

Documentation of the program FCfit

Daniel Krügler and Michael Schmitt

Heinrich-Heine-Universität, Institut für Physikalische Chemie I, 40225 Düsseldorf, Germany

March 8, 2018

Contents

1	Introduction	2
2	Theory	3
2.1	Franck-Condon theory	3
2.1.1	Calculation of the emission and absorption spectra	5
2.2	Franck-Condon-Herzberg-Teller theory	5
2.2.1	Calculation of the emission and absorption spectra	6
3	General program decription	7
3.1	Program philosophy	7
3.2	Program features	7
3.3	The cost function	7
4	<i>Ab initio</i> computational prerequisites	8
5	Program usage	8
5.1	General structure of the configuration file	8
5.2	Valid entries of the configuration file	9
5.2.1	The section [Flags]	9
5.2.2	The section [General]	10
5.2.3	The section [Recursion]	13
5.2.4	The section [Fit]	18
5.2.5	The section [FurtherIsotopomers]	20
5.2.6	The Section [FurtherLines]	21
5.2.7	The section [NL2SOL] (Opt.)	23
5.2.8	The section [Local-NL2SOL] (Opt.)	23
5.2.9	The section [PGA]	23
6	Necessary input files	24
7	Program output files	25
8	Compiling the source code	27
9	Examples	27

Copyright

Copyright (c) 2000,2009 Daniel Krügler, Physikalische Chemie I, Universität Düsseldorf, Germany.

Copyright (c) 2000,2004 Petra Imhof, Physikalische Chemie I, Universität Düsseldorf, Germany.

Copyright (c) 1997,1998 Stephan Schumm, Physikalische Chemie I, Universität Düsseldorf, Germany.

The GNU General Public License is applicable to the programs described in this manual. Under the conditions of the GNU General Public Licence the usage of the FCfit-program is free if references are made to (i) **D. Spangenberg, P. Imhof, and K. Kleinermanns, Phys. Chem. Chem. Phys. 5, 2505 (2003)** and (ii) **[newpublication]**. Separate license agreements and copyrights are applicable to the genetic algorithm package PgaPack. For this we refer to the particular conditions for use that comes with those packages. By installing and/or using the ga-program and accompanying software, you accept the license conditions.

Preface

This manual has been prepared with great care and reflects the version 2.8.15 and the options of the FCfit-program with which it is included. Although the program has been tested for many different input conditions and several platforms, it still might contain bugs or inconsistencies.

If you encounter a problem you are requested to send a report to the authors with (i) the type of error, including the full error message, (ii) the operating system, (iii) the program version, which is displayed in the header of the .log file, (iv) the input files and (v) if present, the output file(s).

Since the *FCfit*-program is still under development the authors greatly appreciate all suggestions and remarks to improve the program, to simplify/modify the input and requests for additional output(s).

Suggestions to extend or improve this manual will also seriously be considered.

If you have made changes and/or extensions to the source code that improve the performance of the program(s) please communicate those to the authors.

All remarks can be sent to daniel.kruegler@gmx.de.

Daniel Krügler and Michael Schmitt
Bremen-Düsseldorf, March 8, 2018

1 Introduction

FCfit is program for the simulation and fit of vibronic absorption and emission spectra based on the evaluation of relative Franck-Condon (FC) factors and/or Franck-Condon-Herzberg-Teller (FCHT) theory. The program computes the FC integrals of multidimensional, harmonic oscillators mainly based on the recursion formula given in the papers of Doktorov, Malkin, and Man'ko.[1, 2] For further theoretical background of the theory as well as similar approaches see also refs. [3–7] A minor correction of a typo in the original publication is given in ref.[8].

The fit of vibronic spectra can greatly be improved if independent information for the geometry change upon electronic excitation is available. This information can be the change of the rotational constants upon electronic excitation. While geometry fits to the rotational constants alone are possible and routinely made, using non-linear fits in internal coordinates as basis for the geometries, the combination of rotational constant changes and vibronic intensities allows for determination of much more geometry parameters. In contrast to the separate geometry fits of the two states using rotational constants, in the combined FC fit geometry changes (relative to one of the states, usually the ground state) are determined in the basis of selected normal modes. The quality of the results depends largely on the experience of the user in selecting the "right" normal modes for a proper description of the geometry changes. It is therefore by no means a "black box" program, but needs a lot of intuition of the user.

2 Theory

2.1 Franck-Condon theory

According to the FC principle the probability of a vibronic transition and thus the relative intensity of a vibronic band depends on the overlap integral of the vibrational wave functions of both electronic states.

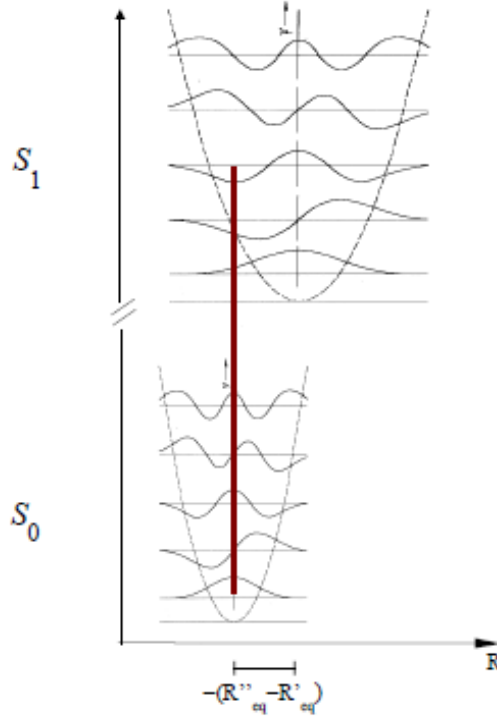


Figure 1: Schematics of the FC principle

This overlap integral is determined by the relative shift of the two potential energy curves connected by the vibronic transition along the normal coordinates Q of both states:

$$FC = \left| \int [\Psi'(Q')]^* \Psi''(Q'') dQ' \right|^2 = \langle v' \dots v'_N | v'' \dots v''_N \rangle^2 \quad (1)$$

where the $\Psi(Q)$ are the N -dimensional vibrational wavefunctions. The normal coordinates Q' of the excited state and Q'' of the ground state are related by the linear orthogonal transformation given by Duschinsky [9]:

$$Q'' = S Q' + \vec{d} \quad (2)$$

where \vec{d} is a displacement vector and S a rotation matrix, which rotates the coordinate system of one state into that of the other state. This matrix is called the Duschinsky matrix. Using the matrix L , which contains the eigenvectors from the diagonalization of the force constant matrix, S can be expressed as:

$$S = (L'')^T L' \quad (3)$$

and \vec{d} as:

$$\vec{d} = (L'')^T M^{1/2} \left(\vec{R}''_{eq.} - \vec{R}'_{eq.} \right) \quad (4)$$

where \mathbf{M} is a diagonal matrix, which contains the square roots of the atomic masses on the diagonal and $\vec{R}_{eq.}$ is the geometry of the respective state.

Doktorov, Malkin und Manko deduced a recursion formula, which allows to compute Franck-Condon integrals considering the Duschinsky effect.[1, 2] Starting point of the recursion is the Franck-Condon integral of the electronic origin:

$$\langle 0''|0'\rangle = 2^{\frac{N_{vib}}{2}} \left[\prod_{j=1}^{N_{vib}} \left(\frac{\omega_j''}{\omega_j'} \right)^{\frac{1}{2}} \right] \det \mathbf{Q}^{\frac{1}{2}} \exp \left[-\frac{1}{2} \delta (1 - \mathbf{P}) \delta \right] \quad (5)$$

The recursion formula for absorption is:

$$\begin{aligned} \langle v''|v', \dots, v_i' + 1, \dots, v_{N_{vib}}'\rangle &= \\ &= 2 \sum_{k=1}^{N_{vib}} R_{ki} \left(\frac{v_k''}{v_i' + 1} \right)^{\frac{1}{2}} \langle v_1'', \dots, v_k'' - 1, \dots, v_{N_{vib}}'' | v_1', \dots, v_i', \dots, v_{N_{vib}}'\rangle + \\ &+ \sum_{j=1}^{N_{vib}} (2\mathbf{Q} - 1)_{ji} \left(\frac{v_j'}{v_i' + 1} \right)^{\frac{1}{2}} \langle v'' | v', \dots, v_i' - 1, \dots, v_{N_{vib}}'\rangle - \\ &- (\mathbf{R}\delta) \left(\frac{2}{v_i' + 1} \right)^{\frac{1}{2}} \langle v'' | v', \dots, v_i', \dots, v_{N_{vib}}'\rangle \end{aligned} \quad (6)$$

and for emission:

$$\begin{aligned} \langle v_1'', \dots, v_i'' + 1, \dots, v_{N_{vib}}'' | v'\rangle &= \\ &= 2 \sum_{k=1}^{N_{vib}} R_{ik} \left(\frac{v_i'}{v_k'' + 1} \right)^{\frac{1}{2}} \langle v_1'', \dots, v_k'', \dots, v_{N_{vib}}'' | v_1', \dots, v_i' - 1, \dots, v_{N_{vib}}'\rangle + \\ &+ \sum_{l=1}^{N_{vib}} (2\mathbf{P} - 1)_{kl} \left(\frac{v_l''}{v_k'' + 1} \right)^{\frac{1}{2}} \langle v_1'', \dots, v_l'' - 1, \dots, v_{N_{vib}}'' | v'\rangle + \\ &+ (1 - \mathbf{P}\delta) \left(\frac{2}{v_k'' + 1} \right)^{\frac{1}{2}} \langle v_1'', \dots, v_i'', \dots, v_{N_{vib}}'' | v'\rangle \end{aligned} \quad (7)$$

The recursive character of Doktorovs equations suggests to save already computed FC integrals in an appropriate manner for further use. An efficient method is realized using so called Hash tables. The entries in a Hash table are administered using keys. The entries of the Hash tables used in FCfit are the energy and the FC factors of a vibronic transition. A quantum string, describing initial and final state is used as key. This quantum string can be compared to an n-digit number, where n is the number of vibrations of the molecule, sorted by their frequency. The i-th position shows, with how many quanta the i-th mode is excited. E.g. the quantum string "001021" represents the vibrational state of a molecule with the third and sixth mode excited with one quantum and the fifth mode with two quanta. Each vibronic transition can be represented by two quantum strings for the initial and final state, respectively. We use the bracket notation, the bra being the initial state, the ket the final state of the transition. The transition from the vibrationless ground state to the vibrational state of the above example would read then: $\langle 000000|001021\rangle$. N.b., that "final" and "initial" refers to the direction of the transition and not to the energetic ordering of states. Using spectroscopic notation this means in absorption $\langle 001021\rangle' \leftarrow \langle 000000\rangle''$ and $\langle 001021\rangle'' \leftarrow \langle 000000\rangle'$ in emission. Thus, the complete description of the transition requires both quantum strings of bra and ket and additionally the information if the transition is in absorption or emission. E.g. $\langle 001000|001000\rangle$ in emission describes the diagonal transition from the singly excited third mode to the respective ground state mode.

2.1.1 Calculation of the emission and absorption spectra

The intensities of the transitions between two vibronic states m and n are proportional to the square of the transition moment M_{mn} and to the frequency ν_{mn} of the transition for absorption spectra and the fourth power of the frequency ν_{mn} for emission spectra, respectively.¹

$$I_{mn} \propto |M_{mn}|^2 \nu_{mn} \quad (Absorption) \quad (8)$$

$$I_{mn} \propto |M_{mn}|^2 \nu_{mn}^4 \quad (Emission) \quad (9)$$

In the present version absorption spectra are calculated only originating from the vibrationless ground state, i.e. a molecular beam experiment is assumed, and no thermal population of vibrational levels of the ground state is taken into account.

2.2 Franck-Condon-Herzberg-Teller theory

Within the frame of the Born-Oppenheimer approximation the vibronic wavefunction is factorized in a vibrational part which depends only on the nuclear coordinates Q and an electronic part which depends on the electronic coordinates q and parametrically on the nuclear coordinates:

$$\Phi_{mv}(q, Q) = \Psi_m(q, Q)v(Q) \quad (10)$$

where m describes the electronic and v the vibrational quantum number.

The transition dipole moment for a transition between an initial electronic state m and a final electronic state n is defined as:

$$M_{mn} = \left\langle v''_{(m)} | \mu_{mn}(Q) | v'_{(n)} \right\rangle \quad (11)$$

with the electronic transition dipole moment $\mu_{mn}(Q)$:

$$\mu_{mn}(Q) = \langle \Psi_m | \mu | \Psi_n \rangle; \quad \mu = \sum_g e r_g \quad (12)$$

where r_g is the position vector of the g th electron. The dependence of the electronic transition dipole moment μ_{mn} from the nuclear coordinates can be approximated by expanding μ_{mn} in a Taylor series about the equilibrium position at Q_0 . Truncation of the expansion after the second expansion term and inserting into equation 11 yields:

$$M_{mn} = \mu_{mn}(Q_0) \left\langle v''_{(m)} | v'_{(n)} \right\rangle + \sum_i \left(\frac{\partial \mu_{mn}}{\partial Q_i} \right)_{Q_0} \left\langle v''_{(m)} | Q_i | v'_{(n)} \right\rangle \quad (13)$$

The displacement of the excited state geometry of the state n along the normal modes Q_i electronically mixes the wavefunctions of all neighboring electronically excited states p that are sufficiently close to the excited state. Through these displacements, the excited state gains orbital character from the perturbing state(s). The derivatives of μ_{mn} are determined numerically from TDDFT/MRCI calculations of the respective state using the method of symmetric finite differences:

¹this refers to energy detecting schemes, while for counting schemes (as for example for the nowadays very common CCD detection in optical multichannel analyzers) the intensity depends on the third power of the frequency ν^3 . [10]

$$\left(\frac{\partial \mu_{mn}}{\partial Q_i} \right)_{Q_0} = \frac{\mu_{mn}(Q_0 + \Delta Q_i) - \mu_{mn}(Q_0 - \Delta Q_i)}{2\Delta Q_i} \quad (14)$$

The Frank-Condon integrals $\langle v''_{(m)} | v'_{(n)} \rangle$ are calculated using the recursive relations of Doktorov [2], which uses the Duschinsky transformation [9] to reduce the integrals over the vibrational ground and vibrational excited state to integrals over the ground state vibrational wave functions only. The integrals $\langle v''_{(m)} | Q_i | v'_{(n)} \rangle$ from the HT terms in equation 13 can be calculated from the FC integrals using the following relation:

$$\begin{aligned} \langle v'_{(m)} | Q_i | v''_{(n)} \rangle &= \sqrt{\frac{\hbar}{2\omega_i}} \left(\sqrt{v_{(n),i} + 1} \langle v'_{(m)} | v''_{(m),i}, \dots, v''_{(n),i} + 1, \dots, v''_{(n),i} \rangle \right. \\ &\quad \left. + \sqrt{v_{(n),i}} \langle v'_{(m)} | v''_{(m),i}, \dots, v''_{(n),i} - 1, \dots, v''_{(n),i} \rangle \right) \end{aligned}$$

2.2.1 Calculation of the emission and absorption spectra

Like for HT spectra, the transitions between two vibronic states m and n are proportional to the square of the transition moment M_{mn} , and to the frequency ν_{mn} of the transition for absorption spectra and the fourth power of the frequency ν_{mn} for emission spectra, respectively. The transition moment is here the sum of the FC (first term in equation 13) and the HT (second term in equation 13) contributions.

The band types of the absorption spectra are calculated from the projections of the transition dipole moment M_{mn} onto the inertial axes of the molecule. In cases where the molecule possesses no symmetry the band types are important indicators to the amount of vibronic coupling. The stronger the coupling between the excited state and the perturbing state, the stronger is the rotation of the transition dipole moment vector towards the orientation in the perturbing state. The orientation of the TDM with respect to the inertial axes can be determined from the relative intensities $I_{a,b,c}$ of the a , b , and c -type transitions in rovibronic spectra using the relations:

$$\begin{aligned} I_a &\propto |M_{mn} \sin \phi \cos \theta|^2 \\ I_b &\propto |M_{mn} \sin \phi \sin \theta|^2 \\ I_c &\propto |M_{mn} \cos \phi|^2 \end{aligned} \quad (15)$$

where θ is the angle between the projection vector of the transition moment on the ab -plane, and the a -axis and ϕ the angle between the TM and the c -axis.

In the case of classical Herzberg-Teller coupling these transitions, which are allowed by vibronic coupling can easily be distinguished from allowed transitions using selection rules of the appropriate point groups, since the pure electronic transition is completely forbidden by symmetry and the vibronic transition gains *all* its intensity (and *all* its band type) through vibronic coupling.[11] Thus, in equation 13 the first term is zero and the intensity of the transition is provided exclusively by the second (and higher) expansion terms. In case of unsymmetric (C_s) molecules, all vibronic transitions are allowed already through the first expansion term in equation 13. This means, that both intensity and band type are Frank-Condon dominated with an additional admixture of Herzberg-Teller type to the intensities and band types. Since the measurement of relative intensities is in general more difficult than the exact determination of the band type of a rovibronic transition, rotationally resolved electronic spectra provide a powerful tool for the assessment of the vibronic coupling.

3 General program decription

3.1 Program philosophy

The input for the program is given via a .ini file, which contains the appropriate parameter settings for the problem. For a listing of the key words and their possible values, see the tables in the following chapters. The transitions in absorption and emission are specified using the Bra/Ket notation described above. Input geometries and Hessian matrices are taken from *ab initio* calculations using program packages like Gaussian, turbomole, MOLCAS, etc. It is important, that both electronic states are treated at the same level of theory. The geometry changes to be fit to vibronic intensities and/or rotational constant changes are expressed as linear combinations of normal mode displacements.

3.2 Program features

FCFit is capable of computing simulations or fits of vibronic spectra in the FC or FCHT approximation for absorption or emission, using a set of transitions, which can be user-defined. If a simulation is to be performed some of the keys needed are placed in the [General] section, while the same keys are placed in the section [Fit] in case of a fit. Furthermore, the geometry change can also be fit to the change of rotational constants from a rotationally resolved electronic spectrum or to both intensities and rotational constant changes (combined fit). Since from a single spectrum only three rotational constants can be extracted, only two geometry parameters can be determined. This limitation can be overcome if rotational constants of more isotopologues are available. For N isotopologues a total of $3N - 1$ geometry parameters can be fit. The fits can be performed with local optimizers as NL2SOL[12] (an extended Levenberg-Marquardt-Algorithm) or PRAXIS [13] (an implementation of the non gradient based method of Brent)[14], or with global optimizers as PGA[15] (a genetic algorithm) or SIMANN[16] (a simulated annealing algorithm). The global minimizer has been extended to allow for intermediate local minimization steps rather than simple cost function evaluations. This technique was originally described by Li and Scheraga as part of a simulated annealing (SA) exploration of the potential hypersurface of proteins. [17] The algorithm, which is also known as *basin-hopping* [18], was successfully used in combination with GA and/or SA to determine minimum-energy structures of fullerenes and atomic clusters. [19]

3.3 The cost function

In a pure FC fit, the geometry of the excited state is varied, until the weighted sum of the squared errors χ^2 of the intensities is minimized:

$$\chi_{Int.}^2 = \sum_i w_i (I_i^{calc.} - I_i^{exp.})^2 \quad (16)$$

Here, the $I_i^{calc.}$ and $I_i^{exp.}$ are the i calculated and experimental intensities, respectively, weighted with individual weight factors w_i . The reduced standard deviation is calculated via:

$$\sigma_{red} = \sqrt{\frac{\chi^2}{n - p}} \quad (17)$$

with n as the number of data points and p the number of parameters.

For a combined fit of intensities and changes of the rotational constants ΔB_g , the total χ^2 is defined as follows:

$$\chi^2 = \chi_{Int.}^2 + \chi_{Rot.}^2 \quad (18)$$

with

$$\chi_{Rot.}^2 = \sum_{g=a}^c w_g (\Delta B_g^{calc.} - \Delta B_g^{exp.})^2 \quad (19)$$

4 *Ab initio* computational prerequisites

For FC simulations of an absorption or emission spectrum several preliminary *ab initio* calculations are necessary: (i) An optimized geometry for each of the electronic states, connected by the transitions in form of cartesian coordinates in units of Å. (ii) The second derivative of the energy (Hessian) for both states, calculated at the same level of theory. Since the output of the hessian matrix differs between the various common *ab initio* program packages two tools are available for conversion of the Hessian into the notation needed for FCfit. Those are: TurboMol2ZmtHess.exe, which expects output from the turbomole program and MolCAS2ZmtHss.exe, which converts MolCAS Hessians to FCfit.

5 Program usage

FCfit is a command line program using a single textual configuration file per program call and should be started by using the syntax:

FCfit file_name

where *file_name* is the full name (relative or absolute) of a configuration file (including any file extension). Additional to such a configuration file, the program needs further input files, which are explained in detail in the section 5.2.

The results of FCfit are written into several files, some of them depending on the contents of the configuration file. The output files are described in section 7

In all cases a log file is written (*file_name.log*), which contains a summary of the program activities.

5.1 General structure of the configuration file

FCfit uses configuration files quite similar to the .ini files known from Windows(R) platforms or the configuration files known from *nix platforms.

These simple text files are divided into several sections, which are identified by the general form

[section_name]

where section_name is a string characterizing a special section.

Each section can contain several entries, which are given by key-value pairs of the style

key_name=value

where *key_name* is a string (possibly with leading and/or trailing spaces), which specifies a special key identifier and value is a (possibly empty) string which fulfills the requirements of this special entry (e.g. it must be a string, an integer value, a boolean value, a floating point value, and so on). Boolean values can use case-insensitively one of the strings "true", "yes", "on", or any nonzero integer value for true. Anything else will be assumed to be "false". Some entries for FCfit are optional, others are required. Empty values of existing keys behave as if the entry has not been provided. Anywhere before, between, or after section lines or entry lines there can be an arbitrary number of empty lines or comment lines. Non-empty comment lines are recognized if the first non-space character is either a semicolon (;) or an ampersand (&). String-like entry values which are introduced with a quotation-mark (") must end with the same " character. Under these conditions the string value is taken as the complete content between these stringizing characters. Otherwise strings are read beginning from the first non-space character and ending with the last non-space character of a line.

5.2 Valid entries of the configuration file

This section explains the possible configuration file sections and entries for FCfit. The following list uses some shortcuts to describe the meaning of entries:

- Boolean, Float, String, Integer, Unsigned: Characterize the expected value type. Unsigned means an unsigned integer value.
- Req.: A required (not optional) entry/section
- Opt.: Optional entry/section. If an entry is optional there is also provided:
- Def.: The default value of an optional value

5.2.1 The section [Flags]

This section is required and contains some general program flags, concerning output of results, and computational details.

Table 1: Valid keys for the section [Flags]: (Req.) This section contains some general program flags.

Keyword	value	comment
"FullInfo":	(Boolean, Opt., Def.: false)	If true, outputs all intermediate program results, otherwise these are ruled by further flags - use this flag carefully, because you can easily create extremely large log files, depending on your project!
"ControlInfo":	(Boolean, Opt., Def.: false)	This flag forces the program to further output additional check values. Its generally not recommended to activate this flag under normal circumstances, because it produces a much larger result file and further dump files. It should be activated if you suspect the program to return errornous results or if it runs under circumstances which lead to an abnormal program termination (Debug mode).
"ReadFreqs":	(Boolean, Opt., Def.: false)	If this flag is not set, the program uses the vibrational frequencies calculated from the mass-weighted Hessian for all frequency-dependent magnitudes, otherwise the existence of two additional files in the source directory (Key "Src-Path", see below) is expected. See general description of the "General" section.
"UseLUDecomp":	(Boolean, Opt., Def.: false)	This flag controls whether the decomposition of the intermediate Q matrix is performed using either the Cholesky method (default) or by means of the LU method.
"InertiaRHS":	(Boolean, Opt., Def.: false)	Determines whether the principal inertia systems of both molecular states are rotated into right systems.

Keyword	value	comment
"ProjectTransRot":	(Boolean, Opt., Def.: false)	Determines whether the program should project out the non-vibrational modes. This leads to a better separation of pure vibrational modes. Furtheron, if you observe severe differences between the non-separated and the separated mode frequencies, you should consider the possibility, that your computed geometry is actually not at a potential minimum.
"PrintHessian":		This flag should be set to false in the currently used version. It will be part of future revisions of the program.

5.2.2 The section [General]

This section (required) contains entries which determine the basic input parameter of the program. The following list shows the basic input files. (\$MolName denotes a file prefix obtained from the "MolName" key and \$LowerStateName and \$UpperStateName denote the state names obtained from the key "LowerStateLabel" and "UpperStateLabel", respectively, cf. Table 2):

1. Geometry files of the two molecular states. Name convention:

\$MolName_\$LowerStateName.zmt and \$MolName_\$UpperStateName.zmt for the lower and upper state, respectively.

Example For the key "MolName" in the section [General] we chose the value: "water". The key "LowerStateLabel" is set to "s0", the key "UpperStateLabel" is set to "s1". The name of the two geometry files must then be: "water_s0.zmt" for the ground state and "water_s1.zmt" for the excited state. Other labels for the excited state like s2, t1, la, lb, etc. are allowed and should be used extensively in case of different excited states.

Each file is a text file which may contain leading comment lines (introduced by a # per line). Each data line must have the format:

NUCLID_SYMBOL X_COORD Y_COORD Z_COORD

where each item is separated from the next one by an arbitrary number of spaces. NUCLID_SYMBOL: Symbol of the atom or nuclid. Either use normal element symbols or nuclid symbols following the nomenclature XXAtomSymbol where XX is the corresponding nuclid number (mass number), e.g. 13C for the carbon nuclid with 13 nucleons. An additionally allowed symbol is D for the 2H nuclid. Note that normal element symbols denote the most abundant nuclid of the corresponding element, e.g. 12C for C, 14N for N and so on. X_COORD, Y_COORD, Z_COORD: x-, y-, and z-coordinate of the corresponding atom in units of Angstrom.

Example The input for water might look like:

```
O 0.000000 0.000000 0.121767
H 0.000000 0.758120 -0.487067
H 0.000000 -0.758120 -0.487067
```

or

```
16O 0.000000 0.000000 0.121767
1H 0.000000 0.758120 -0.487067
1H 0.000000 -0.758120 -0.487067
```

the isotopologue D₂¹⁸O would be:


```

18O 0.000000 0.000000 0.121767
2H 0.000000 0.758120 -0.487067
2H 0.000000 -0.758120 -0.487067

```

Note: FCfit does currently support only molecules of non-linear geometry which possess $3N-6$ vibrational degrees of freedom.

2. Force matrix files of the two molecular states. Name convention:

\$MolName_\$LowerStateName.hss and \$MolName_\$UpperStateName.hss for the lower and upper state, respectively.

Each file is a text file which contains the linearly written (flattened) lower triangle of the symmetric $3N \times 3N$ force matrix (in units of Hartree/Bohr²)

Example The Hessian from a turbomole (RI)MP2 calculation of water looks like:

CARTESIAN FORCE CONSTANT MATRIX (hartree/bohr**2)							
(projected out of translational & rotational space)							
ATOM		1 o			2 h		
		dx	dy	dz	dx	dy	dz
1 o	dx	0.6724700					
	dy	0.0000000	0.0000000				
	dz	0.0000011	0.0000000	0.4955167			
2 h	dx	-0.3362333	0.0000000	-0.1986947	0.3729508		
	dy	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	
	dz	-0.2656678	0.0000000	-0.2477588	0.2321803	0.0000000	0.2288346
3 h	dx	-0.3362367	0.0000000	0.1986935	-0.0367175	0.0000000	0.0334875
	dy	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
	dz	0.2656666	0.0000000	-0.2477579	-0.0334856	0.0000000	0.0189242
ATOM		3 h					
		dx	dy	dz			
3 h	dx	0.3729542					
	dy	0.0000000	0.0000000				
	dz	-0.2321810	0.0000000	0.2288337			

These are the 45 entries from the lower triangle of the 9×9 force constant matrix of water. Using the TurboMol2ZmtHess.exe tool this matrix is converted to a flattened vector of 45 force constants:

```

0.6724700
0.0000000
...
...
...
0.2288337

```

N.b. the file names in the case discussed above would be "water_s0.hss" and "water_s1.hss".

3. If the "ReadFreqs" key from the [Flags] section is set, two files containing the vibrational frequencies must be provided. The names of these files must be:

\$MolName_\$LowerStateName.frq and \$MolName_\$UpperStateName.frq for the lower and upper state, respectively.

Each frequency file contains successively the 3N-6 vibrational modes in cm^{-1} of the corresponding molecular state.

Example The file from the above example "water_s0.frq" would read like:

```
1694.4435
3708.3599
3818.7526
```

If the key "ReadFreqs" is not set, the vibrational frequencies are calculated from the respective Hessian.

Furtheron, the [General] section determines the path for the result files, e.g. the log file, MDL^(R).mol files, HyperChem^(R).ext IR mode files, Molden^(R).mdn files, and the spectral simulation files.

Table 2: Valid keys for the section [General]: (Req.) This section contains some general program flags.

Keyword	value	comment
"AtomCount":	(Integer, Req.)	This positive integer value determines the number of atoms in the molecule which is described by two geometric states. The program user has to ensure, that the input files (geometry, force matrix, frequency etc.) are consistent with this number.
"Title":	(String, Opt., Def.: empty)	Simple textual description, which is written into the log file of the program. This entry does not have any influence on the internal program calculations.
"MolName":	(String, Req.)	This string determines the prefix of the basic input file names, so it forces naming constraints in several input file names like the .hss, .zmt, and .freq files. Whether case sensitiveness must be provided depends on the underlying file system.
"SrcPath":	(String, Opt., Def.: empty)	Platform-dependent description of the source path which provides the basic input files. This description can be either a relative or an absolute path.
"DstPath":	(String, Opt., Def.: empty)	Platform-dependent description of the destination path to which the output files are written. This description can be either a relative or an absolute path.
"LowerStateLabel":	(String, Opt.; "s0")	Identifier for the lower state. Feasible names must only contain letters and digits. The identifier is used to recognize the input files for the lower state and is also used to tag the corresponding output files related to the lower state.
"UpperStateLabel":	(String, Opt.; "s1")	Identifier for the upper state. Feasible names must only contain letters and digits. The identifier is used to recognize the input files for the upper state and is also used to tag the corresponding output files related to the upper state.

Keyword	value	comment
"SingleDistortion":	(Boolean, Opt., Def.: false)	If this flag is activated, the program will perform a single distortion of the molecule in the excited (S1) geometry. Note that this flag - if true - has higher priority than the flag "DoFit", v.i. If single distortion is activated, the following keys do control the program behaviour: "ChangeModeCount", "StateModesUsed", "ModeIndices", and "StartValues".
"ChangeModeCount":	See [Fit]/"ChangeModeCount".	If this key does not exist in the [General] section, [Fit]/"ChangeModeCount" will be expected to exist.
"StateModesUsed":	See [Fit]/"StateModesUsed".	If [General]/"ChangeModeCount" exists, this entry must also exist in the same section and [Fit]/"StateModesUsed" will not be tried.
"ModeIndices":	See [Fit]/"ModeIndices".	If [General]/"ChangeModeCount" exists, this entry must also exist in the same section and [Fit]/"ModeIndices" will not be tried.
"StartValues":	See [Fit]/"StartValues".	If [General]/"ChangeModeCount" exists, this entry must also exist in the same section and [Fit]/"StartValues" will not be tried.
"SaveExtFiles":	(Boolean, Opt., Def.: false)	If this flag is activated, additional MDL .mol of the lower and the upper state geometries and HyperChem .ext files for their corresponding IR spectra will be stored after the computation has finished.
"SaveMoldenFiles":	(Boolean, Opt., Def.: false)	If this flag is activated, additional Molden .mdn of the final geometries will be stored.
"DoFit":	(Boolean, Opt., Def.: false)	If "SingleDistortion" = false and this flag is activated, the program will perform an iterative fit. If "DoFit" == true, the section [Fit] will be evaluated for further control data. Note that fits can use information both from spectra simulations (see [Recursion]/"SpectraCount") and from inertia information of isotopes of the corresponding molecule (see [Fit]/"UseDeltaRotConsts").
"AlignedModes":	See [Fit]/"AlignedModes"	
"AlignedModeN":	(where N is an integral number) See [Fit]/"AlignedModeN"	

5.2.3 The section [Recursion]

This optional section contains entries which determine parameters which control the recursion process, which is necessary to perform the FC calculation. Here, the user can chose the type of spectrum (absorption or emission) to be simulated or fit, the number of spectra, a set of N quantum strings, which define the initial states (N Bras) and the respective N input files which contain the final states (Kets) of the transitions. Using the key words "MaxEnergy", "MaxSingleQuanta", "MaxQuantaSum", "MaxExcitedModes" the performance of the FC calculation can be influenced. "MaxEnergy" gives the highest energy in the calculation. This depends on the range, which is to be simulated and should not exceed 4000 cm^{-1} . "MaxSingleQuanta" gives the maximum excitation of each vibrational mode. "MaxSingleQuanta"=2 means, that only transi-

tions up to first overtone of each mode are taken into account in the recursion. "MaxQuantaSum" gives the maximum of the sum of all excitations for the calculated transitions. E.g. a transition to a Ket state defined by 100200101 has a quanta sum of "5". Any value below "5" for "MaxQuantaSum" would exclude this transition from the recursion. "MaxExcitedModes" gives the cut-off for the maximum number of excited modes. In the preceeding example this number is "4" and any number below this will exclude this transition. Be very carefull with "MaxSingleQuanta" ≥ 3 , "MaxQuantaSum" ≥ 4 and "MaxExcitedModes" ≥ 5 , since these values might cause extremely long computation times.

Table 3: Valid keys for the section [Recursion]: (Opt.) This section contains parameters which control the recursion process

Keyword	value	comment
"SpectraType":	(One of "A", "E", "A/E", or "E/A", Req.)	The spectra type used for the spectra simulation or fit. Feasible values are the following, case-insensitive values: "E" for emission spectra, "A" for absorption spectra, and "A/E" or "E/A" if both spectra type may occur (mixed spectra situation). Please note that mixed spectra have further requirements on other data items: - There must exist a proper TypeN key matching to every (BraN, FileN) entry - The key "NuDependence" is ignored and instead there are a similar number of proper NuDependenceN entries expected - Similar TypeN and NuDependenceN entries are needed inside the [FurtherLines] section for every further spectrum.
"SpectraCount":	(Unsigned integer, Opt., Def.: 0)	This key is required to exist and to give a positive integer value which describe the number of spectra to be simulated, if "DoFit" is true and [Fit]/"UseLineIntents" is true as well.
"CalcFullSpectra":	(Boolean, Opt., Def.: false):	This flag should be used with care, because it will enforce that during the FC computation *all* possible transitions will be computed, which is usually an unnecessarily long process. It's default value of false is normally sufficient, because the algorithm will then only compute the required intermediate transitions that are necessary to compute the wanted spectra lines.

Keyword	value	comment
"BraN":	(A quantum string, Opt.)	If a line spectra fit will be done there need to be exactly M = "SpectraCount" entries "Bra1", "Bra2", ..., "BraM". Each entry must provide a quantum string, that is a sequence of integer values (w/o spaces), with optional bra-brackets, e.g. "0", "01002", or " $\langle 1002 \rangle$ ". The maximum number of digits must be equal to the number of vibrational modes. If less quanta are specified, the rest is expected to be 0. The i-th number corresponds to the vibrational quantum number of the i-th mode (where i is increasing from left to right in the direction of higher energy modes). N.B.: A bra-state corresponds to the initial state of the spectral process, i.e. this will be the state of lower energy in case of absorption spectra and the state of higher energy in emission spectra.

Keyword	value	comment
"FileN":	(A file name, Opt.)	<p>If a line spectra fit will be done, there need to be exactly $M = \text{"SpectraCount"}$ entries "File1", "File2", ..., "FileM". Each entry must provide a file name without path information that is expected to exist in the directory specified with the key [General]/"SrcPath". Each file "FileX" must contain transition data that corresponds to the bra state given by the quantum string of the key "BraX". The files must be text files which contain an arbitrary number of transition data, where each datum has the format:</p> <p>a[b][c][d][e][f][..] x.yz [w.ww] anything_else</p> <p>where a,b,...,f,.. are integer numbers in the range 0..9 (these are quantum strings of the ket transitions belonging to the associated bra state) and x.yz is a floating point number ≥ 0 which represents the relative experimental intensity of this $\langle \rangle$-transition. In this list there <i>*must*</i> exist exactly one \rangle state with an experimental intensity of 1.0, which determines the reference line of the given spectrum. w.ww is optionally read if the fit is configured to use intensity weights ([Fit]/"UseWeights" = true) and these weights shall not forced to be uniform ([Fit]/"UniformIntensityWeight" exists), otherwise we don't try to read this third column. If read, it must be any non-negative floating-point number. After the intensity number (or optionally the weight value), there might additionally be any further information ("anything_else"), which is ignored by the program. Also, there might be any lines containing only whitespaces or comments (those must be initiated by a # character). N.B.: A ket-state corresponds to the final state of the spectral process.</p>

Keyword	value	comment
"NuDependence":	(One of "None", "Nu1", "Nu3", "Nu4", Opt., Def.: "None")	If the spectra type is either "E" or "A" (and *not* a mixed type!), this entry is read. Specifies the applied frequency correction for relative intensity calculations from FC factors. The correction should depend on the spectral type and determines the power of the frequency. Nu1 corresponds to absorption spectra, Nu3 and Nu4 to emission spectra (nu3 for photon counting experiments, nu4 for energy detecting experiments), and None is no frequency dependency correction at all. Please note that one has to provide additional NuBaseXX entries for the corresponding electronic transition energy in cases where NuDependence = None. This also applies to corresponding NuBaseXX entries in the "FurtherLines" section!
"NuBaseN":	(A positive floating-point number)	If a line spectra fit will be done, there need to be exactly $M = \text{"SpectraCount"}$ entries "NuBase1", "NuBase2", .., "NuBaseM", if "NuDependence" != "None". Each entry must then provide a base frequency (in cm^{-1}) that is used as part of the classical frequency dependence formula for intensities. This base frequency is required and corresponds to the electronic energy of the transition, which itself is not required as part of the vibrational analysis. Depending on the spectra type, each calculated vibrational frequency is either added or subtracted from this base energy to get the total energy in frequency units.
"MaxEnergy":	(A floating point number)	This is a skip criterium for every transition, which's overall vibrational energy is higher than this value.
"MaxSingleQuanta":	(An unsigned integer)	During the systematic calculation of required transitions as part of the Doktorov-Malkin-Manko approach this value is used as the maximum vibrational quantum number for each mode. This value strongly influences the performance of the computation and the value should be set as low as possible.
"MaxQuantaSum":	(An unsigned integer)	This is a skip criterium for every transition, which's overall sum of vibrational modes is higher than this value.
"MaxExcitedModes":	(An unsigned integer)	This is a skip criterium for every transition, which's sum of excited vibrational modes is higher than this value. An excited mode has a vibrational quantum number different from 0.

Keyword	value	comment
"HashSize":	(An unsigned integer, Opt.)	This parameter should only be set by experts who know what they do. If different from 0, this value will be set as the initial hash map size to store the computed transitions, otherwise the program will estimate the initial size by a preliminary state count.
"WriteFCFiles":	(Boolean, Opt., Def.: false)	Specifies whether any *.sim and *.spc output is repeated as *.sim.fc and *.spc.fc files which contain the absolute FC factors instead of intensities. This global flag can be overwritten by local WriteFCFiles flag in the [FurtherLines] section.
"WriteOverlapFiles":	(Boolean, Opt., Def.: false)	Specifies whether any *.sim and *.spc output is repeated as *.sim.overlap and *.spc.overlap files which contain the overlap integrals instead of intensities. This global flag can be overwritten by local WriteOverlapFiles flag in the [FurtherLines] section.
"FreqLabelFile":	(Filename, Opt.)	If the key exists, it must specify be a file name without path information that is expected to exist in the directory specified with the key [General]/"SrcPath". This file must contain a sequence of floating-point numbers which shall be used as labels specifying a mode (instead of their calculated frequency). The number of these energy labels must be identical to that of the number of vibrational modes. Note that their nominal values don't influence the internal calculations, but they have influence on the energy column of all output files.

5.2.4 The section [Fit]

This section contains all parameters relevant for the fit control. It is not required to exist, if no fit is done. If a fit shall be done, then at least one of the boolean flags "UseLineIntents" and "UseDeltaRotConsts" must be true. "UseLineIntents"=true performs a FC fit, "UseDeltaRotConsts"=true performs a fit of the changes of rotational constants from a rotationally resolved electronic spectrum. Both parameters=true performs a combined fit.

Table 4: Valid keys for the section [Fit]: (Opt.) This section contains parameters relevant to the fit

Keyword	value	comment
"UseLineIntents":	(Boolean, Opt., Def.: true)	If this value is true we compute a χ^2 value which additionally includes the deviations of relative line intensities.

Keyword	value	comment
"UseDeltaRotConsts":	(Boolean, Opt., Def: false)	If this value is true we compute a χ^2 value which additionally includes the deviations of given changes of the rotational constants. In this case the following other keys are interpreted: "DeltaXWeight" (X = A, B, C), "DeltaX_MHz" (X = A, B, C), and the contents of the section [FurtherIsotopomers] are attempted to read.
"UseWeights":	(Boolean, Opt., Def.: false)	If true, we expect in all experimental files FileX an additional 3rd column, which contains non-negative values for the relative weights of the given intensities. Note that a statistical weight can be realized by using the inverse squared values of the standard deviations of the intensities for these values.
"UniformIntensityWeight":	(Double, Opt., Def.: 0.0)	If "UseWeights" = true, and this key is provided, we do read the weights from our files, but use this uniform weight for *all* intensities! This provided uniform weight must be a non-negative floating-point number. This key is useful if one wants to compare the difference between individual weights, relative weights and equal weights for every line.
"StateModesUsed":	(Either the value of the key "LowerStateLabel" or the value of the key "UpperStateLabel" recognised caselessly, Req.)	Determines whether you want to use lower state or upper state modes for the elongation of the upper state geometry during fit or single distortion.
"ChangeModeCount":	(Unsigned integral value)	Specifies the number of modes that should be elongated during fit or single distortion. If the number is less than the number of all vibrational modes, the program expects the additional key "ModeIndices".
"ModeIndices":	(Comma-separated list of positive integers)	This entry is only read, if "ChangeModeCount" is less than the number of vibrational modes. If read, each number specifies a mode (in ascending order of energy), that should be elongated during the fit or single distortion.
"StartValues":	(Comma-separated list of floating-point numbers, Opt.)	If existing, this entry must contain exactly "ChangeModeCount" values, where the i-th value is used as initial elongation of the i-th mode to be elongated. If these start values are not provided, the initial elongation parameters are set to 0.
"DeltaXWeight" (X = A, B, C):	(Non-negative floating-point number, Opt.)	Each value is used as weighting factor for the corresponding change of rotational constant between the lower and upper state. These entries are required, if "UseDeltaRotConsts" = true.

Keyword	value	comment
"DeltaX_MHz" (X = A, B, C):	(Non-negative floating-point number, Opt.)	Each value is used as the experimental changes of the upper state geometry rotational constants relative to the lower state rotational constants (in MHz), i.e. A(S1) - A(S0), B(S1) - B(S0), C(S1) - C(S0). These entries are required, if "UseDeltaRotConsts" = true.
"FitSolver":	(String, Req.)	Specifies one of the known fit solving algorithms, which's parameters are explained in corresponding sections which have the same name as the algorithm. Feasible values are "NL2SOL", "PRAXIS", "SIMANN", or "PGA" case-insensitive). "NL2SOL" and "PRAXIS" are so-called local minimizers, while "SIMANN" and "PGA" are so-called global minimizers.
"AlignedModes":	(Comma-separated list of unsigned integers, Opt.)	Specifies a list of vibrational indices that shall have a defined sign of it's Hessian eigenvector, i.e. it's normal coordinate. As result of the EV decomposition either sign is equally valid and a priori it's unknown which sign a corresponding normal vector will have. If a special sign of selected modes is important for you, this control parameter is the way to realize that. Just add all to-be-aligned mode indices onto this list (The index order follows the order of ascending vibrational frequencies) and ensure that you provide for the N-the entry a corresponding key "AlignedModeN", see "AlignedModeN".
"AlignedModeN":	(Comma-separated data list, Opt.)	For each index X on the list of entry "AlignedModes" there must exist a corresponding key "AlignedModeX" which provides data alignment information with the following format: AlignedModeMM = AA, CC, SGN where MM is the mode index, AA is a selected atom number (starting with 1), CC is the selected atomic coordinate (x, X, y, Y, z, or Z) and SGN is the sign (+ or -) which this mode component should have.

5.2.5 The section [FurtherIsotopomers]

This section is only relevant, if the program should take into account additional inertia data of isotopomers for fit or simulation. If it exists, the key "Count" **must** be present. If, in total three isotopomers have been measured ("Count"=3), the input for "DeltaRots_MHz_N" might read:

"DeltaRots_MHz_1"= 21.5, 4.2, 7.8

"DeltaRots_MHz_2"= 23.1, 3.8, 6.1

"DeltaRots_MHz_3"= 19.1, 5.8, 8.1

The definition of the isotopomes (again using water as example) is then given as:

"Iso_1":H,O,H
 "Iso_2":D,O,H
 "Iso_3":H,18O,H

Here, "Iso_1" is the normal isotopomer, "Iso_2" is HOD, and "Iso_3" is H₂¹⁸O.
 representing the changes of the rotational constants of the three isotopomers upon excitation.

Table 5: Valid keys for the section [FurtherIsotopomers]: (Opt.) This section contains parameters relevant to the fit

Keyword	value	comment
"Count":	(Unsigned integral, Req.)	Specifies the number of additional isotopomers.
"DeltaRotWeights_N":	(Comma-separated triple of non-negative numbers)	If M = "Count" inertia data are used for fit or simulation, there must exist the keys "DeltaRotWeights_1", "DeltaRotWeights_2", ..., "DeltaRotWeights_M". Each entry shall contain a triple of floating-point numbers that are used as weighting factors for the rotational constant differences A, B, and C, resp.
"DeltaRots_MHz_N":	(Comma-separated triple of non-negative numbers)	If M = "Count" inertia data are used for fit or simulation, there must exist the keys "DeltaRots_MHz_1", "DeltaRots_MHz_2", ..., "DeltaRots_MHz_M". Each entry shall contain a triple of floating-point numbers which are the experimental changes of the upper state geometry rotational constants relative to the lower state rotational constants (in MHz), i.e. $A(S_1) - A(S_0)$, $B(S_1) - B(S_0)$, $C(S_1) - C(S_0)$.
"Iso_N":	(Comma-separated list of strings)	If M = "Count" inertia data are used for fit or simulation, there must exist the keys "Iso_1", "Iso_2", ..., "Iso_M". Each entry of nuclid symbols specifying the corresponding isotopomers. The nuclids must be declared in the same order as inside the geometry files.

5.2.6 The Section [FurtherLines]

This is an optional section that allows the program user to specify **additional** transitions to be computed after the overall fit has been done. These transitions are not taken into account during the fit. The corresponding output files have the extension ".spc".

Table 6: Valid keys for the section [FurtherLines]: (Opt.) This section contains parameters relevant to the fit

Keyword	value	comment
"SpectraCount":	(Unsigned integer, Opt., Def.: 0)	Specifies the number M of additional (i.e. further) input files that shall be used to compute corresponding files of computed spectra. There must exist further keys "Bra1", "Bra2", ..., "BraM" as well as "File1", "File2", ..., "FileM", which describe the wanted bra state and the input files that provide the wanted lines (ket states).
"BraN":	(Quantum string, Opt.)	There need to be exactly M = "SpectraCount" entries "Bra1", "Bra2", ..., "BraM". Each entry must provide a quantum string, for details see [Recursion]/"BraN".
"FileN":	(A file name, Opt.)	There need to be exactly M = "SpectraCount" entries "File1", "File2", ..., "FileM". Each entry must provide a file name without path information that is expected to exist in the directory specified with the key [General]/"SrcPath". Each file "FileX" must contain transition data that corresponds to the bra state given by the quantum string of the key "BraX" (see also [Recursion]/"FileN"). Each transition datum must have the following format: a[b][c][d][e][f][..] anything_else where a,b,...,f,.. are integer numbers in the range 0..9. If this list does not contain the —0i state, which determines the reference line of the given spectrum, this will be additionally computed. After the quantum string, there might additionally be any further information ("anything_else"), which is ignored.
"WriteFullSpectra":	(Boolean, Opt., Def.: false)	This flag forces the program to neglect the contents of the further line-files and instead of this writes the complete spectra into the resulting .spc files.
"WriteFCFiles":	(Boolean, Opt., Def.	Specifies whether any *.spc output is repeated as *.spc.fc files which contain the absolute FC factors instead of intensities. This local flag overrides the global WriteFCFiles flag in the [Recursion] section.
"WriteOverlapFiles":	(Boolean, Opt., Def.	Specifies whether any *.spc output is repeated as *.spc.overlap files which contain the overlap integrals instead of intensities. This local flag overrides the global WriteOverlapFiles flag in the [Recursion] section.

Keyword	value	comment
"TypeN":	(See [Recursion]/"SpectraType", Opt.):	A spectra type specification for the N-th further line spectrum, required for each additional "BraN", if [Recursion]/"SpectraType" was set to a mixed-spectra type.
"NuDependenceN":	(See [Recursion]/"NuDependence", Opt.):	A frequency dependency specification for the N-th further line spectrum, required for each additional "BraN", if [Recursion]/"SpectraType" was set to a mixed-spectra type.
"NuBaseN":	(See [Recursion]/"NuBaseN", Opt.):	The electronic base frequency specification for the N-th further line spectrum, required for each additional "BraN", if "NuDependenceN" was set different from "None".

5.2.7 The section [NL2SOL] (Opt.)

This section provides the control parameters of the local minimizer NL2SOL.

Table 7: Valid keys for the section [NL2SOL]: (Opt.) Parameters relevant to the performance of the local optimizer

Keyword	value	comment
"MaxIter"	(Integer, Req.)	Maximum number of iterations
"MaxFuncEvals"	(Integer, Req.)	Number of fuction evaluations
"RelParamConv"	(Float)	
"RelChiSqrConv"	(Float)	
"AbsChiSqrConv"	(Float)	

5.2.8 The section [Local-NL2SOL] (Opt.)

Optional section used for all global minimizers. It has the same contents than the normal [NL2SOL] section, all missing entries are taken from the required [NL2SOL] section. While the normal [NL2SOL] is used for the last final iteration, this one is used during each global fit step.

5.2.9 The section [PGA]

Global minimizer section (since version 2.8.5) using the genetic algorithm variant basin-hopping. While using this minimizer be aware of the fact, that the program still needs the local minimizer NL2SOL for this option. The program looks for the same number of lower and upper parameter limits, as parameters are defined. It will use default values -0.5 and +0.5, if these limits are not provided by the user. Note that the final integer numbers correspond to the mode numbers which have to be fit:

LowerLimit2 = -0.2
 UpperLimit2 = +0.2
 LowerLimit3 = -0.3
 UpperLimit3 = +0.3
 LowerLimit6 = -0.5
 UpperLimit6 = +0.5

Table 8: Valid keys for the section [PGA]: (Opt.) Parameters relevant to the performance of the global optimizer

Keyword	value	comment
BitsPerParam	(Int., Req.)	Number of bits per parameter used in gene pooling.(Maximum: Number of bits per int, usually 31)

All following parameters are optional. Please read the pgapack documentation for the used default parameters.

"GenerationCount"	(Int.)	Maximum number of iterations (generations)
"PopulationSize"	(Int.)	Size of gene population
"Elitism"	(Real)	Percentage of elitism
"RandomSeed"	(Int., ≥ 1)	Random seed value
"CrossoverType"		Crossover type. Feasible values are (case-insensitive): 1-Pt, 2-Pt, Uniform
"CrossoverProb"	(Real)	Crossover probability
"MutationProb"	(Real)	Mutation probability (%)
"SelectionType"		Selection type. Feasible values are (case-insensitive): Prop, SUS, Trn, PTrn
"FitnessType"		Fitness type. Feasible values are (case-insensitive): Raw, LinNorm, LinRank
"AllowDuplicates"	(Boolean)	Defines, whether duplicate genes are allowed.

6 Necessary input files

Apart from the configuration file and the Hessian (.hss) and geometry(.zmt) files for each state described in detail in section 5.2.2, additionally a number of files, which contain the quantum strings of the kets for a given bra as initial state and the intensities of all $\langle | \rangle$ transitions is needed. The necessary definitions are given in the section [Recursion]. For the simulation of a single (SpectraCount=1) absorption spectrum (SpectraType=A) starting from the vibrationless ground state (Bra1=0) a file is expected to exist in the general path, whose name is defined in [Recursion] (File1=*any filename*). In case of the water example from above the number of normal modes is three and File1 might look like:

```

000  1
100  0.2  1.0  nu_1
010  0.1  1.0  nu_2
001  0.2  1.0  nu_3
200  0.01  0.1  2*nu_1
011  0.01  0.1  nu_2+nu_3

```

The first column (req.) gives 6 quantum strings, describing 6 experimentally observed (and assigned!) transitions, $\langle 000|000 \rangle$ (the electronic origin), $\langle 100|000 \rangle$ (the transition to the ground tone of the bending mode), $\langle 010|000 \rangle$ (the transition to the ground tone of the sym. stretching mode), etc. The next column (req.) gives the experimentally determined intensities. N.b. that one of the transitions is used as reference and must have the intensity 1. This must not necessarily be the electronic origin! The third column (opt.) gives a weighting factor for each intensity. If no weighting factor is given, a uniform weight of 1 is used. The fourth column (opt.) gives a free comment about the transition. Here one should use a notation (or free text), which unequivocally describes the transition. Once the quantum strings get longer (for indole we have e.g. 42...) and overtone and/or combination bands have to be considered, it is quite cumbersome to identify the vibrational levels from the notation in the first line.

If SpectraType=E was chosen, the above example gives the emission spectrum from the vibrationless origin (remember, Bra1 was chosen to be 0) into the ground state vibrational levels given in the table.

A second emission spectrum through via a different excited state band can be added by SpectraCount=2, which requires then a second Bra, e.g. Bra2=100, and a second experimental file, e.g. File2=*any other filename*. This contains then the observed intensities of transitions reached from $\langle 100 |$ in the upper state:

```

000  0.1
100  0.2   1.0  nu_1
010  1.0   1.0  nu_2
001  0.2   1.0  nu_3
200  0.01  0.1  2*nu_1
011  0.01  0.1  nu_2+nu_3

```

Here (following the propensity rule), I have chosen the diagonal transition as reference (intensity 1).

7 Program output files

Depending on the keys which are set in the different sections of the .ini file a multitude of different output files can be generated. Here follows a short description about the content of the various files:

Filename	Example	Description
$\{\text{MolName}\}.\text{log}$	water.log	Contains the log information of the simulation or the fit. The projected Hessians, the eigenvalues, and the Duschinsky matrix are written in the log file.
$\{\text{MolName}\}_{\text{LowerStateName}}.\text{ir}.\text{ext}$	water_S0ir.ext	Hyperchem .ext file for displaying the ground state vibrations of the molecule.
$\{\text{MolName}\}_{\text{LowerStateName}}.\text{mol}$	water_S0.mol	Hyperchem .mol file of the ground state structure of the molecule.
$\{\text{MolName}\}_{\text{LowerStateName}}.\text{mdn}$	water_S0.mdn	Molden frequency file of the ground state.
$\{\text{MolName}\}_{\text{UpperStateName}}.\text{ir}.\text{ext}$	water_S1ir.ext	Hyperchem .ext file for displaying the excited state vibrations of the molecule.
$\{\text{MolName}\}_{\text{UpperStateName}}.\text{mol}$	water_S1.mol	Hyperchem .mol file of the excited state structure of the molecule.
$\{\text{MolName}\}_{\text{UpperStateName}}.\text{mdn}$	water_S1.mdn	Molden frequency file of the excited state.

Filename	Example	Description
$\{\text{MolName}\}_{\text{Bra}\{N\}}.\text{spc.overlap}$	water_Bra1.spc.overlap	Values of the overlap integrals $\langle v' v'' \rangle$ for all transitions, which are computed using the parameters defined in "MaxEnergy", "MaxSingleQuanta", "MaxQuantaSum", "MaxExcitedModes" if "WriteFullSpectra" is true. If "WriteFullSpectra" is false, only overlap integrals are given for transitions defined in the section [Further Lines] through "Bra1"..."BraN" and "File1"..."FileN".
$\{\text{MolName}\}_{\text{Bra}\{N\}}.\text{spc.fc}$	water_Bra1.spc.fc	Values of the Franck-Condon factors $\langle v' v'' \rangle^2$ for all transitions, which are computed using the parameters defined in "MaxEnergy", "MaxSingleQuanta", "MaxQuantaSum", "MaxExcitedModes" if "WriteFullSpectra" is true. If "WriteFullSpectra" is false, only overlap integrals are given for transitions defined in the section [Further Lines] through "Bra1"..."BraN" and "File1"..."FileN".
$\{\text{MolName}\}_{\text{Bra}\{N\}}.\text{spc}$	water_Bra1.spc	Values of the intensities $\frac{\langle v' v'' \rangle^2}{\langle v' v'' \rangle_{ref}^2}$ multiplied with the appropriate ν -dependence for all transitions, which are computed using the parameters defined in "MaxEnergy", "MaxSingleQuanta", "MaxQuantaSum", "MaxExcitedModes" if "WriteFullSpectra" is true. If "WriteFullSpectra" is false, only overlap integrals are given for transitions defined in the section [Further Lines] through "Bra1"..."BraN" and "File1"..."FileN".
$\{\text{MolName}\}_{\text{Bra}\{N\}}.\text{sim.overlap}$	water_Bra1.sim.overlap	Values of the overlap integrals $\langle v v \rangle$, given for transitions defined in the section [Further Lines] through "Bra1"..."BraN" and "File1"..."FileN".

Filename	Example	Description
$\{\text{MolName}\}_{\text{Bra}}\{\text{N}\}.\text{sim}.\text{fc}$	water_Bra1.sim.fc	Values of the FC integrals $\langle v v \rangle^2$, given for transitions defined in the section [Further Lines] through "Bra1"... "BraN" and "File1"... "FileN".
$\{\text{MolName}\}_{\text{Bra}}\{\text{N}\}.\text{sim}.\text{ht}.\text{tdm0}$	water_Bra1.sim.ht.tdm0	Values of the first term $\mu_{mn}(Q_0) \langle v''_{(m)} v'_{(n)} \rangle$ in equation 13, given for transitions defined in the section [Further Lines] through "Bra1"... "BraN" and "File1"... "FileN".
$\{\text{MolName}\}_{\text{Bra}}\{\text{N}\}.\text{sim}.\text{ht}.\text{tdm}$	water_Bra1.sim.ht.tdm	Values of the second term in equation 13 $\sum_i \left(\frac{\partial \mu_{mn}}{\partial Q_i} \right)_{Q_0} \langle v''_{(m)} Q_i v'_{(n)} \rangle$, given for transitions defined in the section [Further Lines] through "Bra1"... "BraN" and "File1"... "FileN".

8 Compiling the source code

FCfit is written in ISO-ANSI 1998 C++ and thus you need a standard C++ compiler for successfully compiling the sources. The sources contain several flags to cope with incomplete standard C++ compilers and the sources have been successfully compiled with

-gcc 2.95 -gcc 3.4 (MinGW32) -Borland C++ compiler 5 (0x0550)

FCfit is provided as a statically bound program, which depends on the following libraries:

- The boost library ($\zeta=1.30.0$, last tested with 1.35.0), see <http://www.boost.org> - The C++ sources of the library cpputility - The C++ sources of the library laulib - The C++ sources of the library port3 - The C++ sources of the library f2cpp - The C++ sources of the library pgapack.cpp - The Lapack and Blas libraries

You have to modify the Makefile.in files if your paths for the above mentioned C++ sources differ from the following:

Boost: /usr/local/lib/boost_1_33_1 cpputility: .. laulib: ../laulib port3: .. f2cpp: ..

On most Linux or Unix systems Makefile's are created after using the console directive "make -f Makefile.in Makefiles".

9 Examples

References

- [1] E. V. Doktorov, I. A. Malkin, and V. I. Man'ko, J. Mol. Spec. **56**, 1 (1975).
- [2] E. V. Doktorov, I. A. Malkin, and V. I. Man'ko, J. Mol. Spec. **64**, 302 (1977).
- [3] T. E. Sharp and H. M. Rosenstock, J. Chem. Phys. **41**, 3453 (1964).
- [4] A. Warshel and M. Karplus, Chem. Phys. Letters **17**, 7 (1972).

- [5] R. Islampour, M. Dehestani, and S. Lin, J. Mol. Spec. **194**, 179 (1999).
- [6] R. Berger, C. Fischer, and M. Klessinger, J. Phys. Chem. A **102**, 7157 (1998).
- [7] J. Dods, D. Gruner, and P. Brumer, Chem. Phys. Letters **261**, 612 (1996).
- [8] B. Crompt, M. Triest, J. T. Carrington, and C. Reber, Spectrochim Acta A, special issue entitled Theoretical Spectroscopy: State of the Science **55**, 575 (1999).
- [9] F. Duschinsky, Acta Physicochimica U.R.S.S. **7**, 551 (1937).
- [10] R. J. LeRoy, J. Mol. Spec. **192**, 237 (1998).
- [11] G. Herzberg and E. Teller, Z. Phys. Chem. B **21**, 410 (1933).
- [12] J. E. Dennis, Jr., D. M. Gay, and R. E. Welsch, ACM Transactions on Mathematical Software **7**, 348 (1981).
- [13] K. R. Gegenfurter, Behav. Res. Meth. Instr. & Comp. **24**, 560 (1992).
- [14] R. P. Brent, *Algorithms for minimisation without derivatives*, Prentice hall, Englewood cliffs, 1973.
- [15] D. Levine, PGAPack V1.0, PgaPack can be obtained via anonymous ftp from: <ftp://ftp.mcs.anl.gov/pub/pgapack/pgapack.tar.z>, 1996.
- [16] W. L. Goffe, G. D. Ferrier, and J. Rogers, J. Econometr. **60**, 65 (1994).
- [17] Z. Q. Li and H. A. Scheraga, Proc. National Academy of Science **84**, 6611 (1987).
- [18] D. J. Wales and J. P. K. Doye, J. Phys. Chem.A **101**, 5111 (1997).
- [19] D. M. Deaven and K. M. Ho, Phys. Rev. Letters **75**, 288 (1995).