Users Manual for **SOCJT 2**

Terrance Codd

December 15, 2013

Contents

0 Preface **3**

1 Introduction **4**
1.1 Program limitations. . . . 4

2 The Jahn-Teller Effect **4**
2.1 Basis set. . . . 5
2.2 Hamiltonian. . . . 5
2.3 PES and definition of customary Jahn-Teller parameters. . . . 5
2.4 The Jahn-Teller quantum numbers $j^{(k)}$ . . . 6
2.5 Eigenvectors. . . . 7
2.6 Ham reduction factor. . . . 7
2.7 Coriolis and spin-rotation parameters. . . . 8

3 Input **8**
3.1 &GENERAL . . . 9
3.2 &MODE_INFO . . . 10
3.3 &SOLVE_INFO . . . 11
3.4 &IO_INFO . . . 13
3.5 &FIT_INFO . . . 13

4 Output **14**
4.1 Output of **SOCJT 2** . . . 14
4.1.1 Preamble . . . 14
4.1.2 Each j block . . . 15
4.1.3 Calculation summary . . . 17
4.2 Fitting output . . . 18
4.3 Auxiliary files . . . 19

5 How to use SOCJT **20**
5.1 Executing the program . . . 20
5.2 Execution times . . . 20
5.3 Recommendations on input parameters . . . 20
5.4 Initial guesses . . . 21
5.5 Degeneracies . . . 21
5.6 Negative eigenvalues . . . 21
5.7 Non-linear least squares fitting . . . 21
5.7.1 The format of $fitfile$ . . . 22

6 SOCRT **22**
6.1 &GENERAL . . . 22
6.2 &MODE_INFO . . . 22
6.3 &SOLVE_INFO . . . 23
6.4 &IO_INFO . . . 23
6.5 &SPECTRA . . . 23
6.6 &UV_INFO . . . 23

7 e2e **23**
7.1 Input . . . 23

8 pi2pi **24**

0 Preface

The program **SOCJT 2 has the following capabilities:**

- **Find the lowest energy solutions to the Jahn-Teller Hamiltonian, including linear and quadratic terms as well as bilinear coupling terms between Jahn-Teller active and inactive modes and cross-quadratic Jahn-Teller coupling between two degenerate modes, and spin-orbit coupling, for any molecule in a non-cubic point group.**
- **Any number of Jahn-Teller active and inactive modes can be accommodated, as well as an arbitrary spin state, whether half-integer or integer.**
- **A non-linear least squares routine is available for the fitting of the parameters to experimental data.**
- **Generate scans of one or more parameters automatically.**

1 Introduction

The Jahn-Teller effect is by now quite old and over a thousand papers have been published on Jahn-Teller related molecules. While many of these papers are on distortions of Jahn-Teller molecules in solid-state crystal structures, an ever increasing number of high resolution, gas-phase measurements of Jahn-Teller active states of molecules are being obtained in the laboratory. The high quality of this data, free of perturbations by extra-molecular interactions, has revealed the true complexity of the Jahn-Teller problem. Of particular importance to our research has been the fact that nearly all Jahn-Teller active states so far observed have been open shell systems, which in principle all have first-order spin-orbit coupling, except for the somewhat rare open shell singlet states. It was the purpose of this research to develop a simple to use, yet fairly complete, set of computer programs for the analysis of the spectroscopic data for Jahn-Teller molecules.

This manual describes the use and background of SOCJT 2 and is a modified version of the manual written for SOCJT by Timothy A. Barckholtz and Terry A. Miller. The program SOCJT 2 (an anacronym for Spin-Orbit Coupling and Jahn-Teller; pronounced "sock-it") diagonalizes the spin-vibronic portion of the Jahn-Teller Hamiltonian, which produces the energies and eigenvectors of the spin-vibronic levels of the state.

This program was written by Terrance Codd between 2011 and 2013 while a graduate student at The Ohio State University, advised by Terry Miller. It was written in C♯ and therefore requires either the .NET framework (Windows) or Mono (OSX and Linux) to run. Most tests were performed using .NET framework 4.0 on two quad-core Xenon processors with 4 GB of RAM running 64-bit Windows 7 Enterprise. Some testing with SOCJT 2 has been done on MNode1, a linux cluster running Red Hat Enterprise 6.1 Linux using Mono 3.2.1.

1.1 Program limitations.

SOCJT 2 , by design, suffers many, though not all, of the limitations of SOCJT which are described below.

- SOCJT 2 cannot handle degeneracies greater than two; i.e., it does not work for the cubic point groups. It also may not work for molecules with a C_4 axis; see the discussion of these point groups in references ? and ?.
- The program can only properly handle one type of Jahn-Teller active mode, with respect to the values of s_2. Examination of Table 3 of reference ? shows that it is possible to have a state with more than one symmetry of quadratic linear Jahn-Teller active modes. In this case, they may have a different value of s_2. However, only one value of s_2 is allowed as input in the program. The ramification of this is that the energies will be calculated accurately, but the Coriolis coupling constants will likely be erroneously computed. This case should be fairly rare, and I have not considered this possibility extensively.
- The program has been extensively test against SOCJT for the C_{3v}/D_{3h} case, and is believed to be accurate for all other non-cubic point groups.

2 The Jahn-Teller Effect

Before describing the input and output of SOCJT 2 , we briefly summarize the Jahn-Teller effect, which serves several purposes. The Jahn-Teller effect is over sixty years old, and numerous different approaches have been taken in the analysis of the vibronic structure of Jahn-Teller active molecules. There thus exists a wide variety of notations and nomenclatures for all of the different parameters and effects involved in the problem. This section outlines our approach and summarizes the most important aspects of the problem. For a full description, we refer the reader to the masterwork on this problem, reference ?. The appendix to this

manual includes a table of all of the different mathematical symbols used in this work, and, where appropriate, the corresponding input parameter to the SOCJT 2 program.

2.1 Basis set.

The basis set used to compute the vibrational structure of Jahn-Teller states is a product of electronic, vibrational, and spin functions. At the symmetric configuration of the nuclei, the electronic wavefunction is degenerate and we use the label $\Lambda = \pm 1$ to distinguish between the two complex components of the ^{2}**E** electronic wavefunction $\mathbf{E}_\pm$. We multiply the electronic basis set by the harmonic oscillator basis functions for the p two-dimensional Jahn-Teller active modes and the $3N - 6 - 2p$ harmonic oscillator basis functions for the non-Jahn-Teller active modes. The vibrational portion of the basis set is characterized by the principal vibrational quantum number v_i and the angular vibrational quantum number l_i for each Jahn-Teller active mode. To include spin-orbit coupling in the calculation, a ket characterized by the projection, Σ, of **S** on the symmetry axis is appended to the basis set.

Only some vibrational modes of a Jahn-Teller active molecule will show the effects of k^{th}-order Jahn-Teller coupling. In the mammoth paper describing this project, we derived a general method for determining which vibrational modes will be Jahn-Teller active. For a given vibrational mode of symmetry e_{s_v} and a given electronic state of symmetry E_{s_e}, a value for s_k of either **0** or **1** can be chosen to satisfy the equation

$$(2s_e + (-1)^{s_k} k s_v) \ \mathbf{mod}\ n = 0, \tag{1}$$

which means the quantity $(2s_e \pm k s_v)$ must be an integer multiple of n, where n is the order of the principal axis of the molecule. The only variable in equation (1) is s_k, and the values of s_k ($k = 1$ and **2**) are required input to **SOCJT** . The reader should refer to Table **2** of reference **?** for guidance in choosing values for s_1 and s_2. See also section **1.1** for some restrictions on these quantities.

2.2 Hamiltonian.

The Hamiltonian for the molecule is the sum of a number of terms,

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_T + \hat{V} + \hat{\mathcal{H}}_{SO}, \tag{2}$$

where $\hat{\mathcal{H}}_T$ is the kinetic energy of the nuclei. The sum of $\hat{\mathcal{H}}_T$ and the potential $\hat{V}$ define the potential energy surface (PES). We refer the reader to our previous papers[?,?] on this topic for the terms of the power series expansion of the potential $\hat{V}$. For our purposes here, it suffices to say that the potential has, in addition to the standard harmonic oscillator terms for the degenerate modes, terms linear and quadratic in the vibrational coordinates of the degenerate modes that account for linear and quadratic Jahn-Teller coupling. The experimentally observed spin-vibronic energy levels of an $\widetilde{X}\,^2$**E** states can be viewed as the eigenvalues of the nuclear motion defined on a PES determined by $\hat{V}$ and $\hat{\mathcal{H}}_{SO}$.

2.3 PES and definition of customary Jahn-Teller parameters.

The customary parameters that are used to characterize the PES are defined in the following way. The spin-orbit Hamiltonian $\hat{\mathcal{H}}_{SO}$ is parameterized by the product $a\zeta_e$, where ζ_e is the projection of the electronic orbital angular momentum on the C_3 axis and a is the spin-orbit coupling constant. The remaining parameters of the PES can most easily be defined by noting that, in the limit $a\zeta_e \to 0$, the form of the potential $\hat{V}$ along the i^{th} normal coordinate is

$$U_{i,\pm} = \tfrac{1}{2}\lambda_i \rho_i^2 \pm \rho_i k_i \left[1 + \tfrac{2 g_{ii}\rho_i}{k_i}\cos 3\phi_i + \tfrac{g_{ii}^2 \rho_i^2}{k_i^2}\right]^{\frac{1}{2}} \tag{3}$$

$$\approx \tfrac{1}{2}\lambda_i \rho_i^2 \pm \left(k_i \rho_i + g_{ii}\rho_i^2 \cos 3\phi_i\right), \tag{4}$$

where in the last equality the expansion of the radical has been truncated at terms quadratic in ρ_i. In these equations, cylindrical coordinates ρ_i and ϕ_i have been used for ith degenerate normal mode that is Jahn-Teller active. The energies of the minima and maxima along this coordinate are given by

$$E_{\min,i} = -\frac{k_i^2}{2\lambda_i(1-K_i)} = -\frac{D_i\omega_{e,i}}{(1-K_i)} \approx -D_i\omega_{e,i}(1+K_i) \tag{5}$$

$$\rho_{\max,i} = \frac{k_i}{\lambda_i(1+K_i)}; \phi_{\max,i} = \frac{\pi}{3}, \pi, \frac{5\pi}{3} \tag{6}$$

$$E_{\max,i} = -\frac{k_i^2}{2\lambda_i(1+K_i)} = -\frac{D_i\omega_{e,i}}{(1+K_i)} \approx -D_i\omega_{e,i}(1-K_i), \tag{7}$$

where D_i is the linear Jahn-Teller coupling constant for the i^{th} mode and K_i is its quadratic Jahn-Teller coupling constant. Both of these coupling constants are dimensionless.

The energies of Eqs. 5 and 7 are relative to the symmetric configuration, which is defined as the zero of energy. The depth, $\epsilon^{(1)}$, of the moat is the linear Jahn-Teller stabilization energy and is a direct measure of the net effect the Jahn-Teller coupling has on the energy of the molecule. From Eqs. 5 and 7, the stabilization energy due to linear Jahn-Teller coupling is obtained by setting $K_i = 0$ and taking the difference, which yields

$$\epsilon_i^{(1)} = D_i\omega_{e,i}. \tag{8}$$

The additional stabilization due to quadratic Jahn-Teller coupling in the mode, $\epsilon_i^{(2)}$, is

$$\epsilon_i^{(2)} = D_i\omega_{e,i}K_i. \tag{9}$$

The barrier to pseudorotation about the moat is then $2D_i\omega_{e,i}K_i$.

A common approximation is that the Jahn-Teller stabilization energy in the state is a sum of the Jahn-Teller effect in the individual modes. Under this assumption, the total Jahn-Teller stabilization energy is a sum over the individual Jahn-Teller stabilization energies,

$$\epsilon_{total} = \sum_i \left[D_i\omega_{e,i}\left(1+K_i\right)\right]. \tag{10}$$

When spin-orbit coupling is non-negligible, the stabilization energy along the i^{th} coordinate is decreased by the presence of spin-orbit coupling to[?]

$$-\epsilon_i^{so} = \begin{cases} -D_i\omega_{e,i} + \frac{(a\zeta_e)^2}{16D_i\omega_{e,i}} & , a\zeta_e < 4D_i\omega_{e,i} \\ 0 & , a\zeta_e \geq 4D_i\omega_{e,i} \end{cases} \tag{11}$$

The total Jahn-Teller stabilization energy is then a sum over all of the individual stabilization energies of Eq. 11.

2.4 The Jahn-Teller quantum numbers $j^{(k)}$

As the Taylor expansion of the Jahn-Teller Hamiltonian is increased, the number of conserved quantum numbers is decreased. Elsewhere[?] we derived the following general form of the "good" Jahn-Teller quantum number, $j^{(k)}$ for the k^{th}-order term of the Jahn-Teller Hamiltonian,

$$j^{(k)} = \frac{1}{k}l_t + \frac{1}{2}(-1)^{s_k}\Lambda, \tag{12}$$

where s_k is chosen according to equation (1). The specific relationships between $j^{(k)}$, l_t, and Λ are given in Table 2 of reference ? for the common point groups. The user's choice of s_1 and s_2 in the input file dictate the definition of $j^{(1)}$ and $j^{(2)}$, according to equation (12). In most circumstances the superscript $^{(k)}$ is dropped, and a value of $k = 1$ is assumed. This is the custom used in the remainder of this manual and in the program itself.

2.5 Eigenvectors.

The general form for the eigenfunctions $|j, n_j, \alpha, \Sigma\rangle$ is

$$|j, n_j, \alpha, \Sigma\rangle = \sum_i \left(c_{i,n_j,\Sigma} |\Lambda_i\rangle \prod_{m=1}^{p} |v_{m,i}, l_{m,i}\rangle |\Sigma_i\rangle \right), \tag{13}$$

where the summation runs over all of the basis functions used in the calculation. Each eigenvector $|j, n_j, \alpha, \Sigma\rangle$ has an associated eigenvalue $E_{j,n_j,\Sigma}$. The notation $|j, n_j, \alpha, \Sigma\rangle$ indicates which j-block the level corresponds to and which eigenvector, n_j, it is from that symmetry block (corresponding to a given j – different j mod 3 when quadratic coupling is included – and Σ combination), with the lowest energy solution of each symmetry block being $n_j = 1$. Because j is not always a good quantum number, we have included into the ket the label α, which is the symmetry species of the state. Note that there are now two subscripts on the quantum numbers ν and l in the basis functions of equation (13). The first, m, corresponds to which vibrational mode the quantum number refers while the second, i, represents the basis function to which the quantum number belongs.

The quantum number Σ is included in the summation only when spin-orbit coupling is included in the Hamiltonian; in its absence, equation (13) becomes

$$|j, n_j, \alpha\rangle |\Sigma\rangle = \sum_i \left(c_{i,n_j} |\Lambda_i\rangle \prod_{m=1}^{p} |v_{m,i}, l_{m,i}\rangle \right) |\Sigma\rangle . \tag{14}$$

In the two limits of small spin-orbit coupling or large Jahn-Teller coupling the spin-vibronic wavefunction is approximately identical to the vibronic wavefunction, i.e., $|j, n_j, \alpha\rangle |\Sigma\rangle \approx |j, n_j, \alpha, \Sigma\rangle$. At these limits, the two components of the spin-orbit doublet have identical vibronic wavefunctions and PES's. Depending on the choice of input parameters (i.e., a zero or non-zero value of $a\zeta_e$), the output of the eigenvectors will be in the form of either equation (13) or (14).

2.6 Ham reduction factor.

An important feature of the SOCJT program is that it has the capability of directly including spin-orbit coupling along with Jahn-Teller coupling in the Hamiltonian. Prior to this program, nearly all Jahn-Teller calculations in the literature have added spin-orbit coupling to the linear Jahn-Teller Hamiltonian "after the fact" via a formula initially derived by Child and Longuet-Higgins.[?] The formula is derived by taking the expectation value of $\hat{\mathcal{H}}_{SO}$ for the vibronic eigenfunction, computed without including $\hat{\mathcal{H}}_{SO}$ in the Hamiltonian:

$$\langle\Sigma| \langle j, n_j, \alpha| \hat{\mathcal{H}}_{SO} |j, n_j, \alpha\rangle |\Sigma\rangle = a\zeta_e d_{j,n_j}\Sigma, \text{ where } d_{j,n_j} = \sum_i \Lambda_i c_{i,n_j}^2. \tag{15}$$

The parameter d_{j,n_j} is often called the Ham reduction factor and is usually not given with subscripts. However, because each vibronic level has a unique value of d, we feel it is appropriate to assign subscripts to this parameter to identify the eigenfunction to which it corresponds. From this formula, the approximate spin-orbit splitting, $\Delta E^{SO}_{j,n_j}$ of the n_j^{th} energy level of the j^{th} symmetry block is

$$\Delta E^{SO}_{j,n_j} = E_{j,n_j,\Sigma=+\frac{1}{2}} - E_{j,n_j,\Sigma=-\frac{1}{2}} = a\zeta_e d_{j,n_j}. \tag{16}$$

The quantity d_{j,n_j} is contained in the output for each eigenvalue and eigenvector computed. These values are printed even when spin-orbit coupling or quadratic Jahn-Teller coupling is included in the calculation. In these cases, the Ham reduction factor may have little relevance to the observed energy levels, and the actual computed energies should be used instead of the energies calculated by equation (16).

2.7 Coriolis and spin-rotation parameters.

One last quantity remains to be defined, and that is the Coriolis coupling constant, which is defined by the Coriolis Hamiltonian,

$$\hat{\mathcal{H}}_{COR} = -2A\left(\hat{L}_z + \hat{G}_z\right)\hat{N}_z. \tag{17}$$

As we discussed in detail elsewhere,[?,?] the Coriolis coupling has a strong dependence on the spin-orbit coupling, and is quite closely related to the spin-rotation parameter ϵ_{aa}. We present only the results of our derivations of the following equations and refer the reader to the original paper[?] for details.

The expectation value ζ_t of $\hat{\mathcal{H}}_{COR}$ over the eigenfunction for a given degenerate vibronic level, in the absence of spin-orbit coupling, is

$$\begin{aligned}
\hat{\mathcal{H}}_{COR} &= -2A\left\langle j, n_j, \alpha = e\right| \hat{L}_z + \hat{G}_z \left|j, n_j, \alpha = e\right\rangle \hat{N}_z & (18)\\
&= -2A\zeta_t \hat{N}_z, & (19)\\
\textbf{where } \zeta_t &= \sum_i \left[c^2_{i,n_j}\left(\Lambda_i\zeta_e + \sum_{m=1}^{p} l_{m,i}\zeta_m\right)\right], & (20)
\end{aligned}$$

where ζ_e is the electronic orbital angular momentum and ζ_i is the Coriolis coupling constant for the i^{th} vibrational mode.

With non-zero spin-orbit coupling, equation (20) becomes

$$\begin{aligned}
\hat{\mathcal{H}}_{COR} &= -2A\left\langle j, n_j, \alpha = e, \Sigma\right| \hat{L}_z + \hat{G}_z \left|j, n_j, \alpha = e, \Sigma\right\rangle \hat{N}_z & (21)\\
&= -2A\zeta_t^{\Sigma} \hat{N}_z, & (22)\\
\textbf{where } \zeta_t^{\Sigma} &= \sum_i \left[c^2_{i,n_j,\Sigma}\left(\Lambda_i\zeta_e + \sum_{m=1}^{p} l_{m,i}\zeta_m\right)\right]. & (23)
\end{aligned}$$

The difference between the Coriolis constant for each spin component, $\Delta\zeta_t^{\pm}$, is

$$\Delta\zeta_t = \zeta_t^{+} - \zeta_t^{-}, \tag{24}$$

where the signs correspond to the sign of $\Sigma = \pm\frac{1}{2}$. The two Coriolis constants will be approximately related by

$$\zeta_t^{\pm} = \zeta_t^{0} \pm \frac{1}{2}\Delta\zeta_t, \tag{25}$$

where ζ_t^0 denotes the Coriolis coupling constant that would result if there were no spin-orbit coupling; i.e., a Coriolis coupling constant corresponding to equation (20). As derived elsewhere,[?] the difference in the two Coriolis coupling constants is related to the spin-rotation interaction by the equality

$$\frac{\epsilon_{aa}^{2v}}{2A} = -\Delta\zeta_t, \tag{26}$$

where A is the rotational constant about the primary axis of the molecule.

For each eigenvalue and eigenvector computed, a value of ζ_t is computed by either equation (20) or (23), which allows the quantities $\Delta\zeta_t$ and then $\frac{\epsilon_{aa}^{2v}}{2A}$ to be computed.

3 Input

The input file is organized into sets of namelist groups, each of which is begun by a &NAMELIST command and ended with a forward slash , as in

```
&GENERAL nmodes = 1
 AZETA = 950.
 MAXJ = 1.5
 TITLE = cdch3
 /
```

Within each &NAMELIST the input variables are assigned their initial values, and separated by returns, spaces or tabs. Each &NAMELIST group is separated from the previous by a '/'. C♯ does not suffer the same weaknesses of FORTRAN for file I/O so the number of spaces and alignment of decimals for the input parameters is unimportant. Where a boolean value is required SOCJT 2 checks for a 'T' or 'TRUE' (in upper or lower case) and if that is not found the default value of FALSE is used. Any input files for SOCJT will work with SOCJT 2 however there are some capabilities in SOCJT not included in SOCJT 2 such as computation of spectra and those sections will be ignored. The order of the variables in the &NAMELIST does not matter, nor does the order of the &NAMELIST groups in the input file itself.

The &NAMELIST groups are separated by their function. There will be one each of &GENERAL, &SOLVE_INFO and &IO_INFO. There will be as many &MODE_INFO groups as there are active modes (specified by the variable *nmodes* in &GENERAL). If any group is not included, a default set of parameters will be used. The &FIT_INFO, &SCAN and &CROSS_TERMS are only necessary to include when needed. Likewise, if more than one &NAMELIST group is included in the input file, only the first will be used.

BEWARE: There is only some error checking in the input file. It is quite possible that the "garbage in garbage out" principle applies. For example, values of S that are not either integer or half-integer may cause the program to crash or give nonsensical results.

3.1 &GENERAL

The &GENERAL namelist group sets the general parameters for the calculation. The default values are

```
&GENERAL
 NMODES = 1
 TITLE = 'TITLE'
 S = 0.5
 AZETA = 0.0
 FIT_AZETA = FALSE
 ZETAE = 1.0
 MAXJ = 7.5
 CALC_DERIV = FALSE
 USE_KAPPA_ETA = FALSE
 S1 = 0
 S2 = 1
 /
```

NMODES Number of Jahn-Teller active modes in the state. Integer.

TITLE Title for the calculation, up to 80 characters long. Does not need to be in quotes (they will be considered part of the name).

S Value of the spin angular momentum for the state, must be zero, half-integer, or integer, although there is no check on this. Real.

AZETA The value of $a\zeta_e$ in a spin-orbit state. If $a\zeta_e$ is zero, then the value of S does not matter. Real.

FIT_AZETA Whether or not to fit $a\zeta_e$ in the fitting routines. Boolean.

ZETAE The value of ζ_e for the state, used only in the calculation of the Coriolis coupling constant. Real.

MAXJ The maximum value of j to be used in the calculation. For linear Jahn-Teller coupling only, this is the maximum value of j for which the diagonalizations will be performed. For quadratic Jahn-Teller coupling calculations, this is the maximum value of j to be included in the matrices. For quadratic Jahn-Teller coupling, maxj should be at least 7.5. This value must be half-integer; if something else is used, I believe the effective value becomes the closest half-integer value, rounding down. Real.

CALC_DERIV Boolean variable that determines whether the derivatives of the energies with respect to the various parameters (vibrational frequencies, coupling constants, etc.) should be calculated. The calculation of the derivatives of the eigenvalues with respect to the parameters has probably not been adequately tested, and no guarantee is made that they are accurate. Boolean.

USE_KAPPA_ETA Boolean variable that is true if κ (KAPPA) and η (ETA) will be used for the linear and quadratic Jahn-Teller coupling terms instead of D and K. This is false by default and may be left out of the input file if D and K are to be used. There is a simple relationship between *kappa* and *eta* and D and K which was derived by John Stanton. Within SOCJT 2 these are converted to D and K. This is purely a matter of user convenience when *kappa* and *eta* would be easier to use. Boolean.

s1 Value of s_1 according to equation (1). This choice has no bearing on the calculated energies of the spin-vibronic levels, only the relative phases of the electronic and vibrational contributions to the Coriolis coupling constant of each level. While the value should be either 0 or 1, any even or odd number will work as if they were 0 or 1, respectively. Integer.

s2 Value of s_2 according to equation (1). See above, and section 1.1 for a discussion of the limitations on this parameter.

3.2 &MODE_INFO

For each vibrational mode, a &MODE_INFO must exist, otherwise default parameters are used that will cause spurious results for the user. There must therefore be NMODES &MODE_INFO namelist groups in the input file. The default values are

```
&MODE_INFO
 MODEVMAX = 0
 MODEOMEGA = 0.0
 MODEWEXE = 0.0
 MODED = 0.0
 MODEK = 0.0
 FIT_OMEGA = FALSE
 FIT_D = FALSE
 FIT_K = FALSE
 FIT_WEXE = FALSE
 ISATYPE = FALSE
 MODEA_OMEGA = 0.0
 MODEZETA = 0.0
 /
```

If using KAPPA and ETA then replace MODED, MODEK, FIT_D, and FIT_K by the following.

```
KAPPA = 0.0
ETA = 0.0
FIT_KAPPA = FALSE
FIT_ETA = FALSE
```

MODEVMAX Maximum value of v for this vibrational mode. In general, the lower the vibrational frequency and the larger the Jahn-Teller coupling constants, the larger this number needs to be. In practice, a value never less than 2 is useful, even for the modes with very large frequencies and low coupling constants (such as C-H stretches often are). Integer.

MODEOMEGA Vibrational frequency, in cm^{-1}, for the mode. Real.

MODEWEXE Anharmonicity for the mode, in cm^{-1}. Real.

MODED Linear Jahn-Teller coupling constant for the mode, as defined in equations (5) to (7). This value is usually considered to be positive. I have not tested the effect of a negative value of *moded* in the program. Dimensionless, real.

MODEK Quadratic Jahn-Teller coupling constant for the mode, as defined in equations (5) to (7). While D_i is positive, K_i can be either positive or negative. The choice has no effect on the calcualted energy levels, but does swap the symmetries of the $j = \frac{3}{2}$ levels split by the quadratic interaction. These symmetries are sometimes not able to be determined experimentally, in which case the sign of K_i has no meaning. Dimensionless, real.

FIT_OMEGA,FIT_D,FIT_K,FIT_WEXE,FIT_KAPPA,FIT_ETA Whether to fit these parameters in the fitting routine. All boolean.

ISATYPE Boolean value to indicate if this mode is a degenerate (default) or nondegenerate (A type). If TRUE, then the mode is treated a nondegenerate for the purposes of building the basis set and any values entered for D or K (ETA or KAPPA) are ignored. Boolean.

MODEA_OMEGA The value of the vibrational frequency of this mode in the non-degenerate state to or from which electronic transitions occur. This parameter is used only in the calculation of the relative intensities of the vibrational progressions in the electronic spectra, and in the files generated to produce simulations of these spectra. Real, cm^{-1}.

MODEZETA The Coriolis coupling constant for this vibrational mode, used only in the calculation of the Coriolis coupling constant for the (spin-)vibronic level. Dimensionless, real.

3.3 &SOLVE_INFO

The &SOLVE_INFO namelist group contains parameters used in the diagonalization routines. In general, the default values should be adequate for the block Lanczos routine, except for M, which is the number of eigenvalues desired, which the user must choose.

There is the option of using either a block Lanczos routine (Underwood method) or a simple Lanczos routine. The block routine uses full reorthogonalization and generates the eigenvectors during the routine automatically. The benefits of this routine are that it is guaranteed to find all of the eigenvalues with no spurious values and if the eigenvectors are requested it does not take any extra time to do so. The drawback of this routine is that it may take up to 100x longer than the simple Lanczos routine. For the block routine K_FACTOR is the size of the blocks used, NOITS is the maximum number of iterations that will run and TOL is the tolerance used to define when an eigenpair has converged. When using the block routine, the runtime may be dependent upon K_FACTOR and while I have found 2 to be adequate in most cases a larger value may run faster. For a larger K_FACTOR less iterations will be

required but they will take longer. The routine completes when either NOITS is exceeded or all eigenvalues requested converge to the specified tolerance.

For the simple Lanczos routine K_FACTOR is not used and the other parameters described above have slightly different uses. NOITS is the number of Lanczos iterations run, and therefore, the size of the Lanczos matrix generated. This routine does not use any orthogonalization so some eigenvalues will be repeats and others will be spurious values which do not correspond to actual eigenvalues of the Hamiltonian. These duplicate and spurious values are removed after diagonalization by a test. For this reason the M used should be larger than the number of eigenvalues actually desired. How much larger depends on the number of iterations as more spurious and duplicate eigenvalues will be generated as more iterations are run. TOL for the simple Lanczos is used in the test to remove duplicate and spurious eigenvalues. Any eigenvalue within TOL of another is removed as a duplicate and this is the same test used to remove spurious values. A value of 10^{-6} works for most cases but it may be necessary to use a smaller value.

Defaults are

```
&SOLVE_INFO
 M = 10
 K_FACTOR = 2
 NOITS = 10000
 TOL = 0.0001
 PARVEC = 1
 PARMAT = 1
 PARJ = 2
 BLOCK_LANCZOS = F
 NEW_RANDOM = F
 /
```

The Following is for the Block Lanczos routine.

M Number of eigenvalues to find in each block. Integer.

K_FACTOR The block Lanczos routine uses blocks with K_FACTOR columns. The larger K_FACTOR is the fewer iterations the routine will take to converge but the longer each iteration will take to run. Not used in the simple Lanczos routine. Integer.

NOITS Maximum number of iterations the eigenvalue routines will perform for each block. If the calculation stops because it has reached *noits*, the output will include the eigenvalues and eigenvectors found to that point. This can then serve as a guide for how many iterations might be necessary. This is the number of Lanczos iterations in the simple Lanczos routine. Integer.

TOL This is the relative tolerance to which the eigenvalues are found, such that $\frac{\Delta\epsilon}{\epsilon}$ <TOL. A value of TOL less than 0.0001 is highly recommended, as this will, in general, result in energies calculated to within 1 cm^{-1}. For the simple Lanczos a value not larger than $10^{(}-6)$ should be used. Real.

PARVEC This is the degree of parallelization to be used in the matrix-vector multiplactions in the Lanczos routines. In practice a value of 2 is only useful for basis sets larger than a few hundered thousand basis functions. Otherwise it will run fastest with the default value of 1. Integer.

PARMAT This is the degree of parallelization to be used in the matrix generation routine for each j block. In practice, as long as there are, a sufficient number of processors, the matrix generation speed will increase almost linearly with the value of PARMAT. Integer.

PARJ This is the degree of parallelization to be used for the different j blocks. Because each j block will call a matrix generation routine and a diagonalization routine the total number of processors used will be PARJ times PARVEC/PARMAT. Integer.

BLOCK_LANCZOS Boolean value to indicate if the block Lanczos routine should be used. Boolean.

NEW_RANDOM Boolean value to indicate if the random number generator from SOCJT should be used or the .NET built in random number generator should be used. Typically fewer iterations are used in the block routine with the new random routine. Boolean.

3.4 &IO_INFO

The &IO_INFO namelist group contains Boolean variables that indicate whether a particular piece of output is desired as well as how the eigenvectors should be printed. Defaults are

```
&IO_INFO
 PRINT_BASIS = FALSE
 PRINT_VEC = FALSE
 PRINT_MATRIX = FALSE
 EV_MIN = 0.2
 /
```

PRINT_BASIS Whether or not to include in the output the basis set for each block. Boolean.

PRINT_VEC Whether or not to include in the output the eigenvectors for each block. The eigenvalues, Ham reduction factors, and Coriolis constants are part of the output whether PRINT_VEC is true or false. Boolean.

PRINT_MATRIX Whether or not to include in the output the Hamiltonian matrix for each block. Boolean.

EV_MIN The minimum value of the weight for a basis function to be printed in the eigenvectors. Double.

3.5 &FIT_INFO

The &FIT_INFO namelist group controls the fitting portion of SOCJT 2 . See section 5.7 for more information. The default values are

```
&FIT_INFO
 FITFILE = fit.fit
 FTOL = 0.0
 XTOL = 0.0
 GTOL = 0.0
 MAXFEV = 25
 FACTOR = 0.001
/
```

FITFILE The name of the file that contains the energy levels to be fit and their assignments. See section 5.7.1 for the formatting of this file. String.

FTOL, XTOL, and GTOL To quote from the lmder.f literature regarding these three parameters:

```
  ftol is a nonnegative input variable. Termination occurs when both
the actual and predicted relative reductions in the sum of squares are
at most ftol. Therefore, ftol measures the relative error desired in
the sum of squares.

  xtol is a nonnegative input variable. Termination occurs when the
relative error between two consecutive iterates is at most
xtol. Therefore, xtol measures the relative error desired in the
approximate solution.

  gtol is a nonnegative input variable. Termination occurs when the
cosine of the angle between fvec and any column of the jacobian is at
most gtol in absolute value. Therefore, gtol measures the
orthogonality desired between the function vector and the columns of
the jacobian.
```

The smaller these three numbers are, the tighter the parameters will be converged. See section 5.7 for more information regarding these parameters. All real.

MAXFEV Maximum number of iterations for the fitting procedure. Integer.

FACTOR Essentially the step size the fitting procedure takes between successive iterations. A value of FACTOR greater than 1 or so will almost invariably cause wild and useless results. A small step size will take longer to converge, and may converge to a solution that is a local minimum. Real.

4 Output

4.1 Output of SOCJT 2

The description of the output file uses as its example the output of the `test1` input file contained in the file distribution. Hopefully the examples and accompanying descriptions are sufficient to explain the output. The name of the output file is `filename.out`, where `filename` is the name of the input file.

4.1.1 Preamble

The output of the main program SOCJT begins with the time the calculation began,

```
 It is 171236.280 on 19990805
```

The time is given in the form `hhmmss.###` to signify the hour, minute, seconds, and fraction of a second. The date is given in the form `yyyymmdd` for the year, month, and day.

Following a header, the &GENERAL and &MODE_INFO namelist groups from the input file are given. Some additional information regarding the input parameters and PES are tabulated:

```
 A*zeta_e =      -25.347877587660860
 Parameters for each Jahn-Teller active mode:
 Mode #  V(min)   V(max)     Omega(E)    wexe    D       K        JTSE  Omega(A)

  1        0        5         573.00      .00    .023    .010    13.18    585.00
  2        0       10         263.00      .00    .435    .000   114.40    239.00

 Total linear Jahn-Teller Stabilization energy =   127.58
```

```
 Total quadratic Jahn-Teller Stabilization energy =      .13
 Total linear JTSE including spin-orbit coupling =   108.94
```

The numbering of the modes in the table are based on the order of the &MODE_INFO namelist groups in the input file. The numbering given in the table will be the numbering of the vibrational modes for the rest of the output file.

The column of the table entitled `JTSE` is the Jahn-Teller stabilization energy for the mode; i.e., it is the energy of equation (8), and does not include the effects of spin-orbit or quadratic Jahn-Teller coupling.

The next few lines regurgitate the remaining input namelist groups, &IO_INFO, &SOLVE_INFO, &SPECTRA, and all of the &UV_INFO groups.

Lastly, a few sentences summarize the type of calculation being performed,

```
Quadratic JT Program
Spin-orbit being added
```

4.1.2 Each j block

At this point, the output begins to cycle through all of the blocks of the Hamiltonian, beginning with $j = \frac{1}{2}$ and working up to $maxj$. Each block is begun by a header, such as

```
 ********************
 ********************
 **                  **
 **     J =  .5      **
 **  Sigma =  -.5    **
 **                  **
 ********************
 ********************
```

followed by the number of basis functions in this block.

Basis set **The next section of the output is the listing of each basis function included in this block of the Hamiltonian, and is only printed if the value of $print\_basis$ has been set to `.true.`. The output is of the form**

```
Basis Fxn #  v(1)  l(1)  v(2)  l(2)  Lam  2*Sig

     1         0     0     1     1     1    -1
     2         0     0     3     1     1    -1
     3         0     0     5     1     1    -1
     4         0     0     7     1     1    -1
```

and continues through all of the basis functions. The columns give the principal quantum number v_i and the vibrational angular quantum number l_i. The column "`Lam`" is the value of Λ for the basis set, and the column "`2*Sig`" is twice the value of Σ. (I realize this is a funny way to do it, but it ensures that the output is uniform for both integer and half-integer spin states.)

Hamiltonian matrix **After the basis set, a line is given that indicates how many non-zero matrix elements there are in this block of the Hamiltonian. This number is actually the number of non-zero matrix elements in the lower half of the matrix. If the value of $print\_matrix$ has been set to `.true.` then all of the non-zero matrix elements will be printed out,**

```
  Col #  Row #    Value     Col #  Row #    Value     Col #  Row #    Value
      1      1    1049.00
```

```
     2      2    1575.00
  .
  .
  .
     1     99     245.31     11     99     122.89     99     99     886.00
     1    100     245.31      2    100     346.92     12    100     122.89
     2    101     346.92      3    101     424.89     13    101     122.89
     3    102     424.89      4    102     490.62     14    102     122.89
```

The column and row numbers correspond to the indices of the matrix, which are the basis functions listed previously in the output. The values are listed in units of cm^{-1}.

Matrix diagonalization information **If the variable *print_monit* was set to `.true.` in the input file, a lot of extra lines are added to the output. This information is included only for the purposes of diagnostics in the programming, and *print_monit* should generally be set to `.false` in the input file.**

Possible diagonalization errors **If the diagonalization was not successful, an error message will be given indicating what the problem was. For the most part, the only error message that will be encountered in normal practice in the message indicating the maximum number of iterations was exceeded. In this case, the calculation should be resubmitted with a larger value of *noits*. It may or may not help to include the eigenvalues and eigenvectors that were found as initial guesses. Even if an error was encountered, the eigenvalues and eigenvectors for the solutions that were found will be listed.**

Eigenvalues and eigenvectors **Hopefully, though, the next lines in the output will be**

```
 Underwood method completed successfully.
 Final results for j =  .5 and Sigma =  -.5
```

("Underwood" is the type of diagonalization routine used.) The next section of the output lists the eigenvalues that were found as solutions,

```
Found 11 eigenvalues
--------------------
     649.146200
     979.415800
    1165.737000
    1253.750000
    1386.468000
    1487.378000
    1526.489000
    1745.356000
    1778.216000
    1792.343000
    1821.780000
```

These eigenvalues are in units of cm^{-1}, and are relative to the undistorted point of the PES being defined as the zero of energy. The final portion of the output file is a list of all of the eigenvalues found, with the zero of energy being set to the energy of the lowest energy (spin-)vibronic solution.

If the value of *print_vec* was set to `.true.` in the input file, then a considerable amount of additional information about each solution is included in the output:

```
 Eigenvalue      1 =     649.15
 Spin-orbit quenching parameter d =  -.1669613
```

```
zeta_t = -.167* zeta_e  +   .016* zeta_ 1 +   .401* zeta_ 2 =   .215

Eigenvector: (Only vectors with coefficient > 0.025 are shown)

    Coefficient  v(1)  l(1)  v(2)  l(2)   Lam  2*Sig
      .6185285     0     0     1     1     1    -1
      .1097540     0     0     3     1     1    -1
      .0255114     1    -1     2     2     1    -1
      .1402043     1     1     0     0     1    -1
      .0363902     1     1     2     0     1    -1
     -.7133587     0     0     0     0    -1    -1
     -.2546745     0     0     2     0    -1    -1
     -.0334868     0     0     4     0    -1    -1
     -.0801715     1    -1     1     1    -1    -1
     -.0411716     1     1     1    -1    -1    -1
```

The Ham reduction parameter d is computed for each level, assuming that ζ_e is unity. The next line computes a value for the Coriolis constant, ζ_t, for the level, both as a formula of its components and as a numerical value, based on the input values for ζ_e and the ζ_i.

The eigenvector is given in tabular form, listing only those basis functions for which the coefficient is greater than $\pm$0.025. The columns are listed just as the basis functions are listed previously in the output file.

Electronic spectra **After the eigenvectors are listed, if desired, the process repeats itself with the next j block. However, if all of the necessary calculations have been performed such that a requested electronic spectrum can now be computed, the data for that UV-Vis spectrum is included in the output at this point. For the electronic transition intensities, the output is of the form**

```
UV-Vis spectrum #           1

 (only lines that are > 1% of intensity of most intense transition
  are listed)

                            E state        A state vib qns
 Energy     Rel. Int.   j    Sigma    p      v(1)   v(2)
---------  ----------- ---  -------  ---     ----   ----
 36063.00        .27    .5    -.5    1        1      2
 35956.00        .23    .5    -.5    1        0      4
 35824.00       1.29    .5    -.5    1        1      1
 35824.00        .34    .5    -.5    1        1      1
 35717.00       2.42    .5    -.5    1        0      3
 35585.00       3.93    .5    -.5    1        1      0
 35478.00      12.92    .5    -.5    1        0      2
 35239.00      75.69    .5    -.5    1        0      1
 35000.00     100.00    .5    -.5    1        0      0
```

This output is self-explanatory in nature. The transition energies are calculated using the values of $modea\_omega$ in the input file and the computed (spin-)vibronic energies.

4.1.3 Calculation summary

After the last j block has been computed, the results of the calculation are summarized.

```
 Final results for all eigenvalues found.

 Eigenvalue    j   Sigma   n_j    zeta_t
        .00    .5   -.5     1      .1201
       4.44    .5    .5     1      .1392
     153.56   1.5   -.5     1      .1900 2.-fold degenerate
     195.84   1.5   -.5     2     -.2301 2.-fold degenerate
.
.
.
     842.58    .5   -.5    10     1.3277
     844.91    .5    .5    10     1.3497
 Program SOCJT exited properly.
 It is 171246.500on 19990805
```

The comments "*2.-fold degenerate*" serves simply as a reminder that the two spin-components of the $j = \frac{3}{2}$ levels are not split by spin-orbit coupling, but are mixed with each other. The last line is simply the time and date the calculation finished.

4.2 Fitting output

When the non-linear least squares routines are used, SOCJT is executed once for each iteration of the fitting routine. At the end of the fitting, output similar to the following is generated:

```
 Done with fitting routine.
 Energies were calculated   4 times.
 Derivatives were calculated   3 times.

 Fitting ended properly.
 Both actual and predicted relative reductions
   in the sum of squares are at most ftol.

 Final calculated parameters.
 ---------------------------
 A*zeta_e =      -25.347877587660860
 Parameters for each Jahn-Teller active mode:
 Mode #  V(min)   V(max)     Omega(E)    wexe    D       K        JTSE  Omega(A)

  1        0        5         603.32      .00    .378    .100   228.25    585.00
  2        0       10         269.96      .00    .439    .214   118.46    239.00

 Total linear Jahn-Teller Stabilization energy =   346.71
 Total quadratic Jahn-Teller Stabilization energy =    48.18
 Total linear JTSE including spin-orbit coupling =   346.20


 Final sum of squares of Delta E =     155.31
 Final rms error =  3.76

  Fit E     j  n_j Sig    Calc E       Err
    7.96    .5  1   .5      4.44      -3.52
  155.30   1.5  1  -.5    153.56      -1.74
  192.68   1.5  2  -.5    195.84       3.16
  298.22    .5  2  -.5    290.56      -7.66
  284.27    .5  2   .5    291.92       7.65
```

```
 376.90    .5  3   .5    374.50      -2.40
 379.14    .5  3  -.5    378.45       -.69
 439.50   1.5  3  -.5    441.05       1.55
 481.72   1.5  4  -.5    480.23      -1.49
 496.59    .5  4   .5    497.88       1.29
 499.06    .5  4  -.5    499.46        .40
```

Following this table, all of the namelist groups are printed with the final fit values, so that these lines can be copied into an input file and the program started from these values.

4.3 Auxiliary files

There are several additional files that may be generated by SOCJT : `filename.uv.#.txt`, `filename.basis.`$j.\Sigma$, **and** `filename.vecs.`$j\Sigma$. **The filenames are constructed by appending brief notations to the** `filename` **of the input file. For the electronic transition intensites, the files are numbered according to the &UV_INFO namelist groups. The basis set and vector files, used in subsequent calculations by SOCJT (as guesses to the diagonalization routines) or by e2e have a slightly more complicated format. Following the appendation of** `.basis.` **or** `.vecs.`**, the value of** j **is appended. If spin-orbit coupling was included in the calculation, a "**`m`**" or "**`p`**" is appended to indicate the sign of the value of** Σ**, which follows, for the basis set or eigenvectors that are being written. While cumbersome, these filenames allow for all of the required information to be stored in a useable format.**

Electron transition simulations For each electronic transition calculated, a file an `(x,y)` **file is generated that simulates the electronic transition to facilitate comparisons with experimental data.**

```
 36073.00      .01675822
 36072.00      .02837989
 .
 .
 .
 35002.00   89.50251000
 35001.00   97.26550000
 35000.00  100.00000000
 34999.00   97.26550000
 34998.00   89.50251000
 34997.00   77.91646000
 34996.00   64.17130000
 34995.00   50.00000000
 .
 .
 .
Parameters
azeta =      100.000000
omega(          1) =      573.000000
D(          1) =    2.300000E-02
K(          1) =    1.000000E-02
A_omega(          1) =      585.000000
omega(          2) =      263.000000
D(          2) =    4.350000E-01
K(          2) =    0.000000E+00
A_omega(          2) =      239.000000
```

The simulation assumes a Gaussian line profile, with a full width at half maximum of $fwhm$ **from the input file. To keep the file length manageable, when the intensity drops below 0.01**

(the maximum is set to 100.0), the data point is deleted from the file. The file concludes with a summary of most of the relevant input parameters to the Jahn-Teller calculation.

Basis set and vector files If the formatting of these files is desired, please contact the authors directly. These files are created so that the initial guess functionality of SOCJT works and so that the program e2e has all of the information it needs to calculate transition intensities between two degenerate states.

5 How to use SOCJT

5.1 Executing the program

To execute SOCJT , you can either open it up from the Windows Explorer, create a shortcut to it for the Start menu, or go to an MS-DOS prompt, change to the directory `socjt.exe` is in, and type `socjt` at the prompt. The program immediately prompts the user with the following question,

```
What is the input filename?
```

Only the first 25 characters are actually used. The filename should be the name of the input file. The names of all the other files generated by SOCJT , such as `filename.out` for the output file. Thus, it is probably not useful to name the input file with a `.inp` appended to it, as the other files will include the "`.inp`" as part of their file names. All of the output and auxiliary files that are generated are plain text files and can be viewed with any standard text editor. There are no binary files associated with SOCJT .

5.2 Execution times

The time required for execution scales exponentially with the size of the basis set. It is recommended that the user perform preliminary calculations with smaller basis sets, and then work up to bigger basis sets. Elsewhere,[?] we have tabulated the time required for a variety of calculations using a 200 MHz Cyrix-686. As an example, a four-mode linear Jahn-Teller calculation with values of $mode\_vmax$ equal to 7,5,3, and 2 took approximately 70 minutes. So far, I have experienced no problems with the program running out of memory, using machines with only 32 MB of RAM.

5.3 Recommendations on input parameters

There are a few general recommendations regarding the choice of basis set. First, in general, the larger the value of either D_i or K_i, the larger that mode's value of $mode\_vmax$ should be. This result is fairly obvious. Second, the smaller the frequency $\omega_{e,i}$ is, the larger the basis set should be. This result is not directly obvious, but is derived from the experimental realization that most information is available for the lower energy spin-vibronic levels, and these levels will be dominated by the vibrational modes with low frequencies.

Lastly, and most importantly, it is important to recognize that the choice of a finite basis set is a potentially severe approximation. Thus, it is very important that a series of calculations needs to be performed, with increasing size of the basis set. The calculations should have converged to the desired accuracy, although in practice a few cm^{-1} is as close as they will converge. For fitting spectra, most calculations can be run with only moderately sized basis sets. As the improvement of the fit to the experimental data improves, the basis set can be slowly increased.

5.4 Initial guesses

The initial guess functionality is intended to improve the convergence speed from one iteration to the next. In the first calculation, the variables *vec_file* and *basis_file* in the &IO_INFO namelist group must both be set to `.true.`. This generates the necessary files to jump start the next calculation. In the next calculation, the variable *guesses* in the &SOLVE_INFO group is set to `.true.`. As indicated in section 3, this calculation only works if the basis sets from the two calculations are identical. In other words, the exactly same values of *mode_vmax* must be used. Also, the calculations must be the same with respect to the terms in the Hamiltonian. For example, one calculation cannot have spin-orbit coupling, while the other does. While the initial guess does work, it is often no faster than starting from scratch. This feature has not been fully tested, and is not guaranteed to work without error.

5.5 Degeneracies

One subtlety that is often not appreciated is that every single level is at least doubly degenerate. However, we only calculate one spin component for each energy. To make things simple, we have calculated only those levels with positive j. There is of course a degenerate component, but we neglect it. See equation 76 of reference ?.

5.6 Negative eigenvalues

Because the eigenvalue-finding routines are specialized for a type of matrix known as sparse, Hermitian, positive-definite, trouble can result if an eigenvalue becomes negative. How can an eigenvalue be negative? Remember that the zero of energy is the "undistorted" point of the PES, and that in the absence of Jahn-Teller coupling the value of the lowest energy eigenvalue is the sum of the vibrational frequencies of the mode. (Because of the degeneracies of the modes, it is the sum of the frequencies rather than the sum of one-half of the frequencies. The frequencies of only the Jahn-Teller active modes are used in the calculation.) Jahn-Teller coupling decreases the energy of the lowest level, and if the coupling constant(s) are large enough, than in the definition of zero energy given above, the energy of this level can actually become negative. In this case the program will crash. The solution is to add an extra Jahn-Teller active vibrational mode to the problem that has a very large vibrational frequency (10 000 cm^{-1} usually is more than sufficient) with zero Jahn-Teller coupling constants. The value of *modevmax* for this mode should be 1. This has the effect of increasing the diagonal Hamiltonian matrix elements by 10 000 cm^{-1}, which is equivalent to redefining the zero of energy to be 10 000 cm^{-1}. In only the rarest cases will the energy of the lowest eigenvalue fall negative. In these cases, a few test calculations should be run to ensure that the added, spurious vibrational mode has no effect on the actually calculated vibrational spacings. This appears to be a problem only for the Cray version of the program; the DOS version has no problem with negative eigenvalues.

5.7 Non-linear least squares fitting

If any of the parameters fit_azeta, fit_omega, fit_d, fit_k, or fit_wexe are true, then the fitting routines will be executed. If they are all false, then the eigenvalues will be computed only once, and the program will be terminated. The fitting routines invoke a non-linear least squares algorithm (written by so-and-so, downloaded from somewhere) to optimize the input parameters. The fitting procedures use as their starting point the values of $a\zeta_e$, $\omega_{e,i}$, and so on that are given in the input file.

It is important to note that for most realistic cases, there is probably not one unique "best fit" of the experimental data. The multimode Jahn-Teller problem is extremely complicated, and it is quite likely that a number of "local minima" exist that can fit the data reasonably well. Therefore, the choice of input parameters will strongly influence the fitting results. We recommend the use of *ab initio* calculations to guide the choice of vibrational frequencies, linear

Jahn-Teller coupling constants, and spin-orbit coupling constant. See reference ? for more details of *ab initio* calculations of the Jahn-Teller surface.

5.7.1 The format of $fitfile$

An additional file, named $fitfile$, is required. It contains a list of the eigenvalues and quantum number assignments of the energy levels to be fit. These energies are all relative to the "vibrationless" energy level being defined as the zero of energy. This file, a fixed format file, is of the format

```
11
␣␣␣␣7.96␣␣␣␣0.5␣␣1␣␣0.5
␣␣155.30␣␣␣␣1.5␣␣1␣-0.5
.
.
.
␣␣496.59␣␣␣␣0.5␣␣4␣␣0.5
␣␣499.06␣␣␣␣0.5␣␣4␣-0.5
```

The first line of the file contains the number of energy levels to be fit, in this case 11. There are then that many lines following with the energy, j value, n_j, and (if $a\zeta_e$ is non-zero) the value of Σ. Each line is of the format

```
format(f8.2,2x,f4.1,1x,i2,1x,f4.1)
```

for $a\zeta_e$ non-zero, or

```
format(f8.2,2x,f4.1,1x,i2)
```

for $a\zeta_e$ equal to zero. The three "quantum numbers" j, n_j, and Σ define each unique energy level, and can be found at the end of the output of a SOCJT run.

6 SOCRT

The program SOCRT is actually only a slight modified version of SOCJT . As discussed in section 1, the Renner-Teller effect for Π states of polyatomic linear molecules can be formulated using the Jahn-Teller Hamiltonian, but with a zero (by symmetry) linear Jahn-Teller effect. Because of the higher symmetry of the linear molecule's PES, a greater number of quantum numbers are conserved. Combined with the slightly different nomenclature used for the Renner-Teller problem, these changes necessitated a unique program for Renner-Teller molecules.

This section discusses the few changes that have been made to the input file for the program SOCRT as compared to that for SOCJT . The output has only cosmetic changes associated with it, and will not be discussed futher. The fitting routines are not available with SOCRT.

6.1 &GENERAL

In the &GENERAL namelist group, the variables $zetae, s1,$ and $s2$ have been deleted, while the variable $maxj$ has been changed to $maxk$, since K is the name given to the conserved vibronic quantum number in Renner-Teller systems. It is an integer, and indicates up to what value of K the blocks should be calculated.

6.2 &MODE_INFO

The Renner-Teller analogue of the Jahn-Teller parameter K_i is ϵ_i. For Renner-Teller molecules, there is no equivalent of the linear Jahn-Teller effect, and so D_i is non existent. Thus, the variables $moded$ and $modek$ have been eliminated from the &MODE_INFO namelist group. In their place the variable $mode\_epsilon$ as been added, which is a real variable with units of cm^{-1}.

6.3 &SOLVE_INFO

The variable *guesses* has been deleted from SOCRT. The initial guess functionality is not available in SOCRT. The Renner-Teller calculations are in general very quick, and this should not cause a major inconvenience to most users.

6.4 &IO_INFO

There have been no changes made to this namelist group.

6.5 &SPECTRA

There have been no changes made to this namelist group.

6.6 &UV_INFO

Only the name of the variable e_j has been changed to e_k, to reflect the change in the nomenclature of the conserved vibronic quantum number.

7 e2e

The program e2e was written to take the output of two Jahn-Teller calculations and calculate the intensities of an electronic transition between the two states. As such, it requires that these two calculations have been successfully completed, with the variables *vec_file* and *basis_file* having been set to `.true.`. All of the output files, along with the e2e executable file, must be in the same directory. As with SOCJT and SOCRT, e2e is executed by simply double clicking the file `e2e.exe` from the explorer or executing it from a DOS prompt.

The two JT calculations have to have the same number of active modes, although it is not necessary for them to have had the same values of *modevmax*. Diagonal Franck-Condon factors are assumed (which is an improvement to be left to a future version). The order of the modes is the same between the two calculations - i.e., the mode listed first in the state 1 SOCJT calculation is the same as the mode listed first in the state 2 SOCJT calculation. Both calculations must be the same with respect to spin-orbit coupling, and this must be properly listed in the input file to e2e. If it is desired for one state to have zero spin-orbit coupling, a very small (ca. 0.1 cm^{-1}) value should be used for *azetae*.

Two reasonable assumptions are made in the calculation, that the selection rules on j and Σ are $\Delta j = 0$ and $\Delta\Sigma = 0$.

This program has not been fully tested, and the program is not foolproof with respect to the input - i.e., garbage-in-garbage-out may apply.

There is an interesting application of the program e2e, unrelated to the calculation of transition intensities. The program e2e can be used to test whether or not a calculation has converged with respect to the size of the basis set. In testing convergence, the only difference between the two calculations is the size of the basis set (i.e., the values of *modevmax*). In this case, a calculation of the transition intensities from one state to another gives an indication of how closely the calculation is converged. The closer the convergence, the closer the intensity of the zero-zero band will be to unity, and the lower the intensities of all other transitions.

7.1 Input

The input file is old-style; i.e., it's not namelist oriented like the input file for SOCJT and SOCRT. The input file begins with the following four lines (type of variable is given in parentheses):

```
filename1␣(character(15))
filename2␣(character(15))
nspectra␣(i2)
spin_orbit␣(logical)
```

The first two lines contain the filenames of the two SOCJT calculations, followed by the number of spectra to be calculated. The last line is a logical variable to indicate whether or not spin-orbit coupling was used in the SOCJT calculations.

After these four lines, there follows a line for each spectrum to be calculated. The input differs slightly depending on whether spin-orbit coupling was included in the calculation. If *spin_orbit* **is true, then these lines take the form**

```
j␣sigma␣nj1␣ttype␣origin␣step_size␣fwhm
```

and if *spin_orbit* **was set to** `.false.`**, then it has the form**

```
j␣nj1␣ttype␣origin␣step_size␣fwhm
```

The variables are

j **The value of** j **for the two vibronic levels involved in the transition. Real.**

$sigma$ **The value of** Σ **for the two vibronic levels involved in the transition. Real.**

$nj1$ **Which eigenvector of state 1 the transition originates from (**$nj1 = 1$ **for lowest energy eigenvector for the state). Integer.**

The variables $ttype, origin, step\_size,$ **and** $fwhm$ **are defined as they are for SOCJT .**

8 pi2pi

This program has not yet been written, but it will come quickly. It isn't that much different than e2e. If you require it, send email to the author (timothy.barckholtz@colorado.edu).