5-2019

Effects of Anharmonicity on Vibrational Frequencies and Molar Absorptivities of Triplet-HON

Andrea Nicole Becker
University of Tennessee, abecke12@vols.utk.edu

Recommended Citation

Becker, Andrea Nicole, "Effects of Anharmonicity on Vibrational Frequencies and Molar Absorptivities of Triplet-HON. " PhD diss., University of Tennessee, 2019.
https://trace.tennessee.edu/utk_graddiss/5429

Effects of Anharmonicity on Vibrational Frequencies and Molar Absorptivities of Triplet-HON

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Andrea Nicole Becker

May 2019

Dedication

I dedicate this dissertation first and foremost to my parents, Robert and Regina Becker, who have provided unwavering support through trials and tribulations and never told me that I ‘couldn’t do it’ , and my grandmother Lois Becker, who has always done the same. Next to my brother, Jeffrey Becker, who discusses the most random of science tangents with me on holidays or late at night. I dedicate this to ‘The Herd’ – Helen, Heather, Samara, Lillie, Katie, and Jessica – because without them I would not have developed into the person I am today; to ‘The Roomies plus One’ – Robin, Rebecca, and Cody – for being my rocks through college and my anchors through graduate school. To my lab mates both past and present: Sara Isbill, Randi Beil, Paul Mott, Gavin McCarver, Kiran Dhah, Matt Dutra, James Smith, Ashleigh Barnes, and Jesse Kern for letting me bounce ideas off of them and swapping stories, jokes, and bad puns. And finally, I dedicate this to my cats Tux and Jupiter, because nothing is better than coming home to two beings who are excited to see you and waiting at the window (even if it’s just because you feed them), or having a purring body decide to use you as a pillow.

Acknowledgments

I would like to thank my advisor Dr. Robert J. Hinde for his invaluable guidance and mentorship during my time working for him, and our collaborator Dr. David Anderson at the University of Wyoming for the motivation of this project as well as the experimental data we compared against. The National Science Foundation has my appreciation for the grant allowing us to pursue this research. Thanks are also extended to the National Institute of Computational Sciences for the use of their Advanced Computing Facility.

I would also like to thank Dr. Bhavya Sharma, Dr. Konstantinos Vogiatzis, and Dr. David Keffer for serving on my doctoral committee. Finally, Dr. Harkiran Dhah and James Smith III, M.S., have my overwhelming gratitude for laying a majority of the groundwork and benchmarking with their research on HNO and HNNO under Dr. Hinde.

Abstract

The reaction between NO and atomic hydrogen to form HNO is one that has been studied both experimentally and theoretically due to the observation of HNO in interstellar matter, as well as HNO's role as an intermediate product in some atmospheric reactions.[5] However, this reaction can also produce the HON molecule when performed in solid argon or *para*-hydrogen matrix environments at extremely low temperatures (10K and below 5K, respectively). [15, 20] Surprisingly, the HNO and HON products are formed at comparable rates in the *para*-hydrogen matrix, even though the reaction to form HNO has no energy barrier whilst the formation of HON must surpass a 12 $^{kcal}/_{mol}$ barrier.[20, 5] The molar absorptivities of these two molecules are required to thoroughly study the kinetics of the $H + NO$ reaction occurring within the matrix; since the IR spectrum is taken *in situ*, these values must be obtained computationally.

The Double Harmonic Approximation (DHA) is a common method for calculating the molar absorptivities of a molecule; however, the DHA does not consider anharmonicity or coupling between vibrational levels. We propose both of these additions as important contributors to the molar absorptivities of HON due to the highly anharmonic character of its O-H vibrational mode, and therefore applied Vibrational Second-Order Perturbation Theory (VPT2). Unfortunately, while the VPT2 method provided better models of the potential energies than the DHA, the dipole moment polynomials were less accurate when compared to the *ab initio* data. Application of the linear variational method instead of VPT2 allowed us to control over whether or not coupling was included in the system, as well as the number of anharmonicity terms added (degree of polynomial), and we found a 10^{th} order polynomial with a linear combination of the harmonic wavefunctions nine lowest energy levels was necessary to best model a single vibrational mode. The combinations of harmonic wavefunctions from each vibrational mode required when modeling the entire system is still unclear, but the inclusion of coupling between the modes has a significant effect on the calculated anharmonic frequencies and should be taken into account in any future analysis.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and General Information

1.1 About HNO and HON

The molecule known as nitroso hydrogen or nitrosyl hydride (HNO) has been studied experimentally due to its role as an intermediate in various atmospheric chemical reactions, and presence in interstellar space. [5, 15] These reactions include the generation of NO pollutants through combustion [15] or the decomposition of ozone. [5] However, theoretical calculations first showed − and experiments later verified − that the molecule HON should be formed along-side HNO, albeit at lesser concentrations depending on the conditions under which it was formed.

The author of a 1994 computational study [14] provided the following suggested guidelines for the experimental generation HON: "Probably the best method of forming the XON isomers is via matrix isolation experiments in which NO and X radicals diffuse, although the mobility of the NO radical especially must be limited. It does not seem likely that any appreciable formation of the XON isomers will occur in the gas-phase, and therefore these species may be safely ignored in the study of atmospheric chemistry." [14] While this conclusion may set to rest any further concerns about the presence of HON alongside HNO in the atmosphere, it still supplies a new question: what happens when HON *is* generated in matrix isolation experiments, as hypothesized by Lee? [14]

1.2 Experimental Background: Anderson Group

Dr. David Anderson at the University of Wyoming has undertaken a series of experiments that aim to provide an answer to that question, amongst others. These experiments are performed in a solid *para*-hydrogen matrix, rather than the more common argon environment. When irradiated, *para*-hydrogen matrix serves as the source of H atoms that ultimately react with the NO radical to form HNO and HON. The apparatus used by the Anderson and his co-workers holds the matrix at extremely low temperatures ranging from 1.8 to 4.3 K, a regime where quantum effects are influential; the Anderson group is interested in learning how the quantum effects of the diffusion of the H atom through the matrix affects the formation of the products.

The technique of using matrix isolation spectroscopy for bimolecular reactions is fundamentally reliant upon species diffusion through the matrix. The bimolecular reaction

$$\mathrm{H} + \mathrm{NO} \longrightarrow \mathrm{HNO}$$

studied by the Anderson group occurs when an H atom diffuses through the solid $\mathrm{pH_2}$ matrix and reacts with a NO radical impurity; therefore the rate of the reaction is determined by the diffusion coefficient of the H atom. [20] This is because much heavier NO reactant is immobilized, or 'stored', within the rare-gas matrix due to its very small diffusion coefficient at these temperatures.

The H atom diffuses through the $\mathrm{pH_2}$ matrix by the reaction

$$\mathrm{H} + \mathrm{H_2} \longrightarrow \mathrm{H_2} + \mathrm{H}$$

which has an exchange reaction barrier of approximately 10 ${}^{kcal}/_{mol}$. [20] However, since the matrix is solid $\mathrm{pH_2}$ and therefore kept at temperatures below 5 Kelvin, the energy barrier cannot be overcome; the H atom instead travels through quantum tunneling. When a NO impurity is encountered, the product HNO should form without delay as the reaction has no energy barrier (seen in Figure 1.1).

The Anderson group produces the NO doped pH_2 matrix through rapid vapor deposition, with NO and pH_2 gases co-deposited onto a BaF_2 optical substrate in a liquid-He cryostat; the pH_2 gas was previously prepared using an ortho/para converter. Throughout the experiment, the temperature of the crystal is measured and intermittent Fourier-transform infrared (FTIR) spectra are taken to track the generation of products over time and thus the kinetics of the reaction. The experiment is initiated by irradiating a small portion of the crystal with a 193 nm 8 ns pulse, which dissociates approximately 30% of the NO radicals; these N and O atoms then react with the neighboring H_2 molecules to form

$$\mathrm{N} + \mathrm{H_2} \longrightarrow \mathrm{NH} + \mathrm{H}$$

$$\mathrm{N} + 2\,\mathrm{H_2} \longrightarrow \mathrm{NH_3} + \mathrm{H}$$

$$\mathrm{O} + \mathrm{H_2} \longrightarrow \mathrm{OH} + \mathrm{H}$$

$$\mathrm{OH} + \mathrm{H_2} \longrightarrow \mathrm{H_2O} + \mathrm{H}$$

The resulting H atoms diffuse through the pH_2 matrix *via* quantum tunneling until they interact with another impurity: this could be a NO radical, another H atom, or one of the photodissociation products. The Anderson group observed NH, NH_3, and H_2O to be the primary products created by the photolysis through the above reactions; the products of the H atom diffusion were identified as HNO, NOH, *trans*-HNOH, and NH_2OH.

1.3 Experimental Measurements: Anderson Group

Throughout the experiment, FTIR spectra are collected to identify and quantify the products created by the reaction. As the reaction proceeds, the Anderson group measured a continual decrease in NO concentration, thus indicating that formation of HNO and HON is due to the diffusing H atoms reacting with NO monomers in the matrix; this decrease can be seen in Figure 1 of Ref. [20]. Between traces a and b shown in that figure, the concentration of

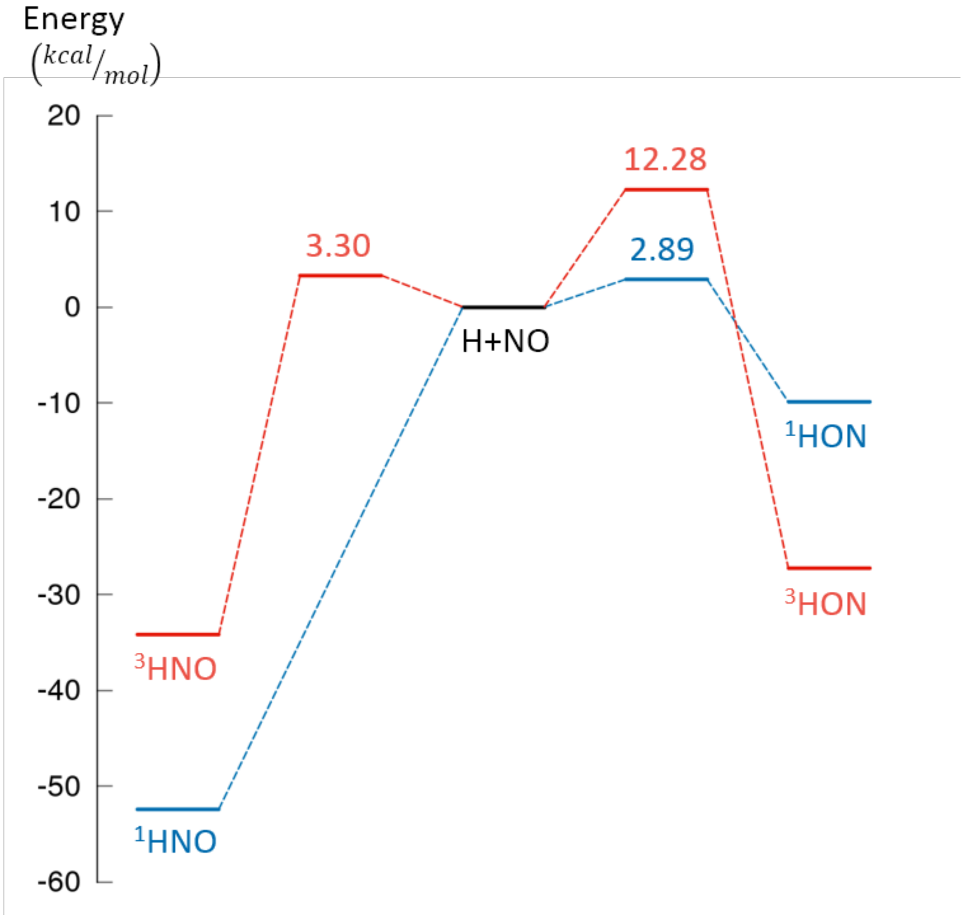


Figure 1.1: The energy profile for the reaction of H + NO; all energy values used were taken from the paper by Bozkaya. [5]

NO decreased by 28%, which yielded the photoproducts NH, NH_3, and H_2O seen in Figure 2 of Ref. [20].

These absorption spectra show the vibrational peaks associated with certain molecules increasing over time. Each vibrational peak is a response associated with a certain molecular vibration; the energy absorbed by the molecule, and thus the radiation frequency, is dependent upon the molecular structure. For example, a non-linear triatomic molecule such as HNO or HON will have three response peaks, as according to the formula 3N-6 where N is the number of atoms in the molecule. These response peaks are generalized as two 'stretching' modes, where the bonds extend and contract, and a 'bending' mode where the angle becomes more obtuse and acute. The frequencies associated with certain vibrational modes have been approximately quantified, such as an O-H bond appearing approximately in the 3200-3600 cm^{-1} range, or a nitro compound N-O bond having a strong response between 1500-1550 cm^{-1}.

The Anderson group identified the products formed by identifying the different peaks present in the IR spectra. Additionally, they were able to calculate the concentration of the NO in the matrix using a correlation between concentration and the measured absorbance. This relation in its most basic form is described by the Beer-Lambert Law, which states that

$$A(\lambda) = \epsilon(\lambda)bc$$

where A is the measured absorbance, ϵ is the molar absorptivity coefficient of the specific molecule, b is the path length the light travels, and c is the concentration of the sample. [17] If the molar absorptivity coefficient of the species is known (and it meets the requirements of the Beer-Lambert law), the species concentration can be calculated. For some molecules, the molar absorptivity coefficient at a given wavelength is easily obtained experimentally through calibration measurements; other molecules can more difficult to produce and isolate, thereby preventing the acquisition of pure samples for calibration.

HNO and HON are both molecules that do not yet have experimentally determined molar absorptivity coefficients. This means that instead of explicitly determining a concentration value for these molecules as they did with NO, the Anderson group must instead turn to computationalists to supply the theoretical molar absorptivity values, which are then used to calculate the concentration. While there is a possibility for error due to assumptions made during the calculation of the these coefficients, theorists can check the accuracy of their work against experimental results.

1.4 Molar Absorptivity Computation

The value of a calculated molar absorptivity coefficient can vary depending on the electronic structure method, level of theory, and basis set specified. The more rigorous and complete the theoretical approach is, the closer the calculated value should be to experiment. Additionally, characteristics of the molecule must also be taken into account: for example, if the molecule exhibits a weaker or longer bond than other similar species, it is likely diffuse functions should be included in the basis set. Due to these possible variations, it is important to compare calculated results against experimentally obtained values. For molecules that have more than one normal mode, though the response peaks may be of different magnitudes, the concentration of the molecule is constant. Therefore a ratio of the integrated absorbance values can be taken to obtain a ratio of the molar absorptivity coefficients.

$$\frac{A(\lambda_2) = \epsilon(\lambda_2)bc}{A(\lambda_1) = \epsilon(\lambda_1)bc} \tag{1.1}$$

By comparing the ratio of calculated molar absorptivity coefficients – and their associated wavelengths – to those of experiment, the computationalist can evaluate whether their calculated values are sufficiently accurate to be used to determine concentration.

As stated earlier, HNO and HON are non-linear triatomic molecules with three vibrational modes. One benefit of examining such small molecules is that more rigorous levels of theory can be employed without being too computationally expensive. A comparison

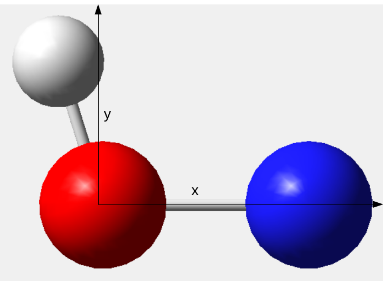


Figure 1.2: The HON molecule oriented on the xy plane. The oxygen is centered at the origin, with the O-N bond along the positive x-axis. The hydrogen is oriented in the positive y-direction.

between different combinations will be examined in the next chapter.

1.5 Kinetics Results: Anderson Group

The Anderson group used molar absorptivity values calculated by Maier *et al.* [15] to determine the concentrations of HNO and HON present in the pH_2 matrix from their corresponding FTIR peak intensities. They found that at 1.8 Kelvin the concentrations of HNO and HON increased at an equal rate after photolysis, until the concentration of H atoms was almost zero at approximately 300 minutes as seen in Figure 5 of Ref. [20]. This result is unexpected as the two products were identified as ^{1}HNO and ^{3}HON; calculations by Bozkaya [5] show ^{1}HNO to have no reaction energy barrier, while ^{3}HON has a reaction barrier of $12^{kcal}/_{mol}$. However, as mentioned above, the product formation was found to be limited by the quantum diffusion limit of the H atoms and not the rate at which the H atom reacted with any orientation of the NO impurity. The Anderson group did note that a change in temperature from 1.8 K to 4.3 K affected the concentration of ^{1}HNO but not that of ^{3}HON, as shown in Figure 6 of Ref. [20].

In order to best and fully quantify the products made, the Anderson group needs molar absorptivities that are as accurate as possible. To do so for ^{3}HON entails thoroughly characterizing the normal modes, and possibly adding additional corrections such as anharmonicity and coupling to the original calculations should they prove inadequate. The next chapter shall discuss literature that employed this strategy for HNO and HON.

Chapter 2

Literature Investigations of HNO and HON

There are three main papers investigating the vibrational properties of the HON molecule, from both experimental and computational standpoints. The results described include harmonic frequencies, dipole moments, reaction energies, barrier heights, molar absorptivities, as well as investigating isotopic effects on these listed properties. The methodologies used to obtain these results varies between the three studies: for example, Lee performed calculations on a closed-shell HON molecule as opposed to the later experimentally observed ^{3}HON, and the transition states between HNO and HON were calculated computationally by Bozkaya while Maier experimentally determined them through ultraviolet irradiation.

2.1 Lee

Lee's calculations were completed for ^{1}HON, rather than the experimentally observed ^{3}HON. However, the conclusions Lee drew from his work regarding the qualitative vibrational frequencies and molar absorptivity ratios can still be compared to those of ^{3}HON, while the transition energies between and barrier heights of closed shell isomers are compared to those calculated by Bozkaya. [5] The CCSD and CCSD(T) levels of theory were used to obtain all values, with the self-constructed TZ2P basis set. This consisted of Dunning's [5s3p/3s] basis functions [9] on H, O, and N, and an additional two sets of d polarization functions for both O and N. [13] Force constants were obtained by applying the finite difference method to analytical first derivatives of the energy.

The portion of Lee's work that focused on HON was aimed at determining whether HON was likely to form experimentally; he accomplished this by first characterizing the molecule's vibrational modes, and then determining its formation energy from the initial reactants of H and NO as well as the barrier height relative to its isomeric counterpart HNO.

2.1.1 Vibrational Frequencies and IR Absorption Intensities

Prior calculations by Lee [13] allowed for comparison between the ordering of the vibrational frequencies of HNO and HON: Lee noted that both molecules placed the stretching mode between the hydrogen and the central atom as the highest frequency, followed by the bending mode, with the lowest frequency consisting of the N-O bond stretch. The harmonic frequencies of ^{1}HON as calculated by Lee are: $\omega_1 = 3319$, $\omega_2 = 1462$, and $\omega_3 = 1238$ cm^{-1}, with respective molar absorptivity values of 27, 13, and 52 $^{km}/_{mol}$. That Lee calculated an approximately 2:1:4 ratio between the molar absorptivity values of the modes lends further credence to the conclusion by experimentalists [20, 15] that ^{3}HON is being generated and measured, as the response peak of the second vibrational mode is extremely low experimentally.

Table 2.1: Variables calculated or recorded by the three authors, as discussed in the following sections.

	Lee (^{1}HON)	Bozkaya	Maier
Computation	**X**	**X**	**X**
Experiment			**X**
Geometry	X	X	X
Frequencies	X	X	X
Barrier Heights	X		X
Dipole Moment	X		
Molar Absorptivities	X		X
Isotopes		X	X

Table 2.2: Mulliken population analysis as calculated by Lee. Lee declared ^{1}HNO the more stable of the two molecules as it "satisfies the octet rule without resorting to the use of partial charges". [14]

	HON			HNO	
	CCSD	CCSD(T)		CCSD	CCSD(T)
X	+0.34	+0.34	X	+0.22	+0.22
O	−0.33	−0.31	N	+0.05	+0.04
N	−0.01	−0.02	O	−0.27	−0.26

2.1.2 Stability and Barrier Height

When comparing HNO and HON, Lee concluded that ^{1}HNO was the significantly more stable molecule of the two due to being able to form without requiring partial charges to satisfy the octet rule; this conclusion is supported by his Mulliken charge analysis, where HNO has significantly less charge separation between the hydrogen and the N-O portion of the molecule. The geometry analysis also showed that the H-O bond length was slightly longer than most but still covalent.

The energy difference between ^{1}HNO and ^{1}HON was calculated to be 42.3 $^{kcal}/_{mol}$ in favor of HNO at 0 K, with a barrier height of 9.3 $^{kcal}/_{mol}$ from ^{1}HON to reach the transition state for isomerization. Therefore Lee concludes that while HNO is much more likely to form first, due to its lower ground-state energy and charge stability, some ^{1}HON should be expected to form in matrix isolation at low temperature. However, while Lee is partially correct that HON is observed at low temperatures, experimentalists actually measure ^{3}HON.

2.2 Bozkaya

Bozkaya's article and calculations focused heavily on the kinetics of all four species of H+NO: their isomerizations, dissociation reactions, isotopic substitutions, in addition to calculating their geometries and harmonic and anharmonic vibrational frequencies. The CCSD(T) level of theory with the cc-pCVQZ basis set was used when calculating all single point energies and geometries, while the reaction energies and barrier heights were determined *via* a series of

Table 2.3: Comparison between values calculated by both Lee [14] and Bozkaya [5] using differing levels of theory for ^{1}HON. Lee used CCSD(T)/TZ2P, while Bozkaya used CCSD(T)/cc-pCVQZ. Despite Bozkaya having a significantly larger basis set, their calculations for the equilibrium geometry, harmonic frequencies, and isomerization reaction energy were all quite close. Whether this trend would continue for the anharmonic frequencies is unfortunately unanswered, as Lee did not continue with anharmonic corrections.

	Lee	Bozkaya
E_h	-130.195 Ha	-130.339 Ha
r_{HO}	0.986 Å	0.9854 Å
r_{ON}	1.276 Å	1.2626 Å
θ_{HON}	109.9°	110.42°
ω_1	3319 cm^{-1}	3317 cm^{-1}
ω_2	1462 cm^{-1}	1473 cm^{-1}
ω_3	1238 cm^{-1}	1289 cm^{-1}
Isomerization Reaction	42.3 $^{kcal}/_{mol}$	42.23 $^{kcal}/_{mol}$
Dissociation Barrier	9.3 $^{kcal}/_{mol}$	7.88 $^{kcal}/_{mol}$

increasing corrections to the Hartree-Fock energy while extrapolating towards the complete basis set limit.

2.2.1 Vibrational Frequencies and Isotopic Substitutions

Bozkaya calculated the harmonic and anharmonic vibrational frequencies, which were obtained using the VPT2 method, for each molecule and transition state that was studied and compared them to other computational results when possible. Additionally, the anharmonic frequencies calculated for the commonly observed ^{1}HNO molecule were compared to experiment with a maximum variation of only 8 cm^{-1}, although Bozkaya did not compare to any experimental results for ^{3}HON.

Analysis indicated that most of the isotopes still showed tunneling effects, and a general summary of Bozkaya's conclusions is that the isotopes decreased the probability of the isomerization or dissociation reactions occurring but did not significantly change the reaction rate. Of the six cases, only two had additional notes: the dissociation reactions of ^{1}HON and ^{3}HON. Since the potential energy curve of ^{1}DON is much steeper than ^{1}HON, the probability of the molecule dissociating decreases which leads Bozkaya to state that ^{1}DON

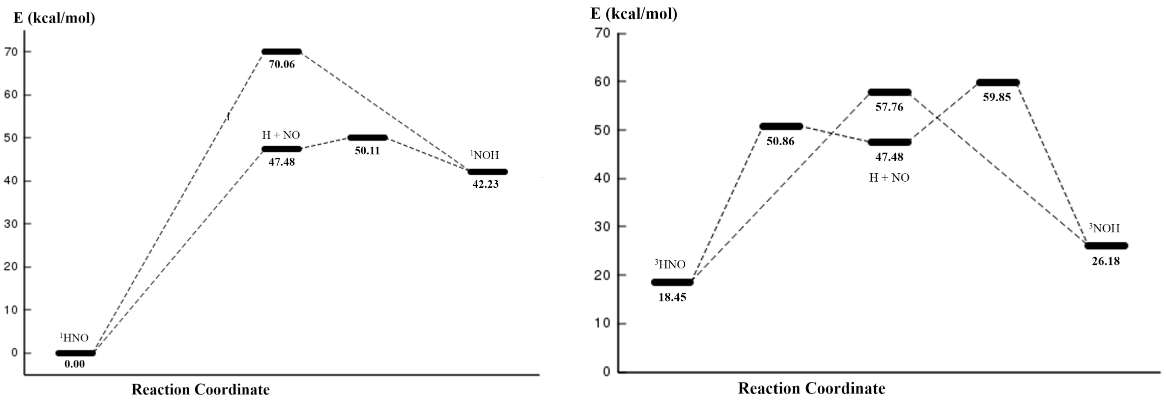


Figure 2.1: Singlet (left) and triplet (right) transition state energy profiles for the HNO to HON transitions and dissociations, as calculated by Bozkaya. All energies are relative to ^{1}HNO. [5]

is more likely to be observed than ^{1}HON. The molecule ^{3}HON is influenced by tunneling effects when below 500 K, while tunneling is only significant for ^{3}DON when below 300 K.

2.2.2 Isomerization and Dissociation Reactions

Bozkaya calculated isomerization reaction and dissociation barrier energies by using successive correlation corrections to extrapolate towards both the Full Configuration Interaction (FCI) and Complete Basis Set (CBS) limit. He found the transition state barrier of the ^{1}HNO → ^{1}HON isomerization (70.06 $^{kcal}/_{mol}$) to be higher than the isomerization reaction energy of 42.23 $^{kcal}/_{mol}$, indicating that a 2-step process of dissociating to H + NO then forming ^{1}HON would be more likely to occur. This is supported by Bozkaya's calculations for dissociation energies showing that the barrier height for the dissociation of ^{1}HON is only 7.88 $^{kcal}/_{mol}$, and the formation of ^{1}HNO has no barrier. An energy profile of this transition and dissociation is shown in Figure 2.1 on the left.

The potential energy surface for the isomerization and dissociations of ^{3}HNO and ^{3}HON is not as clear-cut as that of the singlets. While the energy of the dissociated reactants lies below that of the transition state, the energy barriers to forming either product are close in energy. However, the product distribution kinetics performed for ^{3}HNO and ^{3}HON demonstrate no isomerization from the initial product whatsoever for ^{3}HNO, and only a brief window of opportunity at 400 K for ^{3}HON. At 500 K and above, both molecules decompose to form $\mathrm{H} + \mathrm{NO}$.

Bozkaya concludes his explanations with a comparison of the kinetics analysis of the products distribution for each potential energy surface. The ^{1}HON molecule dissociates at room temperature due to its low activation energy, while ^{3}HON is a stable product until approximately 400 K. The formation of either product from their respective isomer of HNO is not probable, but ^{3}HON may be observed experimentally while ^{1}HON is extremely unlikely to exist. Additionally, Bozkaya states that tunneling effects at low temperatures have a strong influence on the isomerization and dissociation reactions of the molecule, which could play a role in the experimental observations of Anderson and Maier.

2.3 Maier

Maier is unique amongst the three papers discussed here in that he includes both experimental and theoretical results and spectra. The experimental results were collected for molecules created in an argon matrix at 10 K, while the theoretical results were calculated using the Double Harmonic Approximation on the Gaussian suite. Maier compares the experimentally obtained vibrational frequencies and integrated absorption intensities to the calculated harmonic frequencies and molar absorptivities for HON, DON, HO^{15}N, and DO^{15}N. He also used photoisomerization to induce a transition between HNO and HON.

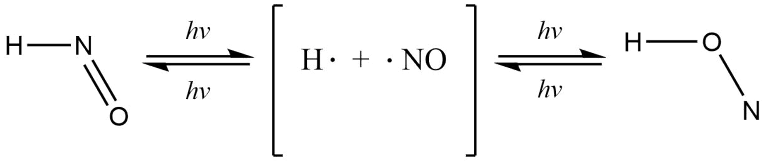


Figure 2.2: Transition between HNO and HON when irradiated with light, as observed by Maier. [15] Light of $\lambda = 313$ nm is required to isomerize from HNO to HON; to go from HON to HNO, light of $\lambda = 254$ nm is applied.

2.3.1 Difference Spectrum and Peak Comparison

When photoisomerization was induced in either direction, the IR spectrum taken measured an increase in the concentration of NO radical; therefore Maier concluded that the transition between HNO and HON passed through a dissociation as seen in Figure 2.2, rather than a transition state structure. The collected spectra were used to generate a 'difference spectrum', as seen in Figures 1 and 2 of Ref. [15]. Maier's calculated vibrational frequencies and molar absorptivities were used to generate theoretical IR spectra to compare to these experimental results, and showed a qualitative level of agreement.

The second vibrational mode of HON (ν_2) was not experimentally measurable due to its low intensity, in addition to being coupled with the third vibrational mode (ν_3). The frequencies and intensities of the modes that were observed, however, showed a somewhat fluctuating agreement with the computational results (Table 2.4). This is likely due to limited modeling parameters, which will be discussed later in detail. Still, the experimental and theoretical values had qualitative agreement.

2.3.2 Isotope Effects

Where HON does not have a measurable second vibrational mode, ν_2 is the highest intensity response peak of DON. Exchanging the hydrogen for its deuterium isotope effectively decouples the two lower frequency modes in addition to significantly changing the absorbance intensity ratios of the three modes, as seen in Table 2.4. By contrast, exchanging N for its ^{15}N isotope has little effect on the vibrational frequencies and absorption intensities of all three modes. This is likely somewhat due to it having only gained an additional ${}^{1}\!/_{14}{}^{th}$ of its original mass, unlike hydrogen which doubled in mass.

Table 2.4: Experimental and calculated values for the four isotopomers of HON. The vibrational frequencies are in units of cm^{-1}, with relative intensities are given in parentheses. [15] [b] BLYP/6-311++G**. [c] QCISD/6-311++G**. [d] Not measurable due to too low intensity. [e] Absolute intensity: 163 $^{km}/_{mol}$. [f] Absolute intensity: 177 $^{km}/_{mol}$. [g] Absolute intensity: 160 $^{km}/_{mol}$. [h] Absolute intensity: 168 $^{km}/_{mol}$. [i] Absolute intensity: 76 $^{km}/_{mol}$. [j] Absolute intensity: 95 $^{km}/_{mol}$. [k] Absolute intensity: 73 $^{km}/_{mol}$. [l] Absolute intensity: 93 $^{km}/_{mol}$.

Vibration		HON	HO^{15}N	DON	DO^{15}N
ν_1	exp.	3467.2 (19)	3466.4 (19)	2563.6 (28)	2564.6 (20)
	BLYP[b]	2479.8 (27)	2479.8 (27)	2532.8 (30)	2532.8 (31)
	QCISD[c]	3780.3 (42)	3780.3 (44)	2751.9 (42)	2751.9 (43)
ν_2	exp.	–[d]	–[d]	1149.0 (100)	1127.9 (100)
	BLYP	1157.7 (0)	1145.7 (0)	1092.2 (100)[i]	1071.6 (100)[k]
	QCISD	1232.1 (1)	1222.4 (5)	1171.1 (100)[j]	1149.2 (100)[l]
ν_3	exp.	1095.6 (100)	1085.6 (100)	868.8 (79)	867.2 (74)
	BLYP	1030.2 (100)[e]	1020.7 (100)[g]	815.2 (70)	813.6 (72)
	QCISD	1142.5 (100)[f]	1129.1 (100)[h]	895.8 (81)	893.9 (52)

2.3.3 Multiplicity Analysis

The calculations done by Maier indicate that the lowest-energy state of HNO has a singlet multiplicity, which is common throughout the literature. The HON molecule, however, is predicted to be a triplet with an energy gap of 20 $^{kcal}/_{mol}$ between the ground states. When Maier compared the vibrational frequencies of ^{3}HON to the experimental spectrum, the peaks proved a better match than with ^{1}HON. This confirmation backs up Bozkaya's conclusion from kinetics thirteen years later, and limits some of the usefulness of Lee's data from five years before.

2.4 Overall Summary and Comparison

Lee's paper in 1994 was a good theoretical analysis as to what was limiting the observation of HON, and how that barrier could be surpassed experimentally. However, his calculations were only for ^{1}HON and an open-shell system was not included despite allusions towards the possibility. Five years later, Maier published what he stated to be the first matrix isolation and identification of HON in a solid argon matrix at 10 K. A transition between HON and HNO was induced through photoisomerization, with the intermediate products determined to be the dissociated H + NO. The calculations that accompanied Maier's experimental IR spectra showed that the products formed were ^{1}HNO and ^{3}HON, and their vibrational frequencies provided the closest matches to experiment.

In 2012, Bozkaya completed a series of kinetics calculations on the dissociation and isomerizations of all four species of the H + NO reaction. While he cited Lee, there was no mention made of Maier's results which would likely have been beneficial in supporting his conclusions. Regardless, the results of both Maier and Bozkaya agree that ^{3}HON is observed experimentally rather than ^{1}HON, and that the molecule will dissociate rather than follow a transition state when isomerizing.

Chapter 3

The Double Harmonic Approximation

When first considering the vibrational motion of a molecule, the most simple approximation to make is one of harmonicity: that the energy of the molecule changes identically with both a positive and negative magnitude of displacement of the normal mode q. This has the effect of truncating the potential energy equation at the second-order term. A harmonic system is also much easier to treat quantum mechanically, as the energy levels are evenly spaced at $\hbar\omega$, and the wavefunctions have a known form. This approximation is the basis of one of the most common methods for calculating molar absorptivity values, the Double Harmonic Approximation (DHA). The DHA uses the simplicity of the shortened equations for potential energy and dipole moment to calculate the molar absorptivity of a normal mode with fewer required calculations than the other, more involved, methods that will be discussed later.

3.1 Molar Absorptivity

No matter the rigor of the method being applied, the basis of calculating the molar absorptivity remains the same: the harmonic frequencies and normal mode displacement vectors of the molecules are determined, the normal mode displacements are used to calculate the transition dipole moments, and finally the frequencies and transition dipole moments are applied to find the molar absorptivity values of the molecule. What does change is the amount and intensity of the calculations between the first and second steps, as will be seen in Chapter Four.

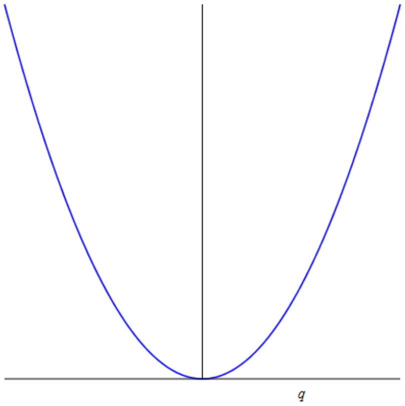


Figure 3.1: The curvature of a harmonic oscillator as a function of normal mode displacement (q).

3.1.1 Determining Normal Modes

The identity of a normal mode is defined by its energy level and molecular displacement, while the vibrational frequency of a normal mode is the difference in energy between that energy level and the ground state. To calculate these components for the three normal modes of the triplet HON molecule, a mass-weighted Hessian matrix was constructed and diagonalized.

A Hessian matrix consists of second-order partial derivatives of a function, in this case of the energy with respect to displacement of the coordinates. Therefore, the first step in obtaining the mass-weighted Hessian matrix is the optimization of the molecule to obtain the equilibrium energy and geometry values. The entries of the Hessian matrix can be numerically calculated by applying the finite difference method, where h is some small

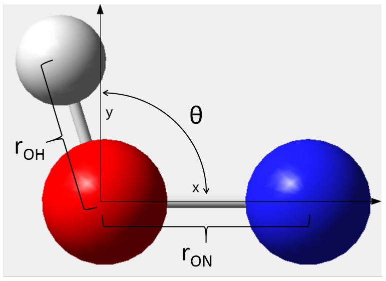


Figure 3.2: The three internal coordinates of the HON molecule.

displacement value.

$$\frac{\partial^2 U}{\partial x^2} = \frac{U(x+h) + U(x-h) - 2U(x)}{h^2} \tag{3.1}$$

$$\frac{\partial^2 U}{\partial x \partial y} = \frac{U\binom{x+h}{y+h} - U\binom{x-h}{y+h} + U\binom{x+h}{y-h} + U\binom{x-h}{y-h}}{4h^2} \tag{3.2}$$

Equation (3.1) is applicable to all 'on-diagonal' terms of the matrix, while Equation (3.2) applies to all 'off-diagonal' terms. While these derivatives could be found using Cartesian coordinates, it would require over 100 calculations whereas defining the system in internal coordinates requires only 19 calculations in total.

$$\begin{bmatrix} \frac{\partial^2 U}{\partial r_{OH}^2} & \frac{\partial^2 U}{\partial r_{OH} \partial r_{ON}} & \frac{\partial^2 U}{\partial r_{OH} \partial \cos\theta} \\ \frac{\partial^2 U}{\partial r_{ON} \partial r_{OH}} & \frac{\partial^2 U}{\partial r_{ON}^2} & \frac{\partial^2 U}{\partial r_{ON} \partial \cos\theta} \\ \frac{\partial^2 U}{\partial \cos\theta \partial r_{OH}} & \frac{\partial^2 U}{\partial \cos\theta \partial r_{ON}} & \frac{\partial^2 U}{\partial \cos\theta^2} \end{bmatrix}$$

However, the coordinate displacements of the vibrational modes are easier to interpret, visualize, and mass-weight when expressed in Cartesian coordinates, so the 3x3 Hessian matrix is converted to a 6x6 Hessian matrix according to the following transformation:

$$F_{cart}(l) = \sum_j \sum_k \frac{\mathrm{d}(int_j)}{\mathrm{d}(cart_l)} \frac{\partial^2 U}{\partial(int_j)\partial(int_k)} \frac{\mathrm{d}(int_k)}{\mathrm{d}(cart_l)}$$

When expressed another way, where x_1 and x_2 are the combination of Cartesian coordinates being considered, and y_1, y_2, and y_3 are the three internal coordinates:

$$\frac{\partial^2 f(y_1, y_2, y_3)}{\partial x_1 \partial x_2} = \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial y_i}{\partial x_1} \frac{\partial^2 f(y_1, y_2, y_3)}{\partial y_i \partial y_j} \frac{\partial y_j}{\partial x_2} \tag{3.3}$$

These derivatives $\frac{\partial y_i}{\partial x_n}$ are contained in what is known as 'Wilson's B Matrix', which for this molecule is

$$\begin{array}{ccccc} \text{A} & & \text{B} & & \text{C} \\ \begin{vmatrix} H_x \\ H_y \\ O_x \\ O_y \\ N_x \\ N_y \end{vmatrix} & \times\text{q} \;+ & \begin{vmatrix} H_x \\ H_y \\ O_x \\ O_y \\ N_x \\ N_y \end{vmatrix} & = & \begin{vmatrix} H_x \\ H_y \\ O_x \\ O_y \\ N_x \\ N_y \end{vmatrix} \end{array}$$

Figure 3.3: The calculation of new coordinates for a normal mode: where A=the displacement coordinates from the eigenvector, q=the magnitude of choice, B=the equilibrium coordinates of the molecule, and C=the new coordinates of the normal mode.

$$\begin{bmatrix} \cos\theta & \sin\theta & -\cos\theta & -\sin\theta & 0 & 0 \\ \frac{\sin^2\theta}{r_{OH}} & -\frac{\sin\theta\cos\theta}{r_{OH}} & -\frac{\sin^2\theta}{r_{OH}} & \frac{\sin\theta\cos\theta}{r_{OH}} - \frac{\sin\theta}{r_{ON}} & 0 & \frac{\sin\theta}{r_{ON}} \\ 0 & 0 & -1 & 0 & 1 & 0 \end{bmatrix}$$

The resultant 6x6 Hessian matrix consists of the H_x, H_y, O_x, O_y, N_x, and N_y Cartesian coordinate derivatives as the rows and columns; each value is then multiplied by $\sqrt{m_1 m_2}^{-1}$ and all units converted to atomic units to obtain the final mass-weighted Hessian matrix.

The diagonalization of the 6x6 Hessian matrix yields six eigenvalues, three of which are near zero, and their associated eigenvectors. The other three eigenvalues correspond to the three vibrational modes of the molecule, where the eigenvalue is equal to the square of the vibrational frequency. The associated eigenvectors contain the displacements of each atom along a Cartesian axis, in the same order as the rows and columns of the Hessian matrix. By adding or subtracting these values from the Cartesian coordinates of the equilibrium geometry, new coordinates are obtained that describe the position of the atoms during the vibration of that normal mode.

3.1.2 Calculating Transition Dipole Moment and Molar Absorptivity

The dipole moment of a molecule is the difference in two charges with opposite signs across a given distance: to calculate the dipole moment of a molecule at a given geometry, the *change* in those charges as a function of the electric field (F_ρ) is determined by calculating the energy of the molecule when placed in oriented electric fields of varying strengths. Then, by taking the following derivative

$$\mu_\rho = -\frac{\partial E}{\partial F_\rho}|_{F_\rho=0}$$

the dipole moment, μ_ρ, of the molecule along a given Cartesian axis ρ can be calculated. The number of points used to calculate this dipole moment can be as few as three, if a linear fit is confirmed.

In order to calculate the *transition* dipole moment, the dipole moments as a function of normal mode must be obtained. This means that for a range of displacements of a normal mode on either side of the equilibrium geometry, the dipole moment is first calculated and then fit as a function of that displacement magnitude q. Once the polynomial fit has been achieved, the linear term is extracted as

$$\frac{\partial \mu_\rho}{\partial q}|_{q=0}$$

which is then used to obtain the transition dipole moment $|M_{ji}|^2$ between vibrational levels $j = 0$ and $i = 1$, where M_{ji} is in units of Debye.

$$|M_{ji}|^2 = \sum_\rho |\langle i|\mu_\rho|j\rangle|^2 \qquad (3.4)$$

Finally, this transition dipole moment $|M_{ji}|^2$ and the corresponding normal mode frequency ν_0, in units of cm^{-1}, are substituted into the following formula to obtain molar absorptivity

$\bar{A}$ in units of $^{km}/_{mol}$: [17]

$$G_{ji} = 41.6238|M_{ji}|^2 \quad \bar{A} = \frac{G_{ji}\nu_0}{16.60540} \tag{3.5}$$

3.2 Assumptions of the DHA

As stated in the name, the Double Harmonic Approximation assumes that the energy of the molecule as a function of the normal mode is harmonic. Since the dipole moment is the negative first derivative of the energy with respect to the electric field, this therefore requires the dipole moment to be linear. These restrictions can be seen when expanding $V(q)$ and $\mu(q)$ *via* a Taylor series, then truncating them at the third and second terms, respectively.

$$V(q) = V(q)_{q=0} + q\frac{\mathrm{d}V(q)}{\mathrm{d}q}\bigg|_{q=0} + \frac{q^2}{2!}\frac{\mathrm{d}^2V(q)}{\mathrm{d}q^2}\bigg|_{q=0} \tag{3.6}$$

$$\mu_\rho(q) = \mu_\rho(q)_{q=0} + q\frac{\mathrm{d}\mu_\rho(q)}{\mathrm{d}q}\bigg|_{q=0} \tag{3.7}$$

The second term of Equation 3.6 is zero at equilibrium and $\frac{\mathrm{d}^2V(q)}{\mathrm{d}q^2}$ is calculated by the Hessian matrix, while $V(q)_{q=0}$ is the equilibrium energy of the molecule. Therefore Equation 3.6 can be rewritten as:

$$V(q) = V(q)_{eq} + \frac{\hbar\omega}{2}q^2 \tag{3.8}$$

This means that the terms of the potential energy equation for the DHA are set even before performing an *ab initio* scan of the energies as a function of normal mode displacement. However, this is not the case for the dipole moment equation. Due to the DHA assumption for dipole moment being a linear term truncation (Equation 3.7), the transition dipole moment term $|M_{ji}|^2 = \sum_\rho |\langle i|\mu_\rho|j\rangle|^2$ is

$$|M_{ji}|^2 = \sum_\rho \left| \langle i|\mu_\rho(q)_{q=0} + q\frac{\mathrm{d}\mu_\rho(q)}{\mathrm{d}q}\bigg|_{q=0}|j\rangle \right|^2$$

This expression can be simplified based on two pieces of knowledge about of the system: first, that the observed IR spectra peaks are for the transition between the $n = 0$ and $n = 1$

energy levels, $j = 0$ and $i = 1$; and second, that since the system is a harmonic oscillator, movement between the energy levels can be described by the ladder operators $\hat{a}^{+}$ and $\hat{a}$. The raising operator $\hat{a}^{+}$, when applied to a harmonic wavefunction $|\phi_n\rangle$, raises the energy level n by one and produces an eigenvalue times that new wavefunction:

$$\hat{a}^{+}|\phi_n\rangle = \sqrt{n+1}\,|\phi_{n+1}\rangle \qquad \hat{a}\,|\phi_n\rangle = n\,|\phi_{n-1}\rangle$$

The same process is true for the lowering operator $\hat{a}$, which lowers the energy level by one; however, n can never be less than 0 which indicates the ground-state energy level. If the lowering operator is applied to ϕ_0, the result is an eigenvalue of 0. The movement of the harmonic oscillator can be either up or down one energy level, so the operator describing this transition is a linear combination of the applied raising and lowering operators:

$$\hat{q} = \frac{\hat{a}^{+} + \hat{a}}{\sqrt{2}} \tag{3.9}$$

When the energy levels of i and j are specified, the first term of $|M_{ji}|^2$ becomes the overlap integral of $\langle 1|0\rangle$ multiplied by a constant which is therefore zero, and second term simplifies to

$$|M_{01}|^2 = \sum_{\rho} \left| \left.\frac{\mathrm{d}\mu_\rho(q)}{\mathrm{d}q}\right|_{q=0} \langle 1|q|0\rangle \right|^2$$

The movement between energy levels is denoted by the term $\langle 1|q|0\rangle$; the lowering operator can be substituted in for q, yielding a final expression for the transition dipole moment that depends only on the calculated dipole moments of the normal mode.

$$|M_{01}|^2 = \sum_{\rho} \left| \sqrt{\frac{1}{2}} \left.\frac{\mathrm{d}\mu_\rho(q)}{\mathrm{d}q}\right|_{q=0} \right|^2 \tag{3.10}$$

3.3 Calculations using DHA

All *ab initio* energy calculations were completed using the NWChem [22] computational program; for these DHA calculations the methods QCISD [19] and CCSD(T) [6] were implemented, with the 6-311++G** Pople [8, 12, 11, 7] and aug-cc-pVTZ Dunning basis sets. [10] All other calculations were done using programs written in Maple, which can be found in the Appendices for reference.

The QCISD and CCSD(T) methods applied here both include electron correlation, which is a correction that accounts for the instantaneous interaction between electrons in a molecule. QCISD stands for quadratic configuration interaction of single and double excitations, and corrects size-consistency errors that occurred with the CISD method; it was developed by the Pople group, and is on par with the CCSD method. Size-consistency is the concept that the energy of two non-interacting systems is equal to the sum of their energies when calculated separately, and is one of the main issues caused by limiting the terms included in configuration interaction. CCSD(T) is an abbreviation for the coupled-cluster method with full treatment for single and double excitations, and perturbative treatment used to estimate triple excitations. Unlike configuration interaction, coupled-cluster has no errors with size-consistency as it uses an exponential ansatz for the wavefunction instead of a linear combination ansatz.

The two basis sets used are by different creators, Pople and Dunning. The Pople basis sets use different numbers of primitive Gaussian functions to model the atomic orbital basis function. The 6-311++G** basis set is a split-valence triple-zeta basis set, with additional d and p polarization functions on non-hydrogen atoms and hydrogen respectively, and diffuse s and p functions for all atoms. For HON, this includes a total of 51 basis functions. Dunning's basis sets are called 'correlation-consistent' and are optimized using CISD wavefunctions; due to this, they are focused at best modeling the correlation energy of valence electrons. All Dunning basis sets contain polarization functions on the valence shells, denoted by 'pV', after which comes the size of the basis set; the aug-cc basis sets have additional diffuse functions added to each atom. the aug-cc-pVTZ basis set is a correlation-consistent polarized valence triple-zeta basis set and contains a total of 115 basis functions for HON. [Paci]

3.3.1 System Specifications

The displacement value h was set at 0.01 when calculating the values of the Hessian matrix. The frequencies extracted from the diagonalization of that matrix were converted to units of cm^{-1}, and the displacement vectors converted to Å. The value of q in Figure 3.3 ranged from -5 to 5 in increments of 0.5; all future data discussed also uses these parameters. The electric fields used to obtain the dipole moments of the molecule had strengths of -0.002 and 0.002 atomic units and were applied along either the x-axis and y-axis for each normal mode, at each value of q.

Although the Double Harmonic Approximation uses the linear term of a polynomial fit to calculate the transition dipole moment of a normal mode, this does not necessarily mean that a linear equation is the best fit. For a given degree of polynomial, the y-intercept of each fitted equation should be equal to the equilibrium dipole moment of the molecule; if this is not the case, higher-order terms are required. Convergence in the y-intercept term occurred between the cubic and quartic fits, and so the linear term of the cubic polynomial was extracted and used to calculate the transition dipole moment $|M_{ji}|^2$ (Equation 3.10).

3.3.2 Comparison with Literature Values

Bozkaya used the CCSD(T) method and a cc-pCVQZ basis set to calculate molecule geometries, energies, and frequencies that can be compared with our results. The cc-pCVQZ basis set contains 174 total basis functions for hydrogen, oxygen, and nitrogen, whereas the aug-cc-pVTZ basis set used for our calculations contains 115. Despite Bozkaya's calculations containing significantly more functions to describe the atomic orbitals, our equilibrium geometries and harmonic frequencies were quite close to their reported values for both HON and its isotopic counterpart DON. However, Bozkaya did not calculate molar absorptivities for the molecules, and so any further comparisons will continue in Chapter Four.

Maier, however, calculated harmonic frequencies and molar absorptivities for HON, DON, and their isotopic nitrogen counterparts using QCISD and the 6-311++G** basis set which

Table 3.1: Left column calculated using CCSD(T)/aug-cc-PVTZ; right column (Bozkaya's) calculated using CCSD(T)/cc-pCVQZ. [5]

	Calculated	Bozkaya's
E_h	-130.271 Ha	-130.417 Ha
r_{HO}	0.972 Å	0.9676 Å
r_{ON}	1.335 Å	1.3255 Å
θ_{HON}	107.3°	107.47°
HON ω_1	3699 cm^{-1}	3728 cm^{-1}
HON ω_2	1232 cm^{-1}	1247 cm^{-1}
HON ω_3	1118 cm^{-1}	1138 cm^{-1}
DON ω_1	2692 cm^{-1}	2713 cm^{-1}
DON ω_2	1151 cm^{-1}	1180 cm^{-1}
DON ω_3	892 cm^{-1}	897 cm^{-1}

has 78 basis functions. As seen in Table 3.2, there is very close agreement between our calculated values and those of Maier, thus reinforcing our conclusion that Maier used the Double Harmonic Approximation to obtain the molar absorptivities listed. Nearly every value is a close match, except for one: the molar absorptivities of the third vibrational mode of DON. We find this value to be strangely off when all other numbers agree so closely, especially considering that the HON and HO^{15}N pair had extremely similar molar absorptivities; it would be expected that DON and DO^{15}N would follow this trend as our own values did.

To ensure that this discrepancy was not due to a miscalculation on our parts, we calculated the harmonic frequencies and molar absorptivities also using CCSD(T)/6-311++G**, and compared both Pople basis set results to those obtained with the CCSD(T)/aug-cc-pVTZ combination in Table 3.3.

Table 3.2: A comparison between our calculated vibrational frequencies and molar absorptivities, and those of Maier [15], using QCISD/6-311++G**.

	Frequencies (cm^{-1})		Molar Intensities ($^{km}/_{mol}$)	
	Calculated	Maier	Calculated	Maier
HON	3778.2	3780.3	75.2	74.3
	1231.8	1232.1	2.69	1.77
	1141.5	1142.5	182	177
$HO^{15}N$	3778.2	3780.3	75.2	73.9
	1222.3	1222.4	9.48	8.40
	1128.0	1129.1	169	168
DON	2750.4	2751.9	39.9	40.
	1169.7	1171.1	94.8	95
	895.9	895.8	**49.1**	**77**
$DO^{15}N$	2750.4	2751.9	39.9	40.
	1147.9	1149.2	92.6	93
	894.0	893.9	48.6	48

Table 3.3: Comparison between the three combinations of method and basis set we used to calculate harmonic frequency and molar absorptivity.

	Frequencies (cm^{-1})			Molar Intensities (${}^{\mathrm{km}}\!/\!{}_{\mathrm{mol}}$)		
	QCISD/	CCSD(T)/	CCSD(T)/	QCISD/	CCSD(T)/	CCSD(T)/
	6-311++G**	6-311++G**	aug-cc-PVTZ	6-311++G**	6-311++G**	aug-cc-PVTZ
HON	3778.2	3744.7	3699.6	75.2	68.7	81.0
	1231.8	1220.6	1231.7	2.69	1.60	4.5
	1141.5	1127.5	1118.1	181.5	183.9	196.0
HO^{15}N	3778.2	3744.3	3699.6	75.2	70.4	81.0
	1222.3	1210.6	1222.8	9.48	11.6	10.9
	1128.0	1114.6	1104.3	169.4	163.8	180.1
DON	2750.4	2725.6	2692.4	39.9	37.8	41.7
	1169.7	1158.8	1150.6	94.8	98.5	86.4
	895.9	884.9	892.5	49.1	49.4	56.6
DO^{15}N	2750.4	2725.6	2692.4	39.9	37.8	41.7
	1147.9	1137.1	1129.0	92.6	96.1	84.1
	894.0	883.1	890.7	48.6	48.8	56.2

The Dunning basis set clearly has an effect on the molar absorptivity values, raising them when compared to the Pople basis set with the exception of mode 2 of DON and DO^{15}N, which decreased instead. This is rather intriguing as mode 2 is the 100% intensity mode for the deuterium isotope according to Maier's experiments, and when comparing our own molar absorptivity values. Still, whether the cc-pVTZ calculations provide more accurate data is debatable as there are no experimental values to compare to; at best, ratios of the molar absorptivity values can be compared to integrated intensities of experimental peaks. However, we can hope that if the frequency values are closer to those of experiment, then the molar absorptivity values may be closer to the true value as well.

Regarding the isotopic substitutions, changing N for ^{15}N had no effect on the vibrational frequencies of mode 1 − which makes sense because it is the H-O stretch − and caused inconsistent fluctuations for modes 2 and 3. Swapping hydrogen for deuterium decreased the vibrational frequency of mode 1 by approximately 1000 cm^{-1}, and increased the separation between modes 2 and 3, hopefully decoupling them. We have yet to further pursue this phenomenon, however.

3.3.3 Comparison with Anderson's Values

The vibrational frequency values measured by the Anderson group each have an associated response peak of some absorption intensity; as explained in Equation 1.1, a ratio of these integrated absorbance intensities is proportional to the ratio of their corresponding molar absorptivity coefficients, thus comparisons between calculations and experiment can be done for the frequencies and molar absorptivity ratios as seen in Table 3.4.

Since mode 2 has such a minimal response as to be unmeasurable, there is no experimental frequency value ν_2, and the only integrated absorbance intensity ratio is between mode 3 and mode 1; we still include the calculated mode 2 values here for the sake of completeness. When comparing accuracy between the methods QCISD and CCSD(T) for a given basis set, CCSD(T) calculates closer vibrational frequencies while QCISD yields a closer molar absorptivity ratio, which is inconclusive as to which method is more accurate. A comparison

Table 3.4: Comparison between our calculated frequencies and experimentally measured frequencies. [20] Bottom line is a ratio between molar absorptivities for calculated values, and integrated absorbance intensities for experimental values.

	QCISD/ 6-311++G**	CCSD(T)/ 6-311++G**	CCSD(T)/ aug-cc-pVTZ	Experiment
ν_1	3778.24	3744.68	3699.61	3475.85
ν_2	1231.80	1220.64	1231.66	—
ν_3	1131.51	1127.48	1118.12	1098.84
$^{\nu_3}/_{\nu_1}$	2.41	2.67	2.42	2.38

between basis sets 6-311++G** and aug-cc-pVTZ for the method CCSD(T), however, shows the Dunning basis set to be more accurate for both vibrational frequencies and molar absorptivity ratio. Additionally, the molar absorptivity ratio calculated by CCSD(T)/aug-cc-pVTZ is extremely close to that of QCISD/6-311++G**. From these comparisons, we can somewhat confidently conclude that the more rigorous method and higher order basis set are more accurately describing the vibrational states of HON.

However, 'more accurately' does not mean that this description is *sufficiently* accurate. A plot of the harmonic energy functions for each normal mode was overlaid with the corresponding *ab initio* energy values in Figure 3.4. While modes 2 and 3 have some harmonic curvature apparent, the offset between the calculated energy values and the harmonic energy function is still significant for points past $q = \pm 2$. By contrast, mode 1 is clearly not parabolic and not at all well-described by the Double Harmonic Approximation.

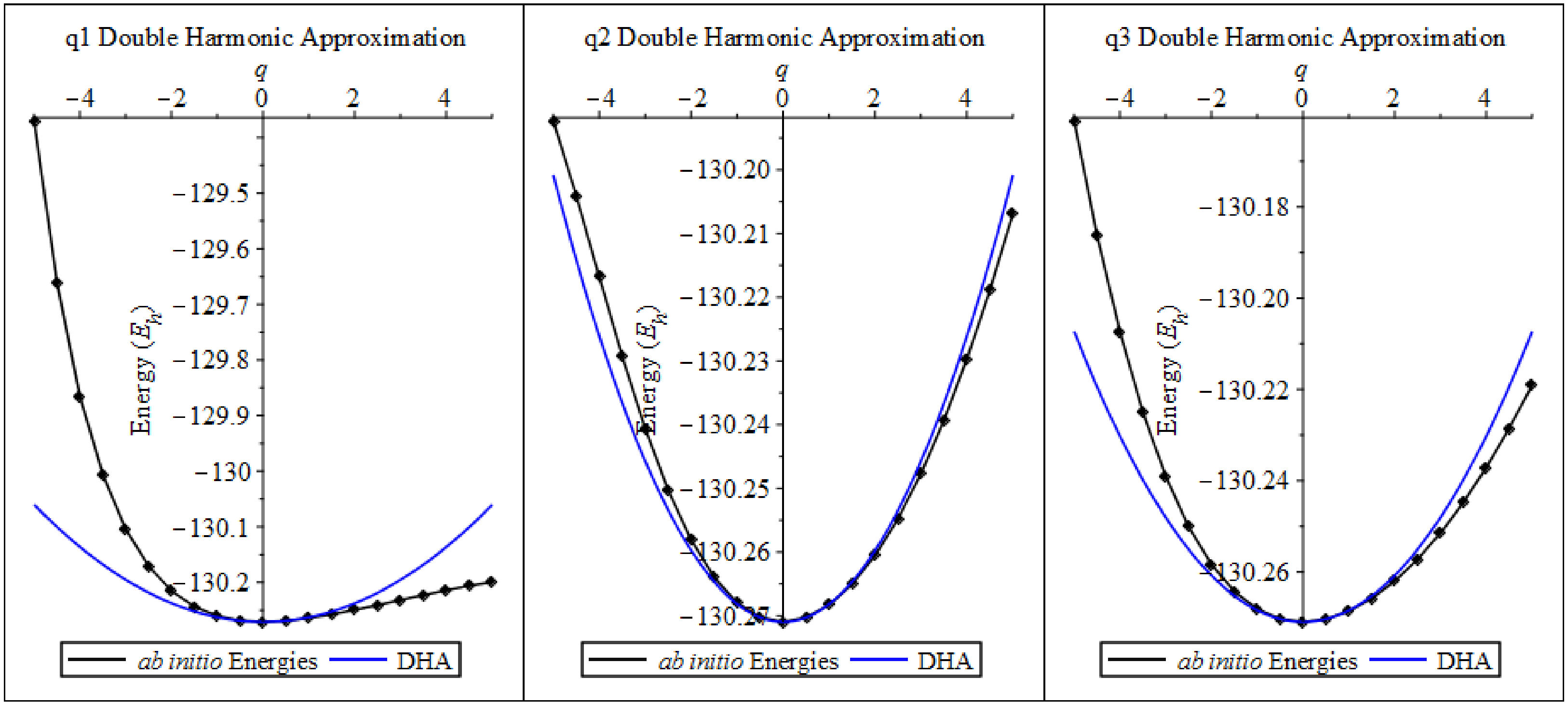


Figure 3.4: The *ab initio* energies (black) and quadratic curves from the Double Harmonic Approximation (blue) as a function of normal mode displacement. While modes 2 (center) and 3 (right) have recognizable harmonic curvature, mode 1 (left) is clearly not well-described by the DHA.

3.4 Conclusions about DHA

The Double Harmonic Approximation can be a good starting point for approximating the molar absorptivity values of a molecule's normal modes. However, while the harmonic energy functions and dipole moment cubic fits for ^{3}HON are in excellent agreement with the *ab initio* data points for the $q = \pm 0.5$ displacements in Figures 3.4 and 3.5, any values of q beyond those limits display some amount of error, depending on the vibrational mode. This is especially prevalent for the potential energy curve of mode 1, and the x- and y-dipole moments of mode 3.

It is clear that higher order polynomials must be employed to describe the system and minimize this error. This is best indicated by mode 1, whose potential energy surface has clear anharmonic curvature; all terms included in the potential energy function beyond the quadratic are anharmonic adjustments. The next concern is, what approach should be taken in order to calculate these anharmonic corrections?

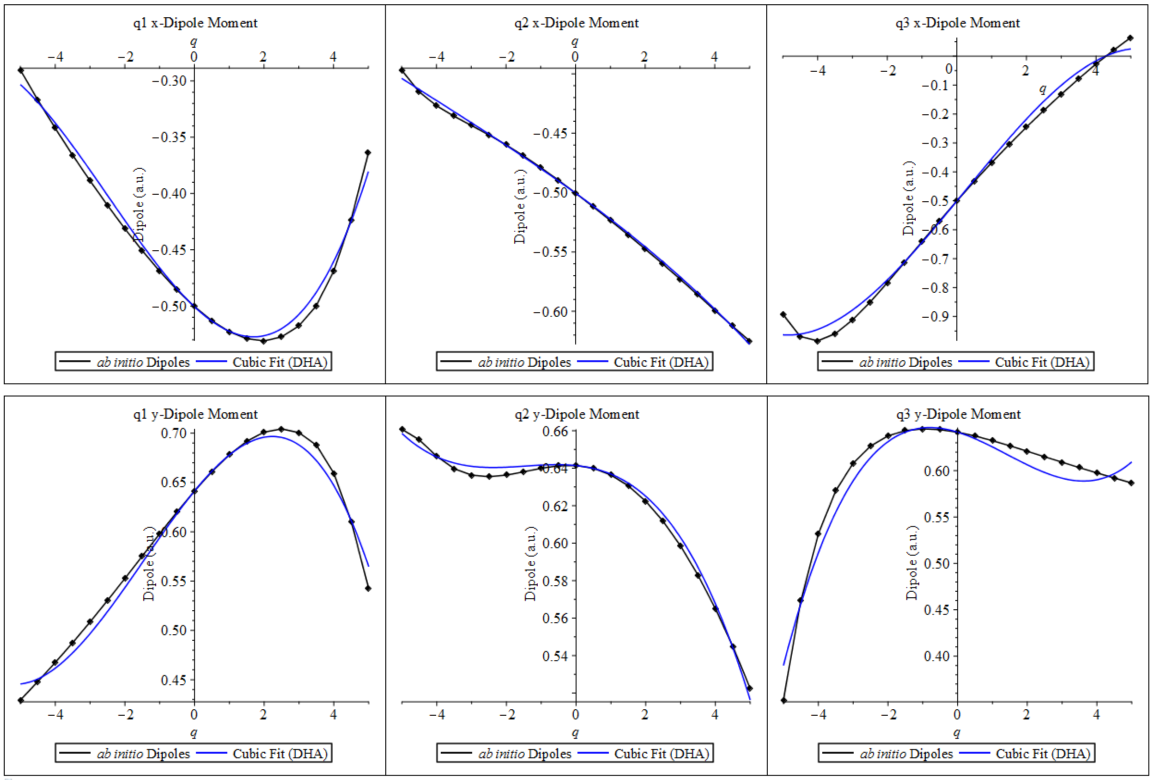


Figure 3.5: The *ab initio* calculated dipole moments (black) and cubic fit equations (blue) used for the Double Harmonic Approximation as a function of normal mode displacement. Top row are dipole moments in the x-direction, bottom row are in the y-direction.

Chapter 4

Beyond the Double Harmonic Approximation

4.1 Anharmonicity

The potential energy curve in Figure 4.1 is an example of an anharmonic oscillator, with a steep rise in energy as the atoms draw closer and a slower rise that levels off as the atoms move apart. As mentioned in Chapter 3, the Double Harmonic Approximation does accurately describe a small portion of the curvature near the origin, and so the potential energy equation of the harmonic oscillator can be used as a starting model. [16] The corrections added on to this model take the form of higher order polynomial terms, whose coefficients can be included in the Hamiltonian operator to obtain the energy levels of the anharmonic oscillator.

As the system is no longer a harmonic oscillator, the energy levels are no longer exactly $\hbar\omega$ apart; instead, as the energy of the system increases the separation between the energy levels decreases. This means that ladder operators cannot be applied to the wavefunctions of the anharmonic oscillator to determine transition probability as they had been with the Double Harmonic Approximation. One of our methods for obtaining the anharmonic molar absorptivities requires explicitly calculating this value and so this issue will be addressed at that point, as well as how it is overcome.

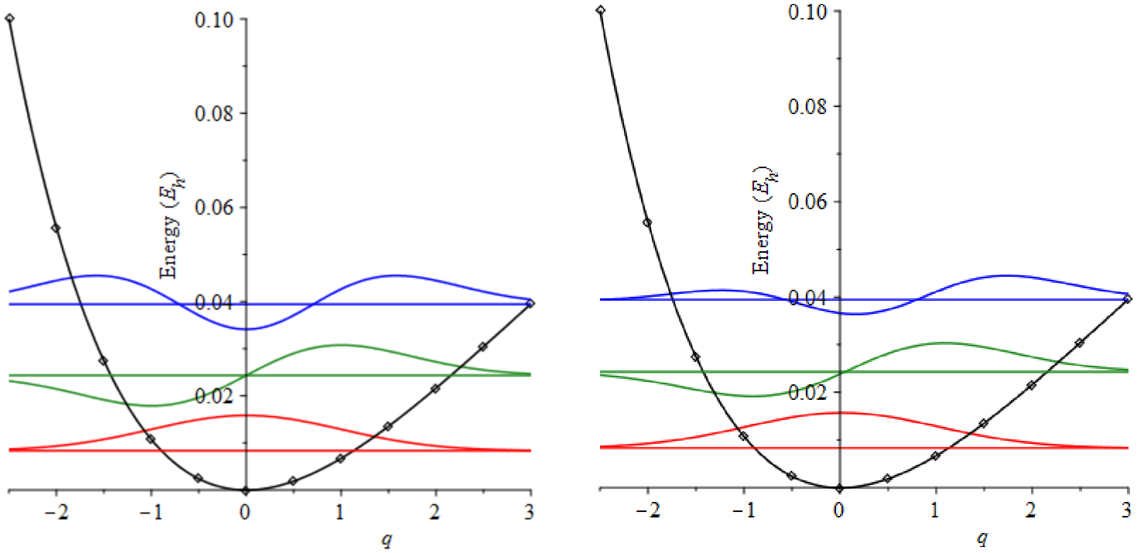


Figure 4.1: A plot of the potential energy as a function of mode 1 for ^{3}HON. Left: harmonic wavefunctions of n=0, n=1, n=2 energy levels; the n=1 and n=2 wavefunctions extend far outside the limits of the potential energy curve on the left. Right: anharmonic wavefunctions of n=0, n=1, n=2 energy levels; linear combinations of harmonic wavefunctions modify the n=1 and n=2 anharmonic wavefunctions to better lie within the limits of the potential energy curve, and thus are more accurate. This is most clearly seen at the n=2 energy level, but is also somewhat apparent at n=1.

4.2 Coupling

While ideally each vibrational mode is distinct and not interacting with other vibrational modes of the molecule, this is unlikely when two modes are close in energy. This interaction between the modes means that it is difficult to separate a specific vibrational movement and assign it to one energy level, for example; the total potential energy of the system is also different from what it would be were the modes non-interacting.

The HON molecule has already provided a reason for including coupling through the literature: while two of the papers (Anderson [20] and Maier [15]) discussed in Chapters 1 and 2 refer to vibrational mode 2 as the O-N stretch and vibrational mode 3 as the bend of the molecule, another (Lee [14]) switches those designations so mode 2 is the bend and mode 3 is the O-N stretch. Our own observations of the vibrational modes yielded inconclusive results as the two modes appeared to experience both movements simultaneously. As these two modes are only 100 cm^{-1} apart in energy, it is plausible that coupling is a contributor to the system and should be considered accordingly when calculating the energy.

$$V(q_1, q_2, q_3) = V_1(q_1) + V_2(q_2) + V_3(q_3) + V_{12}(q_1, q_2) + V_{13}(q_1, q_3) + V_{23}(q_2, q_3) + V_{123}(q_1, q_2, q_3) \tag{4.1}$$

Coupling between each pair of vibrational modes, as well as all three modes simultaneously, should be included to fully model the system; however three-way coupling can be extremely difficult to calculate. The results presented here currently only take two-way coupling into account.

4.3 Vibrational 2nd Order Perturbation Theory (VPT2)

Vibrational second-order perturbation theory (VPT2) is an approximation method for the Schrodinger equation that assumes the potential energy surface of the system can be expressed as a quartic polynomial with respect to the coordinates of the vibrational modes. [25, 2]

$$V(q) = \frac{1}{2}\sum \omega_i q_i^2 + \frac{1}{3!}\sum \phi_{ijk} q_i q_j q_k + \frac{1}{4!}\sum \phi_{ijkl} q_i q_j q_k q_l \tag{4.2}$$

The first term describes the harmonic portion of the potential, while the cubic and quartic terms describe the anharmonic effects; ϕ_{ijk} and ϕ_{ijkl} are the third- and fourth-order derivatives of the potential energy with respect to displacement of the normal mode q. The total Hamiltonian of the system is therefore also a sum of the harmonic approximation of the system and additional cubic and quartic corrections, referred to as perturbations (Equation 4.3).

$$\hat{H} = \hat{T} + \frac{1}{2}\sum \omega_i q_i^2 + \frac{1}{3!}\sum \phi_{ijk} q_i q_j q_k + \frac{1}{4!}\sum \phi_{ijkl} q_i q_j q_k q_l \tag{4.3}$$

The harmonic approximation of the system is known as the unperturbed Hamiltonian operator $\hat{H}^{(0)}$ [16] and includes the first two terms of Equation 4.3, while the cubic terms are included in the first-order Hamiltonian $\hat{H}^{(1)}$ and the quartic terms in the second-order

Hamiltonian $\hat{H}^{(2)}$, as seen in Equation 4.4. [25]

$$\hat{H}^{(0)} = \hat{T} + \frac{1}{2}\sum \omega_i q_i^2 \quad \hat{H}^{(1)} = \frac{1}{3!}\sum \phi_{ijk} q_i q_j q_k \quad \hat{H}^{(2)} = \frac{1}{4!}\sum \phi_{ijkl} q_i q_j q_k q_l \tag{4.4}$$

The cubic and quartic force constants ϕ are calculated from numerical derivatives, which requires some small displacement value h to be defined; the third- and fourth-derivative equations of the finite difference method can be found in the Appendix. Literature includes various suggestions for the magnitude of this variable: some quantum chemical software default to 0.047 a_0 [1, 2], while some authors suggest ranges between 0.006 to 0.02 a_0 [2], or the step size of each mode being proportional to the square-root of the harmonic frequency [24], and others recommend step sizes ranging from 0.12 a_0 to 0.2 a_0 depending on the degree of derivative. [25]

4.3.1 Applying VPT2

The frequencies and molar absorptivities of HON, $HO^{15}N$, DON, and $DO^{15}N$ were calculated using VPT2 by first obtaining 57 *ab initio* energies for use with the finite difference method. A total of 19 different displacement components were used, ranging between 0.1Åto 0.5Åin increments of 0.025Åin order to determine the best displacement value. Once the cubic and quartic force constants had been calculated, they were plugged into the following equations [25] along with the harmonic frequencies (ω) to solve for the anharmonic constants (units of cm^{-1}). The 'on-diagonal' terms (Equation 4.5) are those whose quadratic force constants were taken relative to the same vibrational mode, and so would lie along the diagonal of a matrix, while the opposite is true for the 'off-diagonal' terms (Equation 4.6). [4]

$$16\chi_{ii} = \phi_{iiii} - \sum_j \frac{\left(8\omega_i^2 - 3\omega_j^2\right)\phi_{iij}^2}{\omega_j\left(4\omega_i^2 - \omega_j^2\right)} \tag{4.5}$$

$$4\chi_{ij} = \phi_{iijj} - \sum_k \frac{\phi_{iik}\phi_{jjk}}{\omega_k} + \sum_k \frac{2\omega_k\left(\omega_i^2 + \omega_j^2 - \omega_k^2\right)\phi_{ijk}^2}{\Delta_{ijk}}^{\dagger}$$

$$\Delta_{ijk} = (\omega_i + \omega_j - \omega_k)(\omega_i + \omega_j + \omega_k)(\omega_i - \omega_j + \omega_k)(\omega_i - \omega_j - \omega_k) \quad (4.6)$$

[†] Rotational term is excluded because rotational corrections are not being considered at this time.

These six resultant anharmonic constants ($\chi_{11}, \chi_{22}, \chi_{33}, \chi_{12}, \chi_{13}$, and χ_{23}) were used to calculate the three anharmonic vibrational frequencies of the molecule:

$$\nu_i = \omega_i + 2\chi_{ii} + \frac{1}{2}\sum_{i\neq j}\chi_{ij} \quad (4.7)$$

A similar process was followed to determine the anharmonic transition dipole moments of each vibrational mode, although the equation is too long to place here and therefore can be found in the Appendix. [23] However, the equation required also calculating the linear, quadratic, and cubic derivatives of the dipole moments in the x- and y-directions with respect to the normal modes. These values, the previously calculated quadratic, cubic, and quartic force constants, and the harmonic frequencies were all used to obtain two anharmonic transition dipole moments: one in the x-direction, and the other in the y-direction. These values were plugged into Equation 3.5 for G_{ji} from Chapter 3 and, with the corresponding anharmonic frequency values, used to obtain the anharmonic molar absorptivities of the vibrational modes.

4.3.2 Results of VPT2

Of the 19 displacement values used, the two smallest values of 0.1 and 0.125Å yielded results that were abnormal when compared to the larger displacement values. As the displacement values continued to increase, the frequency and molar absorptivity values both decreased at a roughly linear rate and showed no sign of leveling off, which was concerning. However, there was a slight plateau around 0.3Å, and so all numerical results referred to were calculated using the 0.3Å displacement.

Table 4.1: A comparison of our VPT2 anharmonic frequencies to those calculated by Bozkaya. [5]

	Ours	Bozkaya
HON	3492.44	3523
	1194.68	1215
	1086.87	1108
DON	2563.6	2605
	1125.16	1156
	869.80	876

Before comparing the results of VPT2 to other models, they were first checked against the anharmonic frequencies calculated by Bozkaya. [5] While both sets of calculations used the CCSD(T) level of theory, Bozkaya used a higher level of basis set, cc-pCVQZ, that additionally differs from aug-cc-pVTZ in that it contains core-valence functions but no diffuse functions. It is interesting to note that our calculated anharmonic frequencies are closer to those of experiment than Bozkaya's, despite using a smaller basis set, as seen in Table 4.1. However, it is likely this is due to the lack of diffuse functions included in the cc-pCVQZ calculations which appear important to include due to some of ^{3}HON's unique characteristics.

When comparing the results obtained using VPT2 to values previously discussed, the anharmonic frequencies calculated using VPT2 decreased relative to the harmonic frequencies calculated using the Double Harmonic Approximation, but drew much closer to the experimental values measured by the Anderson group [20] and Maier [15] as can be seen in Table 4.2. VPT2 does seem to over-correct the lower frequency modes however, as mode 3 of HON and mode 2 of DON both changed from being above or approximately equal to the experimental measurements to between 10-25 cm^{-1} below. A comparison with the molar absorptivity ratios of Maier was not done as the intensities of Maier's spectra were generated through isomerization rather than a product-limited reaction.

While calculating the frequencies and molar absorptivities was a good comparison to determine if the method was doing better than the Double Harmonic Approximation, we were also able to check the quality of model another way. Since the cubic and quartic force

Table 4.2: A comparison between the vibrational frequencies and molar absorptivities calculated using the Double Harmonic Approximation (DHA) and VPT2. VPT2 values given were calculated using a displacement value of 0.3Å.

	Frequencies (cm^{-1})			Molar Abs. (km/mol)	
	DHA	VPT2	Maier (Exp)	DHA	VPT2
HON	3699.61	3492.43	3467.2	80.958	62.533
	1231.66	1194.68	—	4.5217	1.9321
	1118.12	1086.87	1095.6	196.01	168.17
$HO^{15}N$	3699.61	3493.17	3466.4	80.972	62.500
	1222.84	1185.34	—	10.953	7.1552
	1104.27	1075.39	1085.6	180.11	160.49
DON	2692.39	2563.6	2592.95	41.718	34.917
	1150.56	1125.16	1149.0	86.423	82.659
	892.46	869.80	868.8	56.582	53.176
$DO^{15}N$	2692.39	2583.02	2564.6	41.712	34.921
	1128.95	1104.41	1127.9	84.094	80.445
	890.67	868.14	867.2	56.204	52.887

constants were calculated using the finite difference method, we substituted the on-diagonal terms into a quartic polynomial for each vibrational mode and plotted the equation against the *ab initio* energies used to obtain the force constants originally; this process was repeated using the dipole moment derivatives and the *ab initio* obtained dipoles. Each of the 19 of the displacement values was plotted for the potential energy and x- and y-dipole moments in Figure 4.2, although there was little variation between the functions.

Just as with the DHA, not all of the data points are modeled by the polynomial. The fit is much better since the anharmonic terms are included, but begins to diverge from the data points for q values of ±4 and beyond when plotting the potential energy. The dipole moment plots show even more deviation, with the y-dipole moments of all three modes ceasing to fit well beyond ±3. The lack of fit for the x-dipole moments is not nearly as blatant, but is still obvious. Since the quartic polynomial does not model the potential energy curve well, it is clear that higher order anharmonic corrections are required, which means going beyond VPT2.

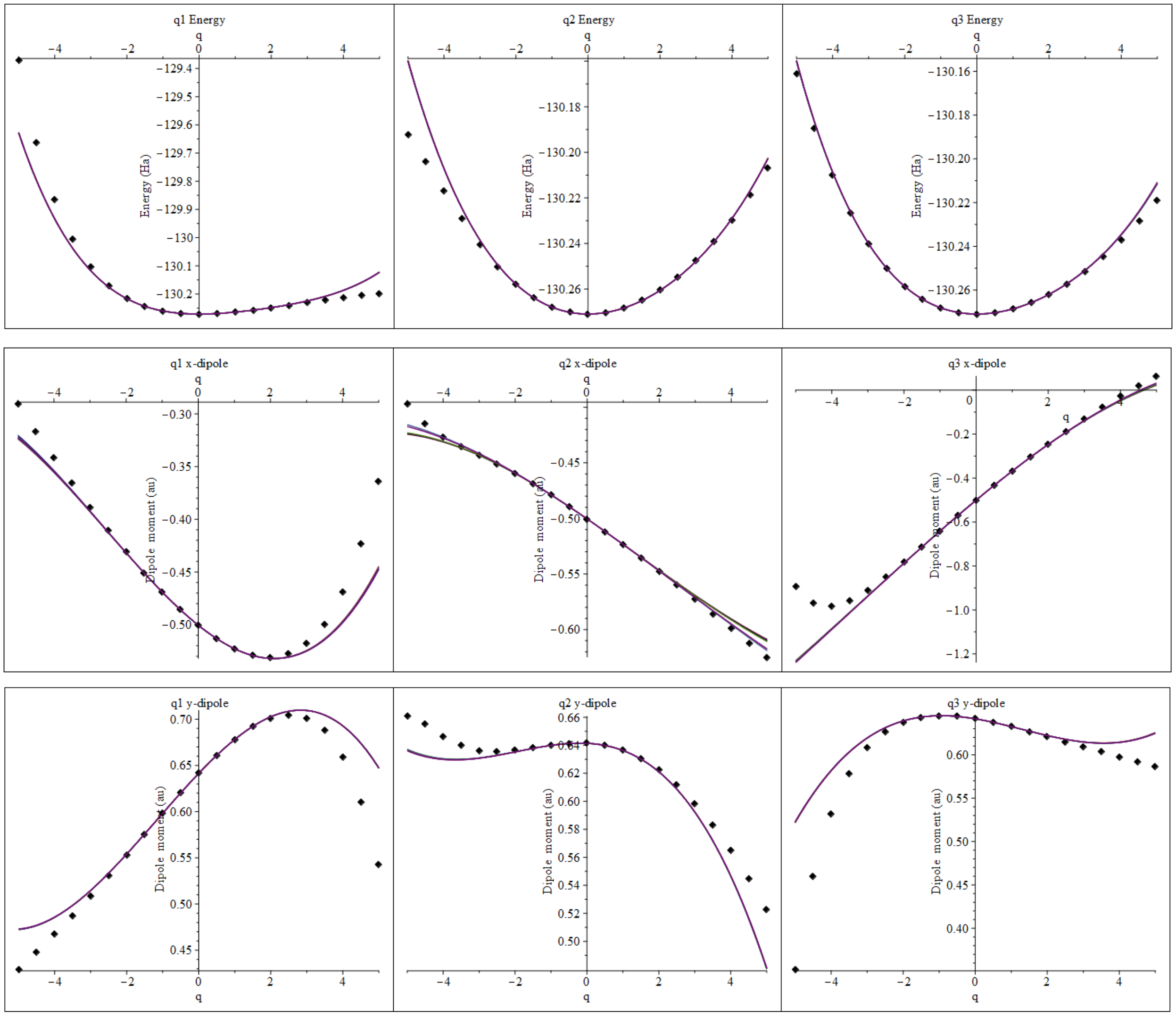


Figure 4.2: Top: VPT2 calculated polynomials plotted against *ab initio* energies (points). Center and Bottom: VPT2 calculated polynomials plotted against *ab initio* obtained dipoles (points), for the x-dipole (center) and y-dipole (bottom) components. Only on-diagonal terms were considered and plotted against their respective modes in each case.

4.4 Variational Method: Fitting of *ab initio* Energies

While the application of perturbation theory allowed for the inclusion of anharmonicity in the system, the 4^{th} order polynomial was not sufficiently accurate. In order to use higher order polynomials to model these vibrational modes and calculate their frequencies and molar absorptivities, a different approximation method is required. The variational method approximates the system with a trial wavefunction, then minimizes the energy with respect to the variational parameters. The energy of this trial wavefunction will always be higher than the true energy of the system, and any variations made that lower the resultant energy are closer to the exact answer. [Sherrill]

Considering these possible higher order terms, the potential energy equation describing this system has the following form:

$$V(q) = V_0 + c_2q^2 + c_3q^3 + c_4q^4 + c_5q^5 + ... + c_nq^n \tag{4.8}$$

The corresponding Hamiltonian operator describing this system is expanded to the same power, where the coefficients of the potential energy equation are also those of the Hamiltonian operator:

$$\hat{H} = \hat{T} + c_2\hat{q}^2 + c_3\hat{q}^3 + c_4\hat{q}^4 + c_5\hat{q}^5 + ... + c_n\hat{q}^n \tag{4.9}$$

To obtain these constants for the Hamiltonian, we calculate *ab initio* potential energy surfaces as a function of each individual vibrational mode, as well as two-dimensional energy surfaces that included vibrational mode coupling. A polynomial equation is then fit to the resultant curve, and the coefficients extracted for use with each respective Hamiltonian; however, since each set of coefficients describes a specific set of conditions regarding which vibrational modes are active or frozen, each Hamiltonian operator is also bound by those same conditions. The summation of these partial descriptions then forms a final operator that describes the entirety of the system.

If coupling is not being considered as part of the system, the overall Hamiltonian matrix would consist of the harmonic oscillator, with the additional terms describing the independent anharmonic contributions arising from modes 1, 2, and 3.

$$\hat{H}_{nc} = \hat{H}_0 + \hat{H}(q_1) + \hat{H}(q_2) + \hat{H}(q_3) \tag{4.10}$$

To calculate the energies of the system when coupling is considered, the Hamiltonian matrices with vibrational modes 1 and 2, 1 and 3, and 2 and 3 simultaneously active are included. The coefficients of these operators are derived from a two-dimensional fit, considering each possible combination where both modes contribute.

$$\hat{H}_c = \hat{H}_0 + \hat{H}(q_1) + \hat{H}(q_2) + \hat{H}(q_3) + \hat{H}(q_1, q_2) + \hat{H}(q_1, q_3) + \hat{H}(q_2, q_3) \tag{4.11}$$

4.4.1 Linear Variation Method

The linear variation method defines the trial wavefunction ψ as a linear combination of basis functions ϕ. [16] As the anharmonic oscillator being considered is a correction to the harmonic oscillator, its trial wavefunction can be defined as a linear combination of harmonic wavefunctions. Therefore, the anharmonic wavefunction of the system $\psi(q_1)$ can be rewritten as $\psi(q_1) = \sum_{\nu_1=0}^{i} c_{\nu_1}\phi_{\nu_1}(q_1)$. When this expansion is continued one step further to a wavefunction dependent upon all three vibrational modes:

$$\psi(q_1, q_2, q_3) = \sum_{\nu_1=0}^{i}\sum_{\nu_2=0}^{j}\sum_{\nu_3=0}^{k} c_{\nu_1,\nu_2,\nu_3}\phi_{\nu_1}(q_1)\phi_{\nu_2}(q_2)\phi_{\nu_3}(q_3) \tag{4.12}$$

The number of harmonic wavefunctions considered for each vibrational mode are denoted by the integers i, j, and k, and so the total number of terms in the linear combination of $\psi(q_1, q_2, q_3)$ is $(i+1) \times (j+1) \times (k+1)$. Since the Hamiltonian is operating on harmonic wavefunctions, the terms of its expansion once again consist of raising and lowering operators (Equation 3.9). A symbolic example of this application is shown in Equation 4.13, yielding

$\hat{q}^2$	0	1	2	3	4
0	**X**		**X**		
1		**X**		**X**	
2	**X**		**X**		**X**
3		**X**		**X**	
4			**X**		**X**

Figure 4.3: Example matrix of the $\hat{q}^2$ operator for harmonic energy levels 0 through 4. Rows are the initial energy level (ν_x) and columns are the final energy level (ν'_x). **X** marks any combinations that yield a non-zero value.

a linear combination of the coupling between the initial and final energy levels n and m.

$$\langle m|\hat{q}^2|n\rangle = \frac{1}{2}\left[\sqrt{n^2 - n}\delta_{m,n-2} + (2n+1)\delta_{m,n} + \sqrt{(n+1)(n+2)}\delta_{m,n+2}\right] \tag{4.13}$$

The application of the $\hat{q}^2$ operator to a harmonic energy level will only yield a non-zero value if the final energy level is -2, 0, or $+2$ relative to the initial. For a system that considers the harmonic energy levels from 0 to 4, for example, this would mean that only the marked elements in the $\hat{q}^2$ matrix (Figure 4.3) contain a non-zero value.

Since the anharmonic wavefunction that $\hat{q}^n$ is being applied to consists of the product of three harmonic wavefunctions, the dimensions of the q matrices are $(i+1) \times (j+1) \times (k+1)$ and each overall matrix element is the product of three separate matrix elements. A symbolic form of this is shown in Equation 4.14, where $n = n_1 + n_2 + n_3$ for the powers of $\hat{q}$.

$$\langle \nu'_1\nu'_2\nu'_3|\hat{q}^n_{1,2,3}|\nu_1\nu_2\nu_3\rangle = \langle \nu'_1|\hat{q}_1^{n_1}|\nu_1\rangle\ \langle \nu'_2|\hat{q}_2^{n_2}|\nu_2\rangle\ \langle \nu'_3|\hat{q}_3^{n_3}|\nu_3\rangle \tag{4.14}$$

An explicit example of Equation 4.14 is written below, for a single matrix element of the $\hat{q}^4$ operator, with i, j, and k each equal to 4.

$$\langle 3|\hat{q}_1^2|1\rangle\ \langle 2|\hat{q}_2^2|0\rangle\ \langle 1|\hat{q}_3^0|0\rangle \tag{4.15}$$

While the first two terms of Equation 4.15 are non-zero, the third term $\langle 1|\hat{q}_3^0|0\rangle$ is does not result in equivalent energy levels and therefore the value of the overall matrix element is

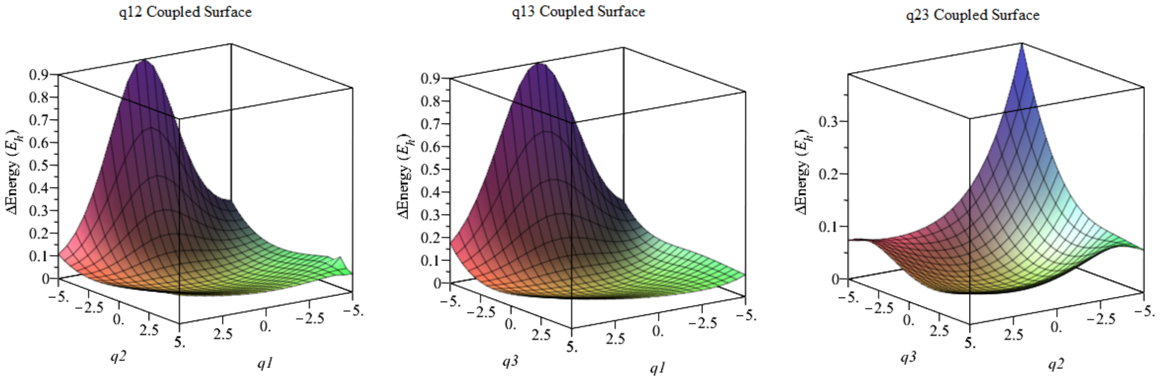


Figure 4.4: The potential energy surfaces of the coupling between modes 1 and 2 (left), 1 and 3 (center), and 2 and 3 (right).

zero; the term would be located in the 2^{nd} row $(1 \times 1 \times 2)$ and 24^{th} column $(4 \times 3 \times 2)$ of the 125×125 matrix $(5 \times 5 \times 5)$. The optimal number of energy levels i, j, and k to include in the approximate wavefunction is determined later in this chapter.

4.4.2 Fitting the Uncoupled and Coupled Systems

In order to determine the importance of including coupling terms in the calculations, energies were calculated both with and without coupling terms included. The non-coupling or uncoupled system included terms that were dependent upon a single vibrational mode, while the coupled terms required single-point energy calculations where two vibrational modes were modified to obtain a two-dimensional potential energy surface as a function of both normal modes.

To determine the most accurate polynomial fit to use when describing the system, a series of increasingly higher-order polynomial equations were applied to the *ab initio* energies of each vibrational mode, and the root mean-squared error (RMSE) values collected. Table 4.3 shows that a 4^{th} order fit, as was used in VPT2, proves to be an extremely poor descriptor as the RMSE value of mode 1 is larger than the value of the energy level itself, 3699 cm^{-1}, thus reinforcing the need for a higher-order fit. A ‘convergence’ in the RMSE values begins with the 10^{th} order for mode 1, and the 8^{th} order for modes 2 and 3; therefore a 10^{th} order polynomial fit is required to best describe the system. This conclusion is also applied to the two-dimensional coupling equations.

For the two-dimensional equations, only the difference between the coupled and uncoupled potential energy surfaces for each pair of vibrational modes was used to avoid fitting extraneous amounts of data. Figure 4.5 contains a visual representation of this for each coupled pair, where the first plot of each row in the figure is a summation the energies of two non-coupled vibrational modes (such as in the first row of Figure 4.6) plotted at all combinations of displacement. The final result is a set of points with values of zero along the x- and y-axes, and the remaining values consisting of the correction to the energy when coupling between those two modes is considered. This also prevents the two-dimensional equation from having to consider any ‘diagonal’ non-coupling polynomial terms, as those are found by fitting the relevant non-coupled vibrational mode.

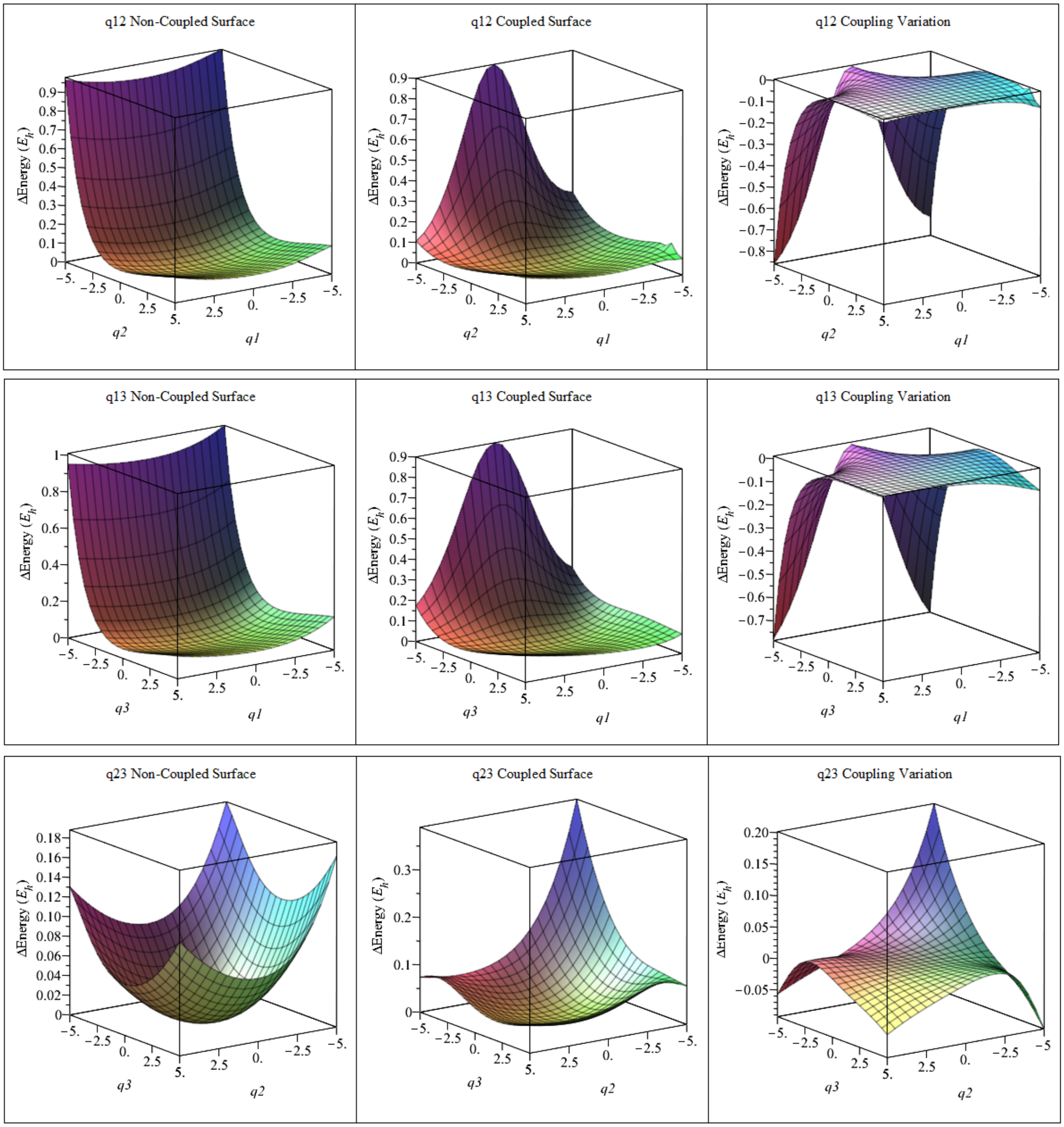
The constants obtained by the fitted equations in Figure 4.6 are applied to the corresponding $\hat{q}$ matrices in the Hamiltonian operators of Equations 4.10 and 4.11, so that the final Hamiltonian matrix can be obtained. However, this first requires that a trial wavefunction for the linear variation method be decided upon so the $\hat{q}$ operator to be applied.

4.4.3 Degree of Variation

As the number of harmonic wavefunctions included in the linear combination of Equation 4.12 increases, the energies obtained by trial wavefunction should converge towards a lowest

Table 4.3: Root Mean-Squared Error (RMSE) of energies (cm^{-1}) for uncoupled modes. Bolded values show where RMSE stopped decreasing by approximately an order of magnitude when fitting polynomial increased.

	q1	q2	q3
4^{th} order	4003.2	356.77	96.304
5^{th} order	1638.6	241.20	98.941
6^{th} order	296.86	12.465	8.5598
8^{th} order	18.087	**1.6857**	**0.77050**
10^{th} order	**1.4155**	1.1134	0.34757
12^{th} order	1.2182	0.95763	0.26636



q12 Non-Coupled Surface

q12 Coupled Surface

q12 Coupling Variation

q13 Non-Coupled Surface

q13 Coupled Surface

q13 Coupling Variation

q23 Non-Coupled Surface

q23 Coupled Surface

q23 Coupling Variation

Figure 4.5: The potential energy surfaces of the coupled modes 1 and 2 (top), 1 and 3 (middle), and 2 and 3 (bottom). The left column shows the summation of the energy surfaces of the individual modes at each combination of displacements; the center column contains *ab initio* coupled energy surface; the surface in the right column is plotted using the difference between the center and left columns, and the coupling polynomial is fit to these data points.

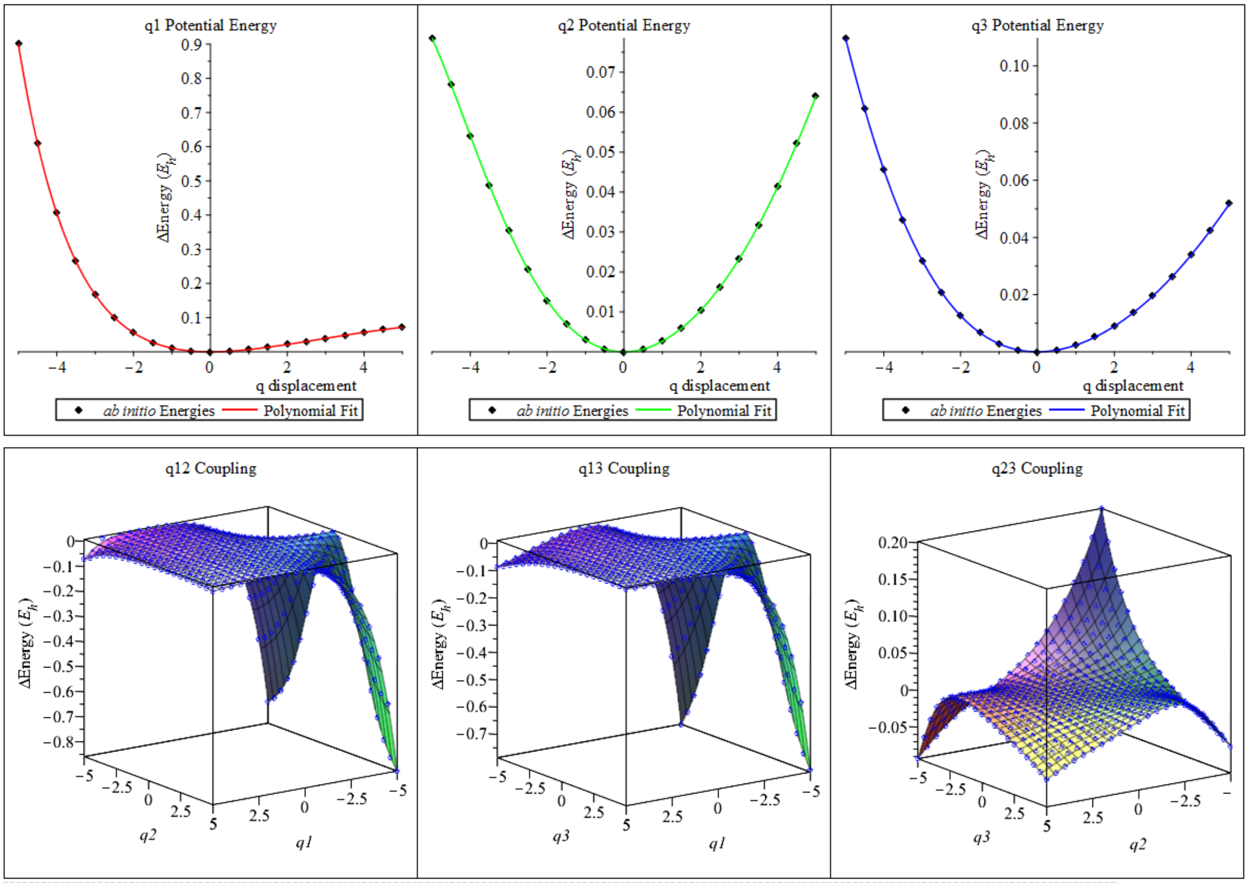


Figure 4.6: The non-coupling (top) and coupling (bottom) fitted potential energy surfaces. The blue dots on the coupling plots are the data points used to fit the two-dimensional polynomial equation.

value. These energy values were obtained by diagonalizing the Hamiltonian matrix, which resulted in a set of eigenvalues and their corresponding eigenvectors. Since there are three vibrational modes, there are three independent degrees of variation to take into account. In order to determine the lowest number of harmonic wavefunctions that should be included while still modeling the system accurately, the Hamiltonian matrix of each separate mode was diagonalized. From these we obtained the ground- and first-excited state energies of each vibrational mode, for a sequentially increasing number of included harmonic energy levels.

The bolded values in Table 4.4 denote the relevant number of energy levels included for each mode before some degree of convergence was reached. As our initial calculations for the linear variation method restrict all three modes to the same final energy level, levels 0 → 8 is the target set of harmonic wavefunctions for linear combination.

4.4.4 Calculated Frequencies and Molar Absorptivities

The notation used in this section is as follows: the maximum energy level included for a vibrational mode refers to all calculations completed with that set of harmonic wavefunctions, and a set of three integers denotes the maximum energy level for modes 1, 2, and 3 in that order. For example, the numbers '5,4,4' would refer to mode 1 including levels 0 → 5, and

Table 4.4: Frequency values (cm^{-1}) individually calculated for the non-coupled modes using the 10th order polynomial fit. Increasing numbers of harmonic energy levels, beginning from the ground-state, were included to see how many terms were required before the energy converged.

ν_n	q1	q2	q3
0 → 4	3533.56	1242.742	1126.896
0 → 5	3514.51	1242.744	1126.875
0 → 6	3512.60	1242.735	1126.867
0 → 7	3511.09	**1242.734**	**1126.864**
0 → 8	**3510.36**	1242.734	1126.864
0 → 9	3510.23	1242.734	1126.864

Table 4.5: All values were calculated using CCSD(T)/aug-cc-pVTZ. The harmonic values were calculated using the Double Harmonic Approximation from Chapter 3. The 'Converged to' row contains the significant figures that had ceased changing as the number of included harmonic levels increased.

Non-coupled	Frequencies (cm^{-1})			Molar Absorptivities ($^{km}/_{mol}$)		
	Mode 1	Mode 2	Mode 3	Mode 1	Mode 2	Mode 3
Harmonic	3699.6097	1231.6635	1118.1235	80.958	4.5217	196.011
2,2,2	3649.1360	1243.2526	1128.0374	57.541	5.0105	164.361
3,3,3	3579.1434	1243.2883	1128.4002	57.809	5.0327	165.759
4,4,4	3533.5693	1242.7421	1126.8955	56.715	5.0314	165.645
5,5,5	3514.5076	1242.7435	1126.8754	56.015	5.0312	165.619
6,6,6	3512.6042	1242.7352	1126.8675	55.900	5.0311	165.621
7,7,7	3511.0856	1242.7342	1126.8642	55.825	5.0311	165.621
8,8,8	3510.3560	1242.7342	1126.8642	55.788	5.0311	165.621
Converged to	3510	1242.7342	1126.8642	55.	5.0311	165.621

modes 2 and 3 each with levels 0 → 4.

As the degree of variation included for each vibrational mode had been determined individually, we repeated the calculations using the sum of the three non-coupled Hamiltonian matrices to obtain the frequencies of each mode relative to the system. The maximum energy level included for each mode began at 2, and continued sequentially up to 8. For these calculations, the maximum energy level of the modes were kept equivalent so that i=j=k. The results of these calculations can be found in Table 4.5, with the frequency and molar absorptivities calculated using the Double Harmonic Approximation included for reference. Modes 2 and 3 both began to converge towards a minimum energy around the 0 → 7 levels, as predicted by the separated calculations; mode 1, however, could likely use even more harmonic energy levels included in the linear combination as it had not quite converged to the 1 cm^{-1} digit. The molar absorptivities reflect these same conclusions, converging at 0 → 7 or even earlier for modes 2 and 3.

A comparison of the anharmonic frequencies and molar absorptivities calculated using the linear variation method to those obtained through the DHA show that while the frequency of mode 1 decreased, the frequency of modes 2 and 3 actually increased. This is opposite the trend observed with VPT2, and so it indicates that a major component of the system is not being included. The molar absorptivity values for modes 1 and 3 are approximately on par with those of VPT2, but mode 2 has again increased where VPT2 decreased. Since these values were calculated using the non-coupling terms of the Hamiltonian operator only, a comparison with values that include coupling is paramount.

With the inclusion of coupling terms, as described in Equation 4.11, the changes in the energies of each mode become more difficult to interpret. One of the most obvious effects of including coupling, as seen in Table 4.6, is that the $0 \rightarrow 8$ linear combination is unstable for all three modes; the energy value increases dramatically in each case, which indicates that the description of the system has failed in some way. This failure is also apparent in mode 1 for the $0 \rightarrow 7$ energy levels, as the value decreases sharply compared to its previous converging trend. Another obvious result, however, is that the addition of the coupling terms decreases the energy of the vibrational modes to be much closer to the values calculated by VPT2 and observed experimentally.

Since mode 1 is no longer stable at the $0 \rightarrow 7$ energy level set, it is apparent that further analysis into the effects of coupling on the system are required. One method of achieving this is to consider different combinations of maximum energy levels across the vibrational modes, rather than all three modes having the same maximum value. Additionally, an analysis of the coefficients in the eigenvector of an anharmonic wavefunction could provide insight into which terms have the most contribution.

Table 4.6: Vibrational frequencies and molar absorptivity values calculated from the non-coupled and coupled Hamiltonian matrices, for an increasing number of included harmonic energy levels.

Freq. (cm^{-1})	Non-coupled			Coupled		
	Mode 1	Mode 2	Mode 3	Mode 1	Mode 2	Mode 3
2,2,2	3649.1360	1243.2526	1128.0374	3625.7499	1214.5696	1087.8806
3,3,3	3579.1434	1243.2883	1128.4002	3538.3740	1208.0945	1077.9936
4,4,4	3533.5693	1242.7421	1126.8955	3493.7363	1206.5255	1073.7876
5,5,5	3514.5076	1242.7435	1126.8754	3473.2576	1206.2840	1073.1313
6,6,6	3512.6042	1242.7352	1126.8675	3463.1233	1206.2436	1072.9261
7,7,7	3511.0856	1242.7342	1126.8642	~~3304.7611~~	1206.2202	1072.8200
8,8,8	3510.3560	1242.7342	1126.8642	~~4715.6069~~	~~2551.5535~~	~~1478.7074~~
Converged to	3510	1242.734	1126.8642	3400	1206.2	1072.

Molar Abs. $({}^{km}/_{mol})$	Non-coupled			Coupled		
	Mode 1	Mode 2	Mode 3	Mode 1	Mode 2	Mode 3
2,2,2	57.541	5.0105	164.361	57.172	4.8949	158.510
3,3,3	57.809	5.0327	165.759	57.151	4.8902	158.354
4,4,4	56.715	5.0314	165.645	56.075	4.8848	157.839
5,5,5	56.015	5.0312	165.619	55.358	4.8835	157.720
6,6,6	55.900	5.0311	165.621	55.113	4.8834	157.693
7,7,7	55.825	5.0311	165.621	~~52.545~~	4.8833	157.678
8,8,8	55.788	5.0311	165.621	~~74.943~~	~~10.3298~~	~~217.333~~
Converged to	55.	5.0311	165.621	55	4.883	157.6

Chapter 5

Conclusions

We have shown that the Double Harmonic Approximation, while a good initial basis for modeling a vibrational system, is not sufficient to end at when describing the ^{3}HON molecule. The anharmonic corrections added by Vibrational Second-Order Perturbation Theory brought the vibrational frequencies closer to those measured experimentally; however, the ratio of the molar absorptivities between modes 3 and 1 for VPT2 was 2.62, which was further away from the Anderson group's calculated 2.38 (personal communication) than the 2.42 of the Double Harmonic Approximation. Additionally, plots of the quartic polynomials showed that the force constants used were still insufficient at describing the entirety of the potential energy curve.

The next step beyond VPT2 was to use the linear variation method, and approximate a trial wavefunction as a linear combination of multiple harmonic wavefunctions. The Hamiltonian operator applied to this trial wavefunction contained force constants up to the 10^{th} derivative, as that was the polynomial order that best fit the potential energy curves. We determined that including the coupling terms in the Hamiltonian matrix made a significant difference when calculating the vibrational frequencies for each mode, but concluded that different combinations of the maximum energy levels of the modes needed to be explored.

5.1 Varying Energy Level Combinations

Of the three vibrational modes, the most anharmonic is clearly mode 1; therefore while performing a systematic analysis of energy level combinations, mode 1 should never remain constant while both modes 2 and 3 change. As an initial first step of this analysis, we calculated the vibrational frequencies of the non-coupled and coupled modes while varying the maximum energy levels of modes 1 and 2 from $i = j = 5$ up to 9, with mode 3 remaining constant at $0 \rightarrow 4$, and while varying modes 1 and 3 with mode 2 constant for the same ranges. The non-coupled and coupled anharmonic frequency values of mode 1 for these combinations are reported in Table 5.1.

There are two key observations that can be drawn from this, relative to the results at the end of Chapter 4. The first is that the frequencies of mode 1 are no longer behaving erratically when increased to a maximum energy level of 9; as long as either mode 2 or mode 3 is of a lower energy level, it appears to provide a stabilizing influence on mode 1 − or no longer be a cause of destabilization. The second observation is that when mode 2 is held constant, as in the table on the right, the coupled frequencies converge over a short range; by contrast, when mode 3 is held constant and mode 2 varies with mode 1, the coupled frequencies experience much larger differences while converging towards a lower energy value. This indicates that the coupling between modes 1 and 2 may be stronger than the coupling between modes 1 and 3, but further investigation into the eigenvector coefficients is recommended to confirm or deny this.

Table 5.1: Anharmonic frequencies (cm^{-1}) of mode 1 calculated using the linear variation method. Left: mode 3 held constant while modes 1 and 2 include increasing numbers of energy levels; Right: mode 2 held constant while modes 1 and 3 include increasing numbers of energy levels.

	Non-coupled	Coupled
5,5,4	3514.51	3476.32
6,6,4	3512.60	3467.84
7,7,4	3511.09	3464.56
8,8,4	3510.36	3462.74
9,9,4	3510.23	3460.78

	Non-coupled	Coupled
5,4,5	3514.51	3488.30
6,4,6	3512.60	3486.81
7,4,7	3511.09	3485.26
8,4,8	3510.36	3484.64
9,4,9	3510.23	3484.45

5.2 Impact of Research

Previous work by Bozkaya [5] has shown the importance of including anharmonic effects when calculating the vibrational frequencies of ^{3}HON, but there is no documentation examining the impact of anharmonicity on the dipole moments of this triatomic molecule. We have found that the inclusion of anharmonicity and coupling, as well as the methods used to calculate these parameters, have significant effect on the dipole moments and thus the molar absorptivity values and ratio of the vibrational modes.

Correct modeling of the dipole moments of the molecule is imperative for obtaining reasonable molar absorptivity values, and we found that accurate vibrational frequencies are not always indicators of a reliable model. Additionally, coupling effects for ^{3}HON must be considered in addition to anharmonicity when calculating vibrational frequencies and dipole moments, as neglecting the coupling terms results in increasing deviations from the experimental molar absorptivity ratio. Whether ^{1}HNO exhibits such impact from coupling terms would be interesting to determine at a later point in time.

5.3 Future Directions

It is clear that there are still many nuances and mysteries in how the coupling between the three vibrational modes of ^{3}HON affect the calculated frequencies and molar absorptivities. Some additional ways to explore these avenues include making contour plots of the ground and excited energy levels for different combinations of vibrational mode coupling, plotting the trial wavefunctions with respect to a given vibrational mode, or analyzing the contributions of each harmonic wavefunction as given in the eigenvector. It is also possible that in addition to the two-way coupling already being considered, three-way coupling may have a non-negligible effect on the energy of system.

There is no shortage of future work to be continued with this project, although the researcher must be cautious to not fall into the trap of over-fitting the system, for modeling a system with too many parameters is just as faulty as modeling it with too few. However, a thorough knowledge of the system being studied will prevent this issue and aid future interpretation of results.

Bibliography

[1] Barone, V. (2004). Vibrational zero-point energies and thermodynamic functions beyond the harmonic approximation. *The Journal of Chemical Physics*, 120(7):3059–3065. 38

[2] Barone, V. (2005). Anharmonic vibrational properties by a fully automated second-order perturbative approach. *J. Chem. Phys.*, 122:014108. 37, 38

[3] Barone, V., Bloino, J., Guido, C. A., and Lipparini, F. (2010). A fully automated implementation of VPT2 Infrared intensities. *Chemical Physics Letters*, 496(1-3):157–161. 64

[4] Bloino, J. (2015). A VPT2 Route to Near-Infrared Spectroscopy: The Role of Mechanical and Electrical Anharmonicity. *Journal of Physical Chemistry A*, 119(21):5269–5287. 38

[5] Bozkaya, U., Turney, J. M., Yamaguchi, Y., and Schaefer, H. F. (2012). The lowest-lying electronic singlet and triplet potential energy surfaces for the HNO-NOH system: Energetics, unimolecular rate constants, tunneling and kinetic isotope effects for the isomerization and dissociation reactions. *Journal of Chemical Physics*, 136(16). v, x, xi, xiii, 1, 4, 7, 9, 12, 13, 27, 40, 56

[6] Cizek, Jiri (2007). *On the Use of the Cluster Expansion and the Technique of Diagrams in Calculations of Correlation Effects in Atoms and Molecules*, chapter 2, pages 35–89. John Wiley & Sons, Ltd. 24

[7] Clark, T., Chandrasekhar, J., Spitznagel, G. W., and Schleyer, P. V. R. (1983). Efficient diffuse function-augmented basis sets for anion calculations. III. The 3-21+G basis set for first-row elements, Li-F. *J. Comput. Chem.*, 4:294–301. 24

[8] Ditchfield, R., Hehre, W. J., and Pople, J. A. (1971). Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic Molecules. *The Journal of Chemical Physics*, 54:724–728. 24

[9] Dunning, T. H. (1971). Gaussian basis functions for use in molecular calculations. iii. contraction of (10s6p) atomic basis sets for the first row atoms. *The Journal of Chemical Physics*, 55(2):716–723. 10

[10] Dunning, T. H. (1989). Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *J. Chem. Phys.)*, 8(1):1007–1023. 25

[11] Hariharan, P. C. and Pople, J. A. (1973). The influence of polarization functions on molecular orbital hydrogenation energies. *Theor. Chim. Acta*, 28:213–222. 24

[12] Hehre, W. J., Ditchfield, R., and Pople, J. A. (1972). Self-Consistent Molecular Orbital Methods. XII. Further Extensions of Gaussian-Type Basis Sets for Use in Molecular Orbital Studies of Organic Molecules. *The Journal of Chemical Physics*, 56:2257–2261. 24

[13] Lee, T. J. (1993). A coupled-cluster study of XNO (X=H,F,Cl): An investigation of weak X-N single bonds. *The Journal of Chemical Physics*, 99(12):9783–9789. 10

[14] Lee, T. J. (1994). A coupled-cluster study of XON (X=H,F,Cl), and the XONXNO transition states. *Chemical Physics Letters*, 223(5-6):431–438. x, 1, 11, 12, 36

[15] Maier, G., Reisenauer, H. P., and Marco, M. D. (1999). Isonitroso Hydrogen (Hydroxy Nitrene, HON). *Angewandte Chemie (International ed. in English)*, 38(1):108–110. v, x, xi, xiii, 1, 7, 10, 14, 15, 16, 28, 36, 40

[16] McQuarrie, D. A. (2008). *Quantum Chemistry.* University Science Books, Sausalito, Calif. 35, 37, 44

[17] Mills, I. (1993). *Quantities, Units, and Symbols in Physical Chemistry: 2rd edition.* 5, 23

[Paci] Paci, I. Basis sets. http://web.uvic.ca/~simmater/Groupweb/Qmodule/basissets.pdf. Accessed: 2019-03-09. 25

[19] Pople, J. A., Head Gordon, M., and Raghavachari, K. (1987). Quadratic configuration interaction. a general technique for determining electron correlation energies. *The Journal of Chemical Physics*, 87(10):5968–5975. 24

[20] Ruzi, M. and Anderson, D. T. (2015). Quantum Diffusion-Controlled Chemistry: Reactions of Atomic Hydrogen with Nitric Oxide in Solid Parahydrogen. *Journal of Physical Chemistry A*, 119(50):12270–12283. v, xi, 2, 3, 5, 7, 10, 31, 36, 40

[Sherrill] Sherrill, D. The Variational Method. http://vergil.chemistry.gatech.edu/notes/quantrev/node28.html. Accessed: 2019-03-09. 43

[22] Valiev, M., Bylaska, E., Govind, N., Kowalski, K., Straatsma, T., Dam, H. V., Wang, D., Nieplocha, J., Apra, E., Windus, T., and de Jong, W. (2010). Nwchem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications*, 181(9):1477 – 1489. 24

[23] Vázquez, J. and Stanton, J. F. (2007). Treatment of Fermi resonance effects on transition moments in vibrational perturbation theory. *Molecular Physics*, 105(1):101–109. 39, 64

[24] Yagi, K., Hirao, K., Taketsugu, T., Schmidt, M. W., and Gordon, M. S. (2004). Ab initio vibrational state calculations with a quartic force field: Applications to H2CO, C2H4, CH3OH, CH3CCH, and C6H6. *The Journal of Chemical Physics*, 121(3):1383–1389. 38

[25] Yu, Q. and Bowman, J. M. (2015). Vibrational second-order perturbation theory (VPT2) using local monomer normal modes. *Molecular Physics*, 113(24):3964–3971. 37, 38

Appendices

A Additional Equations

A.1 Finite Difference Method

Third Derivatives:

$$\phi_{iii} = \frac{V(x_i + 2h) - 2V(x_i + h) + 2V(x_i - h) - V(x_i - 2h)}{2h^3} \tag{1}$$

$$\phi_{iij} = \frac{V\begin{pmatrix} x_i+h \\ x_j+h \end{pmatrix} + V\begin{pmatrix} x_i-h \\ x_j+h \end{pmatrix} - 2V\begin{pmatrix} x_i \\ x_j+h \end{pmatrix} - V\begin{pmatrix} x_i+h \\ x_j-h \end{pmatrix} - V\begin{pmatrix} x_i-h \\ x_j-h \end{pmatrix} + 2V\begin{pmatrix} x_i \\ x_j-h \end{pmatrix}}{2h^3} \tag{2}$$

$$\phi_{ijk} = \frac{\begin{gathered} V\begin{pmatrix} x_i+h \\ x_j+h \\ x_k+h \end{pmatrix} - V\begin{pmatrix} x_i-h \\ x_j+h \\ x_k+h \end{pmatrix} - V\begin{pmatrix} x_i+h \\ x_j-h \\ x_k+h \end{pmatrix} + V\begin{pmatrix} x_i-h \\ x_j-h \\ x_k+h \end{pmatrix} \\ - V\begin{pmatrix} x_i+h \\ x_j+h \\ x_k-h \end{pmatrix} + V\begin{pmatrix} x_i-h \\ x_j+h \\ x_k-h \end{pmatrix} + V\begin{pmatrix} x_i+h \\ x_j-h \\ x_k-h \end{pmatrix} - V\begin{pmatrix} x_i-h \\ x_j-h \\ x_k-h \end{pmatrix} \end{gathered}}{8h^3} \tag{3}$$

Fourth Derivatives:

$$\phi_{iiii} = \frac{V(x_i + 2h) - 4V(x_i + h) + 6V(x_i) - 4V(x_i - h) + V(x_i - 2h)}{h^4} \tag{4}$$

$$\phi_{iijj} = \frac{\begin{gathered} V\begin{pmatrix} x_i+h \\ x_j+h \end{pmatrix} + V\begin{pmatrix} x_i-h \\ x_j+h \end{pmatrix} - 2V\begin{pmatrix} x_i \\ x_j+h \end{pmatrix} + V\begin{pmatrix} x_i+h \\ x_j-h \end{pmatrix} + V\begin{pmatrix} x_i-h \\ x_j-h \end{pmatrix} \\ - 2V\begin{pmatrix} x_i \\ x_j-h \end{pmatrix} - 2\left(V\begin{pmatrix} x_i+h \\ x_j \end{pmatrix} + V\begin{pmatrix} x_i-h \\ x_j \end{pmatrix} - 2V\begin{pmatrix} x_i \\ x_j \end{pmatrix}\right) \end{gathered}}{h^4} \tag{5}$$

$$\phi_{iiij} = \frac{\begin{gathered} V\begin{pmatrix} x_i+2h \\ x_j+h \end{pmatrix} - 2V\begin{pmatrix} x_i+h \\ x_j+h \end{pmatrix} + 2V\begin{pmatrix} x_i-h \\ x_j+h \end{pmatrix} - V\begin{pmatrix} x_i-2h \\ x_j+h \end{pmatrix} \\ - V\begin{pmatrix} x_i+2h \\ x_j-h \end{pmatrix} + 2V\begin{pmatrix} x_i+h \\ x_j-h \end{pmatrix} - 2V\begin{pmatrix} x_i-h \\ x_j-h \end{pmatrix} + V\begin{pmatrix} x_i-2h \\ x_j-h \end{pmatrix} \end{gathered}}{4h^4} \tag{6}$$

$$\phi_{ijkk} = \frac{\begin{gathered} V\begin{pmatrix} x_i+h \\ x_j+h \\ x_k+h \end{pmatrix} + V\begin{pmatrix} x_i-h \\ x_j+h \\ x_k+h \end{pmatrix} - 2V\begin{pmatrix} x_i \\ x_j+h \\ x_k+h \end{pmatrix} - V\begin{pmatrix} x_i+h \\ x_j-h \\ x_k+h \end{pmatrix} - V\begin{pmatrix} x_i-h \\ x_j-h \\ x_k+h \end{pmatrix} + 2V\begin{pmatrix} x_i \\ x_j-h \\ x_k+h \end{pmatrix} \\ - V\begin{pmatrix} x_i+h \\ x_j+h \\ x_k-h \end{pmatrix} - V\begin{pmatrix} x_i-h \\ x_j+h \\ x_k-h \end{pmatrix} + 2V\begin{pmatrix} x_i \\ x_j+h \\ x_k-h \end{pmatrix} + V\begin{pmatrix} x_i+h \\ x_j-h \\ x_k-h \end{pmatrix} + V\begin{pmatrix} x_i-h \\ x_j-h \\ x_k-h \end{pmatrix} - 2V\begin{pmatrix} x_i \\ x_j-h \\ x_k-h \end{pmatrix} \end{gathered}}{4h^4} \tag{7}$$

A.2 VPT2

Transition Dipole Moment[23, 3]:

$$
\begin{aligned}
\langle \mu^\alpha \rangle_{0i} = {} & \frac{1}{\sqrt{2}}\mu_i^\alpha + \frac{1}{4\sqrt{2}}\sum_j \mu_{ijj}^\alpha - \frac{1}{8\sqrt{2}}\sum_{jk}\bigg\{\phi_{ijk}\mu_{jk}^\alpha\Big(\frac{1}{\omega_i+\omega_j+\omega_k} - \frac{1}{\omega_i-\omega_j-\omega_k}\Big) \\
& + \frac{2}{\omega_k}\phi_{kjj}\mu_{ik}^\alpha\bigg\} + \frac{1}{16\sqrt{2}}\sum_{jkl}\bigg\{\phi_{ikl}\phi_{jkl}\mu_j^\alpha\bigg[\frac{4\omega_j(\omega_l+\omega_k)(1-\delta_{ij})(1-\delta_{ik})(1-\delta_{il})}{(\omega_i^2-\omega_j^2)[\omega_i^2-(\omega_l+\omega_k)^2]} \\
& - \frac{(\omega_l+\omega_k)[3\omega_i^2-(\omega_l+\omega_k)^2]\delta_{ij}(1+\delta_{ik})(1-\delta_{il})}{\omega_i[\omega_i^2-(\omega_l+\omega_k)^2]^2} \\
& - \frac{4\omega_j(7\omega_i\omega_k+3\omega_k\omega_j+3\omega_k^2+4\omega_i\omega_j+4\omega_i^2)(1-\delta_{ij})(1-\delta_{jk})\delta_{il}}{\omega_k(2\omega_i+\omega_k)(\omega_i^2-\omega_j^2)(\omega_i+\omega_j+\omega_k)}\bigg] \\
& + \phi_{ijk}\phi_{llk}\mu_j^\alpha\bigg[\frac{\delta_{ij}}{\omega_i\omega_k}\Big(1+\frac{2}{9}\delta_{ik}\delta_{il}\Big) - \frac{4\omega_j(1-\delta_{ij})(1-\delta_{ik})(1-\delta_{il})}{\omega_k(\omega_i^2-\omega_j^2)} \\
& - \frac{4\omega_j\delta_{ik}(1-\delta_{ij})}{\omega_i(\omega_i^2-\omega_j^2)}\Big(1+\frac{2}{3}\delta_{il}\Big)\bigg]\bigg\} - \frac{1}{8\sqrt{2}}\sum_{jk}\phi_{ijkk}\mu_j^\alpha\Big(\frac{1}{\omega_i+\omega_j} - \frac{1-\delta_{ij}}{\omega_i-\omega_j}\Big)
\end{aligned}
\tag{8}
$$

B VPT2 Force and Dipole Constants

The following four sections contain the VPT2 force and dipole constants derived through the Finite Difference Method with a displacement of 0.3Å, for the isotope variations of HON investigated. The terms i, j, k, and l denote the normal modes being considered (1, 2, or 3). The calculations were performed using CCSD(T)/aug-cc-pVXZ.

B.1 HON

Table B.1.1: Cubic force constants ϕ_{ijk} (cm^{-1}) of HON in normal coordinates.

ijk	TZ	QZ
111	-2609.10919	-2637.34200
112	-44.25554	-47.84107
113	33.33172	33.06437
122	450.57048	397.60215
123	-429.81742	-432.01403
133	367.49580	420.41427
222	-224.78955	-244.03171
223	-65.78999	-73.38880
233	-197.32172	-211.11206
333	-294.66784	-256.35382

Table B.1.2: Quartic force constants ϕ_{ijkl} (cm^{-1}) of HON in normal coordinates.

$ijkl$	TZ	QZ
1111	1559.79677	1571.98631
1112	-41.85608	-39.88093
1113	37.89766	38.98173
1122	-518.29186	-469.09857
1123	440.25201	446.87334
1133	-405.84078	-466.07581
1222	80.57730	77.38351
1223	-36.15769	-41.61014
1233	18.96237	25.04999
1333	12.34529	7.63085
2222	164.27952	139.50469
2223	-121.11540	-100.99847
2233	171.91269	176.13945
2333	-60.18118	-80.17454
3333	202.11579	217.86030

Table B.1.3: Linear dipole moment constants μ_i^α (Debye) of HON in normal coordinates.

	x-component		y-component	
i	TZ	QZ	TZ	QZ
1	-0.0274506	-0.0279054	0.0408852	0.0417887
2	-0.0224146	-0.0148872	-0.0012719	-0.0018040
3	0.1367785	0.1378688	-0.0069611	-0.0062095

Table B.1.4: Quadratic dipole moment constants μ_{ij}^α (Debye) of HON in normal coordinates.

	x-component		y-component	
ij	TZ	QZ	TZ	QZ
11	0.0092807	0.0093337	-0.0065246	-0.0063720
12	-0.0011433	-0.0015217	-0.0033666	-0.0032244
13	-0.0049540	-0.0050108	0.0019340	0.0022790
22	-0.0013010	-0.0007294	-0.0066262	-0.0059927
23	0.0060169	0.0058954	0.0037412	0.0037026
33	-0.0082928	-0.0086946	-0.0054203	-0.0059093

Table B.1.5: Cubic dipole moment constants μ_{ijk}^α (Debye) of HON in normal coordinates.

	x-component		y-component	
ijk	TZ	QZ	TZ	QZ
111	0.0036347	0.0039467	-0.0056134	-0.0056517
112	-0.0002081	0.0001013	-0.0003415	-0.0005305
113	-0.0031671	-0.0031202	0.0026741	0.0028657
122	0.0000614	0.0001896	0.0014996	0.0013993
133	0.0040065	0.0045485	-0.0022687	-0.0020136
222	0.0009083	0.0006007	-0.0034633	-0.0031371
223	-0.0002178	-0.0004167	0.0017386	0.0018737
233	-0.0014804	-0.0019289	-0.0012404	-0.0012280
333	-0.0026806	-0.0025136	0.0041354	0.0043834

B.2 $HO^{15}N$

Table B.2.1: Cubic force constants ϕ_{ijk} (cm^{-1}) of $HO^{15}N$ in normal coordinates.

ijk	TZ	QZ
111	-2609.10841	-2637.41115
112	-40.82218	-44.05646
113	36.54653	36.66480
122	525.08632	479.76810
123	-418.05680	-428.14366
133	296.60684	342.04620
222	-206.91871	-222.39694
223	-49.36075	-54.07615
233	-176.26546	-192.17043
333	-333.83102	-304.84180

Table B.2.2: Quartic force constants ϕ_{ijkl} (cm^{-1}) of $HO^{15}N$ in normal coordinates.

$ijkl$	TZ	QZ
1111	1559.78398	1572.03357
1112	-44.64189	-43.06770
1113	34.40030	35.22760
1122	-593.48433	-552.89618
1123	424.99662	440.63452
1133	-332.81325	-385.55566
1222	88.88956	87.40641
1223	-31.61484	-37.89428
1233	11.66576	16.68874
1333	15.79277	13.00374
2222	212.12884	186.18369
2223	-150.02120	-135.68205
2233	158.82298	166.70345
2333	-39.49515	-55.02698
3333	182.15248	188.59710

Table B.2.3: Linear dipole moment constants μ_i^α (Debye) of HO^{15}N in normal coordinates.

	x-component		y-component	
i	TZ	QZ	TZ	QZ
1	-0.0274591	-0.0279143	0.0408845	0.0417884
2	-0.0344271	-0.0281007	-0.00080812	-0.0013495
3	0.1338089	0.1353573	-0.0068854	-0.0061848

Table B.2.4: Quadratic dipole moment constants μ_{ij}^α (Debye) of HO^{15}N in normal coordinates.

	x-component		y-component	
ij	TZ	QZ	TZ	QZ
11	0.0092816	0.0093348	-0.0065244	-0.0063720
12	-0.0006787	-0.0010259	-0.0035170	-0.0034212
13	-0.0049992	-0.0050930	0.0016551	0.0019988
22	-0.0024880	-0.0019465	-0.0072040	-0.0066216
23	0.0065097	0.0065289	0.0036328	0.0036696
33	-0.0072185	-0.0075416	-0.0048098	-0.0052502

Table B.2.5: Cubic dipole moment constants μ_{ijk}^α (Debye) of HO^{15}N in normal coordinates.

	x-component		y-component	
ijk	TZ	QZ	TZ	QZ
111	0.0036353	0.0039475	-0.0056139	-0.0056524
112	-0.0004185	-0.0002209	-0.0001165	-0.0002586
113	-0.0031093	-0.0030920	0.0026494	0.0027822
122	0.0003235	0.0002788	0.0016134	0.0015670
133	0.0038307	0.0044205	-0.0023602	-0.0021596
222	0.0006878	0.0007221	-0.0038381	-0.0036175
223	0.0001241	-0.0000288	0.0015994	0.0018046
233	-0.0013875	-0.0016691	-0.0012670	-0.0013117
333	-0.0030003	-0.0029057	0.0037575	0.0040012

B.3 DON

Table B.3.1: Cubic force constants ϕ_{ijk} (cm^{-1}) of DON in normal coordinates.

ijk	TZ	QZ
111	1618.63707	1636.36731
112	20.88396	176.53105
113	1.75939	2.16267
122	3.18871	23.40674
123	56.09660	47.98549
133	-471.91592	-471.46664
222	479.62243	476.03289
223	-41.24018	-47.81450
233	40.56843	85.15277
333	77.02547	84.82942

Table B.3.2: Quartic force constants ϕ_{ijkl} (cm^{-1}) of DON in normal coordinates.

$ijkl$	TZ	QZ
1111	827.45464	831.12263
1112	13.69073	-66.44363
1113	-22.70245	-22.14776
1122	-14.01186	-11.38250
1123	40.62675	35.91258
1133	-440.62911	-446.68124
1222	6.20898	9.11578
1223	-1.08733	-6.59475
1233	12.12303	54.30648
1333	40.16657	43.42283
2222	173.02589	164.10199
2223	-23.84304	-20.95551
2233	2.36664	-7.63544
2333	-26.80761	-25.78494
3333	219.21690	218.25354

Table B.3.3: Linear dipole moment constants μ_i^α (Debye) of DON in normal coordinates.

	x-component		y-component	
i	TZ	QZ	TZ	QZ
1	0.0259346	0.0263387	-0.0335750	-0.0343085
2	-0.0951206	-0.0954338	0.0053846	0.0085587
3	0.0858225	0.0859994	-0.0076853	-0.0073440

Table B.3.4: Quadratic dipole moment constants μ_{ij}^α (Debye) of DON in normal coordinates.

	x-component		y-component	
ij	TZ	QZ	TZ	QZ
11	0.0070439	0.0070771	-0.0047402	-0.0046157
12	-0.0039127	-0.0046828	-0.0000977	0.0004177
13	0.0014520	0.0014486	-0.0025473	-0.0026206
22	-0.0000605	0.0013224	-0.0024479	-0.0023128
23	0.0042266	0.0038370	-0.0005387	-0.0003213
33	-0.0062924	-0.0065096	-0.0069323	-0.0068511

Table B.3.5: Cubic dipole moment constants μ_{ijk}^α (Debye) of DON in normal coordinates.

	x-component		y-component	
ijk	TZ	QZ	TZ	QZ
111	-0.0022891	-0.0024459	0.0034836	0.0035034
112	-0.0016953	-0.0018298	0.0016717	0.0021237
113	-0.0015074	-0.0014939	0.0011733	0.0012138
122	-0.0016710	-0.0024037	0.0014184	0.0017413
133	-0.0010646	-0.0012582	-0.0005774	-0.0006261
222	0.0031680	0.0036775	-0.0013773	-0.0017896
223	-0.0009426	-0.0016652	0.0013394	0.0015100
233	-0.0003214	-0.0002742	-0.0003117	-0.0003374
333	0.0004794	0.0003973	0.0032284	0.0033470

B.4 DO^{15}N

Table B.4.1: Cubic force constants ϕ_{ijk} (cm^{-1}) of DO^{15}N in normal coordinates.

ijk	TZ	QZ
111	1618.62311	1621.06671
112	20.74181	20.60924
113	1.65828	0.21026
122	2.47374	2.71312
123	58.54682	57.64906
133	-474.69424	-466.13525
222	466.84099	463.80532
223	-38.76942	-38.42457
233	39.48008	39.67383
333	77.32763	83.54789

Table B.4.2: Quartic force constants ϕ_{ijkl} (cm^{-1}) of DO^{15}N in normal coordinates.

$ijkl$	TZ	QZ
1111	827.44656	824.77864
1112	13.69398	13.01988
1113	-22.62218	-22.81597
1122	-14.26133	-16.46174
1123	43.03574	42.77630
1133	-442.84626	-444.53952
1222	6.07683	5.45804
1223	-1.22067	-0.04312
1233	11.79864	10.91846
1333	40.25010	43.81736
2222	166.99453	164.00570
2223	-22.33627	-21.66706
2233	2.76741	-0.15745
2333	-28.21454	-26.83039
3333	222.84034	215.80775

Table B.4.3: Linear dipole moment constants μ_i^α (Debye) of $DO^{15}N$ in normal coordinates.

	x-component		y-component	
i	TZ	QZ	TZ	QZ
1	0.0259275	0.0264407	-0.0335734	-0.0342950
2	-0.0947514	-0.0946644	0.0053614	0.0052339
3	0.0856438	0.0860206	-0.0073813	-0.0072701

Table B.4.4: Quadratic dipole moment constants μ_{ij}^α (Debye) of $DO^{15}N$ in normal coordinates.

	x-component		y-component	
ij	TZ	QZ	TZ	QZ
11	0.0070434	0.0069431	-0.0047403	-0.0045062
12	-0.0038847	-0.0038862	-0.0000809	-0.0000971
13	0.0014452	0.0013576	-0.0025592	-0.0025641
22	-0.0001097	-0.0000101	-0.0024039	-0.0023097
23	0.0042277	0.0042007	-0.0004414	-0.0004925
33	-0.0063293	-0.0065365	-0.0069520	-0.0067731

Table B.4.5: Cubic dipole moment constants μ_{ijk}^α (Debye) of $DO^{15}N$ in normal coordinates.

	x-component		y-component	
ijk	TZ	QZ	TZ	QZ
111	-0.0022886	-0.0024009	0.0034834	0.0034029
112	-0.0016875	-0.0017806	0.0016604	0.0016270
113	-0.0014967	-0.0014847	0.0011554	0.0012326
122	-0.0016552	-0.0020863	0.0014063	0.0013394
133	-0.0010671	-0.0012943	-0.0005892	-0.0006374
222	0.0030961	0.0033219	-0.0013615	-0.0012935
223	-0.0009013	-0.0013301	0.0013175	0.0013094
233	-0.0003237	-0.0002366	-0.0003177	-0.0003636
333	0.0004598	0.0004236	0.0032751	0.0033926

C Maple Programs

The following codes were written in the mathematical program Maple, and used to calculate the normal mode displacements, vibrational frequencies, transition dipole moments, and molar absorptivities discussed. The input data came from single-point energy calculations performed using NWChem, and read in from Excel spreadsheets.

C.1 Normal Modes

This worksheet uses the internal coordinate values of the equilibrium geometry, and the second derivatives of the internal coordinates as determined through the Finite Difference Method. It outputs the harmonic vibrational frequencies of the molecule and their corresponding displacement vectors.

$restart : with(linalg) : with(ExcelTools) :$

#Conversion factors in use: 1 a.u. → *219474.6 cm*$^{-1}$; 1 *amu* → 1822.89 *au*; 0.529177 *Angstroms* → 1 *au* (*Bohr's radius*)

# Enter the masses of the atoms in amu, as will be ordered in Cartesian coordinate Hessian matrix

$masses := [1.007825, 1.007825, 15.994915, 15.994915, 14.003074, 14.003074];$

# Convert derivatives w.r.t. internal coordinates into derivatives w.r.t. Cartesian coordinates

$$chain := matrix\left(3, 6, \left[\cos(th), \sin(th), -\cos(th), -\sin(th), 0, 0, \frac{(\sin(th))^2}{r\_ho}, -\frac{\sin(th)\cdot\cos(th)}{r\_ho}, -\frac{(\sin(th))^2}{r\_ho}, \frac{\sin(th)\cdot\cos(th)}{r\_ho} -\frac{\sin(th)}{r\_on}, 0, \frac{\sin(th)}{r\_on}, 0, 0, -1, 0, 1, 0\right]\right);$$

$derivs := matrix(3, 3, [drho2, drhodcos, drhodron, drhodcos, dcos2, dcosdron, drhodron, dcosdron, dron2]);$

$el := (i, j) \rightarrow sum(sum(derivs[k, l]\cdot chain[k, i]\cdot chain[l, j], l = 1 ..3), k = 1 ..3);$

# Input the internal coordinates for the canonical geometry, with the angle in radians

$$th := \frac{107.28339}{180}\cdot\mathrm{Pi}; r\_on := 1.33539; r\_ho := 0.97160 ;$$

$drho2 := 1.75643429; dcos2 := 0.18120840; dron2 := 1.40078156; drhodcos := -0.02709078; drhodron := 0.04769898; dcosdron := -0.12105048;$

# Assemble the mass-weighted hessian matrix, using the numerical factors 0.529177 and 1822.89 to get everything into atomic units

$$hesmat := matrix\left(6, 6, \left[seq\left(seq\left(\frac{0.529177^2\cdot evalf(el(i, j))}{\mathrm{sqrt}(masses[i]\cdot masses[j])\cdot 1822.89}, j = 1 ..6\right), i = 1 ..6\right)\right]\right);$$

# Obtain and convert eigenvalues (in atomic units) to vibrational frequencies (in inverse centimeters)

$evals := eigenvalues(hesmat); seq(219474.6\cdot\mathrm{sqrt}(\mathrm{abs}(evals[i])), i = 1 ..6);$
$vex := sort([eigenvectors(hesmat)]);$
$ordered := seq(219474.6\cdot\mathrm{sqrt}(\mathrm{abs}(vex[i, 1])), i = 1 ..6);$ **if** $ordered[4] > ordered[5]$ **then** $keepers := [4, 3, 2]$ **else** $keepers := [6, 5, 4]$ **end if**;

#Calculate the dimensionless normal mode vectors

$u1 := vex[keepers[1], 3, 1] : vex[keepers[1], 1] : \mathrm{sqrt}(\%)\cdot 219474.6;$

$$zn1 := add\left(u1[i]^2, i = 1 ..6\right); Matrix\left(6, 1, \left[seq\left(\frac{u1[i]\cdot\mathrm{sqrt}\left(zn1^{-1}\right)}{\mathrm{sqrt}(1822.89\cdot masses[i]\cdot\mathrm{sqrt}(vex[keepers[1], 1]))}, i = 1 ..6\right)\right]\cdot 0.529177\right)^{+} :$$

$u2 := vex[keepers[2], 3, 1] : vex[keepers[2], 1] : \mathrm{sqrt}(\%)\cdot 219474.6;$

$$zn2 \coloneqq add\left(u2[i]^2, i = 1\,..6\right); Matrix\left(6, 1, \left[seq\left(\frac{u2[i]\cdot \mathrm{sqrt}\left(zn2^{-1}\right)}{\mathrm{sqrt}(1822.89\cdot\ masses[i]\cdot \mathrm{sqrt}(vex[keepers[2], 1]))}, i = 1\,..6\right)\right]\cdot 0.529177\right)^{+} :$$

$$u3 \coloneqq vex[keepers[3], 3, 1] : vex[keepers[3], 1] : \mathrm{sqrt}(\%)\cdot 219474.6;$$

$$zn3 \coloneqq add\left(u3[i]^2, i = 1\,..6\right); Matrix\left(6, 1, \left[seq\left(\frac{u3[i]\cdot \mathrm{sqrt}\left(zn3^{-1}\right)}{\mathrm{sqrt}(1822.89\cdot\ masses[i]\cdot \mathrm{sqrt}(vex[keepers[3], 1]))}, i = 1\,..6\right)\right]\cdot 0.529177\right)^{+} :$$

C.2 Double Harmonic Approximation

This worksheet calculates the molar absorptivities of the vibrational modes using the Double Harmonic Approximation. It requires the harmonic vibrational frequencies, and the potential energies of each vibrational mode as a function of displacement for a negative, zero, and positive energy field in each of the Cartesian directions (x and y).

```
 restart : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) : with(PolynomialTools) :
interface(rtablesize = 20) : interface(warnlevel = 0) : interface(displayprecision =- 1) :
path ≔ "/home/andrea/Documents/HON_Maple.xlsx" :
 sheet ≔ "ccPVTZ_HON_Dipoles" :
 Dipoles ≔ Matrix(Import(path, sheet, "A4:S24", emptycell = 0.0)) :

q1x ≔ ⟨Column(Dipoles, [2])|Column(Dipoles, [3 ..4])⟩ : q1y ≔ ⟨Column(Dipoles, [5])
     |Column(Dipoles, [6 ..7])⟩ :
q2x ≔ ⟨Column(Dipoles, [8])|Column(Dipoles, [9 ..10])⟩ : q2y ≔ ⟨Column(Dipoles, [11])
     |Column(Dipoles, [12 ..13])⟩ :
q3x ≔ ⟨Column(Dipoles, [14])|Column(Dipoles, [15 ..16])⟩ : q3y ≔ ⟨Column(Dipoles, [17])
     |Column(Dipoles, [18 ..19])⟩ :
GetFreq ≔ Matrix(Import(path, sheet, "C64:I64")) : Frequencies ≔ ⟨Column(GetFreq, [1])
     |Column(GetFreq, [4])|Column(GetFreq, [7])⟩; charges ≔ [-0.002, 0.000, 0.002] :

# Caclulate dipole moments w.r.t. normal mode displacement

q1Mxscan ≔ ⟨seq(subs(x = 0, diff( -PolynomialFit(2, charges, Row(q1x, [i]), x), x)), i = 1 ..21)⟩ :
q1Myscan ≔ ⟨seq(subs(x = 0, diff( -PolynomialFit(2, charges, Row(q1y, [i]), x), x)), i = 1 ..21)⟩ :
q2Mxscan ≔ ⟨seq(subs(x = 0, diff( -PolynomialFit(2, charges, Row(q2x, [i]), x), x)), i = 1 ..21)⟩ :
q2Myscan ≔ ⟨seq(subs(x = 0, diff( -PolynomialFit(2, charges, Row(q2y, [i]), x), x)), i = 1 ..21)⟩ :
q3Mxscan ≔ ⟨seq(subs(x = 0, diff( -PolynomialFit(2, charges, Row(q3x, [i]), x), x)), i = 1 ..21)⟩ :
q3Myscan ≔ ⟨seq(subs(x = 0, diff( -PolynomialFit(2, charges, Row(q3y, [i]), x), x)), i = 1 ..21)⟩ :

#Obtain linear terms of dipole moment fits

q1Mxfit ≔ PolynomialFit(3, Column(Dipoles, [1]), q1Mxscan, x); q1Myfit ≔ PolynomialFit(3,
     Column(Dipoles, [1]), q1Myscan, x);
q2Mxfit ≔ PolynomialFit(3, Column(Dipoles, [1]), q2Mxscan, x); q2Myfit ≔ PolynomialFit(3,
     Column(Dipoles, [1]), q2Myscan, x);
q3Mxfit ≔ PolynomialFit(3, Column(Dipoles, [1]), q3Mxscan, x); q3Myfit ≔ PolynomialFit(3,
     Column(Dipoles, [1]), q3Myscan, x);
q1Mx ≔ subs(x = 0, diff(q1Mxfit, x)); q1My ≔ subs(x = 0, diff(q1Myfit, x));
q2Mx ≔ subs(x = 0, diff(q2Mxfit, x)); q2My ≔ subs(x = 0, diff(q2Myfit, x));
q3Mx ≔ subs(x = 0, diff(q3Mxfit, x)); q3My ≔ subs(x = 0, diff(q3Myfit, x));

#Caclulate transition dipole moments (Mji) and molar absorptivity values

Mji ≔ [q1Mx, q1My, q2Mx, q2My, q3Mx, q3My];
```

$$\mathit{sumMji} \coloneqq \left[\mathit{seq}\left(0.5\cdot\left(\frac{\mathit{Mji}[i]}{0.393456}\right)^{2}, i = 1\,..6\right)\right] : \mathit{Gji} \coloneqq [\mathit{sumMji}[1] + \mathit{sumMji}[2], \mathit{sumMji}[3] + \mathit{sumMji}[4], \mathit{sumMji}[5] + \mathit{sumMji}[6]]\cdot 41.6238;$$

$$\mathit{Absorbance} \coloneqq \frac{\mathit{seq}(\mathit{Gji}[i]\cdot \mathit{Frequencies}(i), i = 1\,..3)}{16.60540};$$

C.3 Vibrational Second-Order Perturbation Theory (VPT2)

This worksheet executes the processes ‘Calc_Anfreq’ and ‘Calc_AnDM’ to calculate the VPT2 frequencies and transition dipole moments of the molecule for nineteen displacement values, which are then used to determine the molar absorptivities. It requires the harmonic vibrational frequencies, the nineteen displacement values, and the potential energies of fifty-seven different combinations of displacements from the equilibrium geometry ($-2q$, $-q$, 0, $+q$, and $+2q$) for modes 1, 2, and 3; each set of energies must be completed for a negative, zero, and positive energy field in each of the Cartesian directions (x and y).

```
restart : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) : with(PolynomialTools) :
    with(ArrayTools) : with(ListTools) : with(DocumentTools) : with(combinat) : interface(rtablesize
     = 60) : interface(warnlevel = 0) : Digits ≔ 20 :
disp ≔ ⟨0.100|0.125|0.150|0.175|0.200|0.225|0.250|0.275|0.300|0.325|0.350|0.375|0.400|0.425|0.450
    |0.475|0.500⟩;
 path ≔ "C:/Users/Andrea/Documents/HON_Maple.xlsx" : sheet ≔ "ccPVTZ_HON_Dipoles" :
GetFreq ≔ Matrix(Import(path, sheet, "C64:I64")) : Frequencies ≔ ⟨Column(GetFreq, [1])
    |Column(GetFreq, [4])|Column(GetFreq, [7])⟩ :
 charges ≔ [-0.002, 0.000, 0.002]; freq ≔ Frequencies[1]^+ ;

#Use processes to calculate VPT2 frequencies and transition dipole moments

sheet ≔ "VPT2_HON" : path ≔ "C:/Users/Andrea/Documents/HON_Maple.xlsx" :
HON_AnFreq ≔ Calc_Anfreq(path, sheet) :
HON_AnDM ≔ Calc_AnDM(path, sheet) :
 Frequencies ≔ HON_AnFreq[3] : DM_x ≔ HON_AnDM[1] : DM_y ≔ HON_AnDM[2] :

 #Use frequencies and transition dipole moments to obtain molar absorptivities

Absorbance ≔ Matrix(3, 17) :
for i from 1 to 17 do
 Mji ≔ ⟨Column(DM_x, i)|Column(DM_y, i)⟩;
  Gji ≔ [seq(add((Mji[j](k) / 0.393456)^2, k = 1 ..2), j = 1 ..3)·41.6238];
 for j from 1 to 3 do
  Absorbance[j, i] ≔ Gji[j]·Frequencies[j](i) / 16.60540;
 end do:
end do:

 Absorbance^+ ;
```

C.4 VPT2 Processes

This worksheet contains two processes: ‘Calc_Anfreq’ and ‘Calc_AnDM’. ‘Calc_Anfreq’ calculates the cubic and quartic force constants of the molecule and uses them to determine the anharmonic vibrational frequencies. ‘Calc_AnDM’ calculates the cubic and quartic force constants, and the linear, quadratic, and cubic dipole constants of the molecule to determine the transition dipole moments in each of the Cartesian directions (x and y).

```
restart : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) : with(PolynomialTools) :
    with(ArrayTools) : with(ListTools) : with(DocumentTools) : with(combinat) :

 #Caclulate the anharmonic force constants and frequencies using VPT2

Calc_Anfreq :=proc(path, sheetname)
 local AllEnergies, Energies, All_Fconst3, All_Fconst4, All_Anfreq, a, b, g, h, i, j, k, m, n, Anfreq,
    Fconst3, Fconst4, q12, q23, q13, qrem, qrem1, qrem2, qrem3, temp1, temp2, temp3, k4, k4on, k4off,
    k4off1, k4off2, k3on, k3off1, k3off2, krem, k111, k222, k333, k112, k113, k122, k133, k223, k233,
    k123, k1111, k1122, k2222, k2233, k3333, k1133, Anconst, count, iij, kiij, sum_kiin, kiin, iiii, kiiii,
    xii, delijk, iik, jjk, ijk, kiik, kjjk, kijk, sum_kijn, kijn, iijj, kiijj, ii, ij, sum_xij, xij, vi;
 uses ExcelTools, LinearAlgebra, ArrayTools, ListTools, DocumentTools, Statistics, combinat;

  AllEnergies := Matrix(Import(path, sheetname, "A2:R58", emptycell = 0.0));

 g :=proc(x, d)
   Rounding := -infinity;
   op(1, evalf[d](x)) mod 10;
 end proc:

   All_Anfreq := Matrix(3, 17) :
   All_Fconst4 := Matrix(6, 18) :
  All_Fconst3 := Matrix(10, 18) :
for a from 2 to 18 do
   Energies := ⟨Column(AllEnergies, [1])| Column(AllEnergies, [a])⟩; b := a − 1 : h := disp(b) :
     q23 := Array([ ]) : q13 := Array([ ]) : q12 := Array([ ]) : qrem := Array([ ]) :
   for i from 1 to 57 do
     if g(Energies[i](1), 1) = 3 then Append(q23, Energies[i]) end if;
     if g(Energies[i](1), 2) = 3 then Append(q13, Energies[i]) end if;
     if g(Energies[i](1), 3) = 3 then Append(q12, Energies[i]) end if;
     if g(Energies[i](1), 1) ≠ 3 and g(Energies[i](1), 2) ≠ 3 and g(Energies[i](1), 3) ≠ 3
     then Append(qrem, Energies[i]) end if;
   end do;
   q23; q13; q12; qrem;
   temp1 := Array([ ]) : temp2 := Array([ ]) : temp3 := Array([ ]) :
   for i from 1 to 21 do
     if g(q12[i](1), 2) = 3 then Append(temp1, q12[i](2)) end if ;
     if g(q23[i](1), 3) = 3 then Append(temp2, q23[i](2)) end if ;
     if g(q13[i](1), 1) = 3 then Append(temp3, q13[i](2)) end if ;
   end do;
   k4 := ⟨temp1^+|temp2^+|temp3^+⟩ : k4on :=
```

$$seq\left(\frac{(k4[1](i)-4\cdot k4[2](i)+6\cdot k4[3](i)-4\cdot k4[4](i)+k4[5](i))}{h^4}\cdot 219474.63068, i=1..3\right):$$

$$k3on := seq\left(\frac{(k4[5](i)-2\cdot k4[4](i)+2\cdot k4[2](i)-k4[1](i))}{2\cdot h^3}\cdot 219474.63068, i=1..3\right):$$

$$krem := \left(\frac{1}{8\cdot h^3}(qrem[8](2)-qrem[4](2)-qrem[6](2)+qrem[2](2)-qrem[7](2)+qrem[3](2)+qrem[5](2)-qrem[1](2))\cdot 219474.63068\right):$$

```
   k1111 := k4on[1] : k2222 := k4on[2] : k3333 := k4on[3] : k111 := k3on[1] : k222 := k3on[2] :
```

```
  k333 := k3on[3] : k123 := krem :
temp1 := Array([ ]) : temp2 := Array([ ]) : temp3 := Array([ ]) :
for j from 2 to 4 do
  for i from 1 to 21 do
    if g(q12[i](1), 2) = j and g(q12[i](1), 1) ≠ 1 and g(q12[i](1), 1) ≠ 5 then Append(temp1,
  q12[i](2)) end if;
    if g(q23[i](1), 3) = j and g(q23[i](1), 2) ≠ 1 and g(q23[i](1), 2) ≠ 5 then Append(temp2,
  q23[i](2)) end if;
    if g(q13[i](1), 1) = j and g(q13[i](1), 3) ≠ 1 and g(q13[i](1), 3) ≠ 5 then Append(temp3,
  q13[i](2)) end if;
  end do;
 end do; k4 := ⟨temp1^+|temp2^+|temp3^+⟩ :
 k4off := seq( 1/h^4 ((k4[3](i) + k4[1](i) − 2·k4[2](i)) + (k4[9](i) + k4[7](i) − 2·k4[8](i))
   − 2·(k4[6](i) + k4[4](i) − 2·k4[5](i)))·219474.63068, i = 1 ..3) :
 k3off1 := seq( (k4[3](i) + k4[1](i) − 2·k4[2](i) − (k4[9](i) + k4[7](i) − 2·k4[8](i))) / (2·h^3)
   ·219474.63068, i = 1 ..3) :
 k3off2 := seq( (k4[9](i) + k4[3](i) − 2·k4[6](i) − (k4[7](i) + k4[1](i) − 2·k4[4](i))) / (2·h^3)
   ·219474.63068, i = 1 ..3) :
 k1122 := k4off[1] : k2233 := k4off[2] : k1133 := k4off[3] : k112 := k3off1[1] : k223 :=
   -k3off1[2] : k133 := -k3off1[3] : k122 := k3off2[1] : k233 := k3off2[2] : k113 := k3off2[3] :
 Fconst4 := ⟨[1, 1, 1, 1], k1111|[2, 2, 2, 2], k2222|[3, 3, 3, 3], k3333|[1, 1, 2, 2], k1122|[1, 1, 3, 3],
   k1133|[2, 2, 3, 3], k2233⟩^+ ; Fconst3 := ⟨[1, 1, 1], k111|[2, 2, 2], k222|[3, 3, 3], k333|[1,
   1, 2], k112|[1, 1, 3], k113|[2, 2, 1], k122|[3, 3, 1], k133|[2, 2, 3], k223|[3, 3, 2], k233|[1, 2, 3],
   k123⟩^+ ;

#Use force constants to obtain anharmonicity constants

 Anconst := Matrix(6, 2) : count := 1 :
 for i from 1 to 3 do
   sum_kiin := 0 :
   for j from 1 to 3 do
      iij := eval(permute([i, i, j]));
        for m from 1 to 10  do
           if verify(Fconst3[m](1), iij, `member`) then kiij := Fconst3[m](2) end if;
      end do;
     kiin := ((8·freq[i]^2 − 3·freq[j]^2)·kiij^2) / (freq[j]·(4·freq[i]^2 − freq[j]^2)) : sum_kiin := sum_kiin + kiin :
   end do;
    iiii := eval(permute([i, i, i, i]));
   for n from 1 to 6  do
      if verify(Fconst4[n](1), iiii, `member`) then kiiii := Fconst4[n](2) end if;
   end do;
  xii := (kiiii − sum_kiin)/16; Anconst[count, ...] := ⟨[i, i]|xii⟩;
  count := count + 1 :
end do;
  for i from 1 to 2  do
   for j from 2 to 3 do
     sum_kijn := 0 :
    if i ≠ j then
      for k from 1 to 3 do delijk := (freq[i] + freq[j] − freq[k]) · (freq[i] + freq[j] + freq[k])
 · (freq[i] − freq[j] + freq[k]) · (freq[i] − freq[j] − freq[k]); kiik := 0 : kjjk := 0 : kijk := 0 :
      iik := eval(permute([i, i, k])); jjk := eval(permute([j, j, k]));
      if i = k then ijk := eval(permute([i, j, i])); elif j = k then ijk := eval(permute([i, k, k]));
  else ijk := eval(permute([i, j, k])); end if;
        for m from 1 to 10  do
          if verify(Fconst3[m](1), iik, `member`) then kiik := Fconst3[m](2) end if;
          if verify(Fconst3[m](1), jjk, `member`) then kjjk := Fconst3[m](2) end if;
          if i = k then
            if verify(Fconst3[m](1), ijk, `member`) then kijk := Fconst3[m](2) end if;
         elif j = k then
            if verify(Fconst3[m](1), ijk, `member`) then kijk := Fconst3[m](2) end if;
         else
           if verify(Fconst3[m](1), ijk, `member`) then kijk := Fconst3[m](2) end if;
          end if;
        end do;
        kiik; kjjk; kijk;   kijn := - (kiik·kjjk)/freq[k]
 + (2·freq[k]·(freq[i]^2 + freq[j]^2 - freq[k]^2)·kijk^2)/delijk   : sum_kijn := sum_kijn + kijn :
      end do;

       iijj := eval(permute([i, i, j, j]));
      for n from 1 to 6 do
       if verify(Fconst4[n](1), iijj, `member`) then kiijj := Fconst4[n](2) end if;
      end do;
     xij := (kiijj + sum_kijn)/4; Anconst[count, ...] := ⟨[i, j]|xij⟩;
      count := count + 1 :

     end if;
   end do;
 end do;

 #Use anharmonicity constants to obtain anharmonic vibrational frequencies

Anfreq := Vector[row](0) :
printlevel := 0 :
for i from 1 to 3 do
  sum_xij := 0 :
  ii := eval(permute([i, i]));
  for n from 1 to 6 do if verify(Anconst[n](1), ii, `member`) then xii := Anconst[n](2) end if; end
```

```
      do;
      for j from 1 to 3 do
        if i ≠ j then
          ij := eval(permute([i,j]));
          for n from 1 to 6 do  if verify(Anconst[n](1), ij, `member`) then xij := Anconst[n](2) end if;
       end do;
          xij; sum_xij := sum_xij + xij :
         end if;
       end do;
      vi := freq[i] + 2·xii + 1/2·sum_xij; Insert(Anfreq, i, vi);
    end do;
    All_Anfreq[ ..., b] := Anfreq; All_Fconst4[ ..., a] := Column(Fconst4, 2);  All_Fconst3[ ..., a] :=
       Column(Fconst3, 2);
 end do;
 All_Fconst4[ ..., 1] := Column(Fconst4, 1); All_Fconst3[ ..., 1] := Column(Fconst3, 1);
  return All_Fconst3, All_Fconst4, All_Anfreq ;
  end proc:

 #Calculate VPT2 transition dipole moments

  Calc_AnDM :=proc(path, sheetname)
   local no, xneg, yneg, xpos, ypos, Dipoles, AllEnergies, Energies, All_Fconst3, All_Fconst4,
       All_Dconst1, All_Dconst1_x, All_Dconst1_y, All_Dconst2, All_Dconst2_x, All_Dconst2_y,
       All_Dconst3, All_Dconst3_x, All_Dconst3_y, a, b, d, g, h, i, j, k, l, m, n, p, q, r, z, All_x_mom,
       All_y_mom, count, Fconst3, Fconst4, Dconst1, Dconst2, Dconst3, xdipole, ydipole, x_mom, y_mom,
       q12, q23, q13, qrem, qrem1, qrem2, qrem3, temp1, temp2, temp3, k4, k4on1, k4on, k4off1, k4off2,
       k4off3, k4off4, k3on, k3off1, k3off2, krem, k111, k222, k333, k112, k113, k122, k133, k223, k233,
       k123, k1111, k1112, k1122, k1222, k2222, k2223, k2233, k2333, k3333, k1113, k1133, k1333,
       k1123, k1223, k1233, u12, u23, u13, urem, d1, d2, d2on, d2off, d3, d3on, d3off1, d3off2, m1, m2, m3,
       m11, m22, m33, m12, m13, m23, m111, m222, m333, m112, m113, m122, m133, m223, m233,
       sum_jkl, del_jkl_1, del_jkl_2, del_ij, del_ik, del_il, ikl, kikl, jkl, kjkl, ijk, kijk, llk, kllk, mj, sum_jk,
       del_jk_1, del_jk_2, kjj, kkjj, ijkk, kijkk, ik, mik, jk, mjk, sum_j, ijj, mijj, mi, dip_i;

   uses ExcelTools, LinearAlgebra, ArrayTools, ListTools;

 g :=proc(x, d)
     Rounding := -infinity;
    op(1, evalf[d](x)) mod 10;
   end proc:
  xdipole := Matrix(57, 18) : ydipole := Matrix(57, 18) :

  no := Matrix(Import(path, sheetname, "A2:R58", emptycell = 0.0)) :
  xneg := Matrix(Import(path, sheetname, "T2:AK58", emptycell = 0.0)) :
  xpos := Matrix(Import(path, sheetname, "AM2:BD58", emptycell = 0.0)) :
  yneg := Matrix(Import(path, sheetname, "BF2:BW58", emptycell = 0.0)) :
  ypos := Matrix(Import(path, sheetname, "BY2:CP58", emptycell = 0.0)) :

  #Obtain dipole moments with respect to displacement

 xdipole[ ..., 1] := Column(no, 1) : ydipole[ ..., 1] := Column(no, 1) :
```

```
for j from 1 to 57 do
  for i from 2 to 18 do
   xdipole[j, i] := subs(x = 0, diff( -PolynomialFit(2, charges, Row(⟨Column(xneg, [i])
   |Column(no, [i])|Column(xpos, [i])⟩, j), x), x));
    ydipole[j, i] := subs(x = 0, diff( -PolynomialFit(2, charges, Row(⟨Column(yneg, [i])
   |Column(no, [i])|Column(ypos, [i])⟩, j), x), x));
   end do;
end do;

#Calculate force constants

AllEnergies := no :
   All_Fconst4 := Matrix(15, 18) : All_Fconst3 := Matrix(10, 18) : All_Dconst1_x := Matrix(3,
   18) : All_Dconst1_y := Matrix(3, 18) : All_Dconst2_x := Matrix(6, 18) : All_Dconst2_y :=
   Matrix(6, 18) : All_Dconst3_x := Matrix(9, 18) : All_Dconst3_y := Matrix(9, 18) :
   All_x_mom := Matrix(3, 17) : All_y_mom := Matrix(3, 17) :
for a from 2 to 18 do
  b := a − 1 : h := disp(b) :   q23 := Array([ ]) : q13 := Array([ ]) : q12 := Array([ ]) : qrem :=
   Array([ ]) :

  Energies := ⟨Column(AllEnergies, [1])| Column(AllEnergies, [a])⟩ :
 for i from 1 to 57 do
   if g(Energies[i](1), 1) = 3 then Append(q23, Energies[i]) end if;
   if g(Energies[i](1), 2) = 3 then Append(q13, Energies[i]) end if;
   if g(Energies[i](1), 3) = 3 then Append(q12, Energies[i]) end if;
   if g(Energies[i](1), 1) ≠ 3 and g(Energies[i](1), 2) ≠ 3 and g(Energies[i](1), 3) ≠ 3
   then Append(qrem, Energies[i]) end if;
  end do:
 q23; q13; q12; qrem;
  temp1 := Array([ ]) : temp2 := Array([ ]) : temp3 := Array([ ]) :
 for i from 1 to 21 do
   if g(q12[i](1), 2) = 3 then Append(temp1, q12[i](2)) end if;
   if g(q23[i](1), 3) = 3 then Append(temp2, q23[i](2)) end if;
   if g(q13[i](1), 1) = 3 then Append(temp3, q13[i](2)) end if;
 end do;
   k4 := ⟨temp1^+|temp2^+|temp3^+⟩ : k4on :=
   $seq\left(\frac{(k4[1](i) - 4\cdot k4[2](i) + 6\cdot k4[3](i) - 4\cdot k4[4](i) + k4[5](i))}{h^4}\cdot 219474.63068, i = 1..3\right)$ :
   $k3on := seq\left(\frac{(k4[5](i) - 2\cdot k4[4](i) + 2\cdot k4[2](i) - k4[1](i))}{2\cdot h^3}\cdot 219474.63068, i = 1..3\right)$ :
   $krem := \left(\frac{1}{8\cdot h^3}(qrem[8](2) - qrem[4](2) - qrem[6](2) + qrem[2](2) - qrem[7](2)$
   $+ qrem[3](2) + qrem[5](2) - qrem[1](2))\cdot 219474.63068\right)$ :
 k1111 := k4on[1] : k2222 := k4on[2] : k3333 := k4on[3] : k111 := k3on[1] : k222 := k3on[2] :
   k333 := k3on[3] : k123 := krem :
  temp1 := Array([ ]) : temp2 := Array([ ]) : temp3 := Array([ ]) :
   for j from 2 to 4 do
     for i from 1 to 21 do
    if g(q12[i](1), 2) = j and g(q12[i](1), 1) ≠ 1 and g(q12[i](1), 1) ≠ 5 then Append(temp1,
  q12[i](2)) end if;
    if g(q23[i](1), 3) = j and g(q23[i](1), 2) ≠ 1 and g(q23[i](1), 2) ≠ 5 then Append(temp2,
  q23[i](2)) end if;
    if g(q13[i](1), 1) = j and g(q13[i](1), 3) ≠ 1 and g(q13[i](1), 3) ≠ 5 then Append(temp3,
  q13[i](2)) end if;
  end do;
 end do;
 k4 := $\langle temp1^{+}|temp2^{+}|temp3^{+}\rangle$ :
 k4off1 := seq( $\frac{1}{h^4}$ ((k4[3](i) + k4[1](i) − 2·k4[2](i)) + (k4[9](i) + k4[7](i) − 2·k4[8](i))
   − 2·(k4[6](i) + k4[4](i) − 2·k4[5](i)))·219474.63068, i = 1..3) :
 k3off1 := seq( $\frac{(k4[3](i) + k4[1](i) - 2\cdot k4[2](i) - (k4[9](i) + k4[7](i) - 2\cdot k4[8](i)))}{2\cdot h^3}$
   ·219474.63068, i = 1..3) :
 k3off2 := seq( $\frac{(k4[9](i) + k4[3](i) - 2\cdot k4[6](i) - (k4[7](i) + k4[1](i) - 2\cdot k4[4](i)))}{2\cdot h^3}$
   ·219474.63068, i = 1..3) :
 k1122 := k4off1[1] : k2233 := k4off1[2] : k1133 := k4off1[3] : k112 := k3off1[1] : k223 :=
   -k3off1[2] : k133 := -k3off1[3] : k122 := k3off2[1] : k233 := k3off2[2] : k113 := k3off2[3] :
temp1 := Array([ ]) : temp2 := Array([ ]) : temp3 := Array([ ]) :
 for i from 1 to 21 do
   if g(q12[i](1), 1) ≠ 3 and g(q12[i](1), 2) ≠ 3 then Append(temp1, q12[i](2)) end if;
   if g(q23[i](1), 2) ≠ 3 and g(q23[i](1), 3) ≠ 3 then Append(temp2, q23[i](2)) end if;
   if g(q13[i](1), 3) ≠ 3 and g(q13[i](1), 1) ≠ 3 then Append(temp3, q13[i](2)) end if;
end do;
k4 := $\langle temp1^{+}|temp2^{+}|temp3^{+}\rangle$ :
k4off2 := seq( $\frac{1}{4\cdot h^4}$ (k4[12](i) -2· k4[9](i) + 2·k4[5](i) − k4[2](i)- k4[11](i) + 2·k4[8](i)
   − 2·k4[4](i) + k4[1](i))·219474.63068, i = 1..3) :
k4off3 := seq( $\frac{1}{4\cdot h^4}$ (k4[10](i) -2· k4[9](i) + 2·k4[8](i) − k4[7](i)- k4[6](i) + 2·k4[5](i)
   − 2·k4[4](i) + k4[3](i))·219474.63068, i = 1..3) :
k1112 := k4off2[1]; k1222 := k4off3[1]; k2223 := k4off2[2]; k2333 := k4off3[2]; k1113 :=
  k4off2[3]; k1333 := k4off3[3];

 qrem1 := Array([ ]) : qrem2 := Array([ ]) : qrem3 := Array([ ]) :
 for j in [2, 4] do
   for k in [2, 4] do
       for i from 1 to 8 do
         if g(qrem[i](1), 2) = j and g(qrem[i](1), 3) = k then Append(qrem1, qrem[i](2)) end if;
         if g(qrem[i](1), 1) = j and g(qrem[i](1), 3) = k then Append(qrem2, qrem[i](2)) end if;
```

```
        if g(qrem[i](1), 1) = j and g(qrem[i](1), 2) = k then Append(qrem3, qrem[i](2)) end if;
      end do:
    end do:
  end do:

  for i from 1 to 21 do
    for j in [2, 4] do
      for k in [2, 4] do
        if g(q23[i](1), 2) = j and g(q23[i](1), 3) = k then Append(qrem1, q23[i](2)) end if;
        if g(q13[i](1), 1) = j and g(q13[i](1), 3) = k then Append(qrem2, q13[i](2)) end if;
        if g(q12[i](1), 1) = j and g(q12[i](1), 2) = k then Append(qrem3, q12[i](2)) end if;
      end do:
    end do:
  end do:
  k4 := ⟨qrem1^+|qrem2^+|qrem3^+⟩ :
  k4off4 := seq(( 1/(4·h^4) ((k4[8](i) + k4[7](i) − 2·k4[12](i)) − (k4[4](i) + k4[3](i) − 2
    ·k4[10](i)) − (k4[6](i) + k4[5](i) − 2·k4[11](i)) + (k4[2](i) + k4[1](i) − 2·k4[9](i))))
    ·219474.63068, i = 1 ..3) :
    k1123 := k4off4[1] : k1223 := k4off4[2] : k1233 := k4off4[3] :
    Fconst4 := ⟨[1, 1, 1, 1], k1111|[2, 2, 2, 2], k2222|[3, 3, 3, 3], k3333|[1, 1, 2, 2], k1122|[1, 1, 3,
    3], k1133|[2, 2, 3, 3], k2233|[1, 1, 1, 2], k1112|[1, 1, 1, 3], k1113|[1, 2, 2, 2], k1222|[2, 2, 2, 3],
    k2223|[1, 3, 3, 3], k1333|[2, 3, 3, 3], k2333| [1, 1, 2, 3], k1123|[1, 2, 2, 3], k1223|[1, 2, 3, 3],
    k1233⟩ ^+ ;
    Fconst3 := ⟨[1, 1, 1], k111|[2, 2, 2], k222|[3, 3, 3], k333|[1, 1, 2], k112|[1, 1, 3], k113|[1, 2, 2],
     k122|[1, 3, 3], k133|[2, 2, 3], k223|[2, 3, 3], k233|[1, 2, 3], k123⟩ ^+ ;

#Calculate dipole constants

   count := 1 :  x_mom := Vector[row](0) : y_mom := Vector[row](0) :
  for d in [xdipole, ydipole] do
    Dipoles := ⟨Column(d, [1])| Column(d, [a])⟩;
    u23 := Array([ ]) : u13 := Array([ ]) : u12 := Array([ ]) : urem := Array([ ]) :
    for i from 1 to 57 do
      if g(Dipoles[i](1), 1) = 3 then Append(u23, Dipoles[i]) end if;
      if g(Dipoles[i](1), 2) = 3 then Append(u13, Dipoles[i]) end if;
      if g(Dipoles[i](1), 3) = 3 then Append(u12, Dipoles[i]) end if;
    end do:

   temp1 := Array([ ]) : temp2 := Array([ ]) : temp3 := Array([ ]) :
   for j from 2 to 4 do
      for k from 2 to 4 do
         for i from 1 to 21 do
            if g(u12[i](1), 1) = j and g(u12[i](1), 2) = k then Append(temp1, u12[i](2)) end if;
            if g(u23[i](1), 2) = j and g(u23[i](1), 3) = k then Append(temp2, u23[i](2)) end if;
            if g(u13[i](1), 3) = j and g(u13[i](1), 1) = k then Append(temp3, u13[i](2)) end if;
         end do:
```

```
    end do:
  end do:
 $d1 := seq\left(\frac{(d2[8](i) - d2[2](i))}{2 \cdot h}, i=1..3\right)$ :
 $d2 := \langle temp1^{+}|temp2^{+}|temp3^{+}\rangle$ : $d2on := seq\left(\frac{(d2[8](i) + d2[2](i) - 2 \cdot d2[5](i))}{h^{2}}, i=1..3\right)$ : $d2off := seq\left(\frac{(d2[9](i) - d2[3](i) - d2[7](i) + d2[1](i))}{4 \cdot h^{2}}, i=1..3\right)$ :
 m1 := d1[1] : m2 := d1[2] : m3 := d1[3] : m11 := d2on[1] : m22 := d2on[2] : m33 := d2on[3] :
   m12 := d2off[1] : m23 := d2off[2] : m13 := d2off[3] :

 temp1 := Array([ ]) : temp2 := Array([ ]) : temp3 := Array([ ]) :
  for i from 1 to 21 do
    if g(u12[i](1), 2) = 3 then Append(temp1, u12[i](2)) end if;
    if g(u23[i](1), 3) = 3 then Append(temp2, u23[i](2)) end if;
    if g(u13[i](1), 1) = 3 then Append(temp3, u13[i](2)) end if;
  end do;
 $d3 := \langle temp1^{+}|temp2^{+}|temp3^{+}\rangle$ : $d3on := seq\left(\frac{(d3[5](i) - 2 \cdot d3[4](i) + 2 \cdot d3[2](i) - d3[1](i))}{2 \cdot h^{3}}, i=1..3\right)$ :
 m111 := d3on[1] : m222 := d3on[2] : m333 := d3on[3] :
 temp1 := Array([ ]) : temp2 := Array([ ]) : temp3 := Array([ ]) :
  for j from 2 to 4 do
    for i from 1 to 21 do
      if g(u12[i](1), 2) = j and g(u12[i](1), 1) ≠ 1 and g(u12[i](1), 1) ≠ 5 then Append(temp1, u12[i](2)) end if;
      if g(u23[i](1), 3) = j and g(u23[i](1), 2) ≠ 1 and g(u23[i](1), 2) ≠ 5 then Append(temp2, u23[i](2)) end if;
      if g(u13[i](1), 1) = j and g(u13[i](1), 3) ≠ 1 and g(u13[i](1), 3) ≠ 5 then Append(temp3, u13[i](2)) end if;
    end do;
  end do;
 $d3 := \langle temp1^{+}|temp2^{+}|temp3^{+}\rangle$ :  $d3off1 := seq\left(\frac{(d3[3](i) + d3[1](i) - 2 \cdot d3[2](i) - (d3[9](i) + d3[7](i) - 2 \cdot d3[8](i)))}{2 \cdot h^{3}}, i=1..3\right)$ :
 $d3off2 := seq\left(\frac{(d3[9](i) + d3[3](i) - 2 \cdot d3[6](i) - (d3[7](i) + d3[1](i) - 2 \cdot d3[4](i)))}{2 \cdot h^{3}}, i=1..3\right)$ :
 m112 := d3off1[1] : m223 := -d3off1[2] : m133 := -d3off1[3] : m122 := d3off2[1] : m233 :=
   d3off2[2] : m113 := d3off2[3] :
 $Dconst1 := \langle [1], m1|[2], m2|[3], m3\rangle^{+}$ : $Dconst2 := \langle [1, 1], m11|[2, 2], m22|[3, 3], m33|[1, 2], m12|[1, 3], m13|[2, 3], m23\rangle^{+}$ :
 $Dconst3 := \langle [1, 1, 1], m111|[2, 2, 2], m222|[3, 3, 3], m333|[1, 1, 2], m112|[1, 1, 3], m113|[1, 2, 2], m122|[1, 3, 3], m133|[2, 2, 3], m223|[2, 3, 3], m233\rangle^{+}$ :

#Use force and dipole constants to calculate transition dipole moments
```

```
for i from 1 to 3 do
  sum_jkl ≔ 0 :
  for j from 1 to 3 do  del_ij ≔ ifelse(i = j, 1, 0) :
    for k from 1 to 3 do  del_ik ≔ ifelse(i = k, 1, 0) :
      for l from 1 to 3 do  del_il ≔ ifelse(i = l, 1, 0) :
        del_jkl_1 ≔ ((1 − del_ij)·(1 − del_ik)·(1 − del_il)·
    '(4·freq[j]·(freq[l] + freq[k]))/((freq[i]^2 − freq[j]^2)·(freq[i]^2 − (freq[l] + freq[k])^2))') - (del_ij·(1 + del_ik)·(1 − del_il)·
    '((freq[l] + freq[k])·(3·freq[i]^2 − (freq[l] + freq[k])^2))/(freq[i]·(freq[i]^2 − (freq[l] + freq[k])^2)^2)') - ((1 − del_ij)·(1 − del_ik)·del_il·
    '(4·freq[j]·(7·freq[i]·freq[k] + 3·freq[k]·freq[j] + 3·freq[k]^2 + 4·freq[i]·freq[j] + 4
    ·freq[i]^2)) / (freq[k]·(2·freq[i] + freq[k])·(freq[i]^2 − freq[j]^2)·(freq[i] + freq[j]
    + freq[k]))');

        del_jkl_2 ≔ (del_ij/(freq[i]·freq[k]) ·'(1 + 2/9·del_ik·del_il)') − ((1 − del_ij)·(1 − del_ik)·(1
    − del_il)·'(4·freq[j])/(freq[k]·(freq[i]^2 − freq[j]^2))') − (del_ik·(1 − del_ij)·
    '(4·freq[j])/(freq[i]·(freq[i]^2 − freq[j]^2)) ·(1 + 2/3·del_il)');
        ikl ≔ eval(permute([i, k, l])) : jkl ≔ eval(permute([j, k, l])) : ijk ≔ eval(permute([i, j,
    k])) : llk ≔ eval(permute([l, l, k])) :

        for m from 1 to 10 do
          if verify(Fconst3[m](1), ikl, `member`) then kikl ≔ Fconst3[m](2) end if;
          if verify(Fconst3[m](1), jkl, `member`) then kjkl ≔ Fconst3[m](2) end if;
          if verify(Fconst3[m](1), ijk, `member`) then kijk ≔ Fconst3[m](2) end if;
          if verify(Fconst3[m](1), llk, `member`) then kllk ≔ Fconst3[m](2) end if;
        end do;

        for p from 1 to 3 do
          if Dconst1[p](1) = [j] then mj ≔ Dconst1[p](2) end if;
        end do;

        jkl ≔ kikl·kjkl·mj·eval(del_jkl_1) + kijk·kllk·mj·eval(del_jkl_2);  sum_jkl ≔ sum_jkl + jkl;
      end do;
    end do;
  end do;

  sum_jk ≔ 0 :
  for j from 1 to 3 do  del_ij ≔ ifelse(i = j, 1, 0) :
    for k from 1 to 3 do  del_ik ≔ ifelse(i = k, 1, 0) :
    kijk ≔ 0 : kkjj ≔ 0 : kijkk ≔ 0 : mj ≔ 0 : mik ≔ 0 : mjk ≔ 0 :
      del_jk_1 ≔ (1/(freq[i] + freq[j] + freq[k]) − 1/(freq[i] − freq[j] − freq[k])) :
      del_jk_2 ≔ (1/(freq[i] + freq[j]) − (1 − del_ij)·'1/(freq[i] − freq[j])') :
              ijk := eval(permute([i, j, k])) : kjj := eval(permute([k, j, j])) :

           for m from 1 to 10 do
                if verify(Fconst3[m](1), ijk, `member`) then kijk := Fconst3[m](2) end if;
                if verify(Fconst3[m](1), kjj, `member`) then kkjj := Fconst3[m](2) end if;
            end do;

           ijkk := eval(permute([i, j, k, k])) :
            for n from 1 to 15 do
                if verify(Fconst4[n](1), ijkk, `member`) then kijkk := Fconst4[n](2) end if;
            end do;

            for p from 1 to 3 do
               if Dconst1[p](1) = [j] then mj := Dconst1[p](2) end if;
            end do;

           ik := eval(permute([i, k])) : jk := eval(permute([j, k])) :
            for q from 1 to 6 do
                if Dconst2[q](1) = [ik] then mik := Dconst2[q](2) end if;
                if Dconst2[q](1) = [jk] then mjk := Dconst2[q](2) end if;
            end do;
          $jk := kijk \cdot mjk \cdot del\_jk\_1 + \frac{2}{freq[k]} \cdot kkjj \cdot mik + kijkk \cdot mj \cdot eval(del\_jk\_2); sum\_jk := sum\_jk + jk;$

        end do;
     end do;

      sum_j := 0 :
      for j from 1 to 3 do
        mijj := 0 :
        ijj := eval(permute([i, j, j])) :
         for r from 1 to 9 do
            if verify(Dconst3[r](1), ijj, `member`) then mijj := Dconst3[r](2) end if;
         end do;
         mijj; sum_j := sum_j + mijj;
      end do;

      mi := 0 :
      for p from 1 to 3 do
        if Dconst1[p](1) = [i] then mi := Dconst1[p](2) end if;
      end do;

#Must include coefficients in final summation: $\frac{1}{sqrt(2)}, + \frac{1}{4 \cdot sqrt(2)}, - \frac{1}{8 \cdot sqrt(2)}, + \frac{1}{16 \cdot sqrt(2)}$

 $dip\_i := evalf\left( \frac{1}{sqrt(2)} \cdot mi + \frac{1}{4 \cdot sqrt(2)} \cdot sum\_j - \frac{1}{8 \cdot sqrt(2)} \cdot sum\_jk + \frac{1}{16 \cdot sqrt(2)} \cdot sum\_jkl \right);$

 if count = 1 then Insert(x_mom, i, dip_i); elif count = 2 then Insert(y_mom, i, dip_i); end if:
   end do;
 if count = 1 then  All_Dconst1_x[ ..., a ] ≔ Column(Dconst1, 2 ) : All_Dconst2_x[ ..., a ] ≔
     Column(Dconst2, 2 ) : All_Dconst3_x[ ..., a ] ≔ Column(Dconst3, 2 ) :
elif count = 2 then All_Dconst1_y[ ..., a ] ≔ Column(Dconst1, 2 ) : All_Dconst2_y[ ..., a ] ≔
     Column(Dconst2, 2 ) : All_Dconst3_y[ ..., a ] ≔ Column(Dconst3, 2 ) : end if:
count ≔ 2 :
end do;
   All_Fconst4[ ..., a ] ≔ Column(Fconst4, 2 );  All_Fconst3[ ..., a ] ≔ Column(Fconst3, 2 );
     All_x_mom[ ..., b ] ≔ x_mom; All_y_mom[ ..., b ] ≔ y_mom;
 end do;
 All_Fconst4[ ..., 1 ] ≔ Column(Fconst4, 1 ); All_Fconst3[ ..., 1 ] ≔ Column(Fconst3, 1 );
      All_Dconst3_x[ ..., 1 ] ≔ Column(Dconst3, 1 ); All_Dconst3_y[ ..., 1 ] ≔ Column(Dconst3, 1 );
       All_Dconst2_x[ ..., 1 ] ≔ Column(Dconst2, 1 ); All_Dconst2_y[ ..., 1 ] ≔ Column(Dconst2, 1 );
     All_Dconst1_x[ ..., 1 ] ≔ Column(Dconst1, 1 ); All_Dconst1_y[ ..., 1 ] ≔ Column(Dconst1, 1 );
  return All_x_mom, All_y_mom, All_Fconst3, All_Fconst4, All_Dconst1_x, All_Dconst1_y,
     All_Dconst2_x , All_Dconst2_y, All_Dconst3_x , All_Dconst3_y, xdipole, ydipole;
 end proc:
```

C.5 Anharmonic Contributions

This worksheet is used to determine the polynomial order that best describes the vibrational modes, and the number of harmonic energy levels to include when calculating the anharmonic frequencies *via* the Linear Variational Theorem. It fits polynomial equations of 6^{th}, 8^{th}, 10^{th}, and 12^{th} to the *ab initio* potential energy curves of each vibrational mode and extracts the force constants, which are then used to calculate the Hamiltonian matrices for increasing maximum harmonic energy levels. The resultant total Hamiltonian matrices are diagonalized and the eigenvalues (anharmonic frequencies) and eigenvectors (contributions of each harmonic energy level) are extracted. It requires non-coupled energies (as used for the Double Harmonic Approximation) as well as coupled energies; the coupled energies consist of 21×21 matrices for a negative, zero, and positive energy field in each of the Cartesian directions (x and y).

```
restart : with(Physics) : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) :
    with(PolynomialTools) : with(ArrayTools) : with(ListTools) : with(DocumentTools) :
    with(combinat) : with(linalg) :
q0 ≔ (n1, n2) → if n1 = n2 then 1 else 0 fi;
q1 ≔ (n1, n2) →if n1 < 0 then 0 elif n2 < 0 then 0 elif abs(n1 − n2) = 1 then sqrt( max(n1, n2) / 2 )
    else 0 fi;
 q2 ≔ (n1, n2) → add(q1(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q3 ≔ (n1, n2) → add(q2(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q4 ≔ (n1, n2) → add(q3(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q5 ≔ (n1, n2) → add(q4(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q6 ≔ (n1, n2) → add(q5(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q7 ≔ (n1, n2) → add(q6(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q8 ≔ (n1, n2) → add(q7(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q9 ≔ (n1, n2) → add(q8(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q10 ≔ (n1, n2) → add(q9(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q11 ≔ (n1, n2) → add(q10(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q12 ≔ (n1, n2) → add(q11(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
interface(rtablesize = 20) : interface(warnlevel = 0) : interface(displayprecision =-1) :
 path ≔ "HON_Maple.xlsx" :
 sheet ≔ "ccPVTZ_HON_Dipoles" :
 Dipoles ≔ Matrix(Import(path, sheet, "A4:S24", emptycell = 0.0)) :
q1x ≔ ⟨Column(Dipoles, [2])|Column(Dipoles, [3 ..4])⟩ : q1y ≔ ⟨Column(Dipoles, [5])
    |Column(Dipoles, [6 ..7])⟩ :
q2x ≔ ⟨Column(Dipoles, [8])|Column(Dipoles, [9 ..10])⟩ : q2y ≔ ⟨Column(Dipoles, [11])
    |Column(Dipoles, [12 ..13])⟩ :
q3x ≔ ⟨Column(Dipoles, [14])|Column(Dipoles, [15 ..16])⟩ : q3y ≔ ⟨Column(Dipoles, [17])
    |Column(Dipoles, [18 ..19])⟩ :
GetFreq ≔ Matrix(Import(path, sheet, "C64:I64")) : Frequencies ≔ ⟨Column(GetFreq, [1])
    |Column(GetFreq, [4])|Column(GetFreq, [7])⟩; charges ≔ [-0.002, 0.000, 0.002] :

 #Obtain 6th through 12th order polynomial fits

c1 ≔ Column(q1x, 2)[11] : c2 ≔ Frequencies / 219474.6 : E ≔ ⟨Column(q1x, 2)|Column(q2x, 2)
    |Column(q3x, 2)⟩ :

seq(Fit(c1 + 0.5·c2(i)·x^2 + c3·x^3 + c4·x^4 + c5·x^5 + c6·x^6 , Column(Dipoles, 1), Column(E, i), x), i
    = 1 ..3) : qE6 ≔ seq(sort(%[i], x, ascending), i = 1 ..3) : qE6coeff ≔ Matrix(3, 7,
    [seq(Coefficients(qE6[i], x), i = 1 ..3)]); qE6r ≔ seq(Fit(c1 + 0.5·c2(i)·x^2 + c3·x^3 + c4·x^4
    + c5·x^5 + c6·x^6 , Column(Dipoles, 1), Column(E, i), x, output = residualstandarddeviation)
    ·219474.6, i = 1 ..3) :
 seq(Fit(c1 + 0.5·c2(i)·x^2 + c3·x^3 + c4·x^4 + c5·x^5 + c6·x^6 + c7·x^7 + c8·x^8, Column(Dipoles, 1),
    Column(E, i), x), i = 1 ..3) : qE8 ≔ seq(sort(%[i], x, ascending), i = 1 ..3) : qE8coeff ≔
    Matrix(3, 9, [seq(Coefficients(qE8[i], x), i = 1 ..3)]); qE8r ≔ seq(Fit(c1 + 0.5·c2(i)·x^2 + c3·x^3
    + c4·x^4 + c5·x^5 + c6·x^6 + c7·x^7 + c8·x^8, Column(Dipoles, 1), Column(E, i), x, output
    = residualstandarddeviation)·219474.6, i = 1 ..3) :
 seq(Fit(c1 + 0.5·c2(i)·x^2 + c3·x^3 + c4·x^4 + c5·x^5 + c6·x^6 + c7·x^7 + c8·x^8 + c9·x^9 + c10·x^10,
    Column(Dipoles, 1), Column(E, i), x), i = 1 ..3) : qE10 ≔ seq(sort(%[i], x, ascending), i = 1
```

```
  ..3) : qE10coeff := Matrix(3, 11, [seq(Coefficients(qE10[i], x), i = 1 ..3)]); qE10r := seq(Fit(c1
   + 0.5·c2(i)·x^2 + c3·x^3 + c4·x^4 + c5·x^5 + c6·x^6 + c7·x^7 + c8·x^8 + c9·x^9 + c10·x^10,
  Column(Dipoles, 1), Column(E, i), x, output = residualstandarddeviation)·219474.6, i = 1 ..3) :
seq(Fit(c1 + 0.5·c2(i)·x^2 + c3·x^3 + c4·x^4 + c5·x^5 + c6·x^6 + c7·x^7 + c8·x^8 + c9·x^9 + c10·x^10 + c11
  ·x^11 + c12·x^12, Column(Dipoles, 1), Column(E, i), x), i = 1 ..3) : qE12 := seq(sort(%[i], x,
  ascending), i = 1 ..3) : qE12coeff := Matrix(3, 13, [seq(Coefficients(qE12[i], x), i = 1 ..3)]);
  qE12r := seq(Fit(c1 + 0.5·c2(i)·x^2 + c3·x^3 + c4·x^4 + c5·x^5 + c6·x^6 + c7·x^7 + c8·x^8 + c9·x^9
   + c10·x^10 + c11·x^11 + c12·x^12, Column(Dipoles, 1), Column(E, i), x, output
   = residualstandarddeviation)·219474.6, i = 1 ..3) :
```

#Calculate q-matrices for 6th order polynomial, then obtain anharmonic vibrational frequencies (eigenvalues) and harmonic level contributions (eigenvectors)

```
 sum4qmat6vecs := Matrix(3, 6) : sum5qmat6vecs := Matrix(3, 6) : sum6qmat6vecs := Matrix(3, 6) :
 for m from 1 to 3 do
q2mat := Matrix(6, 6) : q3mat := Matrix(6, 6) : q4mat := Matrix(6, 6) : q5mat := Matrix(6, 6) :
   q6mat := Matrix(6, 6) :
for i from 0 to 4 do
for j from 0 to 4 do
 if i = j then q2mat[i + 1, j + 1] := evalf(q2(i, j)·qE6coeff[m, 3]); fi;
 q3mat[i + 1, j + 1] := evalf(q3(i, j)·qE6coeff[m, 4]);
 q4mat[i + 1, j + 1] := evalf(q4(i, j)·qE6coeff[m, 5]);
 q5mat[i + 1, j + 1] := evalf(q5(i, j)·qE6coeff[m, 6]);
 q6mat[i + 1, j + 1] := evalf(q6(i, j)·qE6coeff[m, 7]);
 od: od:
 q2mat + q2mat + q3mat + q4mat + q5mat + q6mat : sum4qmat6 := % :

for i_max in {5} do
for i from 0 to i_max do
for j from 0 to i_max do
if j = i_max or i = i_max then
if i = j then q2mat[i + 1, j + 1] := evalf(q2(i, j)·qE6coeff[m, 3]); fi;
 q3mat[i + 1, j + 1] := evalf(q3(i, j)·qE6coeff[m, 4]);
 q4mat[i + 1, j + 1] := evalf(q4(i, j)·qE6coeff[m, 5]);
 q5mat[i + 1, j + 1] := evalf(q5(i, j)·qE6coeff[m, 6]);
 q6mat[i + 1, j + 1] := evalf(q6(i, j)·qE6coeff[m, 7]); fi;
  od; od;
q2mat + q2mat + q3mat + q4mat + q5mat + q6mat :
if i_max = 5 then sum5qmat6 := % fi:
od;
sum4qmat6vec := sort([eigenvectors(sum4qmat6)]) : sum5qmat6vec :=
   sort([eigenvectors(sum5qmat6)]) :
 for n from 1 to RowDimension(sum4qmat6vec) do sum4qmat6vecs[m, n] := sum4qmat6vec[n] od;
 for n from 1 to RowDimension(sum5qmat6vec) do sum5qmat6vecs[m, n] := sum5qmat6vec[n] od;
od:
```

#Calculate q-matrices for 8th order polynomial, then obtain anharmonic vibrational frequencies (eigenvalues) and harmonic level contributions (eigenvectors)

```
 sum4qmat8vecs ≔ Matrix(3, 8) : sum5qmat8vecs ≔ Matrix(3, 8) : sum6qmat8vecs ≔ Matrix(3,
    8) : sum7qmat8vecs ≔ Matrix(3, 8) :
 for m from 1 to 3 do
q2mat ≔ Matrix(8, 8) : q3mat ≔ Matrix(8, 8) : q4mat ≔ Matrix(8, 8) : q5mat ≔ Matrix(8, 8) :
    q6mat ≔ Matrix(8, 8) : q7mat ≔ Matrix(8, 8) : q8mat ≔ Matrix(8, 8) :
for i from 0 to 4 do
for j from 0 to 4do
 if i = j then q2mat[i + 1, j + 1] ≔ evalf(q2(i, j) · qE8coeff[m, 3]); fi;
 q3mat[i + 1, j + 1] ≔ evalf(q3(i, j) · qE8coeff[m, 4]);
 q4mat[i + 1, j + 1] ≔ evalf(q4(i, j) · qE8coeff[m, 5]);
 q5mat[i + 1, j + 1] ≔ evalf(q5(i, j) · qE8coeff[m, 6]);
 q6mat[i + 1, j + 1] ≔ evalf(q6(i, j) · qE8coeff[m, 7]);
 q7mat[i + 1, j + 1] ≔ evalf(q7(i, j) · qE8coeff[m, 8]);
 q8mat[i + 1, j + 1] ≔ evalf(q8(i, j) · qE8coeff[m, 9]);

 od: od:
 q2mat + q2mat + q3mat + q4mat + q5mat + q6mat + q7mat + q8mat : sum4qmat8 ≔ % :

for i_max in {5, 6, 7} do
for i from 0 to i_max do
for j from 0 to i_max do
if j = i_max or i = i_max then
 if i = j then q2mat[i + 1, j + 1] ≔ evalf(q2(i, j) · qE8coeff[m, 3]); fi;
 q3mat[i + 1, j + 1] ≔ evalf(q3(i, j) · qE8coeff[m, 4]);
 q4mat[i + 1, j + 1] ≔ evalf(q4(i, j) · qE8coeff[m, 5]);
 q5mat[i + 1, j + 1] ≔ evalf(q5(i, j) · qE8coeff[m, 6]);
 q6mat[i + 1, j + 1] ≔ evalf(q6(i, j) · qE8coeff[m, 7]);
 q7mat[i + 1, j + 1] ≔ evalf(q7(i, j) · qE8coeff[m, 8]);
 q8mat[i + 1, j + 1] ≔ evalf(q8(i, j) · qE8coeff[m, 9]); fi;
 od; od;
q2mat + q2mat + q3mat + q4mat + q5mat + q6mat + q7mat + q8mat :
if i_max = 5 then sum5qmat8 ≔ % fi:
if i_max = 6 then sum6qmat8 ≔ % fi:
if i_max = 7 then sum7qmat8 ≔ % fi:
od;
sum5qmat8; sum6qmat8; sum7qmat8;
sum4qmat8vec ≔ sort([eigenvectors(sum4qmat8)]) : sum5qmat8vec ≔
    sort([eigenvectors(sum5qmat8)]) : sum6qmat8vec ≔ sort([eigenvectors(sum6qmat8)]) :
    sum7qmat8vec ≔ sort([eigenvectors(sum7qmat8)]) :
 for n from 1 to RowDimension(sum4qmat8vec) do sum4qmat8vecs[m, n] ≔ sum4qmat8vec[n] od;
 for n from 1 to RowDimension(sum5qmat8vec) do sum5qmat8vecs[m, n] ≔ sum5qmat8vec[n] od;
 for n from 1 to RowDimension(sum6qmat8vec) do sum6qmat8vecs[m, n] ≔ sum6qmat8vec[n] od;
 for n from 1 to RowDimension(sum7qmat8vec) do sum7qmat8vecs[m, n] ≔ sum7qmat8vec[n] od;
od:


    #Calculate q-matrices for 10th order polynomial, then obtain anharmonic vibrational frequencies
    (eigenvalues) and harmonic level contributions (eigenvectors)

  sum4qmat10vecs ≔ Matrix(3, 10) : sum5qmat10vecs ≔ Matrix(3, 10) : sum6qmat10vecs ≔
    Matrix(3, 10) : sum7qmat10vecs ≔ Matrix(3, 10) : sum8qmat10vecs ≔ Matrix(3, 10) :
```

```
   sum9qmat10vecs := Matrix(3, 10) :
 for m from 1 to 3 do
q2mat := Matrix(10, 10) : q3mat := Matrix(10, 10) : q4mat := Matrix(10, 10) : q5mat :=
    Matrix(10, 10) : q6mat := Matrix(10, 10) : q7mat := Matrix(10, 10) : q8mat := Matrix(10, 10) :
    q9mat := Matrix(10, 10) : q10mat := Matrix(10, 10) :
for i from 0 to 4 do
for j from 0 to 4do
 if i = j then q2mat[i + 1, j + 1] := evalf(q2(i, j) · qE10coeff[m, 3]); fi;
 q3mat[i + 1, j + 1] := evalf(q3(i, j) · qE10coeff[m, 4]);
 q4mat[i + 1, j + 1] := evalf(q4(i, j) · qE10coeff[m, 5]);
 q5mat[i + 1, j + 1] := evalf(q5(i, j) · qE10coeff[m, 6]);
 q6mat[i + 1, j + 1] := evalf(q6(i, j) · qE10coeff[m, 7]);
 q7mat[i + 1, j + 1] := evalf(q7(i, j) · qE10coeff[m, 8]);
 q8mat[i + 1, j + 1] := evalf(q8(i, j) · qE10coeff[m, 9]);
 q9mat[i + 1, j + 1] := evalf(q9(i, j) · qE10coeff[m, 10]);
 q10mat[i + 1, j + 1] := evalf(q10(i, j) · qE10coeff[m, 11]);
 od: od:
 q2mat + q2mat + q3mat + q4mat + q5mat + q6mat + q7mat + q8mat + q9mat + q10mat :
    sum4qmat10 := % :

for i_max in {5, 6, 7, 8, 9} do
for i from 0 to i_max do
for j from 0 to i_max do
if j = i_max or i = i_max then
 if i = j then q2mat[i + 1, j + 1] := evalf(q2(i, j) · qE10coeff[m, 3]); fi;
 q3mat[i + 1, j + 1] := evalf(q3(i, j) · qE10coeff[m, 4]);
 q4mat[i + 1, j + 1] := evalf(q4(i, j) · qE10coeff[m, 5]);
 q5mat[i + 1, j + 1] := evalf(q5(i, j) · qE10coeff[m, 6]);
 q6mat[i + 1, j + 1] := evalf(q6(i, j) · qE10coeff[m, 7]);
 q7mat[i + 1, j + 1] := evalf(q7(i, j) · qE10coeff[m, 8]);
 q8mat[i + 1, j + 1] := evalf(q8(i, j) · qE10coeff[m, 9]);
 q9mat[i + 1, j + 1] := evalf(q9(i, j) · qE10coeff[m, 10]);
 q10mat[i + 1, j + 1] := evalf(q10(i, j) · qE10coeff[m, 11]); fi;
 od; od;
q2mat + q2mat + q3mat + q4mat + q5mat + q6mat + q7mat + q8mat + q9mat + q10mat :
if i_max = 5 then sum5qmat10 := % fi:
if i_max = 6 then sum6qmat10 := % fi:
if i_max = 7 then sum7qmat10 := % fi:
if i_max = 8 then sum8qmat10 := % fi:
if i_max = 9 then sum9qmat10 := % fi:
od;
sum5qmat10; sum6qmat10; sum7qmat10; sum8qmat10; sum9qmat10;
sum4qmat10vec := sort([eigenvectors(sum4qmat10)]) : sum5qmat10vec :=
    sort([eigenvectors(sum5qmat10)]) : sum6qmat10vec := sort([eigenvectors(sum6qmat10)]) :
    sum7qmat10vec := sort([eigenvectors(sum7qmat10)]) : sum8qmat10vec :=
    sort([eigenvectors(sum8qmat10)]) : sum9qmat10vec := sort([eigenvectors(sum9qmat10)]) :
 for n from 1 to RowDimension(sum4qmat10vec) do sum4qmat10vecs[m, n] := sum4qmat10vec[n]
    od;
 for n from 1 to RowDimension(sum5qmat10vec) do sum5qmat10vecs[m, n] := sum5qmat10vec[n]
    od;
 for n from 1 to RowDimension(sum6qmat10vec) do sum6qmat10vecs[m, n] := sum6qmat10vec[n]
    od;
 for n from 1 to RowDimension(sum7qmat10vec) do sum7qmat10vecs[m, n] ≔ sum7qmat10vec[n]
     od;
 for n from 1 to RowDimension(sum8qmat10vec) do sum8qmat10vecs[m, n] ≔ sum8qmat10vec[n]
     od;
 for n from 1 to RowDimension(sum9qmat10vec) do sum9qmat10vecs[m, n] ≔ sum9qmat10vec[n]
     od;
od:


    #Calculate q-matrices for 12th order polynomial, then obtain anharmonic vibrational frequencies
    (eigenvalues) and harmonic level contributions (eigenvectors)

  sum4qmat12vecs ≔ Matrix(3, 12) : sum5qmat12vecs ≔ Matrix(3, 12) : sum6qmat12vecs ≔
    Matrix(3, 12) : sum7qmat12vecs ≔ Matrix(3, 12) : sum8qmat12vecs ≔ Matrix(3, 12) :
     sum9qmat12vecs ≔ Matrix(3, 12) : sum10qmat12vecs ≔ Matrix(3, 12) :
 for m from 1 to 3 do
q2mat ≔ Matrix(12, 12) : q3mat ≔ Matrix(12, 12) : q4mat ≔ Matrix(12, 12) : q5mat ≔
    Matrix(12, 12) : q6mat ≔ Matrix(12, 12) : q7mat ≔ Matrix(12, 12) : q8mat ≔ Matrix(12, 12) :
    q9mat ≔ Matrix(12, 12) : q10mat ≔ Matrix(12, 12) : q11mat ≔ Matrix(12, 12) : q12mat ≔
    Matrix(12, 12) :
for i from 0 to 4 do
for j from 0 to 4do
 if i = j then q2mat[i + 1, j + 1] ≔ evalf(q2(i, j)·qE12coeff[m, 3]); fi;
 q3mat[i + 1, j + 1] ≔ evalf(q3(i, j)·qE12coeff[m, 4]);
 q4mat[i + 1, j + 1] ≔ evalf(q4(i, j)·qE12coeff[m, 5]);
 q5mat[i + 1, j + 1] ≔ evalf(q5(i, j)·qE12coeff[m, 6]);
 q6mat[i + 1, j + 1] ≔ evalf(q6(i, j)·qE12coeff[m, 7]);
 q7mat[i + 1, j + 1] ≔ evalf(q7(i, j)·qE12coeff[m, 8]);
 q8mat[i + 1, j + 1] ≔ evalf(q8(i, j)·qE12coeff[m, 9]);
 q9mat[i + 1, j + 1] ≔ evalf(q9(i, j)·qE12coeff[m, 10]);
 q10mat[i + 1, j + 1] ≔ evalf(q10(i, j)·qE12coeff[m, 11]);
 q11mat[i + 1, j + 1] ≔ evalf(q10(i, j)·qE12coeff[m, 12]);
 q12mat[i + 1, j + 1] ≔ evalf(q10(i, j)·qE12coeff[m, 13]);
 od: od:
 q2mat + q2mat + q3mat + q4mat + q5mat + q6mat + q7mat + q8mat + q9mat + q10mat + q11mat
      + q12mat : sum4qmat12 ≔ % :

for i_max in {5, 6, 7, 8, 9, 10} do
for i from 0 to i_max do
for j from 0 to i_max do
if j = i_max or i = i_max then
 if i = j then q2mat[i + 1, j + 1] ≔ evalf(q2(i, j)·qE12coeff[m, 3]); fi;
 q3mat[i + 1, j + 1] ≔ evalf(q3(i, j)·qE12coeff[m, 4]);
 q4mat[i + 1, j + 1] ≔ evalf(q4(i, j)·qE12coeff[m, 5]);
 q5mat[i + 1, j + 1] ≔ evalf(q5(i, j)·qE12coeff[m, 6]);
 q6mat[i + 1, j + 1] ≔ evalf(q6(i, j)·qE12coeff[m, 7]);
 q7mat[i + 1, j + 1] ≔ evalf(q7(i, j)·qE12coeff[m, 8]);
 q8mat[i + 1, j + 1] ≔ evalf(q8(i, j)·qE12coeff[m, 9]);
 q9mat[i + 1, j + 1] ≔ evalf(q9(i, j)·qE12coeff[m, 10]);
 q10mat[i + 1, j + 1] ≔ evalf(q10(i, j)·qE12coeff[m, 11]);
 q11mat[i + 1, j + 1] ≔ evalf(q10(i, j)·qE12coeff[m, 12]);
 q12mat[i + 1, j + 1] ≔ evalf(q10(i, j)·qE12coeff[m, 13]); fi;
 od; od;
q2mat + q2mat + q3mat + q4mat + q5mat + q6mat + q7mat + q8mat + q9mat + q10mat + q11mat
      + q12mat :
if i_max = 5 then sum5qmat12 ≔ % fi:
if i_max = 6 then sum6qmat12 ≔ % fi:
if i_max = 7 then sum7qmat12 ≔ % fi:
if i_max = 8 then sum8qmat12 ≔ % fi:
if i_max = 9 then sum9qmat12 ≔ % fi:
 if i_max = 10 then sum10qmat12 ≔ % fi:
od;
sum5qmat12; sum6qmat12; sum7qmat12; sum8qmat12; sum9qmat12; sum10qmat12;
sum4qmat12vec ≔ sort( [eigenvectors(sum4qmat12) ]) : sum5qmat12vec ≔
    sort( [eigenvectors(sum5qmat12) ]) : sum6qmat12vec ≔ sort( [eigenvectors(sum6qmat12) ]) :
    sum7qmat12vec ≔ sort( [eigenvectors(sum7qmat12) ]) : sum8qmat12vec ≔
    sort( [eigenvectors(sum8qmat12) ]) : sum9qmat12vec ≔ sort( [eigenvectors(sum9qmat12) ]) :
    sum10qmat12vec ≔ sort( [eigenvectors(sum10qmat12) ]) :
 for n from 1 to RowDimension(sum4qmat12vec) do sum4qmat12vecs[m, n] ≔ sum4qmat12vec[n]
    od;
 for n from 1 to RowDimension(sum5qmat12vec) do sum5qmat12vecs[m, n] ≔ sum5qmat12vec[n]
    od;
 for n from 1 to RowDimension(sum6qmat12vec) do sum6qmat12vecs[m, n] ≔ sum6qmat12vec[n]
    od;
 for n from 1 to RowDimension(sum7qmat12vec) do sum7qmat12vecs[m, n] ≔ sum7qmat12vec[n]
    od;
 for n from 1 to RowDimension(sum8qmat12vec) do sum8qmat12vecs[m, n] ≔ sum8qmat12vec[n]
    od;
 for n from 1 to RowDimension(sum9qmat12vec) do sum9qmat12vecs[m, n] ≔ sum9qmat12vec[n]
    od;
 for n from 1 to RowDimension(sum10qmat12vec) do sum10qmat12vecs[m, n] ≔
    sum10qmat12vec[n] od;
od:
```

C.6 Linear Variational Theorem

This worksheet executes the processes ‘Coupled_Fitting’, ‘Coupled_Freq’, ‘Dipole_Fitting’, and ‘Dipole_Cont’ to calculate the anharmonic vibrational frequencies, transition dipole moments, and molar absorptivities of the molecule *via* the Linear Variational Theorem. The anharmonic frequencies are calculated using both the non-coupled and coupled Hamiltonian matrices, and the transition dipole moments are calculated using only the non-coupled Hamiltonian matrices. The number of included harmonic energy levels for each mode are specified, and a for loop used to sequence through multiple combinations of increasing maximum energy level. It outputs the non-coupled and coupled anharmonic frequencies, and the non-coupled and coupled molar absorptivities.

```
restart : with(Physics) : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) :
    with(PolynomialTools) : with(ArrayTools) : with(ListTools) : with(DocumentTools) :
    with(combinat) : with(linalg) : with(plots) : interface(rtablesize = 50) : interface(warnlevel = 0) :

#Calculate anharmonic molar absorptivities for combinations of maximum harmonic energy level for
    q1, q2, q3

for m in [5, 6, 7, 8, 9] do
 for n in [5, 6, 7, 8, 9] do
 if m ≤ n then
 if m > 5 then
qmatlist :="[q1mat, q2mat, q3mat, q4mat, q5mat, q6mat, q7mat, q8mat, q9mat, q10mat]":
for j from 1 to 6 do
HON_efit[j] : seq('%'[i], i = 1 ..nops(HON_efit[j])) : unassign('%');
 HON_mfit[j] : seq('%'[i], i = 1 ..9) : unassign('%');
 if j = 1 then seq(qmatlist[i], i = 1 ..10); unassign('%'); fi
 od:

 v1 ≔ n : v2 ≔ m : v3 ≔ 4 :
path1 ≔ "HON_Maple.xlsx" : sheet ≔ "ccPVTZ_HON_Dipoles" : path2 ≔
    "HON_coupled_fitting.xlsx" :

#Calculate anharmonic frequencies

 HON_efit ≔ Coupled_Fitting(path1, sheet, path2, "Coupling") :
 eterms ≔ [seq(HON_efit[i], i = 1 ..12)] :
 freq ≔ HON_efit[13];
HON_efreq ≔ Coupled_Freq(freq, eterms, v1, v2, v3) :
 uncoup_freq ≔ HON_efreq[8]; coup_freq ≔ HON_efreq[9];
 print(uncoup_freq, coup_freq);

#Calculate anharmonic transition dipole moments and molar absorptivities

 HON_mfit ≔ Dipole_Fitting(path1, sheet) :
mterms ≔ [seq(HON_mfit[i], i = 1 ..12)] :
efreqs ≔ [freq, seq(HON_efreq[i], i = 8 ..10)] : hmats ≔ seq(HON_efreq[i], i = 2 ..4) :
HON_dip ≔ Dipole_Cont(v1, v2, v3, efreqs, mterms, hmats) :
unc_molar_abs ≔ [HON_dip[1 ..3]];
c_molar_abs ≔ [HON_dip[4 ..6]];
print(unc_molar_abs, c_molar_abs);
 fi;od;od;
```

C.7 Coupled Fitting Process

This worksheet contains the process ‘Coupled_Fitting’. It fits the energies of the molecule to a 10^{th} order polynomial for each vibrational mode; the non-coupled energies are fit to a one-dimensional equation, and the coupled energies are fit to a two-dimensional equation. The coefficients of each fit are extracted as the force constants and saved in a text file.

```
restart : with(Physics) : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) :
    with(PolynomialTools) : with(ArrayTools) : with(ListTools) : with(DocumentTools) :
    with(combinat) : with(linalg) : with(plots) : interface(rtablesize = 50) : interface(warnlevel = 0) :

 #Process to calculate force constants (non-coupled and coupled) from fitted equations

Coupled_Fitting ≔ proc(path1, sheetname1, path2, sheetname2)

 local Dipoles, GetFreq, Frequencies, w1, w2, w3, opt, q1energies, q2energies, q3energies, q1fit, q2fit,
    q3fit, q1consts, q2consts, q3consts, q1terms, q2terms, q3terms, dots, q1, q2, q3, uncoupled_plots,
    q12energies, q13energies, q23energies, q12couple, q12diff, q12_plots, q13couple, q13diff,
    q13_plots, q23couple, q23diff, q23_plots, n1, nn, n2, q12fit, q13fit, q23fit, Efit, q12Efit, q13Efit,
    q23Efit, q12consts, q13consts, q23consts, dots2d, Erange, coupled_plots, q12terms, q13terms,
    q23terms, label;

  uses Physics, ExcelTools, LinearAlgebra, ArrayTools, Statistics, PolynomialTools, ListTools,
    DocumentTools, combinat, linalg, plots;

 #Uncoupled
 #path1≔"/home/andrea/Documents/HON_Maple.xlsx":
 #sheetname1≔"ccPVTZ_HON_Dipoles":
 Dipoles ≔ Matrix(Import(path1, sheet, "A4:S24", emptycell = 0.0)) :
 GetFreq ≔ Matrix(Import(path1, sheet, "C64:I64")) : Frequencies ≔ [GetFreq(1), GetFreq(4),
    GetFreq(7)]; w1 ≔ Frequencies[1]; w2 ≔ Frequencies[2]; w3 ≔ Frequencies[3];
 opt ≔ Dipoles[11, 3];

  q1energies ≔ ⟨Column(Dipoles, 1)|Column(Dipoles, 3)-~opt⟩ :
  q2energies ≔ ⟨Column(Dipoles, 1)|Column(Dipoles, 9)-~opt⟩ :
  q3energies ≔ ⟨Column(Dipoles, 1)|Column(Dipoles, 15)-~opt⟩ :
```

$$q1fit \coloneqq Fit\left(\frac{0.5\cdot w1}{219474.6}\cdot x^2 + c3\cdot x^3 + c4\cdot x^4 + c5\cdot x^5 + c6\cdot x^6 + c7\cdot x^7 + c8\cdot x^8 + c9\cdot x^9 + c10\cdot x^{10},\ q1energies, x\right);$$

$$q2fit \coloneqq Fit\left(\frac{0.5\cdot w2}{219474.6}\cdot x^2 + c3\cdot x^3 + c4\cdot x^4 + c5\cdot x^5 + c6\cdot x^6 + c7\cdot x^7 + c8\cdot x^8 + c9\cdot x^9 + c10\cdot x^{10},\ q2energies, x\right);$$

$$q3fit \coloneqq Fit\left(\frac{0.5\cdot w3}{219474.6}\cdot x^2 + c3\cdot x^3 + c4\cdot x^4 + c5\cdot x^5 + c6\cdot x^6 + c7\cdot x^7 + c8\cdot x^8 + c9\cdot x^9 + c10\cdot x^{10},\ q3energies, x\right);$$

```
 q1consts ≔ [ seq(coeff(q1fit, x, n), n = ldegree(q1fit) ..degree(q1fit)) ];
 q2consts ≔ [ seq(coeff(q2fit, x, n), n = ldegree(q2fit) ..degree(q2fit)) ];
 q3consts ≔ [ seq(coeff(q3fit, x, n), n = ldegree(q3fit) ..degree(q3fit)) ];

 q1terms ≔'['k200','k300','k400','k500','k600','k700','k800','k900','k1000']':
 q2terms ≔'['k020','k030','k040','k050','k060','k070','k080','k090','k0100']':
 q3terms ≔'['k002','k003','k004','k005','k006','k007','k008','k009','k0010']':
```

#Coupling terms

$$\mathit{q12energies} := \mathit{Matrix}(\mathit{Import}(\mathit{path2}, \mathit{sheetname2}, \text{"B2:V22"}, \mathit{emptycell}=0.0) - {\sim}\mathit{opt}) :$$
$$\mathit{q13energies} := \mathit{Matrix}(\mathit{Import}(\mathit{path2}, \mathit{sheetname2}, \text{"B25:V45"}, \mathit{emptycell}=0.0) - {\sim}\mathit{opt}) :$$
$$\mathit{q23energies} := \mathit{Matrix}(\mathit{Import}(\mathit{path2}, \mathit{sheetname2}, \text{"B48:V68"}, \mathit{emptycell}=0.0) - {\sim}\mathit{opt}) :$$

$$\mathit{q12couple} := \mathit{Matrix}\left(21, 21, \left[\mathit{seq}\left(\mathit{seq}\left(\left(\mathit{eval}\left(\mathit{q2fit}, x=\mathit{evalf}\left(\frac{i-1}{2}-5\right)\right)+\mathit{eval}\left(\mathit{q1fit}, x=\mathit{evalf}\left(\frac{j-1}{2}-5\right)\right)\right), j=1..21\right), i=1..21\right)\right]\right) : \mathit{q12diff} := (\mathit{q12energies}-\mathit{q12couple}) :$$
$$\mathit{q12\_plots} := \mathit{plots}[\mathit{display}](\mathit{Array}([\mathit{plots}[\mathit{matrixplot}](\mathit{q12couple}, \mathit{mplot12}, \mathit{title}=\text{"q12 Non-Coupled Surface"}), \mathit{plots}[\mathit{matrixplot}](\mathit{q12energies}, \mathit{mplot12}, \mathit{title}=\text{"q12 Coupled Surface"}), \mathit{plots}[\mathit{matrixplot}](\mathit{q12diff}, \mathit{mplot12}, \mathit{title}=\text{"q12 Coupling Variation"})]));$$
$$\mathit{q13couple} := \mathit{Matrix}\left(21, 21, \left[\mathit{seq}\left(\mathit{seq}\left(\left(\mathit{eval}\left(\mathit{q3fit}, x=\mathit{evalf}\left(\frac{i-1}{2}-5\right)\right)+\mathit{eval}\left(\mathit{q1fit}, x=\mathit{evalf}\left(\frac{j-1}{2}-5\right)\right)\right), j=1..21\right), i=1..21\right)\right]\right) : \mathit{q13diff} := (\mathit{q13energies}-\mathit{q13couple}) :$$
$$\mathit{q13\_plots} := \mathit{plots}[\mathit{display}](\mathit{Array}([\mathit{plots}[\mathit{matrixplot}](\mathit{q13couple}, \mathit{mplot13}, \mathit{title}=\text{"q13 Non-Coupled Surface"}), \mathit{plots}[\mathit{matrixplot}](\mathit{q13energies}, \mathit{mplot13}, \mathit{title}=\text{"q13 Coupled Surface"}), \mathit{plots}[\mathit{matrixplot}](\mathit{q13diff}, \mathit{mplot13}, \mathit{title}=\text{"q13 Coupling Variation"})]));$$
$$\mathit{q23couple} := \mathit{Matrix}\left(21, 21, \left[\mathit{seq}\left(\mathit{seq}\left(\left(\mathit{eval}\left(\mathit{q3fit}, x=\mathit{evalf}\left(\frac{i-1}{2}-5\right)\right)+\mathit{eval}\left(\mathit{q2fit}, x=\mathit{evalf}\left(\frac{j-1}{2}-5\right)\right)\right), j=1..21\right), i=1..21\right)\right]\right) : \mathit{q23diff} := (\mathit{q23energies}-\mathit{q23couple}) :$$

#Turn matrices into to vectors for fitting

$$\mathit{n1} := \mathit{convert}\left(\left[\mathit{seq}\left(\mathit{seq}\left(\mathit{evalf}\left(\frac{i-1}{2}-5\right), j=1..21\right), i=1..21\right)\right], \mathit{Vector}[\mathit{column}]\right) : \mathit{nn} := \mathit{nops}(\mathit{n1}) : \mathit{n2} := \mathit{convert}\left(\left[\mathit{seq}\left(\mathit{seq}\left(\mathit{evalf}\left(\frac{j-1}{2}-5\right), j=1..21\right), i=1..21\right)\right], \mathit{Vector}[\mathit{column}]\right) : \left[\mathit{seq}\left(\mathit{seq}\left(\mathit{evalf}\left(\frac{i-1}{2}-5\right), j=1..21\right), i=1..21\right)\right]:$$
$$\mathit{q12fit} := \langle \mathit{n1}|\mathit{n2}|\mathit{convert}(\mathit{q12diff}, \mathit{Vector})\rangle : \mathit{q13fit} := \langle \mathit{n1}|\mathit{n2}|\mathit{convert}(\mathit{q13diff}, \mathit{Vector})\rangle : \mathit{q23fit} := \langle \mathit{n1}|\mathit{n2}|\mathit{convert}(\mathit{q23diff}, \mathit{Vector})\rangle :$$

$$\begin{aligned}\mathit{Efit} := {} & \mathit{k112}\cdot x^{2}y + \mathit{k122}\cdot x^{1}y^{2} + \mathit{k1112}\cdot x^{3}y + \mathit{k1122}\cdot x^{2}y^{2} + \mathit{k1222}\cdot x^{1}y^{3} + \mathit{q41}\cdot x^{4}y + \mathit{q32}\cdot x^{3}y^{2} + \mathit{q23}\cdot x^{2}y^{3} \\ & + \mathit{q14}\cdot x^{1}y^{4} + + \mathit{q51}\cdot x^{5}y + \mathit{q42}\cdot x^{4}y^{2} + \mathit{q33}\cdot x^{3}y^{3} + \mathit{q24}\cdot x^{2}y^{4} + \mathit{q15}\cdot x^{1}y^{5} + \mathit{q61}\cdot x^{6}y^{1} + \mathit{q52}\cdot x^{5}y^{2} \\ & + \mathit{q43}\cdot x^{4}y^{3} + \mathit{q34}\cdot x^{3}y^{4} + \mathit{q25}\cdot x^{2}y^{5} + \mathit{q16}\cdot x^{1}y^{6} + \mathit{q71}\cdot x^{7}y^{1} + \mathit{q62}\cdot x^{6}y^{2} + \mathit{q53}\cdot x^{5}y^{3} + \mathit{q44}\cdot x^{4}y^{4} + \mathit{q35} \\ & \cdot x^{3}y^{5} + \mathit{q26}\cdot x^{2}y^{6} + \mathit{q17}\cdot x^{1}y^{7} + \mathit{q81}\cdot x^{8}y^{1} + \mathit{q72}\cdot x^{7}y^{2} + \mathit{q63}\cdot x^{6}y^{3} + \mathit{q54}\cdot x^{5}y^{4} + \mathit{q45}\cdot x^{4}y^{5} + \mathit{q36}\cdot x^{3}y^{6} \\ & + \mathit{q27}\cdot x^{2}y^{7} + \mathit{q18}\cdot x^{1}y^{8} + \mathit{q91}\cdot x^{9}y^{1} + \mathit{q82}\cdot x^{8}y^{2} + \mathit{q73}\cdot x^{7}y^{3} + \mathit{q64}\cdot x^{6}y^{4} + \mathit{q55}\cdot x^{5}y^{5} + \mathit{q46}\cdot x^{4}y^{6} \\ & + \mathit{q37}\cdot x^{3}y^{7} + \mathit{q28}\cdot x^{2}y^{8} + \mathit{q19}\cdot x^{1}y^{9} :\end{aligned}$$

$$\mathit{q12Efit} := \mathit{Fit}(\mathit{Efit}, \mathit{q12fit}, [x, y], \mathit{output}=\mathit{leastsquaresfunction}) :$$
$$\mathit{q13Efit} := \mathit{Fit}(\mathit{Efit}, \mathit{q13fit}, [x, y], \mathit{output}=\mathit{leastsquaresfunction}) :$$
$$\mathit{q23Efit} := \mathit{Fit}(\mathit{Efit}, \mathit{q23fit}, [x, y], \mathit{output}=\mathit{leastsquaresfunction}) :$$

```
q12consts ≔ [ seq(seq(coeff(coeff(q12Efit, y, m), x, n − m), m = 1 ..n − 1), n = ldegree(q12Efit)
    ..degree(q12Efit))];
q13consts ≔ [ seq(seq(coeff(coeff(q13Efit, y, m), x, n − m), m = 1 ..n − 1), n = ldegree(q13Efit)
    ..degree(q13Efit))];
q23consts ≔ [ seq(seq(coeff(coeff(q23Efit, y, m), x, n − m), m = 1 ..n − 1), n = ldegree(q23Efit)
    ..degree(q23Efit))];

q12terms :='['k210','k120','k310','k220','k130','k410','k320','k230','k140','k510','k420','k330','k240','k150
    ','k610','k520','k430','k340','k250','k160','k710','k620','k530','k440','k350','k260','k170','k810','k720',
    'k630','k540','k450','k360','k270','k180','k910','k820','k730','k640','k550','k460','k370','k280','k190']':
q13terms :='['k201','k102','k301','k202','k103','k401','k302','k203','k104','k501','k402','k303','k204','k105
    ','k601','k502','k403','k304','k205','k106','k701','k602','k503','k404','k305','k206','k107','k801','k702',
    'k603','k504','k405','k306','k207','k108','k901','k802','k703','k604','k505','k406','k307','k208','k109']':
q23terms :='['k021','k012','k031','k022','k013','k041','k032','k023','k014','k051','k042','k033','k024','k015
    ','k061','k052','k043','k034','k025','k016','k071','k062','k053','k044','k035','k026','k017','k081','k072',
    'k063','k054','k045','k036','k027','k018','k091','k082','k073','k064','k055','k046','k037','k028','k019']':

return q1terms, q2terms, q3terms, q12terms, q13terms, q23terms, q1consts, q2consts, q3consts,
    q12consts, q13consts, q23consts, Frequencies, uncoupled_plots, q12_plots, q13_plots, q23_plots,
    coupled_plots;
end proc:
```

C.8 Coupled Frequency Process

This worksheet contains the process 'Coupled_Freq'. It calculates the anharmonic vibrational frequencies of the molecule by determining q-matrices and multiplying them by their respective force constants to yield Hamiltonian matrices; the Hamiltonian matrices contain elements for all possible combinations of initial and final harmonic energy levels for modes 1, 2, and 3 (as considered simultaneously). Both a non-coupled Hamiltonian matrix and a coupled Hamiltonian matrix are calculated and diagonalized to yield the energies and wavefunction coefficients of the anharmonic energy levels. The individual non-coupled and coupled Hamiltonian matrices, total Hamiltonian matrices, and anharmonic frequencies are saved in a text file.

```
restart : with(Physics) : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) :
    with(PolynomialTools) : with(ArrayTools) : with(ListTools) : with(DocumentTools) :
    with(combinat) : with(linalg) : with(plots) : interface(rtablesize = 50) :

#Process to calculate anharmonic frequencies using obtained force constants

Coupled_Freq ≔ proc(freq, terms, kmax, jmax, imax)

  uses Physics, ExcelTools, LinearAlgebra, ArrayTools, Statistics, PolynomialTools, ListTools,
    DocumentTools, combinat, linalg, plots;

 local w1, w2, w3, q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, n1, n2, n3, nn, largest, nq, nqn,
    q1mat, q2mat, q3mat, q4mat, q5mat, q6mat, q7mat, q8mat, q9mat, q10mat, allqmat, h_harm, h_q1,
    h_q2, h_q3, h_q12_m3, h_q12_m4, h_q12_m5, h_q12_m6, h_q12_m7, h_q12_m8, h_q12_m9,
    h_q12_m10, h_q12, h_q13_m3, h_q13_m4, h_q13_m5, h_q13_m6, h_q13_m7, h_q13_m8,
    h_q13_m9, h_q13_m10, h_q13, h_q23_m3, h_q23_m4, h_q23_m5, h_q23_m6, h_q23_m7,
    h_q23_m8, h_q23_m9, h_q23_m10, h_q23, evals_no, v1no, v2no, v3no, uncoupled_freq, evals_pair,
    v1, v2, v3, coupled_freq, row1, row2, row3;
 w1 ≔ freq[1]; w2 ≔ freq[2]; w3 ≔ freq[3];

 q0 ≔ (n1, n2) → if n1 = n2 then 1 else 0 fi;
q1 ≔ (n1, n2) → if n1 < 0 then 0 elif n2 < 0 then 0 elif abs(n1 − n2) = 1 then sqrt(max(n1, n2)/2)
    else 0 fi;
 q2 ≔ (n1, n2) → add(q1(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q3 ≔ (n1, n2) → add(q2(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q4 ≔ (n1, n2) → add(q3(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q5 ≔ (n1, n2) → add(q4(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q6 ≔ (n1, n2) → add(q5(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q7 ≔ (n1, n2) → add(q6(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q8 ≔ (n1, n2) → add(q7(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q9 ≔ (n1, n2) → add(q8(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q10 ≔ (n1, n2) → add(q9(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q11 ≔ (n1, n2) → add(q10(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q12 ≔ (n1, n2) → add(q11(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :

 seq(seq(assign(terms[z][i] = terms[z + 6][i]), i = 1 ..nops(terms[z])), z = 1 ..6);
 #i=q3,j=q2,k=q1
 n1 ≔ [seq(seq(seq(k, k = 0 ..kmax), j = 0 ..jmax), i = 0 ..imax)]; nn ≔ nops(n1);
 n2 ≔ [seq(seq(seq(j, k = 0 ..kmax), j = 0 ..jmax), i = 0 ..imax)];
 n3 ≔ [seq(seq(seq(i, k = 0 ..kmax), j = 0 ..jmax), i = 0 ..imax)];

row1 ≔ [n1[1], n2[1], n3[1]]; row2 ≔ [n1[2], n2[2], n3[2]]; row3 ≔ [n1[3], n2[3], n3[3]];

 largest ≔ max(imax, jmax, kmax); nq ≔ [seq(0 ..largest)] : nqn ≔ nops(nq) :

 q1mat ≔ Matrix(nqn, nqn, [seq(seq(q1(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn)]) :
 q2mat ≔ Matrix(nqn, nqn, [seq(seq(q2(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn)]) :
 q3mat ≔ Matrix(nqn, nqn, [seq(seq(q3(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn)]) :
 q4mat ≔ Matrix(nqn, nqn, [seq(seq(q4(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn)]) :
 q5mat ≔ Matrix(nqn, nqn, [seq(seq(q5(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn)]) :
 q6mat ≔ Matrix(nqn, nqn, [seq(seq(q6(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn)]) :
```

```
q7mat ≔ Matrix(nqn, nqn, [seq(seq(q7(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn) ]) :
q8mat ≔ Matrix(nqn, nqn, [seq(seq(q8(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn) ]) :
q9mat ≔ Matrix(nqn, nqn, [seq(seq(q9(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn) ]) :
q10mat ≔ Matrix(nqn, nqn, [seq(seq(q10(nq[i], nq[j]), j = 1 ..nqn), i = 1 ..nqn) ]) :

allqmat ≔ [q1mat, q2mat, q3mat, q4mat, q5mat, q6mat, q7mat, q8mat, q9mat, q10mat];

# Define the harmonic component of the Hamiltonian matrix

h_harm ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) ·q0(n2[i], n2[j]) ·q0(n3[i], n3[j]) · (w1
    · (n1[i] + 0.5) + w2 · (n2[i] + 0.5) + w3 · (n3[i] + 0.5) ), j = 1 ..nn), i = 1 ..nn) ]) :

# Define the anharmonic contributions without coupling

h_q1 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) ·q0(n3[i], n3[j]) · (q3mat[ (n1[i] + 1), (n1[j]
    + 1) ]·k300 + q4mat[ (n1[i] + 1), (n1[j] + 1) ]·k400 + q5mat[ (n1[i] + 1), (n1[j] + 1) ]·k500
    + q6mat[ (n1[i] + 1), (n1[j] + 1) ]·k600 + q7mat[ (n1[i] + 1), (n1[j] + 1) ]·k700
    + q8mat[ (n1[i] + 1), (n1[j] + 1) ]·k800 + q9mat[ (n1[i] + 1), (n1[j] + 1) ]·k900
    + q10mat[ (n1[i] + 1), (n1[j] + 1) ]·k1000), j = 1 ..nn), i = 1 ..nn) ]) ·~219474.6 :
h_q2 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) ·q0(n3[i], n3[j]) · (q3mat[ (n2[i] + 1), (n2[j]
    + 1) ]·k030 + q4mat[ (n2[i] + 1), (n2[j] + 1) ]·k040 + q5mat[ (n2[i] + 1), (n2[j] + 1) ]·k050
    + q6mat[ (n2[i] + 1), (n2[j] + 1) ]·k060 + q7mat[ (n2[i] + 1), (n2[j] + 1) ]·k070
    + q8mat[ (n2[i] + 1), (n2[j] + 1) ]·k080 + q9mat[ (n2[i] + 1), (n2[j] + 1) ]·k090
    + q10mat[ (n2[i] + 1), (n2[j] + 1) ]·k0100), j = 1 ..nn), i = 1 ..nn) ]) ·~219474.6 :
h_q3 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) ·q0(n2[i], n2[j]) · (q3mat[ (n3[i] + 1), (n3[j]
    + 1) ]·k003 + q4mat[ (n3[i] + 1), (n3[j] + 1) ]·k004 + q5mat[ (n3[i] + 1), (n3[j] + 1) ]·k005
    + q6mat[ (n3[i] + 1), (n3[j] + 1) ]·k006 + q7mat[ (n3[i] + 1), (n3[j] + 1) ]·k007
    + q8mat[ (n3[i] + 1), (n3[j] + 1) ]·k008 + q9mat[ (n3[i] + 1), (n3[j] + 1) ]·k009
    + q10mat[ (n3[i] + 1), (n3[j] + 1) ]·k0010), j = 1 ..nn), i = 1 ..nn) ]) ·~219474.6 :

# Define the anharmonic contributions involving q1,q2 coupling

h_q12_m3 ≔ Matrix(nn, nn, [seq(seq(q0(n3[i], n3[j]) · (q2mat[ (n1[i] + 1), (n1[j] + 1) ]
    ·q1mat[ (n2[i] + 1), (n2[j] + 1) ]·k210 + q1mat[ (n1[i] + 1), (n1[j] + 1) ]·q2mat[ (n2[i]
    + 1), (n2[j] + 1) ]·k120), j = 1 ..nn), i = 1 ..nn) ]) :
h_q12_m4 ≔ Matrix(nn, nn, [seq(seq(q0(n3[i], n3[j]) · (q3mat[ (n1[i] + 1), (n1[j] + 1) ]
    ·q1mat[ (n2[i] + 1), (n2[j] + 1) ]·k310 + q2mat[ (n1[i] + 1), (n1[j] + 1) ]·q2mat[ (n2[i]
    + 1), (n2[j] + 1) ]·k220 + q1mat[ (n1[i] + 1), (n1[j] + 1) ]·q3mat[ (n2[i] + 1), (n2[j] + 1) ]
    ·k130), j = 1 ..nn), i = 1 ..nn) ]) :
h_q12_m5 ≔ Matrix(nn, nn, [seq(seq(q0(n3[i], n3[j]) · (q4mat[ (n1[i] + 1), (n1[j] + 1) ]
    ·q1mat[ (n2[i] + 1), (n2[j] + 1) ]·k410 + q3mat[ (n1[i] + 1), (n1[j] + 1) ]·q2mat[ (n2[i]
    + 1), (n2[j] + 1) ]·k320 + q2mat[ (n1[i] + 1), (n1[j] + 1) ]·q3mat[ (n2[i] + 1), (n2[j] + 1) ]
    ·k230 + q1mat[ (n1[i] + 1), (n1[j] + 1) ]·q4mat[ (n2[i] + 1), (n2[j] + 1) ]·k140), j = 1 ..nn), i
    = 1 ..nn) ]) :
h_q12_m6 ≔ Matrix(nn, nn, [seq(seq(q0(n3[i], n3[j]) · (q5mat[ (n1[i] + 1), (n1[j] + 1) ]
    ·q1mat[ (n2[i] + 1), (n2[j] + 1) ]·k510 + q4mat[ (n1[i] + 1), (n1[j] + 1) ]·q2mat[ (n2[i]
    + 1), (n2[j] + 1) ]·k420 + q3mat[ (n1[i] + 1), (n1[j] + 1) ]·q3mat[ (n2[i] + 1), (n2[j] + 1) ]
    ·k330 + q2mat[ (n1[i] + 1), (n1[j] + 1) ]·q4mat[ (n2[i] + 1), (n2[j] + 1) ]·k240
    + q1mat[ (n1[i] + 1), (n1[j] + 1) ]·q5mat[ (n2[i] + 1), (n2[j] + 1) ]·k150), j = 1 ..nn), i = 1
    ..nn) ]) :
h_q12_m7 ≔ Matrix(nn, nn, [seq(seq(q0(n3[i], n3[j]) · (q6mat[ (n1[i] + 1), (n1[j] + 1) ]
    ·q1mat[ (n2[i] + 1), (n2[j] + 1) ]·k610 + q5mat[ (n1[i] + 1), (n1[j] + 1) ]·q2mat[ (n2[i]
    + 1), (n2[j] + 1)]·k520 + q4mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n2[i] + 1), (n2[j] + 1)]
    ·k430 + q3mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n2[i] + 1), (n2[j] + 1)]·k340
    + q2mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n2[i] + 1), (n2[j] + 1)]·k250 + q1mat[(n1[i]
    + 1), (n1[j] + 1)]·q6mat[(n2[i] + 1), (n2[j] + 1)]·k160), j = 1 ..nn), i = 1 ..nn)]) :
h_q12_m8 ≔ Matrix(nn, nn, [seq(seq(q0(n3[i], n3[j]) · (q7mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n2[i] + 1), (n2[j] + 1)]·k710 + q6mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n2[i]
    + 1), (n2[j] + 1)]·k620 + q5mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n2[i] + 1), (n2[j] + 1)]
    ·k530 + q4mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n2[i] + 1), (n2[j] + 1)]·k440
    + q3mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n2[i] + 1), (n2[j] + 1)]·k350 + q2mat[(n1[i]
    + 1), (n1[j] + 1)]·q6mat[(n2[i] + 1), (n2[j] + 1)]·k260 + q1mat[(n1[i] + 1), (n1[j] + 1)]
    ·q7mat[(n2[i] + 1), (n2[j] + 1)]·k170), j = 1 ..nn), i = 1 ..nn)]) :
h_q12_m9 ≔ Matrix(nn, nn, [seq(seq(q0(n3[i], n3[j]) · (q8mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n2[i] + 1), (n2[j] + 1)]·k810 + q7mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n2[i]
    + 1), (n2[j] + 1)]·k720 + q6mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n2[i] + 1), (n2[j] + 1)]
    ·k630 + q5mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n2[i] + 1), (n2[j] + 1)]·k540
    + q4mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n2[i] + 1), (n2[j] + 1)]·k450 + q3mat[(n1[i]
    + 1), (n1[j] + 1)]·q6mat[(n2[i] + 1), (n2[j] + 1)]·k360 + q2mat[(n1[i] + 1), (n1[j] + 1)]
    ·q7mat[(n2[i] + 1), (n2[j] + 1)]·k270 + q1mat[(n1[i] + 1), (n1[j] + 1)]·q8mat[(n2[i]
    + 1), (n2[j] + 1)]·k180), j = 1 ..nn), i = 1 ..nn)]) :
h_q12_m10 ≔ Matrix(nn, nn, [seq(seq(q0(n3[i], n3[j]) · (q9mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n2[i] + 1), (n2[j] + 1)]·k910 + q8mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n2[i]
    + 1), (n2[j] + 1)]·k820 + q7mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n2[i] + 1), (n2[j] + 1)]
    ·k730 + q6mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n2[i] + 1), (n2[j] + 1)]·k640
    + q5mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n2[i] + 1), (n2[j] + 1)]·k550 + q4mat[(n1[i]
    + 1), (n1[j] + 1)]·q6mat[(n2[i] + 1), (n2[j] + 1)]·k460 + q3mat[(n1[i] + 1), (n1[j] + 1)]
    ·q7mat[(n2[i] + 1), (n2[j] + 1)]·k370 + q2mat[(n1[i] + 1), (n1[j] + 1)]·q8mat[(n2[i]
    + 1), (n2[j] + 1)]·k280 + q1mat[(n1[i] + 1), (n1[j] + 1)]·q9mat[(n2[i] + 1), (n2[j] + 1)]
    ·k190), j = 1 ..nn), i = 1 ..nn)]) :

h_q12 ≔ (h_q12_m3 + h_q12_m4 + h_q12_m5 + h_q12_m6 + h_q12_m7 + h_q12_m8 + h_q12_m9
    + h_q12_m10) ·~219474.6 :

# Define the anharmonic contributions involving q1,q3 coupling

h_q13_m3 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) · (q2mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k201 + q1mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k102), j = 1 ..nn), i = 1 ..nn)]) :
h_q13_m4 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) · (q3mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k301 + q2mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k202 + q1mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k103), j = 1 ..nn), i = 1 ..nn)]) :
h_q13_m5 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) · (q4mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k401 + q3mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k302 + q2mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k203 + q1mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k104), j = 1 ..nn), i
    = 1 ..nn)]) :
h_q13_m6 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) · (q5mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k501 + q4mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k402 + q3mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k303 + q2mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k204
    + q1mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k105), j = 1 ..nn), i = 1
    ..nn)]) :
```

```
h_q13_m7 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) · (q6mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k601 + q5mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k502 + q4mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k403 + q3mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k304
    + q2mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k205 + q1mat[(n1[i]
    + 1), (n1[j] + 1)]·q6mat[(n3[i] + 1), (n3[j] + 1)]·k106), j = 1 ..nn), i = 1 ..nn)]) :
 h_q13_m8 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) · (q7mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k701 + q6mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k602 + q5mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k503 + q4mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k404
    + q3mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k305 + q2mat[(n1[i]
    + 1), (n1[j] + 1)]·q6mat[(n3[i] + 1), (n3[j] + 1)]·k206 + q1mat[(n1[i] + 1), (n1[j] + 1)]
    ·q7mat[(n3[i] + 1), (n3[j] + 1)]·k107), j = 1 ..nn), i = 1 ..nn)]) :
 h_q13_m9 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) · (q8mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k801 + q7mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k702 + q6mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k603 + q5mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k504
    + q4mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k405 + q3mat[(n1[i]
    + 1), (n1[j] + 1)]·q6mat[(n3[i] + 1), (n3[j] + 1)]·k306 + q2mat[(n1[i] + 1), (n1[j] + 1)]
    ·q7mat[(n3[i] + 1), (n3[j] + 1)]·k207 + q1mat[(n1[i] + 1), (n1[j] + 1)]·q8mat[(n3[i]
    + 1), (n3[j] + 1)]·k108), j = 1 ..nn), i = 1 ..nn)]) :
 h_q13_m10 ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) · (q9mat[(n1[i] + 1), (n1[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k901 + q8mat[(n1[i] + 1), (n1[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k802 + q7mat[(n1[i] + 1), (n1[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k703 + q6mat[(n1[i] + 1), (n1[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k604
    + q5mat[(n1[i] + 1), (n1[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k505 + q4mat[(n1[i]
    + 1), (n1[j] + 1)]·q6mat[(n3[i] + 1), (n3[j] + 1)]·k406 + q3mat[(n1[i] + 1), (n1[j] + 1)]
    ·q7mat[(n3[i] + 1), (n3[j] + 1)]·k307 + q2mat[(n1[i] + 1), (n1[j] + 1)]·q8mat[(n3[i]
    + 1), (n3[j] + 1)]·k208 + q1mat[(n1[i] + 1), (n1[j] + 1)]·q9mat[(n3[i] + 1), (n3[j] + 1)]
    ·k109), j = 1 ..nn), i = 1 ..nn)]) :

 h_q13 ≔ (h_q13_m3 + h_q13_m4 + h_q13_m5 + h_q13_m6 + h_q13_m7 + h_q13_m8 + h_q13_m9
    + h_q13_m10) ·~219474.6 :

# Define the anharmonic contributions involving q2,q3 coupling

h_q23_m3 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) · (q2mat[(n2[i] + 1), (n2[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k021 + q1mat[(n2[i] + 1), (n2[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k012), j = 1 ..nn), i = 1 ..nn)]) :
 h_q23_m4 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) · (q3mat[(n2[i] + 1), (n2[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k031 + q2mat[(n2[i] + 1), (n2[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k022 + q1mat[(n2[i] + 1), (n2[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k013), j = 1 ..nn), i = 1 ..nn)]) :
 h_q23_m5 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) · (q4mat[(n2[i] + 1), (n2[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k041 + q3mat[(n2[i] + 1), (n2[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k032 + q2mat[(n2[i] + 1), (n2[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k023 + q1mat[(n2[i] + 1), (n2[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k014), j = 1 ..nn), i
    = 1 ..nn)]) :
h_q23_m6 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) · (q5mat[(n2[i] + 1), (n2[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k051 + q4mat[(n2[i] + 1), (n2[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k042 + q3mat[(n2[i] + 1), (n2[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k033 + q2mat[(n2[i] + 1), (n2[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k024
    + q1mat[(n2[i] + 1), (n2[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k015), j = 1 ..nn), i = 1
    ..nn)]) :
h_q23_m7 := Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j])·(q6mat[(n2[i] + 1), (n2[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k061 + q5mat[(n2[i] + 1), (n2[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k052 + q4mat[(n2[i] + 1), (n2[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k043 + q3mat[(n2[i] + 1), (n2[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k034
    + q2mat[(n2[i] + 1), (n2[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k025 + q1mat[(n2[i]
    + 1), (n2[j] + 1)]·q6mat[(n3[i] + 1), (n3[j] + 1)]·k016), j = 1 ..nn), i = 1 ..nn)]) :
h_q23_m8 := Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j])·(q7mat[(n2[i] + 1), (n2[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k071 + q6mat[(n2[i] + 1), (n2[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k062 + q5mat[(n2[i] + 1), (n2[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k053 + q4mat[(n2[i] + 1), (n2[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k044
    + q3mat[(n2[i] + 1), (n2[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k035 + q2mat[(n2[i]
    + 1), (n2[j] + 1)]·q6mat[(n3[i] + 1), (n3[j] + 1)]·k026 + q1mat[(n2[i] + 1), (n2[j] + 1)]
    ·q7mat[(n3[i] + 1), (n3[j] + 1)]·k017), j = 1 ..nn), i = 1 ..nn)]) :
h_q23_m9 := Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j])·(q8mat[(n2[i] + 1), (n2[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k081 + q7mat[(n2[i] + 1), (n2[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k072 + q6mat[(n2[i] + 1), (n2[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k063 + q5mat[(n2[i] + 1), (n2[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k054
    + q4mat[(n2[i] + 1), (n2[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k045 + q3mat[(n2[i]
    + 1), (n2[j] + 1)]·q6mat[(n3[i] + 1), (n3[j] + 1)]·k036 + q2mat[(n2[i] + 1), (n2[j] + 1)]
    ·q7mat[(n3[i] + 1), (n3[j] + 1)]·k027 + q1mat[(n2[i] + 1), (n2[j] + 1)]·q8mat[(n3[i]
    + 1), (n3[j] + 1)]·k018), j = 1 ..nn), i = 1 ..nn)]) :
h_q23_m10 := Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j])·(q9mat[(n2[i] + 1), (n2[j] + 1)]
    ·q1mat[(n3[i] + 1), (n3[j] + 1)]·k091 + q8mat[(n2[i] + 1), (n2[j] + 1)]·q2mat[(n3[i]
    + 1), (n3[j] + 1)]·k082 + q7mat[(n2[i] + 1), (n2[j] + 1)]·q3mat[(n3[i] + 1), (n3[j] + 1)]
    ·k073 + q6mat[(n2[i] + 1), (n2[j] + 1)]·q4mat[(n3[i] + 1), (n3[j] + 1)]·k064
    + q5mat[(n2[i] + 1), (n2[j] + 1)]·q5mat[(n3[i] + 1), (n3[j] + 1)]·k055 + q4mat[(n2[i]
    + 1), (n2[j] + 1)]·q6mat[(n3[i] + 1), (n3[j] + 1)]·k046 + q3mat[(n2[i] + 1), (n2[j] + 1)]
    ·q7mat[(n3[i] + 1), (n3[j] + 1)]·k037 + q2mat[(n2[i] + 1), (n2[j] + 1)]·q8mat[(n3[i]
    + 1), (n3[j] + 1)]·k028 + q1mat[(n2[i] + 1), (n2[j] + 1)]·q9mat[(n3[i] + 1), (n3[j] + 1)]
    ·k019), j = 1 ..nn), i = 1 ..nn)]) :

h_q23 := (h_q23_m3 + h_q23_m4 + h_q23_m5 + h_q23_m6 + h_q23_m7 + h_q23_m8 + h_q23_m9
    + h_q23_m10)·~219474.6 :
h_no := h_harm + h_q1 + h_q2 + h_q3 :


# Calculate the eigenvalues without coupling between modes
evals_no := eigenvalues(h_no) : seq(evals_no[i], i = 1 ..10) :
# And obtain the vibrational frequencies
v3no := evals_no[2] − evals_no[1]; v2no := evals_no[3] − evals_no[1]; v1no := evals_no[9]
    − evals_no[1];
uncoupled_freq := [v1no, v2no, v3no];

h_tot := h_harm + h_q1 + h_q2 + h_q3 + h_q12 + h_q13 + h_q23 :
# Calculate the eigenvalues including coupling between pairs of modes
evals_pair := eigenvalues(h_tot) : seq(evals_pair[i], i = 1 ..10) :
# And obtain the vibrational frequencies
v3 := evals_pair[2] − evals_pair[1];  v2 := evals_pair[3] − evals_pair[1];  v1 := evals_pair[9]
    − evals_pair[1];
coupled_freq := [v1, v2, v3];
```

```
return h_harm, h_q1, h_q2, h_q3, h_q12, h_q13, h_q23, uncoupled_freq, coupled_freq, allqmat, row1,
    row2, row3, h_no, h_tot;
end proc:
```

C.9 Dipole Moment Fitting Process

This worksheet contains the process 'Dipole_Fitting'. It first calculates the dipole moment of each vibrational mode for a given displacement, and then fits the resultant values to a 9^{th} order polynomial; since only non-coupled energies are included, only a one-dimensional equation is used. The coefficients of each fit are extracted as the dipole constants and saved in a text file.

```
restart : with(Physics) : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) :
    with(PolynomialTools) : with(ArrayTools) : with(ListTools) : with(DocumentTools) :
    with(combinat) : with(linalg) : with(plots) : interface(rtablesize = 50) : interface(warnlevel = 0) :

 #Process to calculate dipole constants (non-coupled only) from fitted equations

Dipole_Fitting ≔ proc(path1, sheet)

  uses Physics, ExcelTools, LinearAlgebra, ArrayTools, Statistics, PolynomialTools, ListTools,
    DocumentTools, combinat, linalg, plots;

 local w1, w2, w3, Dipoles, GetFreq, Frequencies, charges, q1x, q2x, q3x, q1y, q2y, q3y, q1Mxscan,
    q2Mxscan, q3Mxscan, q1Myscan, q2Myscan, q3Myscan, E9, q1Mxfit, q2Mxfit, q3Mxfit, q1Myfit,
    q2Myfit, q3Myfit, q1Mxconsts, q2Mxconsts, q3Mxconsts, q1Myconsts, q2Myconsts, q3Myconsts,
    q1Mxterms, q2Mxterms, q3Mxterms, q1Myterms, q2Myterms, q3Myterms, q1Mx, q2Mx, q3Mx,
    q1My, q2My, q3My, qm1x, qm2x, qm3x, qm1y, qm2y, qm3y, dots, xdip_plots, ydip_plots;

  Dipoles ≔ Matrix(Import(path1, sheet, "A4:S24", emptycell = 0.0)) :
 GetFreq ≔ Matrix(Import(path1, sheet, "C64:I64")) : Frequencies ≔ [GetFreq(1), GetFreq(4),
    GetFreq(7)]; w1 ≔ Frequencies[1]; w2 ≔ Frequencies[2]; w3 ≔ Frequencies[3];; charges ≔ [
    -0.002, 0, 0.002];

 q1x ≔ ⟨Column(Dipoles, [2])|Column(Dipoles, [3 ..4])⟩ : q1y ≔ ⟨Column(Dipoles, [5])
    |Column(Dipoles, [6 ..7])⟩ :
 q2x ≔ ⟨Column(Dipoles, [8])|Column(Dipoles, [9 ..10])⟩ : q2y ≔ ⟨Column(Dipoles, [11])
    |Column(Dipoles, [12 ..13])⟩ :
 q3x ≔ ⟨Column(Dipoles, [14])|Column(Dipoles, [15 ..16])⟩ : q3y ≔ ⟨Column(Dipoles, [17])
    |Column(Dipoles, [18 ..19])⟩ :
 ⟨seq(subs(x = 0, diff(-PolynomialFit(2, charges, Row(q1x, [i]), x), x)), i = 1 ..21)⟩ : q1Mxscan ≔ %
    - ~%[11] :
 ⟨seq(subs(x = 0, diff(-PolynomialFit(2, charges, Row(q2x, [i]), x), x)), i = 1 ..21)⟩ : q2Mxscan ≔ %
    - ~%[11] :
 ⟨seq(subs(x = 0, diff(-PolynomialFit(2, charges, Row(q3x, [i]), x), x)), i = 1 ..21)⟩ : q3Mxscan ≔ %
    - ~%[11] :
 ⟨seq(subs(x = 0, diff(-PolynomialFit(2, charges, Row(q1y, [i]), x), x)), i = 1 ..21)⟩ : q1Myscan ≔ %
    - ~%[11] :
 ⟨seq(subs(x = 0, diff(-PolynomialFit(2, charges, Row(q2y, [i]), x), x)), i = 1 ..21)⟩ : q2Myscan ≔ %
    - ~%[11] :
 ⟨seq(subs(x = 0, diff(-PolynomialFit(2, charges, Row(q3y, [i]), x), x)), i = 1 ..21)⟩ : q3Myscan ≔ %
    - ~%[11] :

E9 ≔ c1·x^1 + c2·x^2 + c3·x^3 + c4·x^4 + c5·x^5 + c6·x^6 + c7·x^7 + c8·x^8 + c9·x^9 :

q1Mxfit ≔ Fit(E9, Column(Dipoles, 1), q1Mxscan, x) :
q2Mxfit ≔ Fit(E9, Column(Dipoles, 1), q2Mxscan, x) :
q3Mxfit ≔ Fit(E9, Column(Dipoles, 1), q3Mxscan, x) :
q1Myfit ≔ Fit(E9, Column(Dipoles, 1), q1Myscan, x) :
q2Myfit ≔ Fit(E9, Column(Dipoles, 1), q2Myscan, x) :
q3Myfit ≔ Fit(E9, Column(Dipoles, 1), q3Myscan, x) :

 q1Mxconsts ≔ [ seq(coeff(q1Mxfit, x, n), n = ldegree(q1Mxfit) ..degree(q1Mxfit))] :
 q2Mxconsts ≔ [ seq(coeff(q2Mxfit, x, n), n = ldegree(q2Mxfit) ..degree(q2Mxfit))] :
```

```
q3Mxconsts := [ seq(coeff(q3Mxfit, x, n), n = ldegree(q3Mxfit) ..degree(q3Mxfit)) ] :
q1Myconsts := [ seq(coeff(q1Myfit, x, n), n = ldegree(q1Myfit) ..degree(q1Myfit)) ] :
q2Myconsts := [ seq(coeff(q2Myfit, x, n), n = ldegree(q2Myfit) ..degree(q2Myfit)) ] :
q3Myconsts := [ seq(coeff(q3Myfit, x, n), n = ldegree(q3Myfit) ..degree(q3Myfit)) ] :

q1Mxterms :='[ 'mx100','mx200','mx300','mx400','mx500','mx600','mx700','mx800','mx900' ]':
q2Mxterms :='[ 'mx010','mx020','mx030','mx040','mx050','mx060','mx070','mx080','mx090' ]':
q3Mxterms :='[ 'mx001','mx002','mx003','mx004','mx005','mx006','mx007','mx008','mx009' ]':
q1Myterms :='[ 'my100','my200','my300','my400','my500','my600','my700','my800','my900' ]':
q2Myterms :='[ 'my010','my020','my030','my040','my050','my060','my070','my080','my090' ]':
q3Myterms :='[ 'my001','my002','my003','my004','my005','my006','my007','my008','my009' ]':

q1Mx := ⟨Column(Dipoles, 1)|q1Mxscan⟩ :
q2Mx := ⟨Column(Dipoles, 1)|q2Mxscan⟩ :
q3Mx := ⟨Column(Dipoles, 1)|q3Mxscan⟩ :
q1My := ⟨Column(Dipoles, 1)|q1Myscan⟩ :
q2My := ⟨Column(Dipoles, 1)|q2Myscan⟩ :
q3My := ⟨Column(Dipoles, 1)|q3Myscan⟩ :

  return q1Mxterms, q2Mxterms, q3Mxterms, q1Myterms, q2Myterms, q3Myterms, q1Mxconsts,
    q2Mxconsts, q3Mxconsts, q1Myconsts, q2Myconsts, q3Myconsts, xdip_plots, ydip_plots;
end proc:
```

C.10 Dipole Moment and Molar Absorptivity Process

This worksheet contains the process 'Dipole_Cont'. It calculates the transition dipole moments of the molecule by determining q-matrices and multiplying them by their respective dipole constants to yield Hermitian matrices; the Hermitian matrices contain elements for all possible combinations of initial and final harmonic energy levels for modes 1, 2, and 3 (as considered simultaneously). Only non-coupled Hermitian matrices are included for the dipole moment. The eigenvectors determined using 'Coupled_Freq' are then read and used with the dipole moment matrices to obtain scalar transition dipole moment values for each combination of initial and final harmonic energy level, and the value for the $0 \rightarrow 1$ transition extracted to calculate the molar absorptivities.

```
restart : with(Physics) : with(ExcelTools) : with(LinearAlgebra) : with(Statistics) :
    with(PolynomialTools) : with(ArrayTools) : with(ListTools) : with(DocumentTools) :
    with(combinat) : with(linalg) : with(plots) : interface(rtablesize = 50) : interface(warnlevel = 0) :

 #Process to calculate transition dipole moments and molar absorptivities

Dipole_Cont ≔ proc(kmax, jmax, imax, freqs, terms, mats)

  uses Physics, ExcelTools, LinearAlgebra, ArrayTools, Statistics, PolynomialTools, ListTools,
    DocumentTools, combinat, linalg, plots;

 local harm_freq, uncoupled, coupled, qmats, q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12,
    qmatlist, largest, nq, nqn, nq1n, nq2n, nq3n, n1, n2, n3, nn, w1, w2, w3, h_q1, h_q2, h_q3,
    h_harm1, h_harm2, h_harm3, h_q1Mx, h_q2Mx, h_q3Mx, h_q1My, h_q2My, h_q3My, q1Mx, q2Mx,
    q3Mx, q1My, q2My, q3My, m_q1, m_q2, m_q3, q1x_dmu, q2x_dmu, q3x_dmu, q1y_dmu, q2y_dmu,
    q3y_dmu, g1, g2, g3, un_eps1, un_eps2, un_eps3, eps1, eps2, eps3;

harm_freq ≔ freqs[1] : uncoupled ≔ freqs[2] : coupled ≔ freqs[3] : qmats ≔ freqs[4] : h_q1 ≔
    hmats[1] : h_q2 ≔ hmats[2] : h_q3 ≔ hmats[3] :

q0 ≔ (n1, n2) → if n1 = n2 then 1 else 0 fi;
q1 ≔ (n1, n2) →if n1 < 0 then 0 elif n2 < 0 then 0 elif abs(n1 − n2) = 1 then sqrt(max(n1, n2)/2)
    else 0 fi;
 q2 ≔ (n1, n2) → add(q1(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q3 ≔ (n1, n2) → add(q2(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q4 ≔ (n1, n2) → add(q3(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q5 ≔ (n1, n2) → add(q4(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q6 ≔ (n1, n2) → add(q5(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q7 ≔ (n1, n2) → add(q6(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q8 ≔ (n1, n2) → add(q7(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q9 ≔ (n1, n2) → add(q8(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q10 ≔ (n1, n2) → add(q9(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q11 ≔ (n1, n2) → add(q10(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 q12 ≔ (n1, n2) → add(q11(n1, i)·q1(i, n2), i = n2 − 1 ..n2 + 1) :
 qmatlist ≔ [q1mat, q2mat, q3mat, q4mat, q5mat, q6mat, q7mat, q8mat, q9mat, q10mat] :
seq(assign(qmatlist[i] = qmats[i]), i = 1 ..nops(qmats));
seq(seq(assign(terms[z][i] = terms[z + 6][i]), i = 1 ..nops(terms[z])), z = 1 ..6);

 #i=q3,j=q2,k=q1
 n1 ≔ [seq(seq(seq(k, k = 0 ..kmax), j = 0 ..jmax), i = 0 ..imax)]; nn ≔ nops(n1);
 n2 ≔ [seq(seq(seq(j, k = 0 ..kmax), j = 0 ..jmax), i = 0 ..imax)];
 n3 ≔ [seq(seq(seq(i, k = 0 ..kmax), j = 0 ..jmax), i = 0 ..imax)];

  largest ≔ max(imax, jmax, kmax); nq ≔ [seq(0 ..largest)] : nqn ≔ nops(nq) : nq1n ≔ nops([seq(0
     ..kmax)]); nq2n ≔ nops([seq(0 ..jmax)]); nq3n ≔ nops([seq(0 ..imax)]);

 #Calculate individual harmonic Hamiltonian matrices
 w1 ≔ harm_freq[1] : w2 ≔ harm_freq[2] : w3 ≔ harm_freq[3] :
h_harm1 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j])·q0(n2[i], n2[j])·q0(n3[i], n3[j])·(w1
     ·(n1[i] + 0.5)), j = 1 ..nn), i = 1 ..nn)]) :
 h_harm2 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j])·q0(n2[i], n2[j])·q0(n3[i], n3[j])·(w2
     ·(n2[i] + 0.5)),j = 1 ..nn), i = 1 ..nn)]) :
 h_harm3 ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) ·q0(n2[i], n2[j]) ·q0(n3[i], n3[j]) ·(w3
     ·(n3[i] + 0.5)),j = 1 ..nn), i = 1 ..nn)]) :

 #Calculate uncoupled Hamiltonian matrices
h_q1Mx ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) ·q0(n3[i], n3[j]) ·(q1mat[(n1[i] + 1), (n1[j]
     + 1)]·mx100 + q2mat[(n1[i] + 1), (n1[j] + 1)]·mx200 + q3mat[(n1[i] + 1), (n1[j] + 1)]
     ·mx300 + q4mat[(n1[i] + 1), (n1[j] + 1)]·mx400 + q5mat[(n1[i] + 1), (n1[j] + 1)]·mx500
     + q6mat[(n1[i] + 1), (n1[j] + 1)]·mx600 + q7mat[(n1[i] + 1), (n1[j] + 1)]·mx700
     + q8mat[(n1[i] + 1), (n1[j] + 1)]·mx800 + q9mat[(n1[i] + 1), (n1[j] + 1)]·mx900),j = 1
     ..nn), i = 1 ..nn)]) :
h_q2Mx ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) ·q0(n3[i], n3[j]) ·(q1mat[(n2[i] + 1), (n2[j]
     + 1)]·mx010 + q2mat[(n2[i] + 1), (n2[j] + 1)]·mx020 + q3mat[(n2[i] + 1), (n2[j] + 1)]
     ·mx030 + q4mat[(n2[i] + 1), (n2[j] + 1)]·mx040 + q5mat[(n2[i] + 1), (n2[j] + 1)]·mx050
     + q6mat[(n2[i] + 1), (n2[j] + 1)]·mx060 + q7mat[(n2[i] + 1), (n2[j] + 1)]·mx070
     + q8mat[(n2[i] + 1), (n2[j] + 1)]·mx080 + q9mat[(n2[i] + 1), (n2[j] + 1)]·mx090),j = 1
     ..nn), i = 1 ..nn)]) :
h_q3Mx ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) ·q0(n2[i], n2[j]) ·(q1mat[(n3[i] + 1), (n3[j]
     + 1)]·mx001 + q2mat[(n3[i] + 1), (n3[j] + 1)]·mx002 + q3mat[(n3[i] + 1), (n3[j] + 1)]
     ·mx003 + q4mat[(n3[i] + 1), (n3[j] + 1)]·mx004 + q5mat[(n3[i] + 1), (n3[j] + 1)]·mx005
     + q6mat[(n3[i] + 1), (n3[j] + 1)]·mx006 + q7mat[(n3[i] + 1), (n3[j] + 1)]·mx007
     + q8mat[(n3[i] + 1), (n3[j] + 1)]·mx008 + q9mat[(n3[i] + 1), (n3[j] + 1)]·mx009),j = 1
     ..nn), i = 1 ..nn)]) :
 h_q1My ≔ Matrix(nn, nn, [seq(seq(q0(n2[i], n2[j]) ·q0(n3[i], n3[j]) ·(q1mat[(n1[i] + 1), (n1[j]
     + 1)]·my100 + q2mat[(n1[i] + 1), (n1[j] + 1)]·my200 + q3mat[(n1[i] + 1), (n1[j] + 1)]
     ·my300 + q4mat[(n1[i] + 1), (n1[j] + 1)]·my400 + q5mat[(n1[i] + 1), (n1[j] + 1)]·my500
     + q6mat[(n1[i] + 1), (n1[j] + 1)]·my600 + q7mat[(n1[i] + 1), (n1[j] + 1)]·my700
     + q8mat[(n1[i] + 1), (n1[j] + 1)]·my800 + q9mat[(n1[i] + 1), (n1[j] + 1)]·my900),j = 1
     ..nn), i = 1 ..nn)]) :
h_q2My ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) ·q0(n3[i], n3[j]) ·(q1mat[(n2[i] + 1), (n2[j]
     + 1)]·my010 + q2mat[(n2[i] + 1), (n2[j] + 1)]·my020 + q3mat[(n2[i] + 1), (n2[j] + 1)]
     ·my030 + q4mat[(n2[i] + 1), (n2[j] + 1)]·my040 + q5mat[(n2[i] + 1), (n2[j] + 1)]·my050
     + q6mat[(n2[i] + 1), (n2[j] + 1)]·my060 + q7mat[(n2[i] + 1), (n2[j] + 1)]·my070
     + q8mat[(n2[i] + 1), (n2[j] + 1)]·my080 + q9mat[(n2[i] + 1), (n2[j] + 1)]·my090),j = 1
     ..nn), i = 1 ..nn)]) :
h_q3My ≔ Matrix(nn, nn, [seq(seq(q0(n1[i], n1[j]) ·q0(n2[i], n2[j]) ·(q1mat[(n3[i] + 1), (n3[j]
     + 1)]·my001 + q2mat[(n3[i] + 1), (n3[j] + 1)]·my002 + q3mat[(n3[i] + 1), (n3[j] + 1)]
     ·my003 + q4mat[(n3[i] + 1), (n3[j] + 1)]·my004 + q5mat[(n3[i] + 1), (n3[j] + 1)]·my005
     + q6mat[(n3[i] + 1), (n3[j] + 1)]·my006 + q7mat[(n3[i] + 1), (n3[j] + 1)]·my007
     + q8mat[(n3[i] + 1), (n3[j] + 1)]·my008 + q9mat[(n3[i] + 1), (n3[j] + 1)]·my009),j = 1
     ..nn), i = 1 ..nn)]) :


 #Remove multitude of zeros from dipole moment Hermetian matrices

 convert(h_q1Mx, list, nested = true) : [seq(remove(t→t = 0, %[i]), i = 1 ..nn)] : MakeUnique(%) :
     q1Mx ≔ convert(%, Matrix);
 convert(h_q2Mx, list, nested = true) : [seq(remove(t→t = 0, %[i]), i = 1 ..nn)] : MakeUnique(%) :
     q2Mx ≔ convert(%, Matrix);
 convert(h_q3Mx, list, nested = true) : [seq(remove(t→t = 0, %[i]), i = 1 ..nn)] : MakeUnique(%) :
     q3Mx ≔ convert(%, Matrix);
 convert(h_q1My, list, nested = true) : [seq(remove(t→t = 0, %[i]), i = 1 ..nn)] : MakeUnique(%) :
```

$q1My := convert(\%, Matrix);$

$convert(h\_q2My, list, nested = true) : [seq(remove(t \rightarrow t = 0, \%[i]), i = 1 ..nn)] : MakeUnique(\%) :$
$q2My := convert(\%, Matrix);$

$convert(h\_q3My, list, nested = true) : [seq(remove(t \rightarrow t = 0, \%[i]), i = 1 ..nn)] : MakeUnique(\%) :$
$q3My := convert(\%, Matrix);$

#Remove zeros from potential energy Hamiltonians, diagonalize, and place contribution vectors into matrix (wavefunction coefficients)

$$convert\left(evalf\left(\frac{h\_harm1 + h\_q1}{219474.6}\right), list, nested = true\right) : [seq(remove(t \rightarrow t = 0, \%[i]), i = 1 ..nn)] : MakeUnique(\%) : convert(\%, Matrix) : sort([eigenvectors(\%)]) : seq(convert(\%[i, 3, 1], list), i = 1 ..nq1n) : \left[seq\left(\mathrm{sqrt}\left(add(\%[i, j]^2, j = 1 ..nq1n)^{-1}\right) \cdot \%[i], i = 1 ..nq1n\right)\right] : m\_q1 := convert(\%, Matrix) :$$

$$convert\left(evalf\left(\frac{h\_harm2 + h\_q2}{219474.6}\right), list, nested = true\right) : [seq(remove(t \rightarrow t = 0, \%[i]), i = 1 ..nn)] : MakeUnique(\%) : convert(\%, Matrix) : sort([eigenvectors(\%)]) : seq(convert(\%[i, 3, 1], list), i = 1 ..nq2n) : \left[seq\left(\mathrm{sqrt}\left(add(\%[i, j]^2, j = 1 ..nq2n)^{-1}\right) \cdot \%[i], i = 1 ..nq2n\right)\right] : m\_q2 := convert(\%, Matrix) :$$

$$convert\left(evalf\left(\frac{h\_harm3 + h\_q3}{219474.6}\right), list, nested = true\right) : [seq(remove(t \rightarrow t = 0, \%[i]), i = 1 ..nn)] : MakeUnique(\%) : convert(\%, Matrix) : sort([eigenvectors(\%)]) : seq(convert(\%[i, 3, 1], list), i = 1 ..nq3n) : \left[seq\left(\mathrm{sqrt}\left(add(\%[i, j]^2, j = 1 ..nq3n)^{-1}\right) \cdot \%[i], i = 1 ..nq3n\right)\right] : m\_q3 := convert(\%, Matrix) :$$

#Use wavefunction coefficients and dipole moments to obtain transition dipole moments

$q1x\_dmu := evalf(Matrix(nq1n, nq1n, [seq(seq(m\_q1[i,..].q1Mx.m\_q1[j,..]^{+}, j = 1 ..nq1n), i = 1 ..nq1n)]));$

$q2x\_dmu := evalf(Matrix(nq2n, nq2n, [seq(seq(m\_q2[i,..].q2Mx.m\_q2[j,..]^{+}, j = 1 ..nq2n), i = 1 ..nq2n)]));$

$q3x\_dmu := evalf(Matrix(nq3n, nq3n, [seq(seq(m\_q3[i,..].q3Mx.m\_q3[j,..]^{+}, j = 1 ..nq3n), i = 1 ..nq3n)]));$

$q1y\_dmu := evalf(Matrix(nq1n, nq1n, [seq(seq(m\_q1[i,..].q1My.m\_q1[j,..]^{+}, j = 1 ..nq1n), i = 1 ..nq1n)]));$

$q2y\_dmu := evalf(Matrix(nq2n, nq2n, [seq(seq(m\_q2[i,..].q2My.m\_q2[j,..]^{+}, j = 1 ..nq2n), i = 1 ..nq2n)]));$

$q3y\_dmu := evalf(Matrix(nq3n, nq3n, [seq(seq(m\_q3[i,..].q3My.m\_q3[j,..]^{+}, j = 1 ..nq3n), i = 1 ..nq3n)]));$

#Calculate molar absorptivities

$$g1 := \left(\left(\frac{q1x\_dmu[1, 2]}{0.393456}\right)^2 + \left(\frac{q1y\_dmu[1, 2]}{0.393456}\right)^2\right) \cdot 41.6238 : un\_eps1 := \frac{(g1 \cdot uncoupled[1])}{16.60540}; eps1 := \frac{(g1 \cdot coupled[1])}{16.60540};$$

$$\mathit{g2} := \left(\left(\frac{\mathit{q2x\_dmu}[1,2]}{0.393456}\right)^{2} + \left(\frac{\mathit{q2y\_dmu}[1,2]}{0.393456}\right)^{2}\right)\cdot 41.6238 : \mathit{un\_eps2} := \frac{(\mathit{g2}\cdot \mathit{uncoupled}[2])}{16.60540};$$

$$\mathit{eps2} := \frac{(\mathit{g2}\cdot \mathit{coupled}[2])}{16.60540};$$

$$\mathit{g3} := \left(\left(\frac{\mathit{q3x\_dmu}[1,2]}{0.393456}\right)^{2} + \left(\frac{\mathit{q3y\_dmu}[1,2]}{0.393456}\right)^{2}\right)\cdot 41.6238 : \mathit{un\_eps3} := \frac{(\mathit{g3}\cdot \mathit{uncoupled}[3])}{16.60540};$$

$$\mathit{eps3} := \frac{(\mathit{g3}\cdot \mathit{coupled}[3])}{16.60540};$$

return *un_eps1*, *un_eps2*, *un_eps3*, *eps1*, *eps2*, *eps3*, *m_q1*, *m_q2*, *m_q3*, *q1x_dmu*, *q2x_dmu*, *q3x_dmu*, *q1y_dmu*, *q2y_dmu*, *q3y_dmu*;
end proc:

Vita

Andrea Becker joined UT in 2014 with an ACS-certified BS in Chemistry and a minor in Mathematics. She has spent the last five years engaged in computational research on molecular systems and has presented her results at various venues, including the Southeast Theoretical Chemistry Association (SETCA), the Southeast Regional Meeting of the ACS (SERMACS), the American Conference on Theoretical Chemistry (ACTC), and the National ACS Meeting and Exposition.