ζ and R_p^{ref} in Eqs. (6)–(8) are fixed at guess values.
- `OPTNONLIN`: optimize c_α and non-linear parameters $\gamma_{p,\alpha}$ at fixed ζ and R_p^{ref}.
- `OPTZETARREF`: optimize c_α, ζ and R_p^{ref} at fixed $\gamma_{p,\alpha}$.
- `OPTALL`: optimize all parameters in basis set contraction (5).

Note: The current version of the program does not allow to optimize η_α, σ_α and β_α in Eqs. (6) and (7). They are fixed at convenient values: $\eta_\alpha = 1$, $\sigma_\alpha = 6$, and $\beta_\alpha = 1/5$.

```
ELSE IF (JOBTYP=FITPOL .OR. JOBTYP= DIRECTFIT) THEN
```

The following options are available:x

- `FIXBAS`: optimize polynomial coefficients in Eqs. (9), (13) or (18). Fix optimum basis set [Eq. (5)] from previous `JOBTYP=OPTBASIS` run.
- `FIXBASNONLIN`: optimize $C_{i\ldots}$ and c_α at fixed $\gamma_{p,\alpha}$, ζ and R_p^{ref} in Eqs. (6)–(8).
- `OPTBASNONLIN`: optimize $C_{i\ldots}$, c_α and non-linear parameters $\gamma_{p,\alpha}$ at fixed ζ and R_p^{ref}.
- `OPTBASZETARREF`: optimize $C_{i\ldots}$, c_α, ζ, and R_p^{ref} at fixed $\gamma_{p,\alpha}$.
- `OPTALL`: optimize all parameters in Eqs. (9), (13) or (18), and basis set contraction (5).

```
         ENDIF
Line #7. IF (JOBTYP=FITPOL .OR. JOBTYP=DIRECTFIT)
         THEN
           READ(1,*) L,NP
         ELSE GO TO Line #8.
         ENDIF
```

`L` is the degree of the polynomial in Eqs. (9), (13) or (18), and `NP` the number of *ab initio* points in two-, three- and four-body fits.

```
Line #8. DO I=1,DEG
           READ(1,*) MBS(I),NPBS(I)
         ENDDO
```

`MBS(I)` is the length of basis set contraction in Eq. (5), and `NPBS(I)` the number of *ab initio* points used for calibrating all (`DEG`) symmetry unrelated degrees of freedom. For A_3, AB_2 and ABC molecules, `DEG` assumes the values of 1, 2, and 3. For A_4, AB_3, A_2B_2, ABC_2 and ABCD, `DEG` is 1, 2, 3, 4 and 6, in the same order.

```
Line #9. DO I=1,DEG
           DO J=1,(2*MBS(I)+2)
             READ(1,*) CBS(1,I)      ! $c_1$
                       CBS(2,I)      ! $c_2$
                       ⋮
                       CBS(M,I)      ! $c_M$
                       CBS(M+1,I)    ! $\gamma_{p,1}$
                       CBS(M+2,I)    ! $\gamma_{p,2}$
                       ⋮
                       CBS(M+M,I)    ! $\gamma_{p,M}$
                       CBS(2M+1,I)   !$R_p^{\mathrm{ref}}$
                       CBS(2M+2,I)   !$\zeta$
           ENDDO
           DO K=1,NPBS(I)
             READ(1,*) (XBS(Q,K,I),Q=1,NX),
             YYBS(K,I),WWBS(K,I)
           ENDDO
         ENDDO
```

`CBS(J,I)` is the `J`th guess coefficient of the basis set contraction [Eq. (5)] for the `I`th degree of freedom (`DEG`). There are a total of $2M+2$ [`2*MBS(I)+2`], where the first M are linear coefficients c_α in the contraction (5), the second M block defines the corresponding decaying parameters $\gamma_{p,\alpha}$ for each primitive $\phi_{p,\alpha}$ in the series, with the last two representing R_p^{ref} and ζ in Eq. (8). `XBS(Q,K,I)`, `YYBS(K,I)` and `WWBS(K,I)` are the coordinates, energies and weights, respectively, to be used in the 1*D* basis set calibration. `NX` defines the $N(N-1)/2$ number of internal (independent) degrees of freedom for the N-atom aggregate. Their order should be read according to the coordinates in Fig. 1. `YYBS(K,I)` will be subtracted by `EZERO` inside the program.

```
Line #10. IF (JOBTYP=FITPOL .OR. JOBTYP=DIRECTFIT)
          THEN
            DO I=1,NCPOL
              READ(1,*) CPOL(I)
            ENDDO
          ENDIF
```

Read all `NCPOL` guess coefficients for polynomials in Eqs. (9), (13) or (18). Their number are determined internally according the maximum degree `L` provided by the user. Note that there are two independent programs inside "CHIPR-4.0_SOURCE_CODE/" to help users in estimating the number of coefficients needed for each `MOLTYP` and `L`, namely `GENERAL_POL_LOOP_TRIATOM.f90` and `GENERAL_POL_LOOP_TETRATOM.f90`.

9.2. "`abinitio_data.txt`" (FILE #2)

```
Line #1. IF (JOBTYP=FITPOL .OR. JOBTYP=DIRECTFIT)
         THEN
           DO I=1,NP
             READ(2,*) (X(I,J),J=1,NX), YY(I), WW(I)
           ENDDO
          ENDIF
```

Read `NP` *ab initio* data: `X(I,J)`, `YY(I)` and `WW(I)`, thence coordinates, energies and weights, in an obvious correspondence. `NX` defines the $N(N-1)/2$ number of internal (independent) degrees of freedom for the N-atom aggregate. Coordinate orderings must follow the convention in Fig. 1. `YY(I)` will be subtracted by `EZERO` inside the program.

9.3. "`mol_gm_guess.txt`" (FILE #3)

```
Line #1. IF (JOBTYP=FITPOL .OR. JOBTYP=DIRECTFIT)
         THEN
           READ(3,*) (CARTX(I),I=1,3*NATOM)
         ENDIF
```

Read 3*N* Cartesian coordinates (in a_0) of PES global minimum, with every atom defined by its x, y, and z components (order must follow the labels given in Fig. 1). This guess geometry will be later set as such for the simulated annealing optimization algorithm. The predicted global minimum of the fitted potential is then output in a Molden file named "`gmin.res`".

9.4. "`spec_data.txt`" (FILE #4)

```
Line #1. IF (JOBTYP=DIRECTFIT .AND. NATOM=2) THEN
           READ(4,*) ICHARGE, IOMEGA
```

`ICHARGE` is the total charge of the diatomic molecule. `IOMEGA` defines the projection of its electronic orbital angular momentum in the molecular axis. These are attributes obtained from LEVEL [32].

```
Line #2.   READ(4,*) REEXP, WREEXP
```

Read experimental equilibrium geometry [R_e^{exp} in Eq. (11)], in a_0, and associated least-squares weight.

Line #3. `READ(4,*) DEEXP, WDEEXP`

Read experimental dissociation energy [$D_e^{\rm exp}$ in Eq. (11)], in E_h, and associated least-squares weight.

Line #4. `READ(4,*) WEEXP, WWEEXP`

Read experimental harmonic vibrational frequency [$w_e^{\rm exp}$ in Eq. (11)], in cm^{-1}, and associated least-squares weight.

Line #5. `READ(4,*) NROVIB`

`NROVIB` is the number of input experimental (ro)vibrational levels.

Line #6.
```
 DO I=1,NROVIB
   J=NP+I
   READ(4,*) IVEXP(I), IJEXP(I), Y(J), W(J)
 ENDDO
ENDIF
```

`IVEXP(I)` and `IJEXP(I)` are input vibrational and rotational quantum numbers. The associated experimental (ro)vibrational levels, in cm^{-1}, and their weights are kept in the `Y(J)` and `W(J)` vectors, respectively. Recall that `NP` is the number of *ab initio* data points.

10. Output files

The following output files are generated by CHIPR-4.0:

10.1. In basis set calibrations (*JOBTYP=OPTBASIS*)

- **out.res**: File containing a summary of the least-square fitting procedure and the final parameters. These are in a.u; RMSDs are in cm^{-1}.
- **dev_basis_abinitio_points.res**: File containing the deviations (in cm^{-1}) between *ab initio* n-body energies ($n = 2, 3, 4$) employed for the calibration of the y_p coordinates and the ones predicted from Eq. (5). Distances are in a_0.
- **plot_basis_abinitio.res**: File containing *ab initio* n-body energies ($n = 2, 3, 4$) employed for the calibration of the y_ps as well as the total interaction *ab initio* energies (with zero equal to `EZERO`) at the calibration distances R_ps. Energies and distances are in a.u.
- **plot_basis.res**: File containing the values of y_p predicted from Eq. (5) for a grid of R_p distances (energies and distances are in a.u.).
- **plot_nodes.res**: File containing origins $R_{p,\alpha}^{\rm ref}$ (or nodes) of each contracted primitive basis in Eq. (8) and values of y_ps at these points (energies and distances are in a.u.).
- **TRIAT_CHIPR.ps, DIAT_CHIPR.ps or TETRA_CHIPR.ps**: Gnuplot postscript files showing the final plots in `PLOT_BASIS_OPT_CHIPR_DIATOM.gnu`, `PLOT_BASIS_OPT_CHIPR_TRIATOM.gnu` or `PLOT_BASIS_OPT_CHIPR_TETRA.gnu`.

10.2. In polynomial fits (*JOBTYP=FITPOL*)

The same files given above (for `OPTTYP`≠`FIXBAS`, both old and new basis sets are printed) plus the following:

- **dev_abinitio_points.res**: Prints deviations (in cm^{-1}) between *ab initio* and fitted total energies. Distances are in a_0.
- **plot_abinitio.res**: Prints total *ab initio* interaction energies with zero set to `EZERO` (energies and distances in a.u.).
- **plot_pot.res**: Prints interaction energies from Eq. (9) (for `NATOM`=2), Eq. (13)+sum-of-two-bodies (for `NATOM`=3) and Eq. (1) (for `NATOM`=4). The zero of energy is set to `EZERO` (energies and distances in a.u.).
- **strat_rmsd.res**: Prints stratified RMSDs with energy strata defined relative to the minimum of the potential. Energies are in cm^{-1}.
- **coeffs_tab_latex.res**: LaTeX readable table of optimized coefficients (in a.u.).
- **sa.res**: Output file of the simulated annealing optimization code.
- **normalmode.res**: Output file from harmonic vibrational analysis module.
- **gmin.res**: Molden [45] file containing optimum geometry and harmonic frequencies of global minimum of fitted PES. Distances in a_0, frequencies in cm^{-1}.
- **CHIPR_DIAT_FUNC.f90, CHIPR_TRIAT_FUNC.f90 or CHIPR_TETRA_FUNC.f90**: CHIPR diatomic, three-body or four-body ready-to-use potential routines with optimum fitted coefficients. Energies and distances in a.u.; see Ref. [1] and accompanying "README.txt" file to assess the general instructions on how to use them.
- **DIAT_CHIPR.ps, TRIAT_CHIPR.ps or TETRA_CHIPR.ps**: postscript files from `PLOT_POL_OPT_CHIPR_DIATOM.gnu`, `PLOT_POL_OPT_CHIPR_TRIATOM.gnu` or `PLOT_POL_OPT_CHIPR_TETRA.gnu`

10.3. Direct-fits for diatomics (*JOBTYP=DIRECTFIT*)

The same files given in Sections 10.1 and 10.2 plus the following:

- **plot_levels.res**: Prints (ro)vibrational levels predicted from final fitted curve, their experimental values and deviations between the latter. Units are in cm^{-1}.
- **level.res**: Output file from LEVEL program [32].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by Foundation for Science and Technology and Coimbra Chemistry Centre, Portugal, in the framework of the project "MATIS - Materiais e Tecnologias Industriais Sustentáveis" (reference: CENTRO-01-0145-FEDER-000014), co-financed by the European Regional Development Fund (FEDER), through the "Programa Operacional Regional do Centro" (CENTRO2020) program. C.M. thanks also the CAPES Foundation (Ministry of Education of Brazil) for a scholarship (BEX 0417/13-0). A.J.C.V. thanks also the support from China's Shandong Province "Double-Hundred Talent Plan" (2018).

References

[1] C.M.R. Rocha, A.J.C. Varandas, Comput. Phys. Comm. 247 (2020) 106913, http://dx.doi.org/10.1016/j.cpc.2019.106913.
[2] J.N. Murrell, S. Carter, S.C. Farantos, P. Huxley, A.J.C. Varandas, Molecular Potential Energy Functions, John Wiley & Sons, Chichester, 1984.
[3] A.J.C. Varandas, J. Chem. Phys. 138 (2013) 054120–054133, http://dx.doi.org/10.1063/1.4788912.
[4] A.J.C. Varandas, J. Chem. Phys. 138 (2013) 134117–134123, http://dx.doi.org/10.1063/1.4795826.
[5] A.J.C. Varandas, in: K. Han, T. Chu (Eds.), Reaction Rate Constant Computations: Theories and Applications, The Royal Society of Chemistry, 2013, pp. 408–445, http://dx.doi.org/10.1039/9781849737753-00408 (Ch. 17).
[6] F.G.D. Xavier, M.M. González, A.J.C. Varandas, J. Phys. Chem. A 123 (2019) 1613–1621, http://dx.doi.org/10.1021/acs.jpca.8b12005.
[7] C.M.R. Rocha, A.J.C. Varandas, J. Phys. Chem. A 123 (2019) 8154–8169, http://dx.doi.org/10.1021/acs.jpca.9b03194.
[8] C.M.R. Rocha, A.J.C. Varandas, Phys. Chem. Chem. Phys. 21 (2019) 24406–24418, http://dx.doi.org/10.1039/C9CP04890A.
[9] A.J.C. Varandas, S.P.J. Rodrigues, V.M.O. Batista, Chem. Phys. Lett. 424 (2006) 425–431, http://dx.doi.org/10.1016/j.cplett.2006.04.084.

[10] F.G.D. Xavier, M. Martínez-González, A.J.C. Varandas, Chem. Phys. Lett. 691 (2018) 421–430, http://dx.doi.org/10.1016/j.cplett.2017.11.050.

[11] R.J.L. Roy, JQSRT 186 (2017) 179–196, http://dx.doi.org/10.1016/j.jqsrt.2016.06.002.

[12] S.N. Yurchenko, L. Lodi, J. Tennyson, A.V. Stolyarov, Comput. Phys. Comm. 202 (2016) 262–275, http://dx.doi.org/10.1016/j.cpc.2015.12.021.

[13] A. Corana, M. Marchesi, C. Martini, S. Ridella, ACM Trans. Math. Software 13 (1987) 262–280, http://dx.doi.org/10.1145/29380.29864.

[14] E.B. Wilson. Jr., J.C. Decius, P.C. Cross, Molecular Vibrations, McGraw Hill, New York, 1955.

[15] B.R.L. Galvão, V.C. Mota, A.J.C. Varandas, J. Phys. Chem. A 119 (8) (2015) 1415–1421, http://dx.doi.org/10.1021/jp512671q.

[16] B.R.L. Galvão, V.C. Mota, A.J.C. Varandas, Chem. Phys. Lett. 660 (2016) 55–59, http://dx.doi.org/10.1016/j.cplett.2016.07.029.

[17] C.M.R. Rocha, A.J.C. Varandas, Phys. Chem. Chem. Phys. 20 (2018) 10319–10331, http://dx.doi.org/10.1039/C7CP06656B.

[18] F.G.D. Xavier, A.J.C. Varandas, Modeling adiabatic cusps using a 2 × 2 diabatic matrix in HO_2^+, 2020, in preparation.

[19] A.J.C. Varandas, Adv. Chem. Phys. 74 (1988) 255–338, http://dx.doi.org/10.1002/9780470141236.ch2.

[20] A. Aguado, C. Tablero, M. Paniagua, Comput. Phys. Comm. 108 (1998) 259–266, http://dx.doi.org/10.1016/S0010-4655(97)00135-5.

[21] A. Aguado, C. Tablero, M. Paniagua, Comput. Phys. Comm. 134 (2001) 97–109, http://dx.doi.org/10.1016/S0010-4655(00)00181-8.

[22] C. Tablero, A. Aguado, M. Paniagua, Comput. Phys. Comm. 140 (2001) 412–417, http://dx.doi.org/10.1016/S0010-4655(01)00289-2.

[23] J.N. Murrell, S. Carter, J. Phys. Chem. 88 (1984) 4887–4891, http://dx.doi.org/10.1021/j150665a016.

[24] A.J.C. Varandas, L.A. Poveda, Theor. Chem. Acc. 116 (2006) 404–419, http://dx.doi.org/10.1007/s00214-006-0092-6.

[25] J. Li, A.J.C. Varandas, J. Phys. Chem. A 116 (2012) 4646–4656, http://dx.doi.org/10.1021/jp302173h.

[26] L.A. Poveda, M. Biczysko, A.J.C. Varandas, J. Chem. Phys. 131 (2009) 044309, http://dx.doi.org/10.1063/1.3176512.

[27] Z. Xie, J.M. Bowman, J. Chem. Theory Comput. 6 (2010) 26–34, http://dx.doi.org/10.1021/ct9004917.

[28] A. Szabo, N.S. Ostlund, Modern Quantum Chemistry: Introduction To Advanced Electronic Structure Theory, Vol. 1, McGraw-Hill Book Co., New York, 1989.

[29] K. Ruedenberg, R.C. Raffenetti, R.D. Bardo, Proceedings of the 1972 Boulder Conference on Theoretical Chemistry, Wiley, 1972, p. 164.

[30] D.F. Feller, K. Ruedenberg, Theoret. Chim. Acta 52 (1979) 231–251, http://dx.doi.org/10.1007/BF00547681.

[31] M.W. Schmidt, K. Ruedenberg, J. Chem. Phys. 71 (1979) 3951–3962, http://dx.doi.org/10.1063/1.438165.

[32] R.J.L. Roy, JQSRT 186 (2017) 167–178, http://dx.doi.org/10.1016/j.jqsrt.2016.05.028.

[33] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran: The Art of Scientific Computing, second ed., Cambridge University Press, New York, 1993.

[34] W. Chen, K. Kawaguchi, P.F. Bernath, J. Tang, J. Chem. Phys. 142 (2015) 064317–064326, http://dx.doi.org/10.1063/1.4907530.

[35] B. Visser, M. Beck, P. Bornhauser, G. Knopp, J.A. van Bokhoven, P. Radi, C. Gourlaouen, R. Marquardt, Mol. Phys. 117 (2019) 1645–1652, http://dx.doi.org/10.1080/00268976.2018.1564849.

[36] W.M. Kosman, J. Hinze, J. Mol. Spectrosc. 56 (1975) 93–103, http://dx.doi.org/10.1016/0022-2852(75)90206-4.

[37] A. Pashov, W. Jastrzebski, P. Kowalczyk, Comput. Phys. Comm. 128 (2000) 622–634, http://dx.doi.org/10.1016/S0010-4655(00)00010-2.

[38] R. Rydberg, Z. Phys. 73 (1932) 376–385, http://dx.doi.org/10.1007/BF01341146.

[39] O. Klein, Z. Phys. 76 (1932) 226–235, http://dx.doi.org/10.1007/BF01341814.

[40] A.L.G. Rees, Proc. Phys. Soc. 59 (1947) 998–1008, http://dx.doi.org/10.1088/0959-5309/59/6/310.

[41] R.J.L. Roy, JQSRT 186 (2017) 158–166, http://dx.doi.org/10.1016/j.jqsrt.2016.03.030.

[42] B.E. Sagan, Combinatorial Algorithms, and Symmetric Functions, second ed., Springer, New York, 2001.

[43] A.J.C. Varandas, Chem. Phys. Lett. 138 (1987) 455–461, http://dx.doi.org/10.1016/0009-2614(87)80540-7.

[44] T. Williams, C. Kelley, et al., Gnuplot 4.6: An interactive plotting program, 2013, http://gnuplot.sourceforge.net/.

[45] G. Schaftenaar, J. Noordik, J. Comput. Aided Mol. Des. 14 (2000) 123–134, http://dx.doi.org/10.1023/A:1008193805436.