

ROTATIONAL EQUILIBRIUM OF C₂ IN INTERSTELLAR CLOUDS

BY

KYLE B. FORD

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Chemistry
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Adviser:

Assistant Professor Benjamin J. McCall

Abstract

In this work the understanding of the rotational equilibrium of C_2 in interstellar clouds is updated. It is critical for this equilibrium to be well understood if C_2 is to be used as a probe of the physical conditions in these clouds. Although new data continues to be published, the model was last updated in 1987. In this work, new data is collected and incorporated into the model using the program Radex, which will provide a standard format for sharing data, facilitate future updates, and enable the model to quickly run for a grid of temperature and density conditions.

TABLE OF CONTENTS

Chapter 1: Introduction.....	1
Chapter 2: Description of C ₂ Structure and Transitions.....	3
Chapter 3: Organization of data and Update Guide.....	9
Chapter 4: Preliminary Results and Future Direction.....	17
Appendix A: Program BandTrans.f.....	18
References.....	26

Chapter 1

Introduction

C_2 was first discovered in the interstellar medium in 1977 when absorption lines of the (1-0) Phillips band were detected in a near-infrared spectrum of Cyg OB2 No. 12 [1]. C_2 is an astronomically important molecule because the rotational levels of the ground state are long lived and their relative populations measurable. Originally, C_2 was expected to be a good optical probe of kinetic temperature in the interstellar medium. However, in 1982, van Dishoeck and Black (hereafter VDB) published their work [2] describing the excitation of C_2 , which pointed out that the radiative pumping and fluorescence cascade plays an important role in the balance of these states. Although this ruled out C_2 as a direct kinetic temperature probe, it opened the possibility of C_2 as a probe of other physical properties such as radiation field and density. Astronomers have used C_2 as a sensitive probe of physical conditions since these discoveries. [3,4]

The use of C_2 as a probe of astrophysical conditions depends on the ability to relate observed line intensities to the distribution of population amongst low-lying rotational states. This distribution depends not only on the physical conditions like cloud density and radiation field, but on the molecular properties of C_2 such as collision cross sections and radiative transfer rates. If the molecular properties are known, the physical properties can be deduced by comparison of the observed relative populations with predictions from a model run while varying these properties.

In their 1982 work [2], VDB provided a such a model of the C_2 fluorescence cascade. However, at that time there were some portions of the cascade which had not yet been studied. For example, there existed no data for $C_2 - H_2$ collision rates, nor for the transition moment of the $F^1\Pi_u - X^1\Sigma_g^+$ band. Therefore, VDB were at times forced to infer and estimate data based on reasonable assumptions.

In 1987, Le Bourlot, Roueff, and Viala [5] (hereafter BRV) updated the study of VDB by adding more modern data for intercombination transitions and collision rates which had been calculated at that point.

Since that time, there have been several studies which have updated understanding of the fluorescence cascade, but there has been no published work of an updated model. This work is motivated by and intended to fill this gap. In this work, the radiative excitation model will be updated again to include modern data. Realizing that the understanding of the C₂ fluorescence cascade will certainly evolve over time, the long-term goal is to create a database for radiative excitation data of C₂ which is easily updated and accessible to the scientific community.

To achieve this goal, Radex, developed by Van der Tak [6] has been used. Radex is a program which calculates the strength of spectroscopic lines from interstellar clouds assumed to be homogeneous. Radex does not assume local thermodynamic equilibrium, and provides options for things such as cloud geometry, collision partners, etc. In addition to the obvious benefit of 'easy use', the data file needed for the program will provide a standard format to facilitate sharing of data.

Chapter 2

Description of C₂ Structure and Transitions

The use of C₂ as a probe of astrophysical conditions requires that the energy levels of C₂ and the molecular properties describing their interactions be well understood. Fortunately, C₂ has been the subject of many studies both experimental and theoretical [7,8,9]. Also, C₂ spectroscopic data prior to 1992 was presented well in the review by Martin [10]. This wealth of established knowledge contains all the information necessary to understand the molecular properties of C₂ which enables the deduction of properties of the astronomical environment from observations.

2.1 Fluorescence cascade

Modeling the C₂ fluorescence cascade requires first an understanding of the electronic states involved in the cascade. C₂ has a $^1\Sigma_g^+$ ground state. The other three singlet states involved in the cascade are the $A^1\Pi_u$, $D^1\Sigma_u^+$, and $F^1\Pi_u$ states. Interstellar C₂ is excited into one of these states by absorption of a photon. The molecule then decays back into the ground state almost immediately through spontaneous emission, often into a different vibrational level. Absorption to other high level states is forbidden due to symmetry constraints and can therefore be ignored.

At this point, when a molecule is vibrationally excited in the ground state, the $a^3\Pi_u$ state becomes significant. Intercombination transitions between the $a^3\Pi_u$ and $X^1\Sigma_g^+$ state are one of the ways that vibrationally excited molecules can cool. Although spin-forbidden and therefore slow, these transitions are important because vibrational transitions within an electronic state are forbidden for homonuclear diatomic molecules.

Therefore, there is no fast relaxation pathway to compete with these slow mechanisms. Finally, molecules in the ground vibrational and electronic state can undergo further rotational cooling by slow quadrupole transitions or through collisional relaxation.

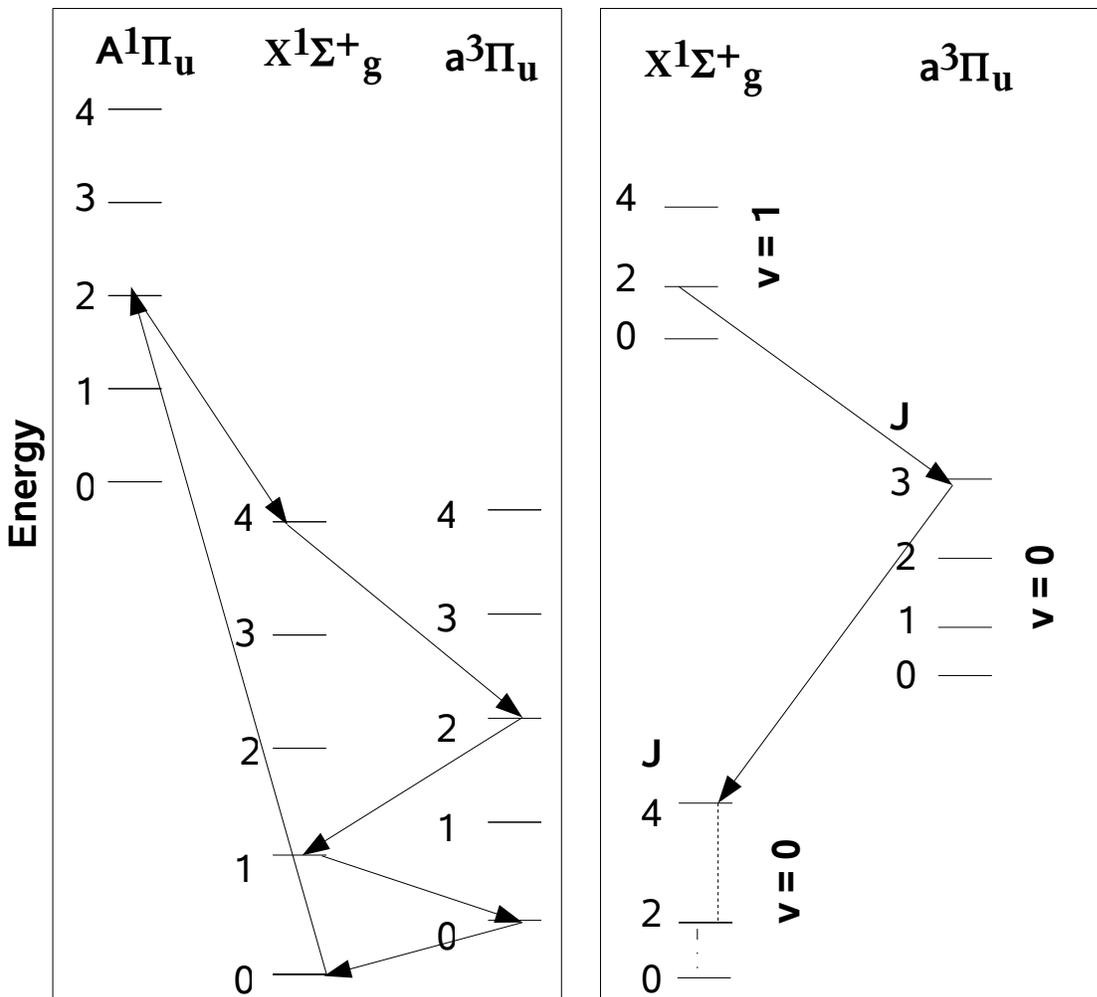


Figure 1: Low lying energy levels of C₂. Adapted from VDB [2]. The left side of the figure shows an overview of the radiative excitation-fluorescence-relaxation cycle which dictates the balance of population in the ground state. Arrows drawn indicate one possible pathway as an illustration. Note that any change Δv is possible for these transitions. The right figure shows rotational structure and illustrates various transitions. The different vertical lines represent the different types of relaxation within the vibrational ground state.

2.2 Energy level structure

Molecular constants presented by Martin [10] were used for each level. Energy levels are calculated according to the usual formula:

$$E = \omega_e(v + 1/2) - \omega_e X_e(v + 1/2)^2 + \omega_e Y_e(v + 1/2)^3 + B_e J(J + 1) - D_e J^2(J + 1)^2 \quad (1)$$

$X^1\Sigma_g^+$ is the ground electronic state of C_2 . In this work the reasoning of VDB and BRV is followed. Namely, states of $J > 20$ and $v > 4$ are ignored. The cutoff for rotational levels is chosen because it can be shown empirically that consideration of higher rotational levels does not effect results. Vibrational levels of $v > 4$ are ignored because these states lie higher in energy than the $A^1\Pi_u$ state and can therefore relax into the lower levels of this state through allowed, fast transitions. It should also be noted that the odd J levels of this state are anti-symmetric with respect to exchange of the two nuclei and because C_2 is a homonuclear diatomic, these levels do not exist.

$a^3\Pi_u$ is the only triplet state considered. This state is very similar in energy to the ground state, being only about 600 cm^{-1} greater in energy. This electronic state simply serves as a decay pathway for the ground state. Since there are no $X^1\Sigma_g^+$ levels higher in energy than this cutoff, states $J > 20$ and $v > 4$ are ignored. In this work, each electronic sub state has been considered separately. Normally, one would expect these sub states to have a double electronic degeneracy. However, each of these lambda doublets consists of a pair of anti-symmetric and symmetric states. As mentioned previously, the anti-symmetric states do not exist. Therefore, the electronic degeneracy is essentially absent.

$A^1\Pi_u$ is the lowest lying singlet excited state of C_2 . Similarly to the ground state, in this work we ignore states $J > 20$ and $v > 4$. This choice is again made empirically. It should also be noted that there exists no $J = 0$ level within this electronic state, as levels with $J < \Omega$ do not exist.

$D^1\Sigma_u^+$ is only considered for levels $v = 0$ and $J < 20$. This approximation is again made from an empirical arguments, as well as following the reasoning of VDB that these higher energy states are of little importance. Note that $D^1\Sigma_u^+$ also is missing anti-symmetric levels. Noting that it is of opposite parity from $X^1\Sigma_g^+$ it can be concluded that in this case, the levels of odd J are missing.

$F^1\Pi_u$ is the highest electronic state considered in this work. As with $D^1\Sigma_u^+$, only $v = 0$ and $J < 20$ states are considered by the same reasoning. Similar to $A^1\Pi_u$, there exist no $J = 0$ levels within this electronic state.

2.3 Spontaneous Emission

All allowed transitions between these levels following the normal selection rules of $\Delta J = 0, \pm 1$ and $g \leftrightarrow u$ are considered for all values of Δv . In addition, the intercombination transitions $X^1\Sigma_g^+ \leftrightarrow a^3\Pi_u$ are considered for all possible transitions $\Delta J = 0, \pm 1$. Finally, quadrupole transitions within $X^1\Sigma_g^+$ which satisfy $v = 0, \Delta J = -2$ are considered. Spontaneous emission rates were calculated from a variety of sources as explained below.

$A^1\Pi_u - X^1\Sigma_g^+$ (Phillips system): Band oscillator strengths taken from Kokkin et. al. [11] Astronomical observations are typically made from transitions of this band.

$D^1\Sigma_u^+ - X^1\Sigma_g^+$ (Mulliken system): Vibrational band spontaneous emission rates were taken from Chabalowski et. al. [12] and translated into band oscillator strengths according to the formula:

$$A_{2 \rightarrow 1} = \frac{8\nu^2 \pi^2 e^2}{m_e c^3} \frac{g_v'}{g_v} f_{v, v'} \quad (2)$$

where ν is the frequency of the transition, e is the charge of an electron, m_e is mass of an electron, c is the speed of light and g represents the degeneracy for each state.

$F^1\Pi_u - X^1\Sigma_g^+$: The oscillator strength used for the 0-0 band is an estimate made by Bruna and Grein. [13]

$a^3\Pi_u - X^1\Sigma_g^+$: Franck-Condon factors calculated by VDB were adopted and translated into band oscillator strengths using:

$$A_{2 \rightarrow 1} = \frac{64\nu^3\pi^4}{3h} \frac{g_v}{g_v'} \sum |R_e|^2 q_{v,v'} \quad (3)$$

The calculated band spontaneous emission can then be easily related to a band oscillator strength through equation (1). In their work, VDB estimated $\sum |R_e|^2$ to be $(1-2) \times 10^{-5}$ for this transition. However, following the studies of Le Bourlot and Roueff [14], 1×10^{-6} is believed to be a better estimate and has been used in this work.

$X^1\Sigma_g^+ - X^1\Sigma_g^+$ (quadrupole transitions): The quadrupole transition rates reported by VDB were used in this work. VDB point out that quadrupole transitions within a Π_u would be orders of magnitude slower than intercombination and they are therefore ignored.

For each transition, with the exception of quadrupole transitions, the calculated band oscillator strength was passed into a Fortran program which calculates spontaneous emission rates for every rovibronic transition that is considered. Quadrupole transitions were not passed into the program because the low number of considered transitions makes a direct copy easier. Within the program, rovibronic transition rates are calculated according to:

$$A_{2 \rightarrow 1} = \frac{8\nu^2 \pi^2 e^2}{m_e c^3} \frac{g_1}{g_2} \frac{S_J}{(2J+1)} q_{v, v'} \quad (4)$$

Where S_J is the rotational line strength (Hönl-London) factor.

2.4 Collisions

Collisional rate data has been graciously provided by Najar from work done in Najar et. al. [9] Collisional excitation and de-excitation is only considered within the vibrational ground state of $X^1\Sigma_g^+$. Collisional transitions between other levels are inefficient and collisions within higher vibrational levels are dwarfed in comparison to the rate of intercombination. Therefore, these transitions are not expected to play a significant role in the balance of ground state population and are ignored.

Chapter 3

Organization of data and Update Guide

One of the main goals of this work is to synthesize data for C_2 in such a way that updates are easy to perform as new data becomes available. This section is intended to describe the details behind this work as well as to document the process of updating the C_2 data file. All relevant files are available from the McCall group upon request.

3.1 Energy Levels and Molecular Constants

The foundational file for this work is the spreadsheet 'C2 energy levels'. Naturally, this contains data relating to the energy levels of C_2 . This file can be used as a base for easily calculating and organizing data before passing it on into other places. A user can update these levels by changing the value of the given molecular constants within the appropriate tabs on the spreadsheet. These tabs within the spreadsheet are labeled by a letter representing each electronic state considered. For example, the letter A represents the $A^1\Pi_u$ electronic state. The list on the right side of the spreadsheet for a given electronic state should be updated by copying the altered levels as numeric values, not using a formula. This will facilitate copying of data later in the procedure.

Once all energy levels have been changed as desired, the next step is to generate an energy level list. Within the same spreadsheet, the user will notice the leftmost tab labeled 'Ordered E'. The user should copy data from all other tabs into these columns in the given order (energy, degeneracy, J, identifier). Again, this is easiest if the rightmost columns in each tab are updated, in which case these columns can simply be directly copied. At this point, the user must decide which states will be considered. The user should only copy desired energy levels into this

list. Next, this list must be ordered as desired by Radex. This can be done by highlighting the numerical values that were added (energy, degeneracy, J, identifier) and using the spreadsheet sort function to organize the data by increasing values of energy. It is also recommended that on this spreadsheet the user record which energy levels were ignored.

The next step is to move the data into input files easily readable by the program 'Bandtrans.f'. The first file of this type is 'Energies.inp'. The user should replace all old data in this file by copying in all data from the 'ordered E' spreadsheet, excluding headers. The other files to be considered are the 'codes' series (Acodes.inp, Xcodes.inp, etc.). These files contain an ordered list of energy level identifiers which will be used by the program. These identifiers are of the form $Zvjj$ where Z is the letter used to identify the electronic state, v is the vibrational quantum number, and jj is the rotational quantum number. For example, A310 is the code for the $v=3, j=10$ level within the $A^1\Pi_u$ electronic state. Each 'codes' file must contain the code for every level to be considered within the given electronic state in order of increasing energy. These can be easily copied from the 'C2 energy levels' spreadsheet.

1	924.1021	1	0	X00
2	934.9683	5	2	X02
3	960.3208	9	4	X04
4	1000.1549	13	6	X06
5	1054.4632	17	8	X08
6	1123.2357	21	10	X010
7	1206.4595	25	12	X012
8	1304.1192	29	14	X014
9	1416.1968	33	16	X016

Table 1: Energy input file sample. The format is: Energy level number (all levels labeled in increasing order), Energy in wavenumber, degeneracy, J, energy level code.

3.2 Spontaneous Emission

The current version of the 'BandTrans.f' program requires data for transition strength to be provided in the form of band oscillator strengths. Data currently being used by 'BandTrans.f' is kept in the 'oscillator_str' series of files. As with other files described up to this point, the

naming convention is straightforward, using the letter symbols associated with each electronic state involved to describe the contents. Intercombination data is the exception to this. It is located in the file 'oscillator_str_aintercom' for intercombination originating in the $a^3\Pi_u$ state and 'oscillator_str_Xintercom' for intercombination originating in $X^1\Sigma_g^+$.

When new data regarding transition strengths becomes available, new oscillator strength input files should be created. These files must be formatted exactly as the old ones. Namely, oscillator strengths must be provided one per line with nothing in between. If necessary, it is recommended that '0' be used as a placeholder to prevent problems with input to the program. If we designate band oscillator strengths as $f_{x,y}$ where x is the vibrational quantum number of the upper electronic state and y is the vibrational quantum number of the lower electronic state, then the band oscillator strengths must be provided in the order: $f_{0,0}$ $f_{0,1}$... $f_{0,max}$ $f_{1,0}$ $f_{1,1}$ $f_{1,2}$... $f_{1,max}$... $f_{max,max}$.

Note that this input file format does not contain any information about which vibrational levels are described. This information will be input directly to the 'BandTrans.f' program which will be described later.

f	v_a	v_x
2.727E-7	0	0
0	0	1
0	0	2
0	0	3
0	0	4
5.212E-8	1	0
2.665E-7	1	1

Table 2: Oscillator strength input file sample. Only the leftmost column is present in the input file. The two right columns have been added for descriptive purposes. This example uses the intercombination oscillator strengths for transitions originating in $a^3\Pi_u$. Some oscillator strengths have been removed to illustrate the idea of using 0 as a placeholder for unused spots between the two values of v_{max} .

3.3 Collisions and Quadrupole Transitions

Data for collisions and quadrupole transitions were not processed by any program, but were directly copied into the Radex data file. Future users who need to alter this data must simply make sure that their input matches the Radex format. Examples of this format can be found in the online guide to Radex. [6]

3.4 BandTrans Program

The program 'BandTrans.f' is used as a tool for manipulating data and easily generating large lists of transitions in a ready-to-use format. Given input from two electronic states, it outputs spontaneous emission rates for every transition from the designated higher state into the lower. It is written in fortran and compiled preferably with f95. Using the g95 compiler causes errors when the program is run. This program uses a series of user-created input files which store data for energy levels, band oscillator strengths, and level identifier codes. While most details are marked with comments within the program, this chapter will present an overview of the program itself. Suggestions for potential improvements will also be discussed.

3.4.1 Variable Definitions

A complete list of variable definitions is provided below. In these definitions, the terms 'high state' and 'low state' will be used to refer to the electronic state of higher energy and the electronic state of lower energy, respectively, of the two states involved in the transition.

- dummy: An unused variable storing unwanted information from a 'read' statement.
- EinA: Einstein A coefficient. This is the spontaneous emission rate.
- Eu: For a given transition, this is the higher state's energy expressed as a temperature (K).
- grabhi: A placeholder variable. Contains the identifier code for the next energy level of the high state to be “grabbed” from the energy level list.
- grablo: A placeholder variable. Contains the identifier code for the next energy level of the low electronic state to be “grabbed” from the energy level list.

- **hicodes:** Identifier code of the v, J level of the high state is given by `hicodes(v+1,J+1)`. This is stored for convenience and is not currently used.
- **hidegens:** Degeneracy of the v, J level of the high state is given by `hidegens(v+1,J+1)`
- **hienergies:** Energy of the v, J level of the high state is given by `hienergies(v+1,J+1)`
- **hilvlnums:** Number of the v, J levels of the high state with respect to the ordered list is given by `hilvlnums(v+1,J+1)`
- **hystatecode:** Energy level identifier code for the high electronic state.
- **HonLon:** Honl-London rotational line strength factor.
- **Jhi:** A placeholder variable which designates the currently considered rotational level of the high state. This is incremented within loop structures.
- **Jhimax:** Highest rotational level of the high state which should be considered.
- **Jhimin:** Lowest rotational level of the high state which should be considered. Some levels are missing their lowest J states.
- **Jhistep:** Step size for J for the high state. Electronic states with J levels missing must be incremented by two. This value is one for all other levels. Must be changed to reflect the state being considered.
- **Jlo:** A placeholder variable which designates the currently considered rotational level of the low state. This is incremented within loop structures.
- **Jlomax:** Highest rotational level of the low state which should be considered.
- **Jlomin:** Lowest rotational level of the high state which should be considered. Some levels are missing their lowest J states.
- **Jlostep:** Step size for J for the low state. Electronic states with J levels missing must be incremented by two. This value is one for all other levels. Must be changed to reflect the state being considered.
- **lambdahi:** Quantum number Λ for the high state. Must be changed to reflect the state being considered.
- **lambdalo:** Quantum number Λ for the low state. Must be changed to reflect the state being considered.
- **locodes:** Identifier code of the v, J level of the low state is given by `locodes(v+1,J+1)`. This is stored for convenience and is not currently used.
- **lodegens:** Degeneracy of the v, J level of the low state is given by `lodegens(v+1,J+1)`
- **loenergies:** Energy of the v, J level of the low state is given by `loenergies(v+1,J+1)`
- **lolvlnums:** Number of the v, J levels of the low state with respect to the ordered list is given by `lolvlnums(v+1,J+1)`
- **lostatecode:** Energy level identifier code for the low state.
- **temp (1-4):** Temporary variables which store information read from input file which is later packed into an array.
- **Ostrs:** Array holding the band oscillator strengths. $f_{hi, low} = ostrs(hi+1, low+1)$ where hi and low are the vibrational levels of the high and low states, respectively.
- **TransE:** Energy of the transition in GHz.
- **vhi:** A placeholder variable which designates the currently considered vibrational level of the high state. This is incremented within loop structures.

- v_{lo}: A placeholder variable which designates the currently considered vibrational level of the low state. This is incremented within loop structures.
- v_{maxhi}: Highest vibrational level of the high state which should be considered.
- v_{maxlo}: Highest vibrational level of the low state which should be considered.

3.4.2 Program overview

This section will describe the functional parts of the program. Details of the program inputs and portions that the user must change will be discussed in detail in the next section.

After declaration of variable and some input statements, line 41 begins a series of 'if' statements. These statements set the minimum J value considered and the step size of J to be used based on the electronic state codes given by the user. This is necessary because of the missing J levels in certain electronic states.

The next section begins around line 85 where the 'codes' files are opened and the first line read. It would have been better to create these codes internally by concatenating strings of the letter portion with the vibrational and rotational quantum numbers. Unfortunately, string concatenation is difficult in Fortran so this method was used as a workaround.

After the files are opened and the first line is read, there is a loop which walks through the energy level input file (master list) using these codes. Each line of the master list is read and the identifier code of the level is compared to the code of the next level for which the program is searching for both electronic states. If the code matches, the data for that level is stored appropriately in an array and the next code is read for that particular electronic state. These codes can be done one by one in order because the format of the energy level input file required by Radex must be in order of increasing energy.

At line 138 the band oscillator strengths are read in from an input file and stored in the array 'ostrs'. As described in section 3.4.1, the ordering of this array follows: $f_{hi, low} = ostrs(hi+1, low+1)$ where hi and low are the vibrational levels of the high and low states, respectively.

Following this, all the necessary data for calculating transitions has been stored locally and the program can begin writing output. The program does this by stepping through the v,J states of the lower electronic state and calculating R,Q, and P transitions as applicable for each

level. Once these transitions have been calculated for every lower state level, the vibrational level of the upper state is incremented and the process repeated until all permutations of levels have been covered. For each transition, a Hönl-London factor, transition energy, and spontaneous emission rate are calculated. The details of this calculation are discussed in chapter 2.3. Data is written to the output as it is calculated. Once this loop is done, the program is finished.

3.4.3 Use of the Program

This section will point out parts of the program that must be considered when used. After declaration of variables as described in section 3.4.1, there is a space where most direct input is done. Currently the user adds this information directly by changing these fields within the code and recompiling. A practical upgrade to the program would be to enable these inputs through command line or some other means. For now, the lines are clearly marked in the code with comments. The user must input the symbol for the two electronic states involved in the transition, the maximum J and v values to be considered for each state, and the quantum number Λ for each state. Note that for intercombination, the program must be run twice for each electronic substate: once with $X^1\Sigma_g^+$ as the high electronic state and again with it as the lower state. This is because there are vibrationally excited levels of $X^1\Sigma_g^+$ higher than $a^3\Pi_u$ substates even though it is the lower electronic state.

The next part which will need to be changed often is at line 85. Upon each use, the user must alter the two lines which open the 'codes' files to reflect the two states being considered. A similar thing must be done at line 138 where the oscillator strength file used must be changes as needed.

Finally, the last thing which must be changed often is the Hönl-London formula. These are found at lines 185, 208, and 223 for the R, Q, and P branches, respectively. The formula must be changed depending on the value of $\Delta\Lambda$ for the transition. For convenience, these formula have been given at the end of the program in a ready-to-use form which can simply be copied into the program as needed.

Once the program has been recompiled in f95 and run, the output is written to a file named 'transitions.out'. This can be further manipulated or copied into the Radex input file as needed. It is recommended that the user skim over this file for errors. For certain transitions, there will be 'NaN' for the spontaneous emission rate. This occurs when a transition does not exist, the Honl-London factor is zero. These lines can be simply deleted from the output file. These lines could be deleted automatically through the program, but they remain so that real errors are not accidentally overlooked as well. When transitions are deleted, the user should ensure that every level is left with at least one transition. If not, this will cause errors in Radex. Any such level should be removed from the master energy level list when discovered. If there are no such errors at this point, with all data copied in, the user should have a fully functional input file for Radex.

3.5 Radex

Radex is the main program upon which this work depends. It is a program written for statistical equilibrium radiative transfer calculations. In short, this work is focused on establishing molecular properties for C_2 . Radex is the program into which all these properties are input. Once an input file is obtained, Radex can quickly calculate output excitation temperature, optical depth, and line intensity minus background for each specified spectroscopic transition. When Radex is run for a grid of temperature and number density, these physical properties for a cloud can be determined by comparison of the observed line intensity ratios with what is found in the grid calculated by Radex.

A user manual and detailed instructions for use can be found on the Radex website [6]. Therefore, that material will not be covered at length in this work. However, there are a few additional comments other users might find useful. The file 'radex.inc' contains parameters included in the main program. Errors can potentially arise if these parameters are not set correctly. For example, there are parameters which set the array size for number of energy levels, radiative transitions, and collisional transitions. Users may need to change the parameters in this file if changes are made to the number of transitions to be considered.

Chapter 4

Preliminary Results and Future Direction

A functional input file for C_2 has been generated and processed using Radex. However, the iterative method used by this program to solve the system of equations which describes the transition balance fails to converge. Parameters for the calculation (located in the file 'radex.inc') were relaxed as loose as 50% relative tolerance on the solution, but there was still no convergence after several thousand iterations.

A possible explanation for this is that the iterative method used in the Radex code does not handle this particular problem very well. In order to solve this issue, it is recommended that the program should be altered to output the main matrix 'A'. This matrix can then be manipulated or input into a program such as Matlab which has the ability to easily run several different iterative algorithms. Once a suitable algorithm is found, it could be coded into Radex.

Ultimately, it would be desirable to establish an easily updated and accessible database which astronomers could use interpret C_2 observations. Radex can be easily scripted to run for a grid of temperature and pressure. Such scripts are included in the Radex package. Once Radex is scripted, a website could be provided which would allow researchers the ability to easily and quickly compare C_2 observations with the most recent model.

Finally, it is highly recommended that the intercombination transition rates be calculated according to the work of Roueff and Le Bourlot [14]. In this work, they point out that the rotational terms do not factor out as is normally done with Hönl-London formulae. Spin – orbit coupling is also considered in their calculations. Unfortunately, this important data is no longer available. Incorporation of this data would likely greatly increase the accuracy of the C_2 rotational equilibrium model.

Appendix A

Program BandTrans.f

```
c-----  
c This program is intended for use in preparing input files for the program Radex. This program  
c takes the inputs of a Radex-formatted energy level list along with molecular constants and  
c outputs a Radex-formatted list of transition levels, Einstein A values, and transition energy.  
  
c IMPORTANT: portions the user will likely need to edit are marked with "*****"  
c-----  
  
      PROGRAM bandtransitions  
      implicit none  
      character :: histatecode, lostatecode  
      character*4 :: grabhi, grablo, temp4  
      integer :: vmaxhi, vmaxlo, vhi, vlo, Jhi, Jlo, lambdahi, Jhistep,  
      & Jlostep, Jhimax, Jlomap, lambdalo, temp1, temp3, Jhimin, Jlomin,  
      & dummy  
      real :: temp2, transE, EinA, HonLon  
c ***** May need to change size of these arrays if v>10 and J>20 are considered.  
      real, dimension (15,15) :: ostrs      !Band oscillator strengths  
      real, dimension(15,30) :: hienergies, loenergies, hidegens,  
      & lodegens, Eu      !Eu is the upper state energy expressed as a temperature.  
      integer, dimension(15,30) :: hilvlnums, lolvlnums  
      character, dimension(15,30) :: hicodes, locodes  
  
c ***** State codes are used to refer to what energy levels we are working with. Find relevant  
c codes in the excel sheet /home/group/Desktop/C2_project/C2 energy levels.ods Examples:  
c ground state = X, first excited singlet = A, etc. Naturally, histatecode corresopnds to the  
c higher energy electronic state  
      histatecode = 'X'  
      lostatecode = 'P'  
c ***** Max value of v considered for upper and lower electronic states.  
      vmaxhi = 4  
      vmaxlo = 4  
c ***** Quantum number lambda of each state. (Sigma = 0, pi = 1, delta = 2, etc.)  
      lambdahi = 0  
      lambdalo = 1
```

c ***** Must set highest J value considered for each state. Usually 20 unless state 'D' is being
c considered. (19 for D)

```
Jhimax = 20  
Jlomax = 20
```

c These if statements set the minimum for J. This is necessary because some states have missing
c J values. See Herzberg diatomics p.220 for explanation.

```
if (hystatecode == 'U' .OR. hystatecode == 'A' .OR.  
&hystatecode == 'F' .OR. hystatecode == 'D' .OR.  
&hystatecode == 'T') then  
  Jhimin = 1  
  else if (hystatecode == 'P') then  
    Jhimin = 2  
  else  
    Jhimin = 0  
endif
```

```
if (lostatecode == 'U' .OR. lostatecode == 'A' .OR.  
&lostatecode == 'F' .OR. lostatecode == 'D' .OR.  
&lostatecode == 'T') then  
  Jlomin = 1  
  else if (lostatecode == 'P') then  
    Jlomin = 2  
  else  
    Jlomin = 0  
endif
```

c This next segment establishes the step size to be used. This is due to the fact that the 'X' state is
c missing odd levels of J and the 'D' state is missing even levels of J.

```
if (hystatecode == 'X' .OR. hystatecode == 'D') then  
  Jhistep = 2  
else  
  Jhistep = 1  
endif
```

```
if (lostatecode == 'X' .OR. lostatecode == 'D') then  
  Jlostep = 2  
else  
  Jlostep = 1  
endif
```

c Setting up initial conditions for the loop that is about to start.

```
open(unit=11,file="Energies.inp")  
vlo = 0  
Jlo = Jlomin
```

```
vhi = 0
Jhi = Jhimin
```

```
c ***** You will need to change the file names below as appropriate. This is the start of a
c silly workaround. It is hard to deal with concatenation of strings with integers in fortran.
c Instead, just make a file of all the codes you will need and read them off. These are files 77 and
c 88.
```

```
open(unit=77,file="Pcodes.inp") !lower electronic state codes
open(unit=88,file="Xcodes.inp") !high electronic state codes
read(88,*) grabhi
read(77,*) grablo
```

```
c Loop through energy level file and grabs relevant data. This takes advantage of the fact that
c even though all levels are mixed up, for any given electronic state, the levels of that state will
c still be in order (with stuff between). The "grab" variables are the codes for the next states that
c we are looking for. This is where the stupid hack is used. Avoiding the fortran integer + string
c concatenation problem, read codes out of the relevant files. step up and read again once a
c match is made.
```

```
do while ((vhi .le. vmaxhi) .OR. (vlo .le. vmaxlo))
  read(11,*) temp1, temp2, temp3, dummy, temp4
```

```
c Compare code found in file with the next ones we want. If it matches, store the data and step up
c the value of J for that particular state.
```

```
if (trim(temp4) == trim(grabhi)) then
  hilvlnums(vhi+1,Jhi+1) = temp1
  hienergies(vhi+1,Jhi+1) = temp2
  hidegens(vhi+1,Jhi+1) = temp3
  hicodes(vhi+1,Jhi+1) = temp4
  Eu(vhi+1,Jhi+1) = temp2/.6950387
  read(88,*) grabhi
  Jhi = Jhi + Jhistep
else if (trim(temp4) == trim(grablo)) then
  lolvlnums(vlo+1,Jlo+1) = temp1
  loenergies(vlo+1,Jlo+1) = temp2
  lodgens(vlo+1,Jlo+1) = temp3
  locodes(vlo+1,Jlo+1) = temp4
  read(77,*) grablo
  Jlo = Jlo + Jlostep
endif
```

```
c This performs clean-up before next loop. If we have grabbed all J values for a given v, step up
c v and reset J.
```

```

    if (Jhi .gt. Jhimax) then
      vhi = vhi + 1
      Jhi = Jhimin
    else if (Jlo .gt. Jlomax) then
      vlo = vlo + 1
      Jlo = Jlomin
    endif
  enddo
close(11)
close(77)
close(88)

```

c Before data processing, should also gather oscillator strength data. Oscillator strengths are read
c in from an appropriately formatted file (see note at bottom of the oscillator str file) and
c stored in a 10x10 array. with the oscillator strength $f(x,y)$ corresponding to $ostrs(x+1,y+1)$

c ***** Will need to change the file name to reflect the appropriate oscillator strengths.

```

    open(unit=22,file="oscillator_str_Xintercom.inp")
    vhi = 0
    vlo = 0
    do while(vhi .le. vmaxhi)
      read(22,*) ostrs(vhi+1,vlo+1)
      vlo = vlo + 1
      if (vlo .gt. vmaxlo) then
        vhi = vhi + 1
        vlo = 0
      endif
    enddo
close(22)

```

c Now all relevant data is stored.

```

    open(unit=33,file="transitions.out")
30  format(i4, 4x, i4, 4x, e12.4, 4x, e12.4, 4x e12.4)
    vhi = 0
    vlo = 0
    Jlo = Jlomin
    do while(vhi .le. vmaxhi)

```

c This loop will walk through every permutation of vhi and vlo, calculating the transition values
c and printing them to a file. The final if statement of the loop will control progression, stepping
c up the J value, or resetting it and changing lower vib state once all J values are finished. Once
c all lower vibrational levels are considered, higher vibration state is change and it all starts
c again. Essentially, it is a loop within a loop within a loop. For each J value, a Q, P, and R
c branch is calculated.

c Begin calculation of R branch. The ugly looking if statement checks to make sure the other
c level is not out of bounds
c or a missing level in either the X(missing odd J) or D(missing even J) state. It also eliminates
c any situation in which the upper electronic level state has lower energy than lower state level.
c This last check only applies to a-X intercombination because these two states have similar
c electronic energies.

```

      if ((hienergies(vhi+1,Jlo+2) .gt. loenergies(vlo+1,Jlo+1))
& .AND. (Jlo .lt. Jlomax) .AND. (Jlo+1 .ge. Jhimin) .AND.
& ((histatecode .ne. 'X') .OR. (MOD(Jlo+1,2) .eq. 0)) .AND.
& ((histatecode .ne. 'D') .OR. (MOD(Jlo+1,2) .eq. 1))) then

```

```

      transE = (hienergies(vhi+1,Jlo+2) -
&loenergies(vlo+1,Jlo+1))*3.0e10

```

c ***** HonLon is the Honl-London factor. This formula changes depending on delta lambda,
c and whether it is a Q, R, or P branch. Honl-London formula can be copy-pasted from the
c bottom of this document. (From Herzberg diatomics p. 208) Remember that delta lambda is
c conventionally denoted for absorption. In other words, if lamdahi=1 and lamdalo =0, delta
c lamda is +1.

c Delta lambda: -1, R Branch

```

      HonLon = real((Jlo+1-lamdahi)*(Jlo+1-1-lamdahi))/
&real(4*(Jlo+1))

```

c EinA is the einstein A coefficient (spontaneous emission rate).

```

      EinA = ((8.*3.1415**2. *transE**2. *(1.602176487e-19)**2.
& *8.99e9*lodegens(vlo+1,Jlo+1))/(9.10938215e-31*299792458.**3.*
&hidegens(vhi+1,Jlo+2))*ostrs(vhi+1,vlo+1)*HonLon/(2.*Jlo +1)

```

```

      write(33,30) hilvlnums(vhi+1,Jlo+2),
&lolvlnums(vlo+1,Jlo+1), EinA, transE, Eu(vhi+1,Jlo+2)
      endif

```

c Begin calculation of Q branch. Honl-London formula changes. Other formula just reference
c different spot (Jlo instead of Jlo+1). Again, ugly if statement makes sure other level exists and
c that the "low" level is not higher than the "high" level.

```

      if ((hienergies(vhi+1,Jlo+1) .gt. loenergies(vlo+1,Jlo+1))
& .AND. Jlo .ge. Jhimin) .AND.
& ((histatecode .ne. 'X') .OR. (MOD(Jlo,2) .eq. 0)) .AND.
& ((histatecode .ne. 'D') .OR. (MOD(Jlo,2) .eq. 1))) then

```

```

      transE = (hienergies(vhi+1,Jlo+1) -
&loenergies(vlo+1,Jlo+1))*3.0e10

```

c ***** Again, may need to change HonLon formula for Q branch.

c Delta lambda: -1, Q Branch

```
HonLon = real((Jlo-lambdahi)*(Jlo+1+lambdahi)*(2*Jlo+1))/  
&real(4*Jlo*(Jlo+1))
```

```
EinA = ((8.*3.1415**2. *transE**2. *(1.602176487e-19)**2.  
&*8.99e9*Iodegens(vlo+1,Jlo+1))/(9.10938215e-31*299792458.**3.*  
&hidegens(vhi+1,Jlo+1))*ostrs(vhi+1,vlo+1)*HonLon/(2.*Jlo +1)
```

```
write(33,30) hilvlnums(vhi+1,Jlo+1),  
&lolvlnums(vlo+1,Jlo+1), EinA, transE, Eu(vhi+1,Jlo+1)  
endif
```

c Begin calculation of P branch. Again, formula are slightly different. Initial check makes sure
c we are not trying to calculate a P branch going to negative values of J or non-existent levels of
c X or D.

```
if ((hienergies(vhi+1,Jlo) .gt. loenergies(vlo+1,Jlo+1))  
& .AND. (Jlo .ge. 1) .AND. (Jlo-1 .ge. Jhimin) .AND.  
& ((histatecode .ne. 'X') .OR. (MOD(Jlo-1,2) .eq. 0)) .AND.  
& ((histatecode .ne. 'D') .OR. (MOD(Jlo-1,2) .eq. 1))) then
```

```
transE = (hienergies(vhi+1,Jlo) -  
&loenergies(vlo+1,Jlo+1))*3.0e10
```

c ***** Again, may need to change Honl-London factor formula for P branch.

c Delta lambda: -1, P Branch

```
HonLon = real((Jlo-1+1+lambdahi)*(Jlo-1+2+lambdahi))/  
&real(4*(Jlo-1+1))
```

```
EinA = ((8.*3.1415**2. *transE**2. *(1.602176487e-19)**2.  
&*8.99e9*Iodegens(vlo+1,Jlo+1))/(9.10938215e-31*299792458.**3.*  
&hidegens(vhi+1,Jlo))*ostrs(vhi+1,vlo+1)*HonLon/(2.*Jlo +1)
```

```
write(33,30) hilvlnums(vhi+1,Jlo),  
&lolvlnums(vlo+1,Jlo+1), EinA, transE, Eu(vhi+1,Jlo)  
endif
```

c This is the "clean-up" part of the loop. This part ensures that this "while" statement runs as a
c nested loop.

```
Jlo = Jlo + Jlostep  
if (Jlo .gt. Jlomap) then
```

```

        Jlo = Jlo_min
        vlo = vlo + 1
    endif

    if (vlo .gt. v_maxlo) then
        vlo = 0
        vhi = vhi + 1
    endif
enddo
close(33)
end program

c *****This section contains Honl-London formulae that can be copy-pasted*****
c Delta lambda: +1, R Branch
c      HonLon = real((Jlo+1+lambdahi)*(Jlo+1-1+lambdahi))/
c      &real(4*(Jlo+1))
c
c Delta lambda: +1, Q Branch
c      HonLon = real((Jlo+lambdahi)*(Jlo+1-lambdahi)*
c      &(2*Jlo+1))/real(4*Jlo*(Jlo+1))
c
c Delta lambda: +1, P Branch
c      HonLon = real((Jlo-1+1-lambdahi)*(Jlo-1+2-lambdahi))/
c      &real(4*(Jlo-1+1))
c
c Delta lambda: 0, R Branch
c      HonLon = real((Jlo+1+lambdahi)*(Jlo+1-lambdahi))/
c      &real(Jlo+1)
c
c Delta lambda: 0, Q Branch
c      HonLon = real((2*Jlo+1)*lambdahi**2)/real(
c      &Jlo*(Jlo+1))
c
c Delta lambda: 0, P Branch
c      HonLon = real((Jlo-1+1+lambdahi)*(Jlo-1+1-lambdahi))/
c      &real(Jlo-1+1)
c
c Delta lambda: -1, R Branch
c      HonLon = real((Jlo+1-lambdahi)*(Jlo+1-1-lambdahi))/
c      &real(4*(Jlo+1))
c
c Delta lambda: -1, Q Branch
c      HonLon = real((Jlo-lambdahi)*(Jlo+1+lambdahi)*(2*Jlo+1))/
c      &real(4*Jlo*(Jlo+1))

```

c Delta lambda: -1, P Branch

c HonLon = real((Jlo-1+1+lambdahi)*(Jlo-1+2+lambdahi))/

c &real(4*(Jlo-1+1))

References

- [1] Souza, S. P. & Lutz, B. L. *Detection of C₂ in the Interstellar Spectrum of Cygnus OB2 Number 12 (VI Cygni Number 12)* ApJL, **1977**, 216, L49-L51
- [2] van Dishoeck, E. F. & Black, J. H. *The excitation of interstellar C₂* ApJ, **1982**, 258, 533-547
- [3] Lambert, D. L.; Sheffer, Y. & Federman, S. R. *Hubble Space Telescope Observations of C₂ Molecules in Diffuse Interstellar Clouds* ApJ, **1995**, 438, 740-749
- [4] Cecchi-Pestellini, C. & Dalgarno, A. *C₂ Absorption-line Diagnostics of Diffuse Interstellar Clouds* MNRAS, **2002**, 331, L31-L34
- [5] Le Bourlot, J.; Roueff, E. & Viala, Y. *Rotational equilibrium of C₂ in diffuse interstellar clouds. I - Static model: the case of Zeta Ophiuchi* A&A, **1987**, 188, 137-144
- [6] Van der Tak, F.F.S., Black, J.H., Schöier, F.L., Jansen, D.J., van Dishoeck, E.F. *A Computer Program for Fast non-LTE Analysis of Interstellar Line Spectra* **2007**, A&A 468, 627-635.
- [7] K. P. Huber and G. Herzberg, *Molecular Spectra and Molecular Structure IV. Constants of Diatomic Molecules*, Van Nostrand-Reinhold, New York, **1979**.
- [8] M. Douay, R. Nietmann and P. F. Bernath, *New observations of the A¹Π_u – X¹Σ_g⁺ Transition (Phillips System) of C₂* JMS **1988**, 131, 250.
- [9] Najar, F.; Abdallah, D. B.; Jaidane, N.; Lakhdar, Z. B.; Chambaud, G. & Hochlaf, M. *Rotational excitation and de-excitation of C₂ (X¹Σ_g⁺) by para-H₂ (j = 0)* JCP, **2009**, 130, 204305
- [10] Martin, M. *C₂ Spectroscopy and Kinetics*. Journal of Photochemistry and Photobiology A: Chemistry, **1992**, 66, 263 - 289
- [11] Kokkin, D. L.; Bacskay, G. B. & Schmidt, T. W. *Oscillator Strengths and Radiative Lifetimes for C₂: Swan, Ballik-Ramsay, Phillips, and d³π_g ← c³Σ_u⁺ Systems*. JCP, **2007**, 126, 084302
- [12] Chabalowski, C. F.; Peyerimhoff, S. D. & Buenker, R. J. *The Ballik-Ramsay, Mulliken, Deslandres-d'Azambuja and Phillips systems in C₂: a theoretical study of their electronic transition moments*. Chem. Phys., **1983**, 81, 57 - 72
- [13] Bruna, P. J. & Grein, F. *Spectroscopy of the C₂ molecule: Valence and Rydberg states in the 7-10 eV region. An ab initio study* Canadian Journal of Physics, **2001**, 79, 653-671

[14] Bourlot, J. L. & Roueff, E. *Intercombination transitions between levels $X^1\Sigma_g^+$ and $a^3\Pi_u$ in C_2* . JMS, **1986**, 120, 157 – 168.