Ab initio REMPI spectra of HCl and HF

ERLENDUR JÓNSSON
kt. 110683-3169
17. október 2008

45 eininga meistaraverkefni í
EFNAFRÆÐI
Leiðbeinandi: Dr. Ágúst Kvaran
Meðleiðbeinandi: Dr. Ingvar H. Árnason

HÁSKÓLI ÍSLANDS RAUNVÍSINDADEILD

Yfirlýsing

Hér með lýsi ég því yfir að ritgerð þessi er samin af mér og að hún hefur hvorki að hluta né í heild verið lögð fram áður til hærri prófgráðu.

_____

Erlendur Jónsson

Abstract

The HCl molecule is a very popular molecule for spectroscopic studies. At the Science Institute of the University of Iceland, HCl has been studied for a few years with the REMPI (resonance enhanced multiphoton ionization) technique. This technique enables the study of previously unknown excited states.

Parallel to that research are the *ab-initio* calculations, where potentials of the excited states are calculated. From the potentials it's possible to find the spectroscopic parameters of each excited state. Several excited states of the HCl molecule have been calculated.

Some calculations were also run for the HF molecule.

The capability to simply do useful calculations for excited states of molecules is fairly recent. The method, which is used here, is the equation-of-motions (EOM) formulations for the coupled cluster methods (CC).

Ágrip

HCl sameindin er vinsæl í rannsóknum með litrófsgreiningu. Sameindin hefur verið rannsökuð í nokkurn tíma við Raunvísindastofnun Háskólans með REMPI aðferðinni (Resonance Enhanced Multiphoton Ionization). Þessi aðferð gerir kleift að rannsaka áður óþekkt örvuð ástönd sameindarinnar.

Samhliða þessum rannsóknum eru *ab-initio* útreikningar, þar sem hægt er að reikna mættisferla örvuðu ástandanna. Út frá þessum mættisferlum er hægt að reikna litrófsfasta hvers ástands fyrir sig. Nokkur örvuð ástönd HCl sameindarinnar hafa verið reiknuð.

Einhverjir útreikningar voru einnig keyrðir fyrir HF sameindina.

Getan til að beita útreikningum einfaldlega á örvuð ástönd sameinda er fremur nýleg. Aðferðinni sem hér er beitt er equations of motion (EOM), sem er stækkun á coupled cluster (CC) aðferðinni.

Contents

List of Figures

List of Tables

Listings

The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation.

Paul A. M. Dirac [1]

1 Introduction

Hydrogen halides are popular molecules to use in spectroscopic studies. They have been used in the photochemistry research group of Ágúst Kvaran at the Science Institute of the University of Iceland for some years now. The main focus has been on spectroscopic studies using the REMPI spectroscopic technique. Recently, interest in the *ab-initio* side of these experiments has increased. This thesis and, at the moment, one article by Kvaran, et al. [2], is a results of this interest.

The application of *ab-initio* calculations for these studies has three aims. One is aiding the identification of states, i.e. what state corresponds to each peak in the experimental spectra since there can be very many peaks in a REMPI spectra. Second is to make sense of ionization pathways as they depend on potential-interactions between states. The third aim is to predict states that haven't been found yet or have been mislabeled.

Ab-initio calculations of hydrogen halides has a long history as hydrogen fluoride is often used as a benchmark molecule [3] and since hydrogen chloride has a wealth of experimental results, e.g. the work of Green et al. [4, 5, 6]. Some *ab-initio* work has already been done on HCl. However, since HCl has eight more electrons than HF, *ab-initio* calculations are considerably more challenging because of the computational complexity of the available *ab-initio* methods. But nonetheless some work has been done on calculations for both molecules.

Let us focus on HCl. The oldest calculation on its excited states, that I'm aware of, were done in 1980 by Hirst and Guest [7], which calculated the $X^1\Sigma^+$ ground-state and the following excited states, $a^3\Pi$, $A^1\Pi$ and $t^3\Sigma^+$, using configuration interaction. Bettendorff et al. did some further calculations in 1982 [8]. Those calculations included the $C^1\Pi$ and $F^1\Delta$ states. Previously Bettendorff et al. did some multi-reference configuration interaction calculations on HF [9].

The only subsequent *ab-initio* work on the excited states of HCl to my knowledge is a very recent article by Pitarch-Ruiz, et al. published this year [10]. The article only calculates vertical excitation energies using the coupled cluster method.

Ab-initio methods that are capable of handling calculations of spectroscopic quality are fairly expensive computationally. A lot of work in recent years has been done to both simplify calculations and segment them to enable parallelization. Modern computational chemistry is getting more inexpensive because of better computers, algorithms and programs. There are still plenty of difficulties entailed in these high-performance computings [11], but these hurdles can be overcome.

One way which is used here at the University of Iceland, is using a Beowulf cluster[1]. They enable small universities to run high-level *ab-initio* calculations such as the ones described in this thesis.

In the future, it will be possible to use graphics processing units (GPU), the video cards, of computers to run calculations. Yasuda published this year two articles using GPUs in quantum chemistry, both to evaluate two-electron integrals [12] and to accelerate density functional calculations [13].

2 REMPI

Resonance enhanced multi-photon ionization is a spectroscopy technique where multiple photons are used to probe the physical properties of molecules. As multiple photons are absorbed in this spectroscopic technique, the usual selection rules are used multiple times. This gives rise to spectra that are not viewable with the more normal single-photon methods. Hence more spectroscopic information can be found with this technique.

A number of molecules has been studied using this technique, such as acetylene, [14], water [15, 16] and of course a plethora of hydrogen halides. Our research group, the Kvaran photochemistry research group, has looked at a number of them, such as HCl, DCl, HBr, HI [17, 18, 19, 20, 21, 22, 23]. Interestingly in the process of studying HF, Kvaran, et al. discovered that the hydrogen bonded complex of $(HF)_2$ can be seen in REMPI-TOF experiments [24]. Recently some *ab-initio* calculations have been incorporated into the analysis of REMPI-TOF experimental results [2].

Other groups have used other REMPI techniques in their studies of HCl, such as the "mass-resolved" REMPI technique of Chichinin et al. [25], which looks specifically at the photodissociation and photoionization processes of HCl.

2.1 Simulation of REMPI spectra

As can be seen in [26] and [19], it is quite simple to find the line positions of rovibrational lines, as they are given with (entirely in cm^{-1})

$$\nu_{J',\nu'\leftarrow J'',\nu''} = \nu^0_{\nu',\nu''} + \Delta\bar{E}_{J',J''} \tag{1}$$

where $\nu^0_{\nu',\nu''}$ is the band origin of a vibrational band defined as:

$$\nu^0_{\nu',\nu''} = T'_e + \left\{\omega'_e(\nu' + \frac{1}{2}) - \omega_e x'_e(\nu' + \frac{1}{2})^2\right\} - \left\{\omega''_e(\nu'' + \frac{1}{2}) - \omega_e x''_e(\nu'' + \frac{1}{2})^2\right\} \tag{2}$$

and T'_e is the excitation energy from the bottom of the ground state potential to the bottom of the excited state potential. ω''_e and ω'_e are the vibrational frequencies of the ground state and excited state, respectively. So the $''$ marks the ground state and $'$ marks the excited state. $\omega_e x_e$

[1] A Beowulf cluster is a group of inexpensive computers that are used in high-performance parallel calculations. Beowulf is from the eponymous Old English poem, he slew the monster Grendel.

is the anharmonic constant and v is the vibrational quantum number. $\Delta\bar{E}_{J',J''}$ is the difference in rotational energies.

$\Delta\bar{E}_{J',J''}$ is a function of the rotational constants of these non-rigid linear rotors[2] (B', B'', D', D'') and the rotational quantum numbers (J', J'') as can be seen:

$$\Delta\bar{E}_{J',J''} = (J+\Delta J)(J+\Delta J+1)B' - (J+\Delta J)^2(J+\Delta J+1)^2D' - J(J+1)B'' + J^2(J+1)^2D'' \quad (3)$$

In this equation, $J = J''$ and $\Delta J = J' - J''$.

These equations reveal the line positions of the REMPI spectra, but they don't show the line intensities (I_{rel}). However, that can be attained by using

$$I_{rel} = CS_{\Delta\Omega}\exp\left(\frac{-\bar{E}(J'')hc}{kT}\right) \quad (4)$$

which is the product of the Boltzmann distribution and the n-photon absorption strength, $S_{\Delta\Omega}$. C is a factor that is independent of the rotational quantum numbers J' and J'':

$$C(v', v'') = KF(v', v'')P^n\sigma_1(v') \quad (5)$$

where $F(v', v'')$ is the Franck-Condon factor of the $v' \leftarrow v''$ transition, K is a parameter that depends on geometric and electronic structure and sample strength of the molecule, and P^n is laser power over n photons [17].

The REMPI absorption strength, $S_{\Delta\Omega}$, was originally derived in 1976 by Bray and Hochstrasser [27] for the two-photon case. Halpern et al. further developed this theory to include three-photon excitations [28]. The algebraic form of $S_{\Delta\Omega}$ is fairly complex as it depends on Clebsch-Gordan coefficients and number of photons. A simplified version, where the ground-state is assumed to be a Σ-state was demonstrated by Kvaran et al., first for three-photons [19] and then for two- and three-photons [21].

3 *Ab-initio* calculations

Many methods exist for *ab-initio* calculations of both, ground and excited states. Hartree-Fock (SCF[3]) is ever so useful, originally proposed by Hartree and later refined by Fock. It remains popular to this day, especially as a starting point for more advanced methods as the full variational calculations of many methods would be even more expensive if it weren't for a fairly good guess for a starting point.

The main drawback of the Hartree-Fock method is the complete lack of electron correlation so a lot of new methods have been designed with this correlation in mind. They are usually

[2]The difference between the rigid and non-rigid is the inclusion of the centrifugal distortion constant which tries to account for bond stretching at high rotations.

[3]Self-consistent field is often used instead of Hartree-Fock as HF can easily be confused with hydrogen fluoride. Though sometimes in literature, you can see FH for hydrogen fluoride

referred to as post Hartree-Fock methods. These methods are e.g. configuration interaction (CI), Møller-Plesset perturbation theory [4] (MPn, $n = 2, 3, \ldots$) and coupled cluster (CC). All of these methods are more accurate than Hartree-Fock, but then again, they are a lot more computationally expensive.

It is assumed that solutions to the Hartree-Fock can be written as a single Slater determinant. These Slater determinants are defined in a orbital basis $\chi_j(\mathbf{x}_j)$[5] for electrons $\mathbf{x}_l$:

$$|\Phi_i\rangle = \Phi_i(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\mathbf{x}_1) & \chi_2(\mathbf{x}_1) & \cdots & \chi_N(\mathbf{x}_1) \\ \chi_1(\mathbf{x}_2) & \chi_2(\mathbf{x}_2) & \cdots & \chi_N(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ \chi_1(\mathbf{x}_N) & \chi_2(\mathbf{x}_N) & \cdots & \chi_N(\mathbf{x}_N) \end{vmatrix}. = |\chi_1(\mathbf{x}_1)\chi_2(\mathbf{x}_2)\cdots\chi_N(\mathbf{x}_N)\rangle \tag{6}$$

The Slater determinant is a simple way to get an antisymmetric wavefunction, so it fulfills the condition:

$$\Phi_i(\ldots, \mathbf{x}_i, \ldots, \mathbf{x}_j, \ldots) = -\Phi_i(\ldots, \mathbf{x}_j, \ldots, \mathbf{x}_i, \ldots) \tag{7}$$

so it fulfills the Pauli exclusion principle.

3.1 Many-body perturbation theory

The many-body perturbation theory is a correction to the Hartree-Fock energies by perturbation. The Hamiltonian of the MPn has an additional term

$$\hat{H} = \hat{H}_0 + \lambda\hat{V} \tag{8}$$

where V is a small perturbation and λ is an arbitrary parameter. H_0 is the Fock operator, so the calculations start with known eigenvectors and eigenvalues. The $\hat{V}$ is the electronic correlation potential.

$$\hat{V} = \hat{H} - \hat{F} \tag{9}$$

The wavefunction and energies are then expanded into a power series. The highest power in the series is n, and so we have many-body perturbation theory of the n-th order (MPn). MP2 was originally developed by Møller and Plesset in 1934 [29], where it was also demonstrated that the contribution of the MP1-energy is zero (i.e. the perturbation is zero) so the HF energy is the same as the total MP1 energy. For more information about the formulation, see Helgaker, et al. [30] or Szabo and Ostlund [31].

Unlike CI and CC, the accuracy of MPn does not necessarily increase with an increase in n. Hirata and Bartlett show this clearly with calculations on hydrogen fluoride, with MPn, $n = 2, \ldots, 20$. [32]

[4] Also known as the many-body perturbation theory, which fortunately has the same abbreviation.

[5] It is not necessary to have N orbitals for N electrons, so Φ_i could be $|\chi_i \cdot \chi_k\rangle$.

Big-O notation	Input		
$\boldsymbol{O}(N)$	1	2	10
$\boldsymbol{O}(N^2)$	1	4	100
$\boldsymbol{O}(N^6)$	1	64	1,000,000
$\boldsymbol{O}(N!)$	1	2	3,628,800

Table 1: The effect of increasing the size of an input, e.g. the number of atoms, by a factor of 2 and 10, on computer time. $\boldsymbol{O}(N^6)$ is the computational complexity of CCSD.

3.2 Configuration interaction

Configuration interaction (CI) is one of the more powerful post Hartree-Fock methods but it has a fairly high computational complexity. If Φ_0 is a solution to the Hartree-Fock equations (it's a Slater determinant). Then the exact (non-relativistic) wavefunction, Ψ, is a linear combination of Slater determinants as follows:

$$|\Psi\rangle = c_0|\Phi_0\rangle + \sum_{ar} c_a^r|\Phi_a^r\rangle + \sum_{\substack{a<b\\r<s}} c_{ab}^{rs}|\Phi_{ab}^{rs}\rangle + \sum_{\substack{a<b<c\\r<s<t}} c_{abc}^{rst}|\Phi_{abc}^{rst}\rangle + \dots \tag{10}$$

where $|\Phi_a^r\rangle$ is a singly excited Slater determinant. One orbital in it is different from Φ_0. So the $|\Phi_{ab}^{rs}\rangle$ is a doubly excited Slater determinant and so on. In a full configuration interaction (FCI) calculation, all the Slater determinants, Φ_0 and $|\Phi_{a\dots}^{r\cdots}\rangle$, are included in the calculations. This is unfortunately prohibitively expensive to calculate as the computational complexity is $\boldsymbol{O}(n!)$, so it is very impractical in use. On the other hand, it's very handy in the benchmarking of other quantum chemistry methods as it gives the exact non-relativistic solution, provided that you use a large enough basis.

Computational complexity is usually described by the big O notation, such as $\boldsymbol{O}(n!)$. This means that the dominant factor of the calculations grows factorially with increasing input. Other factors can be huge but as $n \to \infty$, this will become dominant. Table 1 explores this in detail.

The configuration interaction equation (10) can be simplified by truncating it. One can, for example, assume that only singly and doubly excited determinants are relevant to the calculations. Hence the equation becomes:

$$|\Psi\rangle = c_0|\Phi_0\rangle + \sum_{ar} c_a^r|\Phi_a^r\rangle + \sum_{\substack{a<b\\r<s}} c_{ab}^{rs}|\Phi_{ab}^{rs}\rangle \tag{11}$$

This method is called the configuration interaction singles and doubles (CISD). It is variational[6], but it is not size-extensive. [31]

Size-extensivity is the property of some quantum chemistry methods, where two distinct non-interacting systems can be calculated separately, i.e. the calculations give the same result when system A and B are calculated simultaneously and seperate. Then the energy of the total system would be [30]

$$E_{AB} = E_A + E_B \tag{12}$$

[6]A variational method is a method where there is a lower bound, which is the solution. So it is possible to try use trial functions and attempt to minimize them.

CI can be extended to calculate excited states. [33] That method is referred to as CI-Singles. The method creates excited states by replacing an occupied spin-orbital with a virtual spin-orbital. This is then used to create singly excited determinants. This method does not handle electron correlation properly, so its handling of excited states can be very uneven, as the amount of electron correlation in each state can vary. [34] It could be called excited state "Hartree-Fock".

3.3 Coupled cluster

Coupled cluster is another post Hartree-Fock methods and possibly one of the more important ones. It's application on quantum chemistry problems was originally developed by Čižek in 1966. [35] Coupled cluster assumes that the wavefunction, $|\Psi\rangle$ of a molecule can be written as

$$|\Psi\rangle = e^{\hat{T}}|\Phi_0\rangle \tag{13}$$

where $|\Phi_0\rangle$ is a trial wavefunction (Slater Determinant), usually implemented as the converged Hartree-Fock wavefunction of the molecule in question. $\hat{T}$ is the excitation operator

$$\hat{T} = \sum_{n=1}^{N_{el}} \hat{T}_n \tag{14}$$

$\hat{T}_n$ is the n-fold excitation operator, which expresses all n-fold excitations, so $\hat{T}_1$ is the operator of all single excitations. The size of the calculations with this operator depends entirely on the number of orbitals which it effects, so $\hat{T}_1$ would tend to ∞ as the basis tends to ∞. This finite sum has a upper bound of terms, i.e. the number of electrons in the molecule, N_{el}, as you cannot excite more electrons than are in it. However, that's impractical because of the computational complexity, so the sum is usually trunctated to a few terms. This excitation operator can be visualized, as is seen in figure 1.

One version of coupled cluster theory is the CCSD method, which is an acronym for coupled cluster singles and doubles, so the $\hat{T}_n$ is:

$$\hat{T}_n = \hat{T}_1 + \hat{T}_2 \tag{15}$$

and CCSDTQ, coupled cluster, singles, doubles, triples and quadruples would have the excitation operator:

$$\hat{T}_n = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \hat{T}_4 \tag{16}$$

As this is a convergent sum, it is possible to increase the accuracy with each additional excitation operator term, albeit with diminishing returns. Hirata and Bartlett demonstrate this [32], with calculations where FH, H_2O and F^- are used as test cases. In their article excitations from singles to octuples are used, whereas the octuple excitation is formally exact as each molecule has eight electrons. That corresponds to a full configuration interaction calculation on the molecules.

Unlike truncated CI calculations, such as CISD, the corresponding CCSD calculation are size-extensive, making coupled cluster more preferable in all thermodynamic calculations. Additionally coupled cluster calculation are, depending on choice of a reference wavefunction, size consistent. However CCSD is not a variational method.

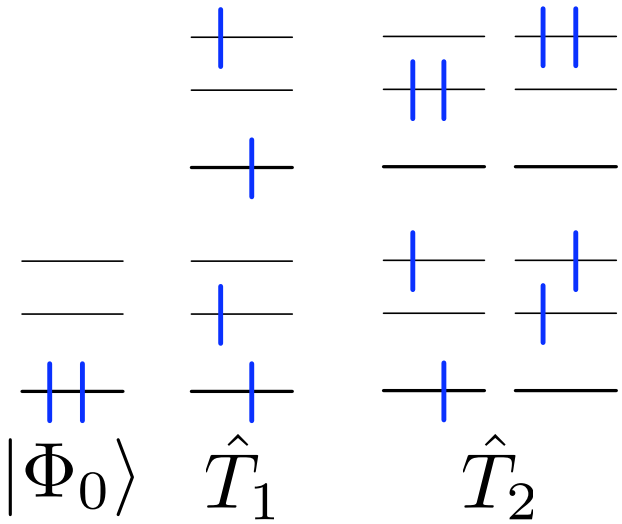


Figure 1: This is a conceptual drawing of the effect of the $\hat{T}_1$ and $\hat{T}_2$ operators on a $|\Phi_0\rangle$ wave-function. This drawing ignores electron spin entirely. Note that the upper configuration of the $\hat{T}_1$ operator is the same as the lower left corner of the $\hat{T}_2$ operator. One might even naively assume that this demonstrates the size-extensivity of the coupled cluster method, far from it, this is similar to the configuration method. The difference is the exponential in equation 13 which is the key to the size-extensivity of the CC method [30].

One inherent problem with this approach is the computational complexity of each additional excitation operator as CCSD has $\boldsymbol{O}(n^6)$, CCSDT $\boldsymbol{O}(n^8)$, CCSDTQ $\boldsymbol{O}(n^{10})$, and so forth. [36] So it follows that we either need some further approximations or to confine ourselves to very small molecules despite the so called Moore's Law of the semiconductor industry.[7] One method that's often called the golden standard of quantum chemistry is CCSD(T), the T within parentheses is an approximation of the triple excitation operator and is included via perturbation. But the computational complexity of CCSD(T) is $\boldsymbol{O}(n^7)$, so it's just in the middle of CCSD and CCSDT. Interestingly, in some cases, one sees better results from CCSD(T) than from CCSDT. For some molecules CCSDT isn't even enough, e.g. ozone, O_3, seems to require quadruple excitations. [37] Ozone has a good amount of static correlation[8], so a single Slater determinant does not handle its ground-state properly. This can be remedied by using a multi-reference method[9], though the renormalized methods, mentioned later in this thesis aim to fix this [40]. Further developments have included CCSD(TQ_f) which extends CCSD(T) with pertubative quadruple excitations with a factorization approximation. This approach has the same computational complexity as CCSD(T) [41].

One key disadvantage of the coupled cluster methods is the fact that the methods don't handle

[7]The amount of transistors that can be placed on an integrated circuit doubles every 18 to 24 months. Though this is not law, but an observation of a trend this has held true for more than four decades.

[8]Static correlation (also known as near-degeneracy correlation) is the long range part of electron correlation, which can be though of as the effect of molecular dissociation on electrons. [38]

[9]See Bartlett [39] for an introduction into multi-reference calculations.

non-equilibrium bond lengths well. Some recent developments have focused on fixing that disadvantage.

One development was the Method of Moments CC or MMCC. Its basic idea is to correct a CC calculation with a non-iterative correction δ. This correction is an attempt to get the FCI energy.[42] Special cases of this method are the renormalized coupled cluster methods. One of these special cases is the completely renormalized CCSD(T) (CR-CCSD(T)) method. It can give better results than CCSDT for large internuclear distance, Piecuch, et al.[3] look at the application of that method to remedy the deficiencies of a single-reference coupled cluster method with relation to the vibrational spectra of HF. Unfortunately these methods are not size-extensive and thereby lose one of the more important advantages of the coupled cluster theory.

However, there are attempts underway to correct this flaw of the renormalized methods. One result of this, is the development of the locally renormalized CCSD(T), LR-CCSD(T), method [43, 40] which is an attempt to regain the size extensivity property of the normal CCSD(T) method.

Kowalski and Piecuch [44] look at the dissociation of the N_2 triple bond, which is one of the classic problems to tackle, because of the significant complexity of addressing the triple bond itself. It usually requires a multi-reference calculation.

Further improvements to handle systems that aren't at equilibrium, e.g. the potential energy surface of explosives, especially the reaction mechanism of the explosive process, have resulted in the ΛCCSD(T), [45, 46]. The authors of [45] explore a transition state of the explosive RDX[10].

There have also been efforts in reducing the computational complexity of coupled cluster methods, e.g. the natural linear scaling coupled cluster method [47]. However, that method does not simplify the calculations when you have a small molecule, as this is a method where one tries to divide the molecule into smaller parts, similar to a unit of a polymer is representative of the whole. This means that larger molecules can be tackled, provided that they have the correct structure for this method.

Hence, one can say that the coupled cluster method is very powerful, and has even been used in modeling artificial atoms, i.e. quantum dots [48].

For a comprehensive review of this theory, see Bartlett and Musial [49] and Helgaker et al. [30].

3.4 Basis sets

The selection of Gaussian basis sets is a fairly complex issue as the number of available basis sets is large with all kinds of uses and logic behind them. It is important to select carefully. A review by Davidson and Feller [50] shows the rationale behind selection of basis sets.

Currently the correlation consistent basis sets [51, 52] are a very popular set of bases, as they are fairly accurate and it is easy to get better results with them because of their systematic method of construction. This even enables the use of extrapolation to approximate the energy values at the complete basis set limit.

[10]RDX is also known as cyclonite. Its IUPAC name is 1,3,5-trinitroperhydro-1,3,5-triazine 1,3,5-trinitro-1,3,5-triazacyclohexane.

If we look at the correlation consistent basis sets, the smallest basis set in it is cc-pVDZ. cc-pVDZ is an acronym for correlation consistent-polarization valence double zeta. The multiplicity of zeta is a measure of the size of each basis, which goes in most cases from double to sextuple. These basis are usually referred to by their acronyms, cc-pVnZ, (n = D,T,Q,5 and 6).

The zeta of these correlation consistent basis sets is a measure of how many s orbitals are defined in each basis set. So for the hydrogen atom, cc-pVDZ has [2s,1p], cc-pVTZ [3s,2p,1d], cc-pVQZ [4s,3p,2d,1f] and cc-pV5Z [5s,4p,3d,2f,1g]. However this is not applicable to all basis sets.

As good as these basis sets are, they aren't capable of handling some calculations as they lack the necessary diffuse functions that are needed. If we looked at calculations involving e.g. fluorine, the cc-pVnZ basis sets are unable to handle the very electronegative atom. To remedy this, it's possible to get an augmented version of the basis sets [53]. They are usually referred to as aug-cc-pVnZ (n = D, T, Q, 5 and 6), similar to the cc-pVnZ basis sets. The more diffuse functions enable calculations with electron polarizabilities and since they are less biased towards the ground state, they enable better excited state calculations [54]. So for size comparison: the augmented version of the basis for hydrogen, aug-cc-pVDZ has [3s,2p], aug-cc-pVTZ [4s,3p,2d], aug-cc-pVQZ [5s,4p,3d,2f] and aug-cc-pV5Z [6s,5p,4d,3f,2g].

When further accuracy is needed, it's possible to add functions to treat core-core and core-valence correlation effects [55]. This results in the aug-cc-pCVnZ basis sets, where C is for core. This unfortunately overestimates the core-core correlation energy so the weighted core valence basis sets is the result from taking this into consideration [56].

Other systematic basis sets exist, such as the polarization consistent basis sets of Jensen [57, 58, 59, 60, 61, 62]. These basis sets are usually referred to as pc-n (n = 0, 1, 2, 3 and 4). This n is measure of size similar to the n in the correlation consistent basis sets. In the pc-1 basis, the hydrogen atom is [2s,1p] and pc-2 basis is [3s,2p,1d], so the polarization consistent basis sets have a dissimilar numbering scheme from the correlation consistent basis. These basis sets have been optimized for use in DFT calculations.

It has a similar structure as the correlation consistent basis sets and it's also possible to extrapolate results from calculations with them to a complete basis set (CBS) limit. Shahbazian and Zahedi demonstrate this clearly in their article [63], which looks at calculations at the SCF-level. Kupka and Lim [64] in turn look at calculations at the MP2 and CCSD(T) level. They also look at spectroscopic parameters and their basis set dependence. There exists also an expansion of the basis sets, which additional tight s, p, d and f function, these are the pcJ-n basis sets [65]. This expansion is specifically designed to calculate spin-spin coupling constants with DFT.

Figure 2 shows how the CBS limit is reached by the choice of both basis and method.

3.5 Excited states

3.5.1 Time dependent density functional theory

Density functional theory (DFT), is one of the most common theories used today in quantum chemistry. This is because of it's low computational complexity, $\boldsymbol{O}(n^4)$,[11] which makes it applica-

[11] This does depend on the functional, as they add all kinds of calculations in the mix.

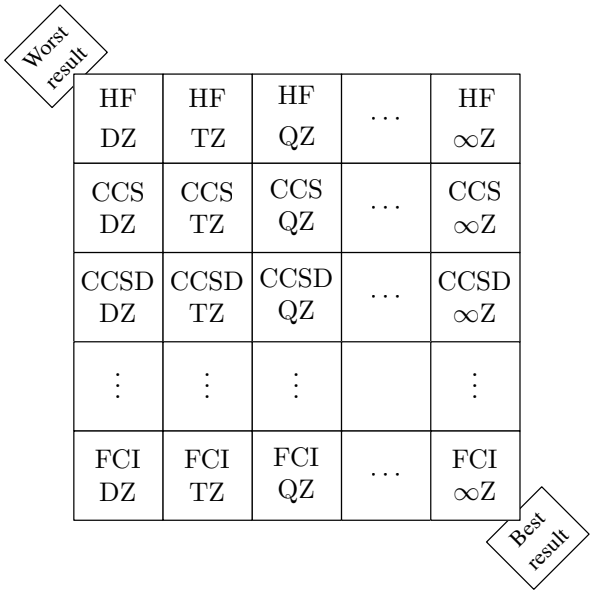


Figure 2: The effect of increasing size of an arbitrary basis set and the level of calculation. By going from the upper left corner down to bottom right corner, the results of calculations are improved, however the cost is great. Note that the FCI calculation with an ∞Z basis set give the exact non-relativistic energy.

ble to fairly large problems. It relies upon reformulating the Schrödinger equation, so that it's a function of the electron density, $\rho(\mathbf{r})$,

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} |\psi_i(\mathbf{r})|^2 \tag{17}$$

This equation (and others) were originally published by Hohenberg and Kohn [66] and form the core of DFT as they show that for a ground-state density $\rho_0(\mathbf{r})$, it is possible to calculate the corresponding ground-state wavefunction $\Psi_0(\mathbf{r_1}, \ldots, \mathbf{r_N})$. Despite having a proof of existence of that relation, we do not have a constructive proof, so we are forced to use numerical methods to approximate the solution.

Kohn and Sham introduced the usage of a noninteracting (single-body) [67] system in a potential $v_s(\mathbf{r})$. This potential was chosen to be

$$v_s(\mathbf{r}) = v(\mathbf{r}) + v_H(\mathbf{r}) + v_{XC}(\mathbf{r}) \tag{18}$$

where $v(\mathbf{r})$ is a potential, $v_H(\mathbf{r})$ is the Hartree potential

$$v_H(\mathbf{r}) = \int \mathrm{d}\mathbf{r}' \frac{\rho(\mathbf{r}')}{\sqrt{(\mathbf{r} - \mathbf{r}')^2}} \tag{19}$$

and $v_{XC}(\mathbf{r})$ is the exchange-correlation (XC) potential. It's supremely important as it gives all of the many-particle effects, yet unattainable as there doesn't exist an exact functional for it.

However there do exist many approximate functionals, such as LDA, GGA [67] and B3LYP [68] which has seen great use in quantum chemistry as it gives impressive results in geometric structure optimization despite it's flaws in energies. An introduction to the density-functional theory can be found in Parr's and Yang's book. [69]

Despite the many advantages of DFT, it doesn't handle excited states, so time-dependent density functional theory (TD-DFT) was formulated. [70] TD-DFT tries to solve the time-dependent Schrödinger equation with the DFT formalism. It is a powerful theory, applicable to many kinds of calculations. One such application is the calculation of the interaction between a laser pulse and acetylene. One disadvantage is the fact that Rydberg states tend to be unbound. [71]

The exchange-correlation potential of the ground-state has inaccuracies, which in turn, gives wrong eigenvalues. These eigenvalues are influenced by the behaviour of the approximated Kohn-Sham potentials for they decay exponentially, instead of $(-1/r)$. Some functionals exist to correct this, i.e. asymptotically correct functionals. But this means that this thesis will focus more on other methods. For more insight into TD-DFT look at a review by Elliott et al. [72]

3.5.2 Equation-of-motion

The coupled cluster method can be extended to handle excited states. One such extension is the equation-of-motion coupled cluster method, EOM-CC. It has similar characteristics as the normal coupled cluster method, i.e. size-extensive. [30] The usual usage of an EOM-CC method is to calculate first a CC ground-state and then try to find the corresponding excited state(s). This can be seen from

$$|\Psi_x\rangle = \mathcal{R}|\Psi_g\rangle \tag{20}$$

where Ψ_x is an excited state and Ψ_g is the ground state. Since this ground-state is a CC wavefunction, it can be expressed algebraically

$$|\Psi_g\rangle = e^{\hat{T}}|\Phi_0\rangle \tag{21}$$

which is the same as equation 13. $\mathcal{R}$ is an excitation operator, similar to CI excitations,

$$\mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1 + \dots \tag{22}$$

where

$$\mathcal{R}_n = \frac{1}{n!^2} \sum r_{ijk\cdots}^{abc\cdots} a^\dagger i b^\dagger j c^\dagger k \cdots \tag{23}$$

For further details on the mathematics see both Stanton and Bartlett [73] and Bartlett and Musial [49].

If the EOM-CC calculations are to be practical, both the ground state $\hat{T}$ and excited state $\mathcal{R}$ excitation operators need both to be truncated. They are usually truncated to the same level, e.g. CCSD has a corresponding EOM-CCSD for excited states, where

$$\mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1 + \mathcal{R}_2 \tag{24}$$

Each method does have a similar computational complexity as its corresponding ground-state method, i.e. EOM-CCSD and CCSD, both have $\boldsymbol{O}(n^6)$. Interestingly $\mathcal{R}$ and $\hat{T}$ do not need to be truncated at the same level. The authors of [74] go through those possibilities for CH^+ and CH_2. This article shows in turn the effect of increasing the level of the method, i.e. going from EOMCCS to EOMCCSD to ... to EOM-CCSDTQPH which is a FCI calculation for CH_2 (when the lowest orbital is frozen).

Hirata has incidentally developed a general method to develop parallel programs which calculate EOM-CC up to an arbitrary level. The method is encapsulated in a program called the Tensor Contraction Engine [75] which then handles all of the parallelization and memory details. It even exploits symmetries to minimize some calculations. NWChem has built in support for EOM-CCSD, EOM-CCSDT and EOM-CCSDTQ because of the program. [76]

However, to be practical, one should limit oneself to only a few excitations, otherwise one would have to wait for too long. EOM-CCSD and EOM-CCSD(T) are two methods that are practical for a number of calculations and have a tolerable computational complexity. However, they do not address excited states where two-electron transitions dominate so few methods have been designed to take care of this. An example is the CCSDt/EOM-CCSDt method, which uses triple excitations within active orbitals. A system that requires this treatment is the beryllium trimer, Be_3 [77].

Excited-state calculations have similar issues to ground-state calculations, such as the problem of multi-reference determinants. This problem can arise quite naturally with small molecules. For example nitrogen, can be handled fairly well close to the equilibrium bond length with single-reference methods, but as the bond is stretched more, the multi-reference character takes over. Larsen et al. [78, 79] have done a FCI benchmark calculation of N_2, utilizing also coupled cluster methods, including response theory calculations, to calculate excited states. They show in their article that as the internuclear bond length increases, the results of the CC methods get worse, i.e. compared to the FCI calculations.

Multi-reference calculations are inherently more complicated to implement and use compared to single-reference calculations, some efforts have been focused on trying to get a multi-reference character into single-reference calculations, i.e. try to have your cake and eat it too. One of these attempts is the application of the method of moments to EOM-CC, similar to its application in ground-states. The main idea of the MM-EOM-CC method is to use a non-iterative energy correction to correct an EOM-CC energy to approximate the exact FCI energy. These ideas were developed by Kowalski and Piecuch [80] and then developed by them [81]. However, the methods that they developed first are not entirely comparable with the more common EOM-CC methods as they use a higher level of $\mathcal{R}$ than $\hat{T}$.

This method of moments theory was developed further and yielded the completely renormalized equation-of-motion coupled cluster singles, doubles and perturbative triples, CR-EOM-CCSD(T) [82]. This method corresponds to the CR-CCSD(T) ground-state method [42]. The CR-EOM-CCSD(T) method is capable of handling both excited states which have strong double excitations and states with multi-reference character. This includes molecules such as N_2, C_2, O_3 [83] and the CH-radical [84], results of course vary, but they are better than with the more common EOM-CCSD method.

One caveat with the MM-EOM-CC methods, is that they are not size-extensive, so some

work has gone into fixing that liability. One method which results from that work is the locally renomalized equations of motion coupled cluster singles and doubles method, LR-EOM-CCSD of Kowalski [85]. That method alleviates some of the single-reference problems of EOM-CCSD, but not all of them.

4 Ab initio REMPI spectra

The results of coupled cluster and EOM-CC calculations are unfortunately not directly comparable with experimental results for they are energies and excitation energies as a function of bond length, i.e. potentials. As the *ab initio* calculations are neede for experimental purposes, this is not an optimal scenario. Hence we need to compare the experimental spectra with either the calculated spectroscopic constants or simulated spectra.

The two methods, that are used in this thesis, are to fit the potentials with a function and to use a numerical method called Fourier grid Hamiltonian.

4.1 Fits

Igor Pro 6.03 was employed to fit the potentials with a Morse potential function

$$U(r) = T_e + D_e \left(1 - e^{-\beta(r-r_e)}\right)^2 \tag{25}$$

The resulting parameters, the excitation energy from a lower potential to this one (bottom of the potential well), T_e, the dissociation energy (from the bottom of the potential well), D_e, the empirical constant, β, and the equilibrium bond length, r_e, can easily be used to calculate the more convential spectroscopic parameters of diatomic molecules, such as the vibrational frequency

$$\omega_e = \frac{\beta}{0.12177}\sqrt{\frac{D_e}{\mu}} \tag{26}$$

the anharmonic constant

$$\omega_e x_e = \frac{\omega_e^2}{4D_e} \tag{27}$$

the rotational constant

$$B_e = \frac{\hbar^2}{8\pi^2 \mu r_e^2} \tag{28}$$

the centrifugal distortion constant

$$D_v = \frac{4B_v^3}{\omega_e^2} \tag{29}$$

μ is the reduced mass of the molecule, given by

$$\mu = \frac{m_1 m_2}{m_1 + m_2} \tag{30}$$

where m_1 is the mass of atom 1, i.e. hydrogen, and m_2 is the mass of atom 2, i.e. chlorine.

To find the vibrational bond length $\langle r_v \rangle$, one finds the weighted average of the vibrational wavefunction:

$$\langle r_v \rangle = \int_0^\infty r\psi_v \, \mathrm{d}r = \frac{\sum_{i=1}^{\infty} r_i |\psi_i|^2}{\sum_{i=1}^{\infty} |\psi_i|^2} \tag{31}$$

and the rotational constant of each vibrational level:

$$B_v = \frac{\hbar^2}{8\pi^2 \mu \langle r_v \rangle^2} \tag{32}$$

4.2 Fourier Grid Hamiltonian

The Fourier Grid Hamiltonian is a numerical method which uses a potential at a discrete set of grid points [86]. The potential which is used, can be, for example, a Morse potential or the resulting calculations from some *ab-initio* method. This method gives the bound state eigenvalues and eigenfunctions, which are the vibrational levels and wave-functions. The Hamiltonian matrix elements were first derived by Marston and Balint-Kurti [86] and subsequently extended by Balint-Kurti, et al. [87] to use a Fast Fourier Transform to handle better larger grid sizes.

We look at a potential, $V(x)$ on a grid of x_i-values, with a fixed grid spacing of Δx, this grid has an even number of points, n_x. Let's define $n = nx/2$. The Hamiltonian matrix elements are

$$H_{ij} = \frac{1}{\Delta x} \left\{ \left[\sum_{l=1}^{n-1} 2 \cos\left(\frac{l 2\pi (i-j)}{n_x} \right) \cdot T_l \right] + (-1)^{(i-j)} T_n \right\} + V(x_i)\delta_{ij} \tag{33}$$

where

$$T_l = \frac{\hbar^2}{2\mu} \left(l \frac{2\pi}{n_x \Delta x} \right)^2 \tag{34}$$

After this matrix is constructed, the only thing left is to find its eigenvectors and eigenvalues.

4.3 AIREMPICalc 1.0

Ab-initio REMPI Calc is a Python program, more easily referred to as AIREMPICalc. It is designed to enable simple calculations and analysis on the *ab-initio* potentials, since the results of tables 6 and 9, which were first published by Kvaran, et al. [2], are extremely labor intensive.

They required a workflow, where one needed first to fit a potential with a Morse curve (T_e,D_e,β and r_E) with the Igor Pro (versions 4.01 and 6.03 were used), these numbers were then moved into Microsoft Excel (part of Microsoft Office 2004) and used to calculate the missing numbers (ω_e, $\omega_e x_e$) for evaluation with a custom set of macros for Igor Pro called FCF 1 [2]. These macros generated the vibrational wavefunction at a set of bond lengths. The wavefunction was then copied into Excel to calculate the vibrational bond length of each vibrational level with equation 31. After that it's straigthforward to calculate B_v with equation 32.

This workflow is error-prone and time-consuming, hence I wrote AIREMPICalc to take care of it. AIREMPICalc is both a program and a set of utility modules that handle most of the work.

This program and its modules are written in Python (version 2.5.1), using SciPy (version 0.6.0) and NumPy (version 1.0.4). [12] SciPy is a package for scientific programming with functions that e.g. fit data and integrate numerically. NumPy is a package that implements N-dimensional arrays and numerical operations with them.

The utility modules are:

`potent` Handles loading of potentials, can also create a Morse potential function from spectroscopic parameters.

`fitter` Fits an *ab-initio* potential curve with a function, such as the Morse potential.

`fgh` Implements the Fourier Grid Hamiltonian (FGH), using both a copy of `FGHEVEN` of Balintkurti, et al. [87] and a re-implementation of it in Python. This module, both has an atomic unit version and one in Å and cm^{-1}.

`spec` Calculates r_v, B_v, ω_e and $\omega_e x_e$ from the FGH results.

Since Python has an interactive interpreter, it is possible to use all of the module functionality without having to run the AIREMPICalc program. However, the program does nearly all of the work for you, since the only thing you need to do, to use it, is a setup file and a potential in a text file.

The AIREMPICalc program does the following:

1. Loads the potentials that have been calculated with the methods described in section 3.
2. If it is supposed to fit the potentials, then it fits (section 4.1) and creates a Python function from the results.
3. If it should interpolate the potentials, then it creates a Python function that uses a spline to interpolate the potential curve.
4. Uses the Fourier grid hamiltonian method (section 4.2) and a Python function to calculate the vibrational wavefunctions and vibrational levels.
5. Calculates the vibrational bond lengths, r_v, (equation 31) and rotational constants, B_v (equation 32). Then it fits the vibrational levels to find the anharmonic vibrational constant, $\omega_e x_e$. It also calculates the vibrational frequence ω_e
6. It should then use the theory of section 2.1 to calculate the REMPI spectra. However, this is not implemented yet.

[12] SciPy and NumPy are both open source and can be downloaded, free of charge, at http://scipy.org/

5 Results

Most program packages support only few point groups. One group, that wasn't supported in the programs, that were used in this thesis, is the $C_{\infty v}$ group. To be able to run all the calculations, a subgroup of $C_{\infty v}$ was used, the C_{2v} group. This heavily impacts the excited state calculations, as the excited states are grouped by symmetry. By looking at the a1, a2, b1 and b2 symmetries, it is possible to see that a1 = Σ, a2 = Δ and b1 = b2 = Π.

5.1 HCl

The calculations of HCl were run on Jötunn, a computer cluster at the University of Iceland, owned by the University Computing Service[13]. They were done with ACESII [88] and NWChem (both version 5.0 [89] and version 5.1 [90]) program packages.

The general procedure was to calculate the ground state and excited states at a set of bond lengths. This yields of course a set of potential curves, a subset of which can be seen in figure 3. Both *ab initio* programs give their ground state results in atomic units, i.e. hartree, so the units need to be converted to cm^{-1} and then shifted so that the minima of the ground state potential is 0.0 cm^{-1}.

When NWChem runs EOMCCSD calculations, it calculates the excitation energy, in eV, as opposed to the energy of the state. This means that, to calculate the excited state potentials, one needs first to convert the excitation energy into cm^{-1}, $E_{excitation}(r)$ and then add it to ground state potential, $U_{gs}(r)$. For a bond length, r_0, the excited state potential would be:

$$U_{ex}(r_0) = U_{gs}(r_0) + E_{excitation}(r_0) \tag{35}$$

The alternate setup are the CR-EOMCCSD(T) calculations of NWChem and EOMCCSD calculations of ACESII, where all results are in hartrees. One only needs to convert the energies into cm^{-1} and shift all the potential curves so that the ground state has a minima of 0.0 cm^{-1}.

The bond length sets are usually centered on r = 1.27 Å. One problem with the bond length sets is that sometimes, one particular bond length simply doesn't work, i.e. I am unable to get the programs to converge to a solution for a given calculation method, bond length and basis. Therefor it's sometimes necessary to tweak a lot of program settings manually to be able to calculate a particular state. The sets often have gaps in them, either states or entire bond lengths. This can be seen in figure 3.

Interestingly these convergence problems seem to depend on the basis, method and the computer the calculation is run on. For instance, should one bond length prove to be stubborn on computer A, you could simply try the calculations on another machine. I have tried this by moving a set of calculations that didn't work on Jötunn to my laptop, where they did work. This is suggestive that it is due to some numerical instabilities, for example in the linear algebra library. The resources to eliminate all the other possible cause were not available.

[13] Reiknistofnun Háskóla Íslands, RHÍ.

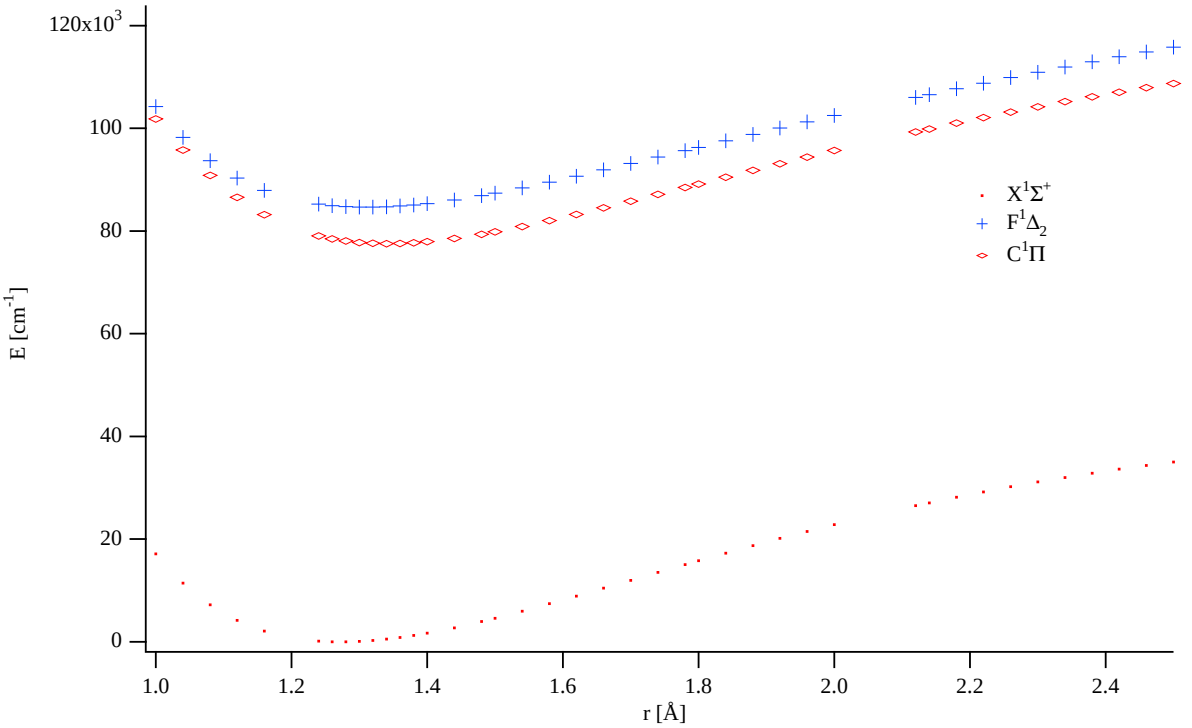


Figure 3: HCl potentials where the ground state, $X^1\Sigma^+$, was calculated with CCSD/aug-cc-pVQZ and the excited states, $F^1\Delta_2$ and $C^1\Pi$ were calculated with CR-EOMCCSD(T)/aug-cc-pVQZ.

5.1.1 FGH implementation

As the Fourier Grid Hamiltonian (FGH) is a numerical method, one needs to run comparisons to find out what its parameters should be to get results that are converged. The grid, that's used in the FGH method, which is very important, has three variables. The start point, r_{min}, end point, r_{max}, and number of grid points, n_x.

Since the most important part of the potential is around the equilbrium bond length, r_e, it would be prudent to focus on the area surrounding it in the potential curve. One should focus on the area bounded by $r_{min} < r_e < r_{max}$, but if we look first at the number of grid points and it's effect on the results. In figure 4, $r_{min} = 0.8r_e$, $r_{max} = 2.0r_E$ and $r_e = 1.27$Å. In this figure, it is easy to see the effect of n_x on the resulting vibrational frequency. The figure shows the ratio

$$\text{Ratio} = \frac{\omega_{e\ \text{calculated}}}{\omega_{e\ \text{experimental}}} = \frac{\omega_{e\ \text{calculated}}}{2990.946\ \text{cm}^{-1}} \tag{36}$$

as a function of n_x. It's quite clear that increasing the number of points increases the accuracy of the calculation. However, the quality of the final results (when n_x is very large) depends highly on which potential curve is used.

Let's look similarly at the other spectroscopic constants, i.e. as a ratio of calculated to experimental, B_v, r_V, ω_e and $\omega_e x_e$ in figure 5. Interestingly the anharmonic vibrational constant, $\omega_e x_e$, doesn't show the same n_x-dependent improvement as seen in figure 4, but it gets worse as

n_x gets higher. Unfortunately the n_x-dependency effect of $\omega_e x_e$ is so large compared to everything else, that no other effects are seen in figure 5(a) so if we look at figure 5(b) where $\omega_e x_e$ has been removed.

Since r_v seems not be affected by the selection of n_x, same holds for B_v as it is a function of r_v. Similar effect is also visible for ω_e as in figure 4. So the selection of n_x effects the anharmonic constant, $\omega_e x_e$ the most, but has also some impact on the vibrational frequency ω_e. Hence, it has little impact on the vibrational bond length, r_v, and the rotational constant, B_v.

Now if we look at the outer limits of the grid, i.e. r_{min} and r_{max}, we can see figure 6. In it, we have `rmin_factor` and `rmax_factor`, defined thusly:

$$r_{min} = r_e \cdot \texttt{rmin\_factor} \qquad r_{max} = r_e \cdot \texttt{rmax\_factor} \tag{37}$$

where r_e is the equilibrium bond length of the $X^1\Sigma^+$ state of HCl, which is $r_e = 1.27$Å. This is a contour map of the ratio of ω_e as described before. The figure shows clearly that being in the bottom of the potential well is not optimal, but that's within the boundary of $r_{min} = 0.99$Å and $r_{max} = 1.55$Å. By going to at least `rmin_factor`=0.7 and `rmax_factor`=1.3, optimal results are obtained for this state. To be absolutely sure, it would be prudent to increase `rmax_factor`, as that end of the potential has a more gradual curve. `rmax_factor` needs to be a bit higher to get all of its information.

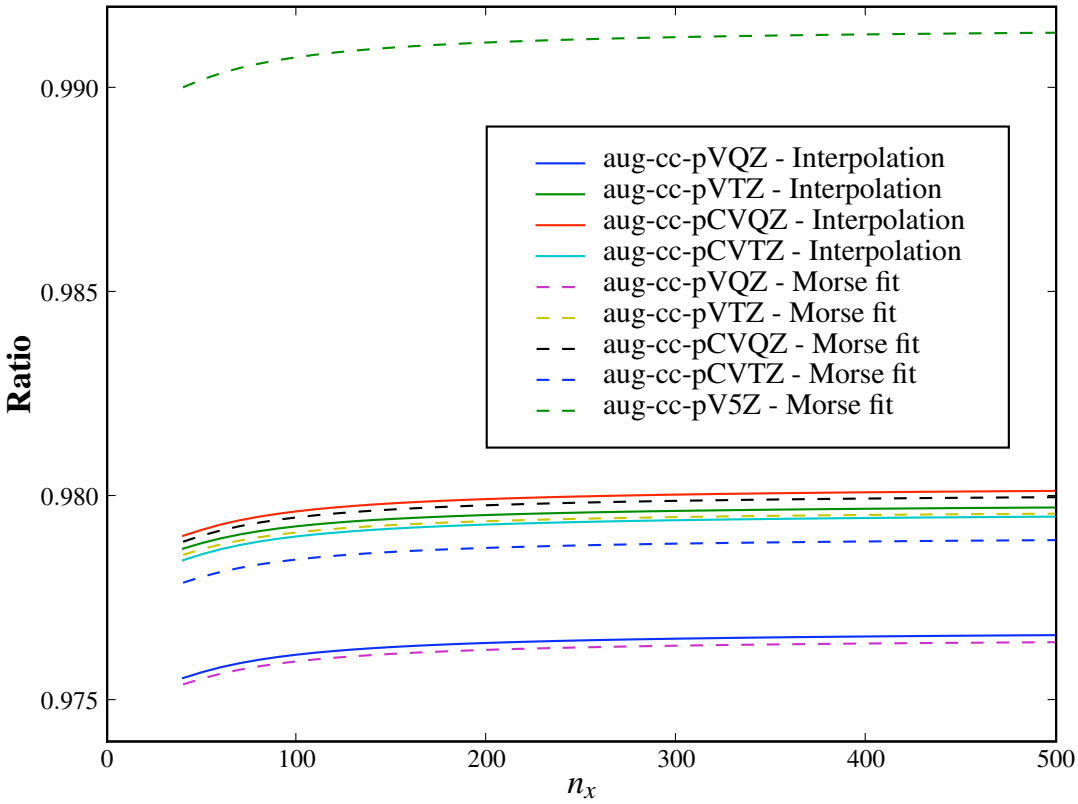
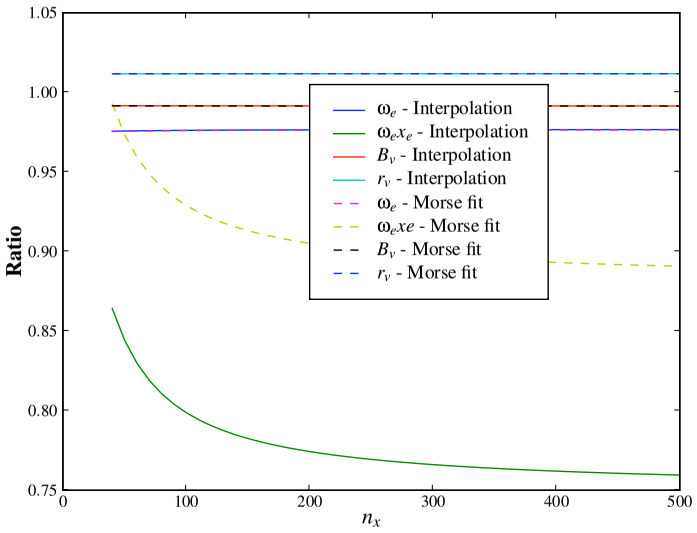
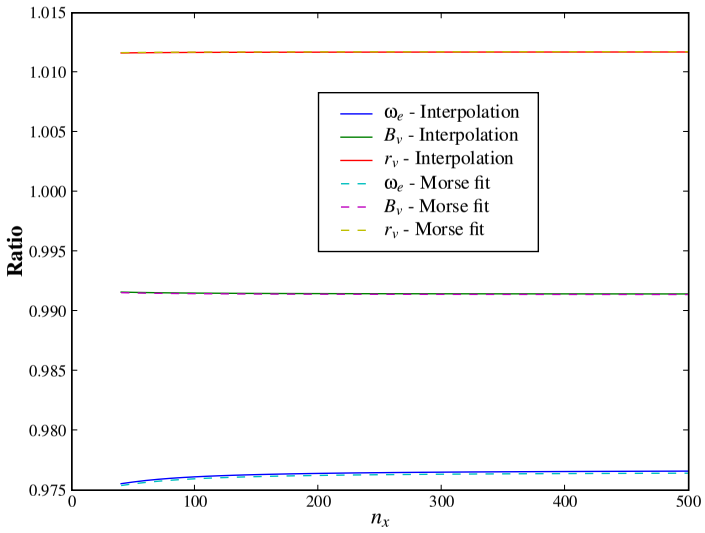


Figure 4: The ratio of the calculated vibrational frequency, ω_e, and experimental frequency as a function of the Fourier grid Hamiltonian grid points, n_x. This is calculated for the $X^1\Sigma^+$ state of HCl at the CCSD level of theory. This graph was made with AIREMPICalc.



(a) The ratio of spectroscopic constants, B_v, r_V, ω_e and $\omega_e x_e$ to their experimental values.



(b) The ratio of spectroscopic constants, B_v, r_V and ω_e to their experimental values. Here $\omega_e x_e$ is not shown to show the other spectroscopic constants clearer.

Figure 5: A look at the spectroscopic constants of the $X^1\Sigma^+$ state of HCl as a function of Fourier grid Hamiltonian grid points. All the constants are shown as a ratio to their experimental counterpart. The potential used was the CCSD/aug-cc-pVQZ point set, both as an interpolated function, also as a fitted function. These graphs were made with AIREMPICalc

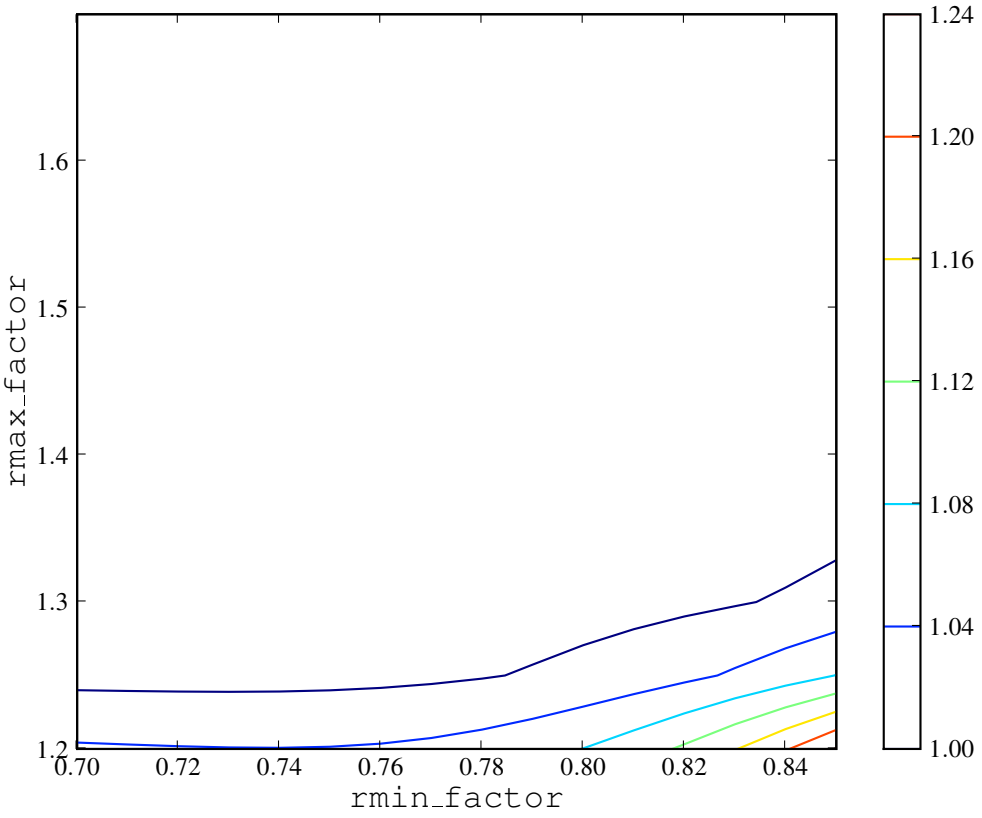


Figure 6: This contour map shows the effect of changing the `rmin_factor` and `rmax_factor` of the AIREMPICalc program on the ratio of the calculated vibrational frequency and experimental vibrational frequency ($\omega_{e\ \mathrm{calc}}/\omega_{e\ \mathrm{exp}}$). These two variables control the outer points of the FGH grid. The potential used was the CCSD/aug-cc-pVQZ point set fitted with a Morse potential. The number of grid point, n_x was 200.

5.1.2 The $X^1\Sigma^+$ state

The ground state of HCl is the $X^1\Sigma^+$ state. The potential curves were fitted with a Morse potential with a threshold of 5000 cm^{-1} [14]. The results can be seen in table 2. These results were obtained with the help of Igor Pro and a homegrown program used in the research group of Ágúst Kvaran, called FCF1. For more info on this methodology, see [2]. That program calculates the vibrational wavefunctions of the Morse potential. These can then be used to find the vibrational bond lengths (equation 31) and rotational constant (equation 32).

One noticeable feature of these calculations, is the lack of complete basis set (CBS) limit for the spectroscopic parameters in the case of CCSD/aug-cc-pVxZ, x = T,Q and 5, as one would

[14] I.e. all values that were lower than 5000 cm^{-1}.

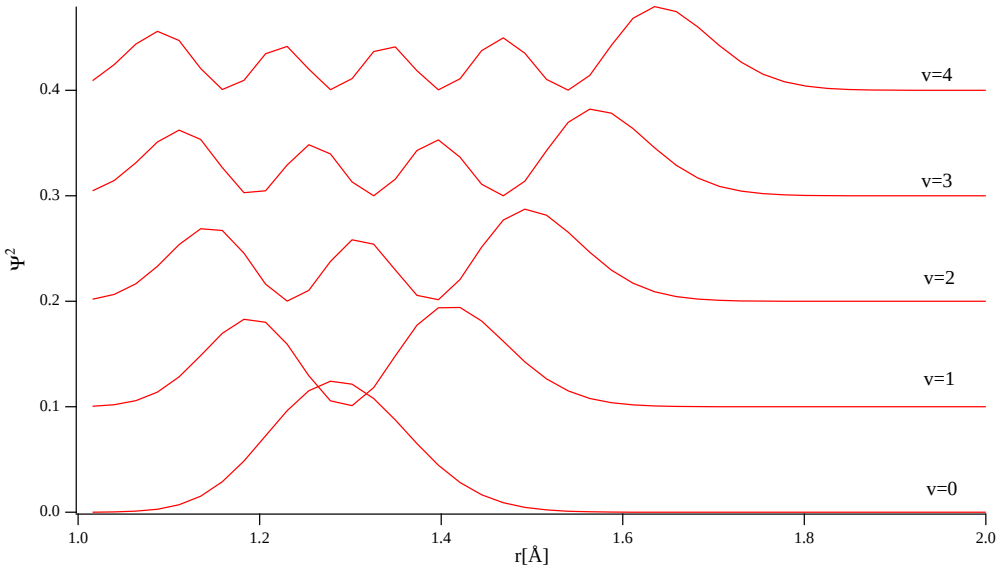


Figure 7: The resulting vibrational wavefunction, of CCSD/aug-cc-pVQZ calculations of the $X^1\Sigma^+$ ground state of HCl. They have been ordered in ascending order, so the lowest vibrational level is the lowest one on the graph. Each subsequent levels is offset by 0.1.

expect the values to follow an exponential curve. This is likely due to the fitting. However, to be get proper CBS-limit results, it would be necessary to first extrapolate with the aug-cc-pVxZ results to an infinite zeta and then use a core-valence correction and calculate the spectroscopic parameters. The resulting spectroscopic parameters would then be the parameters at the CBS-limit.

The AIREMPICalc program automates this methodology, it's even quite simple to visualize the vibrational wavefunctions, which can be seen in figure 7. The results of running the program with the calculated potential curves can be seen in table 3. Instead of having to calculate the spectroscopic constants manually as was done in table 2, this is done automatically. A demo output from the program can be seen in section A.3.7.

The fitting threshold used in the AIREMPICalc program was 5000.0 cm^{-1}, the same as in the manually fitted section. This means that the results should be the same, or at least very close[15]. Here it's quite clear that the results are the same, so the program works for fitting potentials when using a threshold.

Table 3 shows also results of fitting when a threshold isn't used. They are quite similar, however there is a small improvement when this threshold is used. This can best be seen for the anharmonic constant $\omega_e x_e$

It is not necessary to fit the potential curve to be able to use the FGH method to calculate vibrational levels and wavefunctions. There are two other options that I explored. One of which is to use a spline function to interpolate missing values in the potential curve, i.e. missing values for the evaluation of the FGH matrix elements. The resulting spectroscopic parameters can be seen in

[15]There are often some minor numerical differences in implementations of the same algorithm, in this case, a least squares fit.

Method	r_e [Å]	$B_{v=0}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]
Experiments	1.27455[1]	2990.946[1 2]	52.8186[1 2]	10.439826[3]
B3LYP/ aug-cc-pVTZ	1.284	2957	57	10.19
B3LYP/ aug-cc-pVQZ	1.282	2955	57	10.21
MPW1PW91/ aug-cc-pVQZ	1.278	3007	56	10.27
MP2/ aug-cc-pVTZ	1.271	3070	56	10.40
MP2/ aug-cc-pVQZ	1.272	3055	55	10.39
MP4/ aug-cc-pVTZ	1.275	3025	58	10.33
MP4/ aug-cc-pVQZ	1.276	3009	57	10.32
CCSD/AQZ	1.273	3030	57	10.36
CCSD/aug-cc-pVTZ	1.273	3053	53	10.36
CCSD/aug-cc-pVQZ	1.274	3042	51	10.53
CCSD/aug-cc-pV5Z	1.268	3079	61	10.44
CCSD/aug-cc-pCVQZ	1.272	3050	52	10.38
CCSD(T)/ aug-cc-pVTZ	1.275	3016	58	10.32
CCSD(T)/ aug-cc-pVQZ	1.276	3002	57	10.31
CR-CCSD(T)/ aug-cc-pVTZ	1.246	3022	58	10.33
CR-CCSD(T)/ aug-cc-pVQZ	1.256	3008	56	10.32

Table 2: Spectroscopic parameter of the $X^1\Sigma^+$ state of HCl. They were obtained with various *ab initio* calculations and experiment. These results are based on fitted potential curves, that were fitted with a Morse potential. The parameters were then calculated using the methods described in [2] where they were previously published.
The *ab-initio* calculations that are in **bold** were run by Andras Bodi, however I did the subsequent calculation of spectroscopic parameters.
1) ref. [91] ; 2) ref. [17]; 2) $B_0 = B_e - \alpha_e * (1/2)$; for B_e and α_e in ref. [91];

Method	r_e [Å]	T_e [cm^{-1}]	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]	$B_{v=2}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]	D_e
Experiment [6, 91]	1.27455	0	10.4401	10.1361	9.8345	2990.0	52.0	42981
CCSD/aug-cc-pVTZ	1.276	-33.0	10.326	9.838	9.341	2950.1	45.6	43938.5
CCSD/aug-cc-pVQZ	1.277	-48.2	10.306	9.827	9.338	2943.2	43.1	45288.6
CCSD/aug-cc-pV5Z	1.267	-6.2	10.446	9.914	9.372	2960.9	55.7	38797.7
CCSD/aug-cc-pCVTZ	1.276	-5.8	10.322	9.839	9.346	2927.0	42.7	45056.9
CCSD/aug-cc-pCVQZ	1.275	-30.2	10.342	9.856	9.359	2949.9	44.2	44693.9
With a threshold								
CCSD/aug-cc-pVTZ	1.273	-4.1	10.361	9.848	9.326	2929.3	51.4	40290.0
CCSD/aug-cc-pVQZ	1.274	-8.4	10.350	9.843	9.326	2919.9	49.4	41093.1
CCSD/aug-cc-pV5Z	1.268	-2.3	10.441	9.932	9.408	2964.4	48.7	42292.0
CCSD/aug-cc-pCVTZ	1.273	-2.4	10.367	9.854	9.332	2927.4	51.0	40419.8
CCSD/aug-cc-pCVQZ	1.272	-3.3	10.381	9.867	9.343	2930.5	50.8	40519.8

Table 3: Spectroscopic parameter for the $X^1\Sigma^+$ state of HCl. They were obtained with various *ab initio* calculations and experiment. The AIREMPICalc program was used to fit the *ab-initio* potential curves with a Morse potential and then the Fourier Grid Hamiltonian was used (200 grid points). The entire calculated curve was used is some cases and in others, there was used a threshold of 5000.0 cm^{-1}, i.e. all values where the energy was below 5000.0 cm^{-1} were used.

table 4. The *ab-initio* calculations that use the aug-cc-pV5Z do not have a large enough potential curve, as it only goes out to $r = 1.48$ Å. The impact of core-valence correlation does not seem to impact the calculations at the triple zeta level as the difference between the CCSD/aug-cc-pVTZ and CCSD/aug-cc-pCVTZ is minor. There is, however, a larger difference at the quadruple zeta level.

Method	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]	$B_{v=2}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]
Experiment [6, 91]	10.4401	10.1361	9.8345	2990.0	52.0
CCSD/aug-cc-pVTZ	10.361	9.848	9.335	2929.8	42.9
CCSD/aug-cc-pVQZ	10.351	9.842	9.333	2920.4	41.5
CCSD/aug-cc-pCVTZ	10.367	9.852	9.335	2929.1	42.6
CCSD/aug-cc-pCVQZ	10.381	9.867	9.352	2930.8	41.5

Table 4: Spectroscopic parameter of the $X^1\Sigma^+$ state of HCl. They were calculated using the AIREMPICalc program. The potential curves used a spline to interpolate values on the potential curve for the Fourier Grid Hamiltonian calculations, which used 200 grid points. The CCSD/aug-cc-pV5Z potential curve is not large enough to use this method, as its outer limit is $r = 1.48$ Å.

Another method used to evaluate the FGH matrix elements, is simply to use the *ab-initio* potential directly in the construction of the FGH matrix. Table 5, looks at that method, using a specifically generated point set for that purpose. That point set uses CCSD/aug-cc-pVTZ level of theory and is available with a Δx of 0.005 Å. A subset of these points are obviously then available with differing Δx which were used in the FGH method.

The effect of increasing the size of the grid, i.e. increasing n_x and decreasing Δx, is minor for the rotational constants, albeit it does increase with each vibrational level, but in each case the values move a bit closer to the experimental one. The opposite of this, is found in the case of ω_e and $\omega_e x_e$.

Method	n_x	Δx [Å]	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]	$B_{v=2}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]
Experiment [6, 91]			10.4401	10.1361	9.8345	2990.0	52.0
CCSD	0.0250	84	10.360	9.852	9.365	2960.4	51.2
CCSD	0.0200	104	10.360	9.854	9.368	2953.8	50.9
CCSD	0.0150	140	10.361	9.855	9.371	2946.7	50.7
CCSD	0.0100	210	10.362	9.857	9.374	2939.9	50.4
CCSD	0.0050	420	10.362	9.859	9.376	2933.2	50.2

Table 5: Spectroscopic parameter of the $X^1\Sigma^+$ state of HCl. The *ab-initio* method used here was CCSD/aug-cc-pVTZ. The spectroscopc parameters were calculated using the AIREMPICalc program. The points of the potential curve were used directly in the Fourier Grid Hamiltonian method.

5.1.3 The $C^1\Pi$ state

The $C^1\Pi$ state was measured originally by Tilford, et al. [92], where it's noted that the structure is very diffuse and accurate measurements are difficult. This is also noted by Green, et al. [4]. This means that the spectroscopic constants might not be as precise as one would like them to be.

AIREMPICalc was used to calculate the spectroscopic constants for the $C^1\Pi$ state. It used three methods that are available in it, i.e. fits 6, interpolation with spline functions 7 and a direct potential evaluation 8.

The table of Morse fits, table 6, shows that limiting oneself to potential values below a certain energy threshold improves the resulting calculations somewhat. This was also seen for the $X^1\Sigma^+$ state. The results of the CR-EOM-CCSD(T) calculations are generally better than the corresponding EOM-CCSD calculations. Interestingly, all of the EOM-CCSD $T_{v=0}$ values are approximately 1000 cm^{-1} higher than the experimental value of 77485.3 cm^{-1}. The best of the CR-EOM-CCSD(T) results, 77468.0 cm^{-1} use aug-cc-pCVTZ basis, without a fit threshold, are less than 20 cm^{-1} from the experimental value. This is a very impressive result.

The anharmonic constants are impacted heavily by the fit threshold, as without it, the error being from 9 % up to more than 50 %. When the fit threshold is used, all the anharmonic constants of the *ab-initio* calculations deviate less than 10 % from the experimental value of 66.6 cm^{-1}.

Let's look at the results that use a spline interpolation in table 7. There is again visible difference between the EOM-CCSD and CR-EOM-CCSD(T) methods of around 1000 cm^{-1}. If we look at the $T_{v=0}$ value, the best results is 77498.6 cm^{-1} for the CR-EOM-CCSD(T)/aug-cc-pVQZ calculations, which is 13.3 cm^{-1} more than the experimental value, so it's even better than the best fitted value. This set of *ab-initio* calculations also has the best rotational constant, $B_{v=0}$ = 9.332 cm^{-1} a difference of 0.001 cm^{-1}! The vibrational frequencies, ω_e, are then generally better when using CR-EOM-CCSD(T), but the opposite is true for $\omega_e x_e$.

The previous comments are also applicable to table 8 of direct evaluations.

Method	$T_{v=0}$ [cm^{-1}]	$T_{v=1}$ [cm^{-1}]	$T_{v=2}$ [cm^{-1}]	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]	$B_{v=2}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]
Experiment [6]	77485.08	—	—	9.3417	—	—	—	—
	77485.3	—	—	9.333	—	—	—	—
Experiment [92]	77485.3	80169.3	(82725)	9.33	9.29	(9.0)	2817.5	66.6
EOM-CCSD/aug-cc-pVTZ	78958.5	81557.3	84065.8	9.104	8.690	8.287	2598.8	44.7
EOM-CCSD/aug-cc-pVQZ	78687.5	81452.5	84096.4	9.450	8.974	8.510	2765.0	60.7
EOM-CCSD/aug-cc-pV5Z	78623.7	81584.2	84485.9	9.475	9.158	8.847	2960.5	21.5
EOM-CCSD/aug-cc-pCVTZ	78898.7	81559.2	84119.4	9.225	8.790	8.368	2660.4	49.9
EOM-CCSD/aug-cc-pCVQZ	78934.8	81618.4	84199.5	9.348	8.904	8.471	2683.6	50.7
CR-EOM-CCSD(T)/aug-cc-pVTZ	77955.7	80613.7	83185.3	9.160	8.759	8.370	2658.0	42.3
CR-EOM-CCSD(T)/aug-cc-pVQZ	77315.9	80442.0	83363.3	10.061	9.462	8.872	3126.1	102.7
CR-EOM-CCSD(T)/aug-cc-pV5Z	77195.2	80175.3	83095.4	9.505	9.186	8.872	2980.1	22.0
CR-EOM-CCSD(T)/aug-cc-pCVTZ	77468.0	80171.9	82782.8	9.247	8.833	8.430	2703.9	45.7
CR-EOM-CCSD(T)/aug-cc-pCVQZ	77299.4	80019.7	82642.6	9.366	8.937	8.519	2720.3	47.7
With threshold								
EOM-CCSD/aug-cc-pVTZ	78803.4	81530.5	84132.7	9.155	8.688	8.233	2727.1	62.9
EOM-CCSD/aug-cc-pVQZ	78691.2	81421.5	84026.7	9.284	8.807	8.343	2730.4	63.1
EOM-CCSD/aug-cc-pV5Z	78631.0	81417.9	84084.1	9.459	8.987	8.526	2786.9	60.4
EOM-CCSD/aug-cc-pCVTZ	78851.0	81575.3	84176.1	9.264	8.791	8.330	2724.3	62.3
EOM-CCSD/aug-cc-pCVQZ	78869.5	81607.2	84220.5	9.391	8.908	8.438	2737.8	62.6
CR-EOM-CCSD(T)/aug-cc-pVTZ	77821.0	80596.8	83249.7	9.216	8.757	8.308	2775.8	61.9
CR-EOM-CCSD(T)/aug-cc-pVQZ	77495.5	80263.0	82908.8	9.330	8.862	8.406	2767.6	61.3
CR-EOM-CCSD(T)/aug-cc-pV5Z	77203.5	80015.2	82705.5	9.492	9.021	8.560	2811.7	60.8
CR-EOM-CCSD(T)/aug-cc-pCVTZ	77446.5	80211.5	82855.2	9.306	8.841	8.387	2765.0	61.1
CR-EOM-CCSD(T)/aug-cc-pCVQZ	77249.7	80018.6	82665.5	9.418	8.944	8.481	2768.8	61.2

Table 6: Spectroscopic parameter of the $C^1\Pi$ state of HCl. They were obtained with various *ab initio* calculations and experiment. The AIREMPICalc program was used to fit the *ab-initio* potential curves with a Morse potential and then the Fourier Grid Hamiltonian method was used (200 grid points). The entire calculated curve was used is some cases and in others, there was used a threshold of 9000.0 cm^{-1}, i.e. all values, where the energy was less than 9000.0 cm^{-1} more than the lowest value in the potential curve, were used.

Method	$T_{v=0}$ [cm^{-1}]	$T_{v=1}$ [cm^{-1}]	$T_{v=2}$ [cm^{-1}]	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]	$B_{v=2}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]
Experiment [6]	77485.08	—	—	9.3417	—	—	—	—
	77485.3	—	—	9.333	—	—	—	—
Experiment [92]	77485.3	80169.3	(82725)	9.33	9.29	(9.0)	2817.5	66.6
EOM-CCSD/aug-cc-pVTZ	78803.0	81530.8	84124.5	9.159	8.687	8.248	2727.8	56.0
EOM-CCSD/aug-cc-pVQZ	78693.3	81420.4	84021.8	9.286	8.806	8.355	2727.1	54.2
EOM-CCSD/aug-cc-pVCTZ	78850.5	81578.7	84174.9	9.268	8.787	8.339	2728.2	54.0
EOM-CCSD/aug-cc-pVCQZ	78871.2	81607.4	84219.1	9.394	8.902	8.442	2736.3	53.3
CR-EOM-CCSD(T)/aug-cc-pVTZ	77821.8	80595.5	83241.4	9.220	8.755	8.327	2773.7	50.8
CR-EOM-CCSD(T)/aug-cc-pVQZ	77498.6	80261.0	82903.5	9.332	8.859	8.416	2762.4	49.2
CR-EOM-CCSD(T)/aug-cc-pVCTZ	77447.7	80213.1	82853.1	9.309	8.836	8.399	2765.4	49.0
CR-EOM-CCSD(T)/aug-cc-pVCQZ	77252.8	80017.3	82662.4	9.421	8.937	8.485	2764.4	48.4

Table 7: Spectroscopic parameter of the $C^1\Pi$ state of HCl. They were obtained with various *ab initio* calculations and experiment. These parameters were calculated using the AIREMPICalc program. The potential curves used a spline to interpolate values on the potential curve for the Fourier Grid Hamiltonian calculations, which used 200 grid points. The CCSD/aug-cc-pV5Z potential curve is not large enough to use this method, as its outer limit is r = 1.48 Å.

Method	Δx [Å]	n_x	$T_{v=0}$ [cm^{-1}]	$T_{v=1}$ [cm^{-1}]	$T_{v=2}$ [cm^{-1}]	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]	$B_{v=2}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]
Experiment [6]	—	—	77485.08	—	—	9.3417	—	—	—	—
	—	—	77485.3	—	—	9.333	—	—	—	—
Experiment [92]	—	—	77485.3	80169.3	(82725)	9.33	9.29	(9.0)	2817.5	66.6
EOM-CCSD	0.0250	84	78802.2	81561.2	84182.7	9.156	8.679	8.236	2759.0	57.5
EOM-CCSD	0.0200	104	78802.4	81555.4	84171.4	9.156	8.680	8.238	2753.0	57.3
EOM-CCSD	0.0150	140	78802.6	81549.1	84159.4	9.157	8.682	8.241	2746.5	57.0
EOM-CCSD	0.0100	210	78802.9	81543.1	84147.8	9.157	8.684	8.244	2740.2	56.8
EOM-CCSD	0.0050	420	78803.1	81537.1	84136.2	9.158	8.685	8.246	2734.0	56.6
CR-EOM-CCSD(T)	0.0250	84	77821.3	80626.8	83301.2	9.217	8.748	8.315	2805.5	53.1
CR-EOM-CCSD(T)	0.0200	104	77821.4	80620.8	83289.6	9.218	8.749	8.317	2799.3	52.9
CR-EOM-CCSD(T)	0.0150	140	77821.6	80614.3	83277.2	9.218	8.751	8.320	2792.7	52.7
CR-EOM-CCSD(T)	0.0100	210	77821.8	80608.2	83265.2	9.219	8.752	8.322	2786.4	52.5
CR-EOM-CCSD(T)	0.0050	420	77822.0	80602.0	83253.4	9.219	8.754	8.325	2780.0	52.3

Table 8: Spectroscopic parameter of the $C^1\Pi$ state of HCl. The spectroscopc parameters were calculated using the AIREMPICalc program. The points of the potential curve were used directly in the Fourier Grid Hamiltonian method.

5.1.4 The $F^1\Delta_2$ state

The $F^1\Delta_2$ state has been studied by Green et al. [4, 5, 6] and more recently in the photochemistry group of Ágúst Kvaran [2]. It is the lowest *Delta* state of HCl.

Table 9 shows the results of using the previously mentioned manual Morse potential fit methodology, with a threshold of 9000 cm^{-1}. This table was previously published in [2] where a description of the homegrown Igor Pro macro set, FCF1, is found. Andras Bodi calculated TD-DFT *ab-initio* potential curves that were used to make a part of this table. If we look at the results in this table, the best $T_{v=0}$ is clearly the EOM-CCSD/AQZ value, where the difference between it and the experimental value is only around 160 cm^{-1}.

The CR-EOM-CCSD(T) $T_{v=0}$ values are yet again slightly better than the EOM-CCSD values. However, there is a lot more variance in the results here, than for the $C^1\Pi$ and $X^1\Sigma^+$ states. However, there is a general trend for the EOM-CC methods to overestimate the ω_e of the state. The TD-DFT methods are even higher for ω_e. The same holds for the anharmonic constant, $\omega_e x_e$. However, the equilibrium bond length and the rotational constants are somewhat better for the TD-DFT calculations.

If we look now at the results of using the AIREMPICalc program. Table 10 shows the result of using a Morse potential to fit the *ab-initio* potential curves, both without and with a threshold of 9000 cm^{-1}. Most of the spectroscopic constants are not affected much by the threshold, but the dissociation energy, D_e, is affected to a larger extent than the others. The largest deviation from it, in case of the fitted parameters without the threshold is approximately 16000 cm^{-1}, but in the case of the fit with a threshold, the largest deviation is around 4000 cm^{-1}.

When we look at the results from AIREMPICalc when using a interpolated spline function, it is clear that the effect of going from EOM-CCSD to CR-EOM-CCSD(T) is slight, but not very much. The $T_{v=0}$ and ω_e get closer to their experimental values, $B_{v=0}$ goes only slightly closer to its experimental value. However $\omega_e x_e$ deviates from the experimental value.

The same is seen in table 12 where a direct evaluation method is used for AIREMPICalc. The effect of using a larger grid is very little.

Method	$T_{v=0}$ [cm^{-1}]	r_e [Å]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]	B_e [cm^{-1}]	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]
Experiments	81555.3875[3]	1.295[1]	2608.3[1]	49.35[1]	10.415	10.3246	10.143
					10.412[4]	10.3228[2]	10.1447[2]
Bettendorff et al.[5]	79930[5]	1.314[5]	2715[5]		9.96[5]		
			2813[56]	69[56]	9.68[5 6]	9.42[56]	8.90[56]
TD-DFT B3LYP/ aug-cc-pVTZ	77810	1.304	2916	75	10.12	9.83	9.27
TD-DFT B3LYP/ aug-cc-pVQZ	76079	1.303	2881	73	10.14	9.86	9.30
TD-DFT MPW1PW91/ aug-cc-pVQZ	78037	1.300	2906	68	10.18	9.90	9.36
EOM-CCSD/AQZ	81391	1.317	2681	58	9.92	9.65	9.13
EOM-CCSD/aug-cc-pVTZ	84924	1.311	2731	58	10.02	9.75	9.24
EOM-CCSD/aug-cc-pVQZ	84219	1.314	2711	58	9.97	9.71	9.19
EOM-CCSD/aug-cc-pV5Z	84023	1.308	2736	63	10.06	9.79	9.25
EOM-CCSD/aug-cc-pCVQZ	84635	1.315	2719	59	10.01	9.74	9.22
CR-EOM-CCSD(T)/aug-cc-pVTZ	84058	1.309	2758	55	10.05	9.79	9.29
CR-EOM-CCSD(T)/aug-cc-pVQZ	83141	1.312	2736	56	9.99	9.74	9.24
CR-EOM-CCSD(T)/aug-cc-pV5Z	82707	1.306	2755	61	10.09	9.81	9.28
CR-EOM-CCSD(T)/aug-cc-pCVQZ	83109	1.311	2740	57	10.02	9.76	9.25

Table 9: Spectroscopic parameter of the $F^1\Delta_2$ state of HCl. They were obtained with various *ab initio* calculations and experiment. These results are based on fitted potential curves, that were fitted with a Morse potential. The parameters were then calculated using the methods described in [2] where they were previously published.
The *ab-initio* calculations that are in **bold** were performed by Andras Bodi, however I did the subsequent calculation of spectroscopic parameters.
1) ref. [18]; 2) ref. [4]; 3) $T_0 = \nu_0 - (\omega_e/2 - omega_e x_e/4)$; ν_0 from ref. [4]; $omega_e$, $\omega_e x_e$ from ref. [18]; 4) derived from fitting B_v vs. v according to $B_v = B_e - \alpha_e * (v + 1/2)$ for B_v from ref. [4]; 5) ref. [8]; 6) derived from Morse potential fitting of a published potential curve in reference [8]

Method	$T_{v=0}$ [cm^{-1}]	$T_{v=1}$ [cm^{-1}]	$T_{v=2}$ [cm^{-1}]	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]	$B_{v=2}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]	D_e [cm^{-1}]
Experiment [6]	82847.2	85363.7	87772.4	10.3246	10.143	10.256	2624	54	31877
Bettendorff [8, 2]	79930	—	—	9.42	8.90	—	2715	69 2813	
EOM-CCSD/aug-cc-pVTZ	86233.7	88875.2	91406.3	9.724	9.224	8.734	2641.5	53.8	33990
EOM-CCSD/aug-cc-pVQZ	85659.3	88177.5	90612.5	9.479	9.039	8.608	2518.2	38.5	40130
EOM-CCSD/aug-cc-pV5Z	85339.2	87934.4	90396.8	9.789	9.229	8.680	2595.2	66.0	27903
EOM-CCSD/aug-cc-pCVTZ	86575.1	89187.8	91699.2	9.692	9.209	8.735	2612.7	48.6	35915
EOM-CCSD/aug-cc-pCVQZ	85936.7	88559.5	91073.3	9.704	9.205	8.716	2622.8	53.1	33926
CR-EOM-CCSD(T)/aug-cc-pVTZ	85368.7	88047.2	90624.2	9.743	9.267	8.799	2678.5	48.4	37565
CR-EOM-CCSD(T)/aug-cc-pVQZ	84617.5	87147.3	89606.8	9.450	9.047	8.650	2529.8	30.2	47080
CR-EOM-CCSD(T)/aug-cc-pV5Z	84033.6	86650.3	89136.2	9.814	9.260	8.716	2616.7	64.9	28712
CR-EOM-CCSD(T)/aug-cc-pCVTZ	85274.3	87908.8	90453.5	9.692	9.238	8.791	2634.4	41.4	40557
CR-EOM-CCSD(T)/aug-cc-pCVQZ	84408.4	87059.3	89609.6	9.709	9.233	8.766	2650.9	48.0	37164
With a threshold									
EOM-CCSD/aug-cc-pVTZ	86254.6	88870.0	91370.5	9.753	9.237	8.730	2615.4	56.3	32178
EOM-CCSD/aug-cc-pVQZ	85552.6	88141.4	90618.5	9.706	9.194	8.694	2588.8	54.7	32361
EOM-CCSD/aug-cc-pV5Z	85337.3	87948.0	90434.3	9.789	9.248	8.718	2610.7	61.6	29862
EOM-CCSD/aug-cc-pCVTZ	86608.6	89223.2	91719.8	9.758	9.234	8.720	2614.6	58.1	31405
EOM-CCSD/aug-cc-pCVQZ	85959.0	88560.5	91044.7	9.737	9.214	8.701	2601.5	57.7	31278
CR-EOM-CCSD(T)/aug-cc-pVTZ	85400.3	88055.9	90598.9	9.791	9.283	8.785	2655.6	54.8	33693
CR-EOM-CCSD(T)/aug-cc-pVQZ	84488.2	87105.3	89616.3	9.735	9.238	8.751	2617.1	51.3	34541
CR-EOM-CCSD(T)/aug-cc-pV5Z	84031.6	86664.5	89175.6	9.814	9.280	8.757	2632.9	60.1	30894
CR-EOM-CCSD(T)/aug-cc-pCVTZ	85319.7	87963.8	90494.9	9.788	9.277	8.775	2644.1	55.2	33265
CR-EOM-CCSD(T)/aug-cc-pCVQZ	84443.8	87070.5	89584.7	9.760	9.249	8.748	2626.8	55.0	32997

Table 10: Spectroscopic parameter of the $F^1\Delta_2$ state of HCl. They were obtained with various *ab initio* calculations and experiment. The AIREMPICalc program was used to fit the *ab-initio* potential curves with a Morse potential and then the Fourier Grid Hamiltonian was used (200 grid points). The entire calculated curve was used is some cases and in others, there was used a threshold of 9000.0 cm^{-1}, i.e. all values, where the energy was less than 9000.0 cm^{-1} more than the lowest value in the potential curve, were used.

Method	$T_{v=0}$ [cm^{-1}]	$T_{v=1}$ [cm^{-1}]	$T_{v=2}$ [cm^{-1}]	$B_{v=0}$ [cm^{-1}]	$B_{v=1}$ [cm^{-1}]	$B_{v=2}$ [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]
Experiment [6]	82847.2	85363.7	87772.4	10.3246	10.143	10.256	2624	54
Bettendorff [8, 2]	79930	—	—	9.42	8.90	—	2715	69
								2813
EOM-CCSD/aug-cc-pVTZ	86257.6	88866.5	91369.2	9.756	9.231	8.724	2608.9	46.8
EOM-CCSD/aug-cc-pVQZ	85553.9	88136.2	90616.5	9.710	9.190	8.688	2582.3	45.7
EOM-CCSD/aug-cc-pCVTZ	86611.9	89217.1	91714.8	9.759	9.231	8.721	2605.2	46.3
EOM-CCSD/aug-cc-pCVQZ	85964.1	88552.4	91036.0	9.738	9.211	8.705	2588.3	45.7
CR-EOM-CCSD(T)/aug-cc-pVTZ	85406.3	88046.9	90588.6	9.794	9.280	8.786	2640.6	39.8
CR-EOM-CCSD(T)/aug-cc-pVQZ	84489.8	87099.2	89613.9	9.741	9.231	8.742	2609.4	38.8
CR-EOM-CCSD(T)/aug-cc-pCVTZ	85324.1	87956.7	90488.7	9.790	9.272	8.775	2632.6	39.6
CR-EOM-CCSD(T)/aug-cc-pCVQZ	84450.1	87060.9	89574.3	9.762	9.245	8.750	2610.8	38.9

Table 11: Spectroscopic parameter of the $F^1\Delta_2$ state of HCl. They were obtained with various *ab initio* calculations and experiment. These parameters were calculated using the AIREMPICalc program. The potential curves used a spline to interpolate values on the potential curve for the Fourier Grid Hamiltonian calculations, which used 200 grid points. The CCSD/aug-cc-pV5Z potential curve is not large enough to use this method, as its outer limit is r = 1.48 Å.

Method	Δx [Å]	n_x	$T_{v=0}$ [cm^{-1}]	$T_{v=1}$ [cm^{-1}]	$T_{v=2}$ [cm^{-1}]	$B_{v=0}$ [Å]	$B_{v=1}$ [Å]	$B_{v=2}$ [Å]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]
Experiment [6]			82847.2	85363.7	87772.4	10.3246	10.143	10.256	2624	54
Bettendorff [8, 2]			79930	—	—	9.42	8.90	—	2715 2813	69
EOM-CCSD	0.0250	84	86256.1	88894.6	91423.4	9.753	9.224	8.718	2638.5	50.2
EOM-CCSD	0.0200	104	86256.5	88889.2	91412.6	9.753	9.225	8.721	2632.7	50.0
EOM-CCSD	0.0150	140	86256.9	88883.3	91400.9	9.754	9.227	8.724	2626.4	49.7
EOM-CCSD	0.0100	210	86257.2	88877.6	91389.8	9.755	9.229	8.727	2620.4	49.5
EOM-CCSD	0.0050	420	86257.6	88872.0	91378.7	9.755	9.231	8.730	2614.4	49.3
CR-EOM-CCSD(T)	0.0250	84	85405.0	88075.5	90643.2	9.791	9.273	8.782	2670.5	44.5
CR-EOM-CCSD(T)	0.0200	104	85405.3	88069.9	90632.1	9.791	9.275	8.785	2664.6	44.3
CR-EOM-CCSD(T)	0.0150	140	85405.7	88063.9	90620.2	9.792	9.277	8.788	2658.2	44.1
CR-EOM-CCSD(T)	0.0100	210	85406.0	88058.1	90608.8	9.793	9.279	8.791	2652.1	43.9
CR-EOM-CCSD(T)	0.0050	420	85406.3	88052.4	90597.4	9.793	9.281	8.794	2646.0	43.7

Table 12: Spectroscopic parameter of the $F^1\Delta_2$ state of HCl. The spectroscopc parameters were calculated using the AIREMPICalc program. The points of the potential curve were used directly in the Fourier Grid Hamiltonian method.

5.2 HF

Hydrogen fluoride is another of the hydrogen halides that has been studied at the Science Institute by the Kvaran research group. However the experimental results are inconclusive since the photodissociation processes that HF undergoes seem to be very fast. So HF dissociates too quickly to be usable in REMPI studies, so the amount of experimental results on the HF molecule is little. However the HF dimer was discoverd by Kvaran, et al. in 2006 [24].

Older experiments used absorptions in the electronic spectra to probe the structure of HF. Di Lonardo and Douglas did a study in 1973 on the $B^1\Sigma^+$ state [93]. Not much more work has been done on the excited states of HF because of experimental difficulties.

I performed *ab-initio* calculations using the ACESII program for HF. These results were then fed in the AIREMPICalc program. I focused on three states. The $X^1\Sigma^+$ ground state which all the calculations found easily. The $C^1\Pi$ state was selected as the potential was without gaps and jumps in most of the calculated curves. The $B^1\Sigma^+$ was then select as its distinguishing features are easily found in the results of the calculations, i.e. it's easy to see the ion-pair state amongst a bundle of Rydberg states. Only results from the calculations that have an experimental counterpart are displayed in the result tables.

Table 13 shows the results of using a Morse potential to fit the *ab-initio* potential curve of the $X^1\Sigma^+$ ground state, with and without a threshold of 5000.0 cm^{-1}. These fits are then fed into the FGH method to calculate the spectroscopic constants. The effect of using a threshold depends on what spectroscopic parameter one looks at, as the ω_e and $\omega_e x_e$ get worse (deviates more from the experimental value) for it. The r_e, B_e and B_0 are all improved by the usage of a threshold.

Next we look at table 14, where a spline interpolation is used instead of a Morse potential fit. The results are similar to the fitted results (w/threshold), however $\omega_e x_e$ is considerably better.

Next we turn our attention to the lower Rydberg state, the $C^1\Pi$ state. The experimental values are not of a high quality but it is interesting to look at tables 15 and 16. Table 15 uses a similar fit methodology as before (threshold is 9000.0 cm^{-1}). The threshold fits are a bit closer for B_e, similar for r_e. However, the ω_e is a lot closer and the same holds for T_e. Table 16 shows the results when using spline interpolation methodology for this state. It has similar results to the fitted results (w/threshold).

The $B^1\Sigma^+$ state, seems to be a bit more of a challenge, since all the fits I tried failed, i.e. the results were non-physical, such as negative excitation energies and bond lengths. But nonetheless the spline interpolation methodology performed fairly as is seen in table 17. Despite the odd negative values of $\omega_e x_e$, the magnitude of most of them are correct. There is a possibility that this is a subtle bug in AIREMPICalc as the ω_e values are around 10% from the experimental value. The T_0 is also off by 10%.

There is a possibility that the potential curves for the $B^1\Sigma^+$ state are incorrect since there is a fair amount of potential crossing between it and other excited states. This means that there could be some errors in the ordering of excited states of some bond lengths. This would impact the results to some degree. It could be interesting to look at this phenomenon in the future.

Method	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]	r_e [Å]	B_e [cm^{-1}]	B_0 [cm^{-1}]
Experiment[1]	4138.32	89.88	0.91680	20.9557	20.5567
CCSD/aug-cc-pV5Z	4198.7	82.8	0.926	20.054	19.951
CCSD/aug-cc-pVQZ	4200.6	83.3	0.927	20.045	19.940
CCSDT/aug-cc-pVQZ	4123.4	86.2	0.929	19.937	19.814
CCSD/d-aug-cc-pV5Z	4198.1	82.8	0.926	20.053	19.950
CCSD/d-aug-cc-pVQZ	4198.4	83.1	0.927	20.036	19.932
CCSD/t-aug-cc-pVQZ	4198.9	83.2	0.927	20.038	19.934
UHF-CCSD/aug-cc-pV5Z	4198.9	82.8	0.926	20.055	19.952
UHF-CCSD/aug-cc-pVQZ	4200.6	83.3	0.927	20.045	19.940
With threshold					
CCSD/aug-cc-pV5Z	4015.7	101.3	0.913	20.630	20.424
CCSD/aug-cc-pVQZ	4019.1	100.8	0.914	20.603	20.400
CCSDT/aug-cc-pVQZ	3967.3	101.8	0.917	20.477	20.264
CCSD/d-aug-cc-pV5Z	4015.5	101.0	0.913	20.629	20.424
CCSD/d-aug-cc-pVQZ	4017.8	101.3	0.914	20.604	20.399
CCSD/t-aug-cc-pVQZ	4018.3	101.3	0.914	20.605	20.401
UHF-CCSD/aug-cc-pV5Z	4015.7	101.3	0.913	20.630	20.424
UHF-CCSD/aug-cc-pVQZ	4019.1	100.8	0.914	20.603	20.400

Table 13: The $X^1\Sigma^+$ state of HF. These spectroscopic constant were calculated with AIREMPICalc using a Morse potential to fit *ab-initio* potentials and then inputing that into the Fourier Grid Hamiltonian method (200 grid points). Note that UHF means that the calculations were open-shell calculations. The threshold used was 5000.0 cm^{-1}.
[1] Obtained from the NIST Webbook http://webbook.nist.gov/cgi/cbook.cgi?ID=C7664393&Units=SI&Mask=1000

Method	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]	B_0 [cm^{-1}]
Experiment[1]	4138.32	89.88	29.5567
CCSD/aug-cc-pV5Z	4025.2	77.2	20.422
CCSD/aug-cc-pVQZ	4027.5	77.3	20.398
CCSDT/aug-cc-pVQZ	3976.0	79.9	20.263
CCSD/d-aug-cc-pV5Z	4024.9	77.2	20.422
CCSD/d-aug-cc-pVQZ	4026.3	77.3	20.397
CCSD/t-aug-cc-pVQZ	4026.8	77.3	20.399
UHF-CCSD/aug-cc-pV5Z	4025.2	77.2	20.422
UHF-CCSD/aug-cc-pVQZ	4027.5	77.3	20.398

Table 14: The $X^1\Sigma^+$ state of HF. These spectroscopic constant were calculated with AIREMPICalc using a spline interpolation of an a *ab-initio* potential in the Fourier Grid Hamiltonian method (200 grid points). Note that UHF means that the calculations were open-shell calculations.
[1] Obtained from the NIST Webbook http://webbook.nist.gov/cgi/cbook.cgi?ID=C7664393&Units=SI&Mask=1000

	T_e [cm^{-1}]	ω_e [cm^{-1}]	r_e [Å]	B_e [cm^{-1}]
Experiment[1]	105280	2636	1.04	16.0
EOM-CCSD/aug-cc-pV5Z	104432	3219.2	1.032	16.2
EOM-CCSD/aug-cc-pVQZ	104823	3171.8	1.045	15.8
EOM-CCSDT/aug-cc-pVQZ	105623	3178.4	1.044	15.8
EOM-CCSD/daug-cc-pV5Z	103429	3284.7	1.015	16.7
EOM-CCSD/daug-cc-pVQZ	103049	3282.7	1.017	16.6
EOM-CCSD/taug-cc-pVQZ	103029	3285.2	1.016	16.7
With a threshold				
EOM-CCSD/aug-cc-pV5Z	106597	2746.0	1.035	16.1
EOM-CCSD/aug-cc-pVQZ	107083	2702.9	1.049	15.6
EOM-CCSDT/aug-cc-pVQZ	107888	2679.7	1.052	15.5
EOM-CCSD/daug-cc-pV5Z	105260	2844.1	1.019	16.6
EOM-CCSD/daug-cc-pVQZ	104938	2841.3	1.019	16.6
EOM-CCSD/taug-cc-pVQZ	104912	2841.9	1.019	16.6

Table 15: The $C^1\Pi$ state of HF. These spectroscopic constant were calculated with AIREMPICalc using a Morse potential to fit *ab-initio* potentials and then inputing that into the Fourier Grid Hamiltonian method (200 grid points). The threshold used was 9000.0 cm^{-1}.
[1] Obtained from the NIST Webbook http://webbook.nist.gov/cgi/cbook.cgi?ID=C7664393&Units=SI&Mask=1000

Method	$T_{v=0}$ [cm^{-1}]	ω_e [cm^{-1}]
Experiment[1]	105090.8	2636
EOM-CCSD/aug-cc-pV5Z	105973.7	2707.7
EOM-CCSD/aug-cc-pVQZ	106417.4	2683.7
EOM-CCSDT/aug-cc-PVQZ	107229.2	2657.8
EOM-CCSD/daug-cc-PV5Z	104694.4	2806.7
EOM-CCSD/daug-cc-PVQZ	104368.5	2801.3
EOM-CCSD/taug-cc-PVQZ	104343.3	2801.4

Table 16: The $C^1\Pi$ state of HF. These spectroscopic constant were calculated with AIREMPICalc using a spline interpolation of an a *ab-initio* potential in the Fourier Grid Hamiltonian method (200 grid points).
[1] Obtained from the NIST Webbook http://webbook.nist.gov/cgi/cbook.cgi?ID=C7664393&Units=SI&Mask=1000

Method	T_0 [cm^{-1}]	ω_e [cm^{-1}]	$\omega_e x_e$ [cm^{-1}]	B_0 [cm^{-1}]
Experimental	83304.96	1159.18	18.005	4.0203
EOM-CCSD/aug-cc-pV5Z	91143.3	1069.9	-17.5	4.416
EOM-CCSD/aug-cc-pVQZ	91085.1	1071.5	-17.9	4.427
EOM-CCSDT/aug-cc-pVQZ	94235.5	1167.2	0.3	5.026
EOM-CCSD/daug-cc-pV5Z	91127.6	1069.2	-15.3	4.416
EOM-CCSD/daug-cc-pVQZ	91048.0	1071.0	-14.6	4.432

Table 17: The $B^1\Sigma^+$ state of HF. These spectroscopic constant were calculated with AIREMPICalc using a spline interpolation of an a *ab-initio* potential in the Fourier Grid Hamiltonian method (200 grid points).
[1] Obtained from the NIST Webbook http://webbook.nist.gov/cgi/cbook.cgi?ID=C7664393&Units=SI&Mask=1000

6 REMPIControl

REMPIControl is a set of programs that controls the excimer/dye laser setup, at Raunvísindastofnun Háskólans, that's currently being used for REMPI-TOF experiments. They were programmed in the LabVIEW [16] development environment. I used LabVIEW 8.0 for MacOSX and Windows.

Each program and subroutine in LabVIEW is contained in a file called a virtual instrument (VI). This programming work was based on the manual of the LeCroy 9310A digital oscilloscope [94] and the Scanmate*Pro* dye LASER manual [95].

6.1 Oscilloscope

The VIs provided by the manufacturer of the oscilloscope are designed to work with a General Purpose Interface Bus (GPIB). However the lab computer does not have such a port, so the oscilloscope is connected through the serial port (COM10). This meant that to be able to communicate with the oscilloscope, a new driver for Labview was a necessity.

So I developed two VI modules to handle the communcations with the oscilloscope. The first one `oscilloscope_setup` sets up the serial port for the program and initializes the oscilloscope with the communication parameters, such as the data transfer rate, what characters signify a line break, i.e. the carriage return or line feed characters. This module also sets the oscilloscope to the remote mode, i.e. blocks tampering with the settings on the oscilloscope. When the program finishes a run it releases the oscilloscope from this mode.

The other VI module, `oscilloscope_read` handles all the grunt work. It reads data from the oscilloscope and then clears the math buffer[17] of the oscilloscope. A lot of work went into this part as the Labview and oscilloscope manuals were somewhat ambigious on some points.

One interesting issue that had to be settled was interaction between the endianness of the oscilloscope, computer and Labview. Endianness is the ordering of integer values in computer memory. The number one thousand, two hundred thirty four, 1234 is stored as a sequence of

[16] Short for Laboratory Virtual Instrumentation Engineering Workbench

[17] The math buffer, also known as Display A in this case, stores an average of the measurements, though this is possible to configure on the oscilloscope itself.

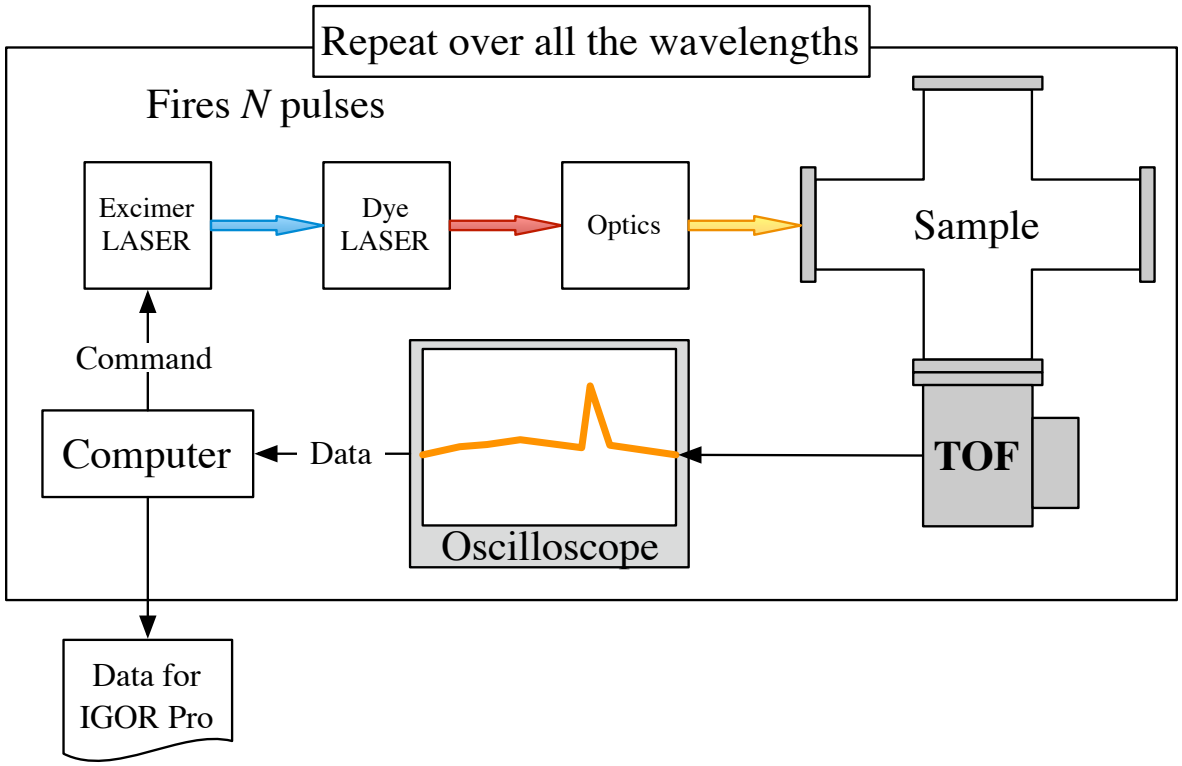


Figure 8: REMPIControl is a program that controls the experimental setup in the picture. The computer sets up the oscilloscope and dye LASER. The dye LASER then handles the excimer LASER. Then the computer oversees the dye LASER and oscilloscope and controls the experimental setup.

smaller numbers, if one would write it out in the more natural way (at least in English and Icelandic), it would be, `one`, `two`, `three`, `four`. However if it were a little-endian ordering, it would be `four`, `three`, `two`, `one`. This is understandably an issue when transferring integer values (the oscilloscope stores the data that way) of the oscilloscope and onto a computer.

It is possible to set the endianness of the oscilloscope to either little- or big-endian. The CPU of the computer is an Intel x86 architecture chip, so it is little-endian. However Labview works in big-endian. This means that one needs to take a serious look at this issue or the result will be subtle nonsense, but nonsense none the less.

6.2 Dye LASER

Labview VIs were included with the Scanmate Pro. These were used to set up most of the functionality of the REMPIControl suite. The VIs handle the intricacies of communicating with the dye laser. Two modules were programmed, `REMPI_Record` and `REMPI3`.

The `REMPI3` module is based on the `Scan` example module, provided by the manufacturer of the dye laser. It was modified to read data from an oscilloscope during scanning. This also

displays the data retrieved in each scan step on the computer screen. There is also a grap of the integral of the data. This integral is numerically integrated over the oscilloscope data. It is possible to select a region of the data to focus on.

The other module, `REMPI_Record`, handles the experimental setup during a repeating firing on a single wavelength, i.e. repeats the LASER firing sequence at a single wavelength.

This module was based on the `Record` VI, provided by the manufacturer of the dye LASER. Interestingly, the manual [94] was incorrect in stating that to trigger the dye laser, the `N` serial command would be needed. So if 50 laser shots were repeated 10 times, the dye laser would trigger the excimer laser 500 times at the chosen frequency. This is clearly a bug! However, this is a bug that's easy to circumvent, as it's simple to simply set it so that `REMPI_Record` would fire n shots once and then repeat the procedure by calling `setup_record` again.

7 Conclusions

This thesis looked at *ab-initio* calculations of excited states, both the theory and its application to the excited states of hydrogen chloride and hydrogen fluoride. The spectroscopic constants of the excited states that the Kvaran's research group has interest in were evaluated with both manual calculations and a specifically designed program.

The AIREMPICalc program, which was implemented using Python and Scipy, is capable of using *ab-initio* potentical curves of diatomic molecules to calculate their spectroscopic constants in very simple fashion, i.e. a setup file and a potential file are input into a program which then outputs the spectroscopic constants.

These calculations result in spectroscopic constants that can be close to the experimental values. This depends somewhat on the level of *ab-initio* calculations, however which internal calculations methodology is used in AIREMPICalc is also important. However, this program gives a simple method to use with new high-level *ab-initio* calculations.

This program was used to evaluate the spectroscopic constants of the following states: the $X^1\Sigma^+$, $C^1\Pi$ and $F^1\Delta_2$ states of HCl and the $X^1\Sigma^+$, $C^1\Pi$ and $B^1\Sigma^+$ states of HF. The results are fairly good and it should be straightforward to improve them with future calculations.

This thesis also describes the development and function of a control program for the REMPI experimental setup at the University of Iceland's Science Institue. The control program was implemented in Labview.

A Appendices

A.1 Acronyms

Methods

Acronym	Meaning
AIREMPICalc	*Ab initio* Resonance enhanced multi-photon ionization calculation program
B3LYP	Becke 3-parameter Lee-Yang-Parr
CBS	Complete basis set
CCSD	Coupled cluster singles doubles
CCSD(T)	Coupled cluster singles doubles perturbative triples
CCSDT	Coupled cluster singles doubles triples
CCSDTQ	Coupled cluster singles doubles triples quadruples
CI	Configuration Interaction
CISD	Configuration Interaction Singles Doubles
CR-CCSD(T)	Completely renormalized coupled cluster singles doubles perturbative triples
CR-EOM-CCSD(T)	Completely renormalized equation-of-motion coupled cluster singles doubles perturbative triples
EOMCCSD	Equation-of-motion coupled cluster singles doubles
FCI	Full Configuration Interaction
FGH	Fourier Grid Hamiltonian
HF	Hydrogen fluoride (not Hartree-Fock within this thesis)
GGA	Generalized gradient approximation
LDA	Local density approximation
MP2	Møller-Plesset perturbation theory to the second order
MP4	Møller-Plesset perturbation theory to the fourth order
MR	Multi-reference
SCF	Self-consistent field
REMPI	Resonance enhanced multi-photon ionization
VI	Virtual Instrument

Table 18: Various acronyms that are used in this thesis.

A.2 Raw data

A.3 AIREMPICalc - Source code

Ab Initio REMPI Calc is a program that calculates REMPI (2+1) and REMPI (3+1) spectra by using diatomic potentials.

This program implements the Fourier grid hamiltonian method of section 4.2. This is actually a re-implementation as Balint-Kurti et al. [87] show a implementation of the method called `FGHEVEN`. That program was in Fortran 77 and used atomic units throughouts. Here it is implemented as a Python function using cm^{-1} (wavenumbers) as the energy units and angstrom as the unit of length.

Please note that tabs are important for the syntax of the Python programming language, so tabs are printed here with the → character.

A.3.1 airempi.py

```
1  #!/usr/bin/python
2  from numpy import *
3  from scipy import linalg
4  import airempi
5  import math
6  import potent
7  import fitter
8  import fgh
9  import spec
10
11 #from pylab import *
12 #from scipy import interpolate
13
14 def main():
15 → """main"""
16 → # Load the setup file
17 → import ConfigParser
18 → import string
19
20 → debug = 1
21
22 → config = ConfigParser.ConfigParser()
23
24 → config.read("setup")
25
26 → ## GENERAL POTENTIAL SETUP
27 → number_of_potentials = config.getint('potentials','number')
28 → pot_files = config.get('potentials','pot_file').split(',')
29 → do_fit = config.getboolean('potentials','fit') # Are we fitting the
      potential ?
30 → do_threshold = config.getboolean('potentials','threshold') # when we fit,
       do we use a threshold ?
31 → do_inter = config.getboolean('potentials','interpolate') # should we
      interpolate U(r) ?
```

```
32  → do_direct = config.getboolean('potentials','direct') # are we using
      straight values from the potential file ?
33  → no_fgh_points = config.getint('fgh','points')
34  → no_fgh_vib_levels = config.getint('fgh','vib_levels')
35  → rmin_factor = config.getfloat('fgh','rmin_factor')
36  → rmax_factor = config.getfloat('fgh','rmax_factor')
37  → print 'rmin_factor = ', rmin_factor
38  → print 'rmax_factor = ', rmax_factor
39
40  → ## MOLECULE SETUP
41  → mol_name = config.get('molecule','name') # What molecule?
42  → mol_red_mass = config.getfloat('molecule','reduced_mass') # The reduced
      molecular mass
43  → mol_r0 = config.getfloat('molecule','r0')
44
45  → ## GROUND STATE
46  → gs_name = config.get('ground_state','name')
47  → gs_threshold = config.getfloat('ground_state','threshold')
48
49  → ## EXCITED STATES
50  → ex_name = config.get('excited_states','name').split(',')
51  → no_ex = config.getint('excited_states','no_ex')
52  → ex_pos = [int(i) for i in config.get('excited_states','pos').split(',')]
53  → ex_threshold = config.getfloat('excited_states','threshold')
54
55  → ## Doing sanity check
56  → if ((no_ex != len(ex_pos)) or (no_ex != len(ex_name))) and (no_ex != 0):
      # do the number match up?
57  → → print 'Mismatch in no_ex: ', no_ex, '; ex_pos: ', len(ex_pos),';
      ex_name: ', len(ex_name)
58  → → return
59  → if do_fit: # if we are fitting
60  → → fit_type = config.get('fit_setup','fit_type') # what kind of a
      diatomic potential are we using, a morse potential ?
61  → → fit_guess = [float(i) for i in config.get('fit_setup','fit_guess').
      split(',')]
62  → → if fit_type == '': # do we have a fitting type?
63  → → → print 'No defined fit type'
64  → → → return
65
66  → if not (do_fit or do_inter or do_direct):
67  → → print 'There is nothing to do!'
68  → → return
69
70  → #if (do_fit and do_inter) or (do_fit and do_direct) or (do_inter and
      do_direct):
71  → if sum([do_fit,do_inter,do_direct]) >= 2: # Prettier than the syntax
      above
72  → → print 'Doing multiple passes!'
73
74  → ## Our setup should be sane, so we move forward
```

```
75  → ## However this doesn't check the validity of the data in the text files
76  → ## But let's begin the calculations
77
78  → print 'Starting calculations'
79  → for i in range(number_of_potentials):
80  → → print 'Running on potential file: ', pot_files[i]
81  → → r, gs, ex, state_names = potent.loader(pot_files[i])
82  → → # print '  GS name: ', gs_name, ', in file: ', state_names[1]
83  → → # for j in range(no_ex):
84  → → #→ print '  EX name: ', ex_name[j], ', in file: ', state_names[ex_pos[
        j]+2]
85  → → if do_fit: # start
86  → → → if fit_type.lower() == 'morse':
87  → → → → print "  Fitting with a morse potential"
88  → → → → fit_pot = fitter.morse_fitter
89  → → → else:
90  → → → → print "Not a defined fit type!"
91  → → → → return
92  → → → ## Now do the fitting
93  → → → print '  GS name: ', gs_name, ', in file: ', state_names[1]
94  → → → if do_threshold:
95  → → → → gs_fit, gs_pot = fitter.small_fitter(fit_pot,gs_threshold,r,gs,
        fit_guess)
96  → → → else:
97  → → → → gs_fit, gs_pot = fit_pot(r,gs,fit_guess)
98  → → → print '    Fitted parameters: ', gs_fit[0]
99
100 → → → fitted_list = [gs_pot]
101 → → → for j in range(no_ex):
102 → → → → print '  EX name: ', ex_name[j], ', in file: ', state_names[
        ex_pos[j]+2]
103 → → → → if do_threshold:
104 → → → → → ex_fit, ex_pot = fitter.small_fitter(fit_pot,ex_threshold,r,
        ex[:,ex_pos[j]],fit_guess)
105 → → → → else:
106 → → → → → ex_fit, ex_pot = fit_pot(r,ex[:,ex_pos[j]],fit_guess)
107 → → → → print '    Fitted parameters: ', ex_fit[0]
108 → → → → fitted_list.append(ex_pot)
109 → → → ## Now do the FGH
110 → → → r_vs = []
111 → → → B_vs = []
112 → → → wes = []
113 → → → wexes = []
114 → → → T_vs = []
115 → → → for f in fitted_list:
116 → → → → val, vec, x = fgh.wave(f, mol_red_mass, mol_r0,no_fgh_points,
        rmin_factor,rmax_factor)
117 → → → → r_v, B_v, we, wexe = spec.constants(x,mol_red_mass,vec,
        no_fgh_vib_levels,val)
118 → → → → T_vs.append(val[0:no_fgh_vib_levels])
119 → → → → r_vs.append(r_v)
```

```
120  → → → → B_vs.append(B_v)
121  → → → → wes.append(we)
122  → → → → wexes.append(wexe)
123  → → → ## Outputting the spectroscopic parameters
124  → → → print '  GS name: ', gs_name, ', in file: ', state_names[1]
125  → → → print '   r_v: ', r_vs[0]
126  → → → print '   B_v: ', B_vs[0]
127  → → → print '   wes: ', wes[0]
128  → → → print '   wexes: ', wexes[0]
129  → → → print '   T_v: ', T_vs[0]
130  → → → for j in range(no_ex):
131  → → → → print '  EX name: ', ex_name[j], ', in file: ', state_names[
       ex_pos[j]+2]
132  → → → → print '   r_v: ',   r_vs[j+1]
133  → → → → print '   B_v: ',   B_vs[j+1]
134  → → → → print '   wes: ',   wes[j+1]
135  → → → → print '   wexes: ', wexes[j+1]
136  → → → → print '   T_v: ', T_vs[j+1]
137
138  → → # do_fit end
139  → → if do_inter: # start do_inter
140  → → → print "  Making interpolated functions"
141  → → → print '    GS name: ', gs_name, ', in file: ', state_names[1]
142  → → → gs_pot = potent.make_spline(r,gs)
143  → → → inter_list = [gs_pot]
144  → → → for j in range(no_ex):
145  → → → → print '    EX name: ', ex_name[j], ', in file: ', state_names[
       ex_pos[j]+2]
146  → → → → ex_pot = potent.make_spline(r,ex[:,ex_pos[j]])
147  → → → → inter_list.append(ex_pot)
148  → → → ## Now do the FGH
149  → → → r_vs = []
150  → → → B_vs = []
151  → → → wes = []
152  → → → wexes = []
153  → → → T_vs = []
154  → → → for f in inter_list:
155  → → → → val, vec, x = fgh.wave(f, mol_red_mass, mol_r0,no_fgh_points,
       rmin_factor,rmax_factor)
156  → → → → r_v, B_v, we, wexe = spec.constants(x,mol_red_mass,vec,
       no_fgh_vib_levels,val)
157  → → → → r_vs.append(r_v)
158  → → → → B_vs.append(B_v)
159  → → → → wes.append(we)
160  → → → → wexes.append(wexe)
161  → → → → T_vs.append(val[0:no_fgh_vib_levels])
162  → → → print '  Spectroscopic constants from interpolation'
163  → → → ## Outputting the spectroscopic parameters
164  → → → print '  GS name: ', gs_name, ', in file: ', state_names[1]
165  → → → print '   r_v: ', r_vs[0]
166  → → → print '   B_v: ', B_vs[0]
```

```
167  → → → print '   wes: ', wes[0]
168  → → → print '   wexes: ', wexes[0]
169  → → → print '   T_v: ', T_vs[0]
170  → → → for j in range(no_ex):
171  → → → → print '  EX name: ', ex_name[j], ', in file: ', state_names[
       ex_pos[j]+2]
172  → → → → print '   r_v: ',   r_vs[j+1]
173  → → → → print '   B_v: ',   B_vs[j+1]
174  → → → → print '   wes: ',   wes[j+1]
175  → → → → print '   wexes: ', wexes[j+1]
176  → → → → print '   T_v: ', T_vs[j+1]
177  → → # do_inter end
178  → → if do_direct: # start do_direct
179  → → → print "  Using direct potentials !"
180  → → → ## Now do the FGH
181  → → → r_vs = []
182  → → → B_vs = []
183  → → → wes = []
184  → → → wexes = []
185  → → → T_vs = []
186  → → → print '  Spectroscopic constants from direct evaluation'
187  → → → ## Outputting the spectroscopic parameters
188  → → → val, vec = fgh.wave_direct(gs, mol_red_mass, r)
189  → → → r_v, B_v, we, wexe = spec.constants(r,mol_red_mass,vec,
       no_fgh_vib_levels,val)
190  → → → r_vs.append(r_v)
191  → → → B_vs.append(B_v)
192  → → → wes.append(we)
193  → → → wexes.append(wexe)
194  → → → print '  GS name: ', gs_name, ', in file: ', state_names[1]
195  → → → print '   r_v: ', r_vs[0]
196  → → → print '   B_v: ', B_vs[0]
197  → → → print '   wes: ', wes[0]
198  → → → print '   wexes: ', wexes[0]
199  → → → print '   T_v: ', T_vs[0]
200  → → → for j in range(no_ex):
201  → → → → val, vec = fgh.wave_direct(ex[:,ex_pos[j]], mol_red_mass, r)
202  → → → → r_v, B_v, we, wexe = spec.constants(r,mol_red_mass,vec,
       no_fgh_vib_levels,val)
203  → → → → r_vs.append(r_v)
204  → → → → B_vs.append(B_v)
205  → → → → wes.append(we)
206  → → → → wexes.append(wexe)
207  → → → → T_vs.append(val[0:no_fgh_vib_levels])
208  → → → → print '  EX name: ', ex_name[j], ', in file: ', state_names[
       ex_pos[j]+2]
209  → → → → print '   r_v: ',   r_vs[j+1]
210  → → → → print '   B_v: ',   B_vs[j+1]
211  → → → → print '   wes: ',   wes[j+1]
212  → → → → print '   wexes: ', wexes[j+1]
213  → → → → print '   T_v: ', T_vs[j+1]
```

```
214
215
216
217
218
219 if __name__ == "__main__":
220 → main()
221
222
223
224 def rempi():
225 → """docstring for rempi"""
226 → pass
```

A.3.2 fgh.py

```
1  from numpy import *
2  from scipy import linalg
3  #import airempi
4  import math
5  #import potent
6  #import fitter
7
8  def hf_potential_hartree(x):
9  → """hf_potential - returns the value of the potential at the x values (
       bohr)
10 → This is the Morse potential of HF ground-state - gotten from the original
       FGHEVEN program"""
11
12 → r0 = 1.7329 # The equilibrium bond length of HF [bohr]
13 → beta = 1.1741 # The exponential parameter of HF
14 → diss = 0.2250073497 # The dissociation energy of HF [hartree]
15
16 → return diss * (exp(-beta*(x-r0))-1.0)**2
17
18
19 def hcl_potential_morse(x):
20 → """hcl_potential_morse - returns the value of the potential at the x
       values (angstrom)
21 → These numbers are from NIST (http://webbook.nist.gov/cgi/cbook.cgi?ID=
       C7647010&Units=SI&Mask=1000#Diatomic)
22 → beta was calculated from them, """
23
24
25 → r_e = 1.27455 # The equilibrium bond length of HCl [angstrom]
26 → beta = 1.8663889082348262# The exponential parameter of HCl
27 → # calculated from NIST values:
28 → # beta = we * 0.12177 * sqrt(mju/De)
29 → # mju_HCl = 0.9795925068
30 → # we = 2990.946
31 → diss = 37302.584352507351 # The dissociation energy of HCl [cm^-1], found
       by 7.41e-19/1.60217733e-19 * 8065.5 (found at page 409 in Physical
```

```
    Chemistr by Engel & Reid)
32
33  → return diss * (exp(-beta*(x-r_e)) - 1.0)**2
34
35
36  def fgh_hf(the_potential):
37  → """docstring for fhg_hf"""
38  → zma = 1837.9822 # Mass of hydrogen
39  → zmb = 34629.61319 # Mass of fluorine
40  → r0 = 1.7329 # Equilibrium bond length, used to decide the grid
41  → print 'The mass of molecule A = ', zma
42  → print 'The mass of molecule B = ', zmb
43
44  → zmu = (zma*zmb) / (zma+zmb)
45
46  → return fgh_atomic(the_potential, zmu, r0)
47
48
49  def atomic(the_potential, zmu, r0):
50  → """fgh - Fourier Grid Hamiltonian - in atomic units
51  → the_potential is the potential to be used
52  → mju is the reduced mass
53  → r0 is the equilibrium bond length
54  → returns wch,zr and xa
55  → """
56  → ########
57  → ########
58  → # This function is a direct write-up of the FGHEVEN of G. Balint-Kurti,
59  → # C. Ward, and C. Clay Marston. Two computer programs for solving the
60  → # Schrodinger equation for bound-state eigenvalues and eigenfunctions
61  → # using the Fourier grid Hamiltonian method.
62  → #    Computer Physics Communications, 67(2):285-292, 1991.
63  → # The difference between FGHEVEN and this function is that FGHEVEN was
      written in Fortran 77
64  → ########
65  → ########
66
67  → nx = 64 # number of grid points
68  → ar = zeros((nx,nx),dtype=float) # The Hamiltonian matrix
69
70  → cevau = 27.211648 # Conversion in to electron volts, NOT USED
71  → nwrite = 10 # Number of eigenvalues/eigenfunctions to be printed
72  → rmin = r0*0.8
73  → rmax = r0*2.3
74
75  → print 'The first ', nwrite, ' energy levels for the relevant molecule'
76
77  → if nx % 2 != 0:
78  →  → print "nx isn't even"
79  →  → return
80  → print 'Number of grid points in 1D grid = ', nx
```

```
81
82  → nhalf = nx/2
83  → nham1 = nhalf - 1
84  → print 'nhalf:',nhalf,'  nahm1:',nham1
85
86 #→ Compute the coordinate grid spacing dz
87  → zl = rmax - rmin
88  → print 'Grid lenght = ', zl
89  → dx = zl / nx
90  → print 'Grid spacings = ', dx
91 #→ Compute reduced mass zmu
92  → print 'Reduced mass of the given molcule = ', zmu
93
94 #→ Now set up the hamiltonian matrix ar(i,j)
95  → darg = 2.0 * pi/nx
96  → targ = 4.0 * ((pi/zl)**2) / (zmu*nx)
97
98  → x = rmin # copy
99  → fv1 = zeros(nx,dtype=float)
100 → fv2 = zeros(nx,dtype=float)
101 → xa  = zeros(nx,dtype=float)
102
103 → for i in range(0,nx):
104 → → fv1[i] = targ * ((i+1)**2)
105 → → fv2[i] = cos(darg*(i+1))
106
107 #→ Initialize variables
108 #→ const - ((-1)**(i-j))*T(nx/2)
109 #→ inij  - (i-j)
110 #→ sum   - total summation of equation within sum
111
112 → const = 0.0
113 → equij = 0.0
114
115 → for i in range(0,nham1):
116 → → equij = equij + fv1[i]
117
118 → for i in range(0,nx):
119 → → xa[i]=x
120 → → for j in range(0,i+1):
121 → → → ij = i - j
122 → → → inij = ij
123 → → → summa = 0.0
124 → → → if ij == 0:
125 → → → → summa = equij
126 → → → else:
127 → → → → for l in range(0,nham1):
128 → → → → → summa = summa + fv1[l]*fv2[ij-1]
129 → → → → → ij = ij + inij
130 → → → → → # cosine is periodic, so only need value within one cycle/
    period
```

```
131  → → → → → if ij > nx:
132  → → → → → → ij = ij % nx
133
134  → → → # Add in T(nx/2)
135  → → → const = ((-1)**inij)*fv1[nhalf]/2
136  → → → ar[i,j] = summa + const
137  → → # Add the potential value when kronicker delta function equals 1, i.e.
         when i and j are equal
138  → → ar[i,i] = ar[i,i] + the_potential(x)
139  → → x = x + dx
140  → #Filling out the Hamiltonian matrix
141  → for i in range(0,nx):
142  → → for j in range(0,i):
143  → → → ar[j,i] = ar[i,j]
144
145  → # now call eigenvalue solver
146  → wch, zr = linalg.eigh(ar)
147
148  → return wch, zr, xa
149
150
151 def wave(the_potential, zmu, r0,nx,rmin_factor=0.8,rmax_factor=2.0):
152  → """fgh - Fourier Grid Hamiltonian - in wavenumbers and angstrom
153  → the_potential is the potential to be used
154  → mju is the reduced mass
155  → r0 is the equilibrium bond length
156  → returns wch,zr and xa
157  → """
158
159  → n = (nx / 2)
160  → h_bar = 1.05459e-34 # in Joule*seconds
161
162  → H = zeros((nx,nx),dtype=float) # The Hamiltonian matrix
163  → T_l = zeros(n,dtype=float)
164  → rmin = r0*rmin_factor
165  → rmax = r0*rmax_factor
166
167  → zl = rmax - rmin # in angstrom
168  → dx = zl / nx
169  → zmu = zmu/(1000*6.022e23) # in kilograms per single atom
170  → dk = (2 * math.pi) / zl; # in angstrom
171  → dk = dk * 1e10 # in meter
172  → T_l = (arange(1,n+1)*dk)**2 * ((h_bar*h_bar)/(2*zmu)) # in Joule
173  → T_l = T_l*5.035e22 # in cm^-1
174  → x = dx*arange(nx) + rmin # in angstrom
175
176  → V = the_potential(x) # in cm^-1
177  → from scipy import weave
178
179  → for i in range(1,nx+1): # note that for memory storage the indices are 1
       lower than they should be according to the mathematical equations
```

```
180  →  →  for j in range(1,nx+1):
181  →  →  →  H[i-1,j-1] = sum((2*cos(arange(1,n)*2*math.pi * (i-j)/nx)) * T_l
          [0:(n-1)])
182  →  →  →  H[i-1,j-1] = (H[i-1,j-1] + (-1)**(i-j)*T_l[n-1])/nx
183  →  →  →  if (i == j):
184  →  →  →  →  H[i-1,j-1] = H[i-1,j-1] + V[i-1]
185
186  →  # now call eigenvalue solver
187  →  eig_vals, eig_vecs = linalg.eigh(H)
188
189  →  return eig_vals, eig_vecs, x
190
191  def wave_direct(V, zmu, r):
192  →  """fgh - Fourier Grid Hamiltonian - in wavenumbers and angstrom
193  →  the_potential is the potential to be used
194  →  mju is the reduced mass
195  →  r0 is the equilibrium bond length
196  →  returns wch,zr and xa
197  →  """
198  →  if (len(r) % 2):
199  →  →  print '******'
200  →  →  print '**  nx is not even'
201  →  →  print '******'
202  →  →  return
203  →  n = (len(r) / 2)
204  →  h_bar = 1.05459e-34 # in Joule*seconds
205  →  nx = len(r)
206  →  H = zeros((nx,nx),dtype=float) # The Hamiltonian matrix
207  →  T_l = zeros(n,dtype=float)
208
209  →  rmin = r[0]
210  →  rmax = r[-1] # the last point in r
211
212  →  zl = rmax - rmin # in angstrom
213  →  #dx = zl / nx
214  →  dx = r[0] - r[1]
215  →  zmu = zmu/(1000*6.022e23) # in kilogramms per single atom
216  →  dk = (2 * math.pi) / zl; # in angstrom
217  →  dk = dk * 1e10 # in meter
218  →  T_l = (arange(1,n+1)*dk)**2 * ((h_bar*h_bar)/(2*zmu)) # in Joule
219  →  T_l = T_l*5.035e22 # in cm^-1
220
221  →  #V = the_potential(x) # in cm^-1
222
223
224  →  for i in range(1,nx+1): # note that for memory storage the indices are 1
          lower than they should be according to the mathematical equations
225  →  →  for j in range(1,nx+1):
226  →  →  →  H[i-1,j-1] = sum((2*cos(arange(1,n)*2*math.pi * (i-j)/nx)) * T_l
          [0:(n-1)])
227  →  →  →  H[i-1,j-1] = (H[i-1,j-1] + (-1)**(i-j)*T_l[n-1])/nx
```

```
228  →  →  → if (i == j):
229  →  →  →  → H[i-1,j-1] = H[i-1,j-1] + V[i-1]
230
231  → # now call eigenvalue solver
232  → eig_vals, eig_vecs = linalg.eigh(H)
233
234  → return eig_vals, eig_vecs
```

A.3.3 fitter.py

```
1  from scipy import *
2  from scipy.optimize import leastsq
3
4  def morse_fitter(r,U,guess):
5  → """docstring for morse_fitter"""
6  → def morse_residuals(p,r,U):
7  → → """docstring for morse_residuals"""
8  → → return U - morse(r,p)
9
10 → def morse(r,p):
11 → → """p[0] = T_e
12 → → p[1] = D_e
13 → → p[2] = beta
14 → → p[3] = r_e"""
15 → → return p[0] + p[1] * (exp(-p[2]*(r-p[3])) - 1.0)**2.0
16
17 → final = leastsq(morse_residuals,guess,args=(r,U),full_output=1)
18
19 → return final, lambda r: morse(r,final[0])
20
21
22 def small_fitter(fitter, threshold, r, U, guess):
23 → """fitter is the fitter that is being used
24 → threshold is the energy threshold for fittings, e.g. 5000 cm^-1 for
      ground states
25 → r are the bond lengths
26 → U are the energies
27 → guess is the fitting guess"""
28
29 → bonds = []
30 → energies = []
31 → the_corrected_threshold = min(U) + threshold
32
33 → for i in range(len(r)):
34 → → if U[i] <=the_corrected_threshold:
35 → → → energies.append(U[i])
36 → → → bonds.append(r[i])
37
38 → bonds = array(bonds)
39 → energies = array(energies)
40
41 → return fitter(bonds,energies,guess)
```

A.3.4 potent.py

```
from numpy import *
from scipy import interpolate
import math
import potent

def main():
 → r,gs,ex = potent.aug_cc_pvqz()

if __name__ == "__main__":
 → main()

def make_spline(r,U):
 → """docstring for make_spline"""
 → return interpolate.UnivariateSpline(r,U)

def make_morse(T_e,D_e,beta,r_e):
 → """docstring for make_morse"""
 → return lambda r: T_e + D_e * (exp(-beta*(r-r_e)) - 1.0)**2.0

def loader(the_file):
 → f = open(the_file,'r')
 → lines = f.readlines()
 → f.close()

 → state_names = lines[0].split()

 → n_stuff = len(lines[1].split())

 → a = array([])

 → for i in lines:
 → → a = concatenate((a,array(i.split())))

 → b = a.reshape(len(a)/n_stuff,n_stuff)[1:,:]
 → b = b.astype(float)


 → r = b[:,0]
 → gs = b[:,1]
 → ex = b[:,2:]

 → return r, gs, ex, state_names
```

A.3.5 spec.py

```
from numpy import *
from scipy import linalg
import math

def find_wexe(we):
```

```
6  → """find_wexe - fits vibrational quantum number vs. we
7  → returns the we (the factor at x**2)"""
8
9  → v = arange(0.5,len(we)+0.5,dtype=float64) # the (shifted) vibrational
       quantum number, 0.5, 1.5, ...
10 → # we are fitting to e_total = e_{J,v} = BJ(J+1) + (v+1/2)we - wexe(v+1/2)
       ^2 + (T_e?)
11
12 → fits = polyfit(v,we,2) # fit the data to: -(v+1/2)^2wexe + (v+1/2)we
13 → return -1.0 * fits[0]
14
15
16 def find_r_v(r,psi):
17 → """find_r_v, finds the weighted average of the wavefunction, which is in
       this case the vibrational wavefunction
18 → so we find the vibrational bond length of this wavefunction
19 → r are the r_i values where the wavefunction is know at, in angstrom
20 → psi is the vibrational wavefunction, not squared"""
21 → psi_sq = psi**2
22 → return sum(dot(r,psi_sq))/sum(psi_sq)
23 → #return sum(map(lambda x,y:x*y,r,psi**2))/sum(psi**2)
24
25
26 def find_B_r(r,mju,psi_many,first):
27 → """find_B_r is function that returns both r_{v=0} and B_{v=0} for the
       first values that are asked for
28 → r are the r_i values where the wavefunction is known at, in angstrom
29 → mju is the weighted mass of the diatomic molecule that is being
       calculated, in g/mol
30 → psi_many is an array that contains the wavefunctions of the first
       vibrational levels, it is not the squared wavefunction
31 → first is how many vibrational levels should be looked at"""
32
33
34 → c = 29979200000 # The speed of light in cm/s
35 → N_A = 6.0221e23# Avogadros number in 1/mol
36 → h_bar = 6.6262E-27 # Planck's constant in erg?/s
37
38 → r_v = zeros(first,dtype=float)
39 → B_v = zeros(first,dtype=float)
40 → B_factor = (h_bar*N_A*1.0e16)/(c*8.0*math.pi*math.pi*mju)
41
42 → for i in range(0,first):
43 → → r_v[i] = find_r_v(r,psi_many[:,i]) # get each vibrational bond length
44 → → B_v[i] = B_factor/(r_v[i]**2) # The rotational constant of each bond
       length
45
46 → return r_v, B_v
47
48 def constants(r,mju,psi_many,vib_levels,energ_vib):
49 → """docstring for constants"""
```

```
50  → r_v, B_v = find_B_r(r,mju,psi_many,vib_levels)
51  → wexe = find_wexe(energ_vib[range(vib_levels)])
52  → we = energ_vib[1] - energ_vib[0]
53  → return r_v, B_v, we, wexe
```

A.3.6 Setup file

This is a setup file from the AIREMPICalc program

```
[potentials]
number: 2
pot_file: aug-cc-pvqz_cm-1.txt,aug-cc-pv5z_cm-1.txt
fit: True
interpolate: True
direct: false
threshold: true
do_rempi: False

[fit_setup]
fit_type: Morse
fit_guess: 0.0, 40000.0, 2.0, 1.27

[molecule]
name: HCl
reduced_mass: 0.9795925068
# not needed when doing direct evaluations
r0: 1.27

[ground_state]
name: x1sigma+
threshold: 5000.0
symmetry: Sigma

[excited_states]
no_ex: 2
name: c1pi,f1delta
#remember that 0 is the first position, not zero!
pos: 21,10
threshold:9000.0
#Available states: sigma,pi,delta,phi
symmetry: Pi,Delta

[rempi]
n+1:2

[fgh]
# needs to be a multiple of 2
points: 64
# This is how many vibrational levels should be used for the
# spectroscopic constant calculations
vib_levels: 10
```

A.3.7 Demo output

This is a demo of the AIREMPICalc output, when run on the $X^1\Sigma^+$ CCSD/aug-cc-pVTZ ground state and $C^1\Pi$ and $F^1\Delta_2$ CR-EOMCCSD(T)/aug-cc-pVTZ excited states

```
Starting calculations
Running on potential file:  aug-cc-pvtz_cm-1.txt
  Fitting with a morse potential
  GS name:  x1sigma+ , in file:  cr_tz_gs
    Fitted parameters:  [ -4.05921672e+00   4.02899876e+04   1.82575121e+00   1.27292748e+00]
  EX name:  c1pi , in file:  cr_tz_b1_2
    Fitted parameters:  [  7.78895130e+04   3.41943617e+04   1.88918570e+00   1.34937443e+00]
  EX name:  f1delta , in file:  cr_tz_a2_1
    Fitted parameters:  [  8.55324838e+04   3.36932736e+04   1.81798592e+00   1.30852041e+00]
  GS name:  x1sigma+ , in file:  cr_tz_gs
   r_v:  [ 1.28870084  1.32155732  1.35708355  1.39642722  1.44037263  1.48889899
  1.54138754  1.597142    1.65582072  1.71683174]
   B_v:  [ 10.36222588   9.85338084   9.34424257   8.82512093   8.29483121
   7.76295035   7.24325186   6.74637091   6.27668959   5.83850705]
   wes:  2927.74577643
   wexes:  52.0281934577
   T_v:  [  1502.21056024   4429.95633667   7247.84630138   9961.24894645
  12575.11228624  15090.98669771  17506.00779804  19814.23628419
  22008.60641956  24083.73613469]
  EX name:  c1pi , in file:  cr_tz_b1_2
   r_v:  [ 1.3664639   1.40187247  1.439201    1.47865356  1.52047116  1.56494067
  1.6124116   1.66323318  1.71858021  1.77446232]
   B_v:  [ 9.21639221  8.75669575  8.30834211  7.87090027  7.44390643  7.02686358
  6.61919916  6.22086813  5.82663298  5.46542215]
   wes:  2775.79816287
   wexes:  61.9010319692
   T_v:  [ 79323.48707993  82099.28524281  84752.19513074  87282.146135
  89688.97015949  91972.34098527  94131.70244974  96166.36481172
  98073.75801564  99862.74480656]
  EX name:  f1delta , in file:  cr_tz_a2_1
   r_v:  [ 1.32573771  1.36143247  1.39926036  1.43979952  1.48379098  1.53194708
  1.58475959  1.64232895  1.70563611  1.76866766]
   B_v:  [ 9.79133833  9.28463994  8.78941997  8.3014361   7.81649086  7.33279853
  6.8522074   6.38023947  5.91540538  5.50129338]
   wes:  2655.30501396
   wexes:  55.5623198129
   T_v:  [  86902.73510367   89558.04011763   92100.14742691   94529.90346402
   96848.44622233   99056.73002133  101154.90660572  103142.14791332
  105014.60278505  106781.21361102]
```

A.4 Jötunn utilities

The two parsing scripts of sections A.4.2 and A.4.3 both assume that the output files come from NWChem and that the files is `prefix.bond.o11111`, where `o11111` is the job number on Jötunn and bond is the bond length of the diatomic molecule.

A.4.1 NWChem file

This is a NWChem job file which uses the aug-cc-pVTZ basis to calculate nine excited states, in the a1, a2, b1 and b2 symmetries of the C_{2v} group. It uses a CCSD ground state calculation and the performs the EOM-CCSD calculations. After it has converged to a solution it starts the CR-EOM-CCSD(T) calculations to find the δ correction.

```
start job_sing_AVTZ_1.270

memory global 3700 mb heap 150 mb stack 200 mb

scratch_dir /scratch/erlendj
permanent_dir /scratch/erlendj


geometry
symmetry c2v
H 0.0 0.0 0.0
Cl 0.0 0.0 1.270
end

echo

basis spherical
H library aug-cc-pvtz
Cl library aug-cc-pvtz
end

title "HCl CR-EOM-CCSD(T) - singlet"

scf
rohf
end

tce
io ga
tilesize 20
CREOMSD(T)
nroots 9
maxiter 50
print high
end

task tce energy
```

A.4.2 Parsing script - EOMCCSD calculations

This files outputs a file with the following column structure
Bond length [Å] Ground state energy [hartree] Excited states [eV], all symmetries, . . .

```
1 files=$(ls HCl_sin.AVDZ_RO.*)
```

```
2
3  for i in $files; do
4  → echo $i
5  → echo $i | gnused -e 's/HCl_sin.AVDZ_RO.//g' -e 's/.o[0-9]....//g' | tr '\
     n' '\t' >> output_file
6  → grep 'CCSD total energy' $i |  gnused -e 's/-/\n-/g' | grep [0-9] | tr '\
     n' '\t'  >> output_file
7  → # The following line assumes that nine excited states were calculated, i.
     e. nroots 9
8  → cat $i | grep Iterations -B 11 | grep vectors -A 9 | grep [0-9] | grep -v
      vectors | gnused -e 's/./\n gnu/48' | grep -v gnu | gnused -e 's/./gnu\n
     /39' | grep -v gnu | tr '\n' '\t'>> output_file
9  → echo '' >> output_file
10 done
```

A.4.3 Parsing script - CR-EOMCCSD(T) calculations

This files outputs a file with the following column structure

Bond length [Å] Ground state energy [hartree] Excited states [hartree], all symmetries, ...

```
1  files=$(ls HCl_sin.AVDZ_RO.*)
2
3  for i in $files; do
4  → echo $i
5  → echo $i | gnused -e 's/HCl_sin.AVDZ_RO.//g' -e 's/.o[0-9]....//g' | tr '\
     n' '\t' >> output_file
6  → grep 'CCSD total energy' $i |  gnused -e 's/-/\n-/g' | grep [0-9] | tr '\
     n' '\t'  >> output_file
7  → # The following line assumes that nine excited states were calculated, i.
     e. nroots 9
8  → cat $i | grep Iterations -B 11 | grep vectors -A 9 | grep [0-9] | grep -v
      vectors | gnused -e 's/./\n gnu/48' | grep -v gnu | gnused -e 's/./gnu\n
     /39' | grep -v gnu | tr '\n' '\t'>> output_file
9  → echo '' >> output_file
10 done
```

A.5 REMPIControl - Source code

The following sections show the source code of the REMPIControl program. Since LabVIEW uses the G graphical programming language, all of the source code is pictures.

A.5.1 `oscilloscope_setup`

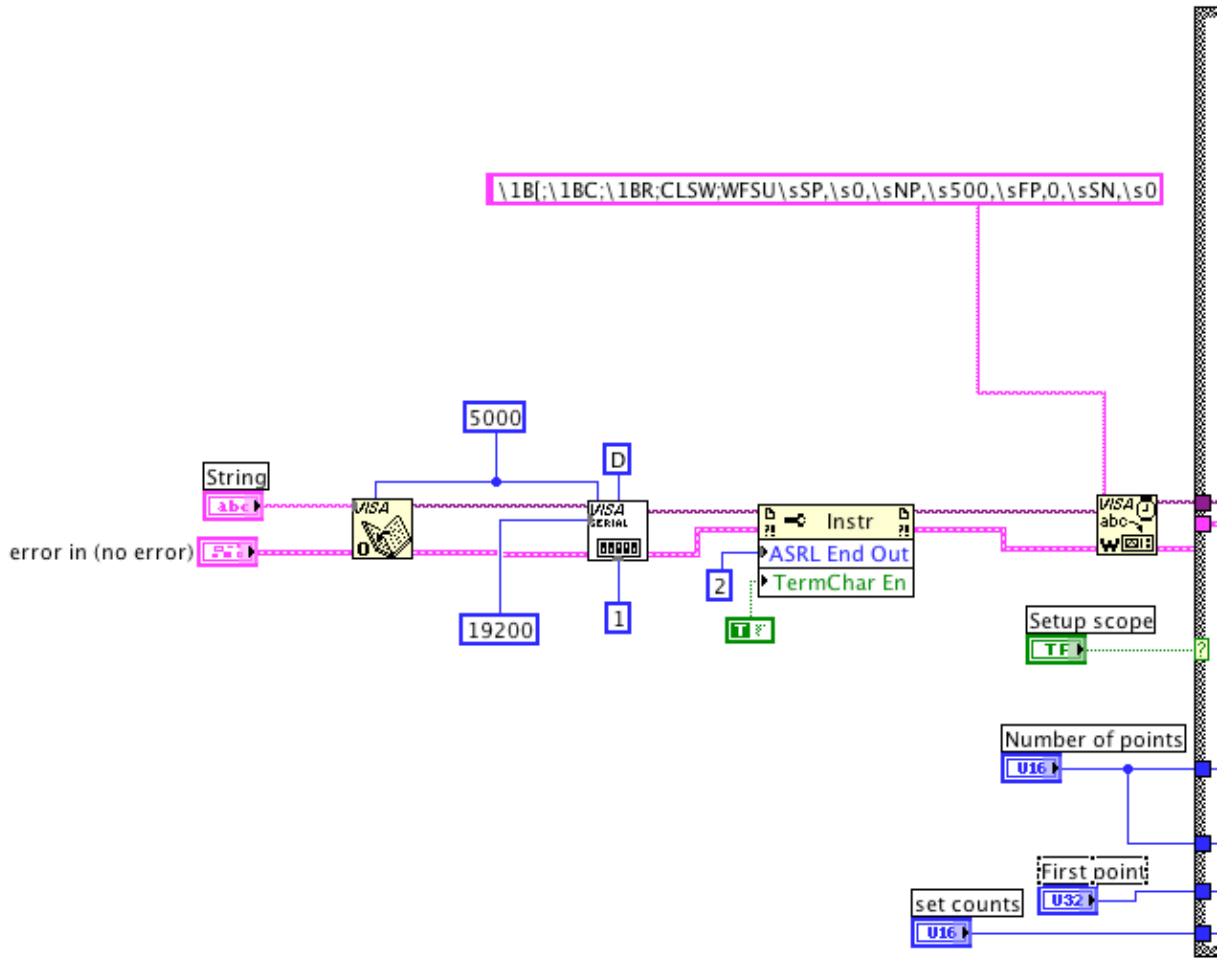


Figure 9: `oscilloscope_setup` - Start

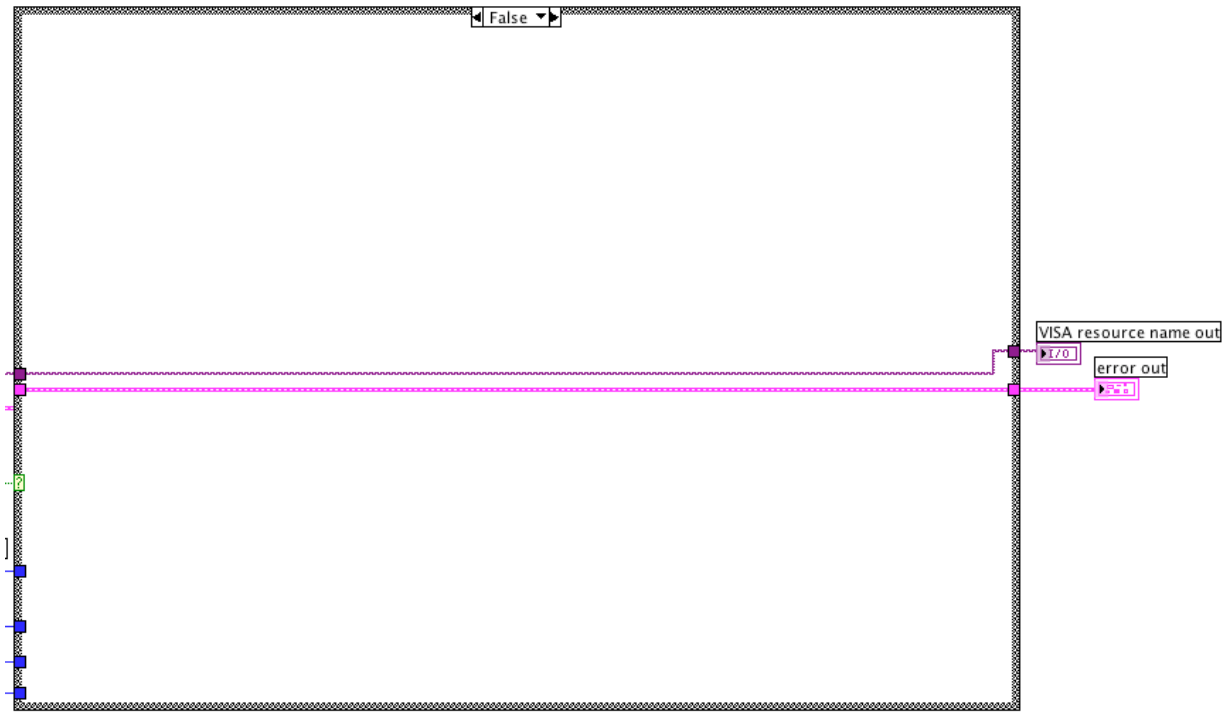


Figure 10: `oscilloscope_setup` - If false

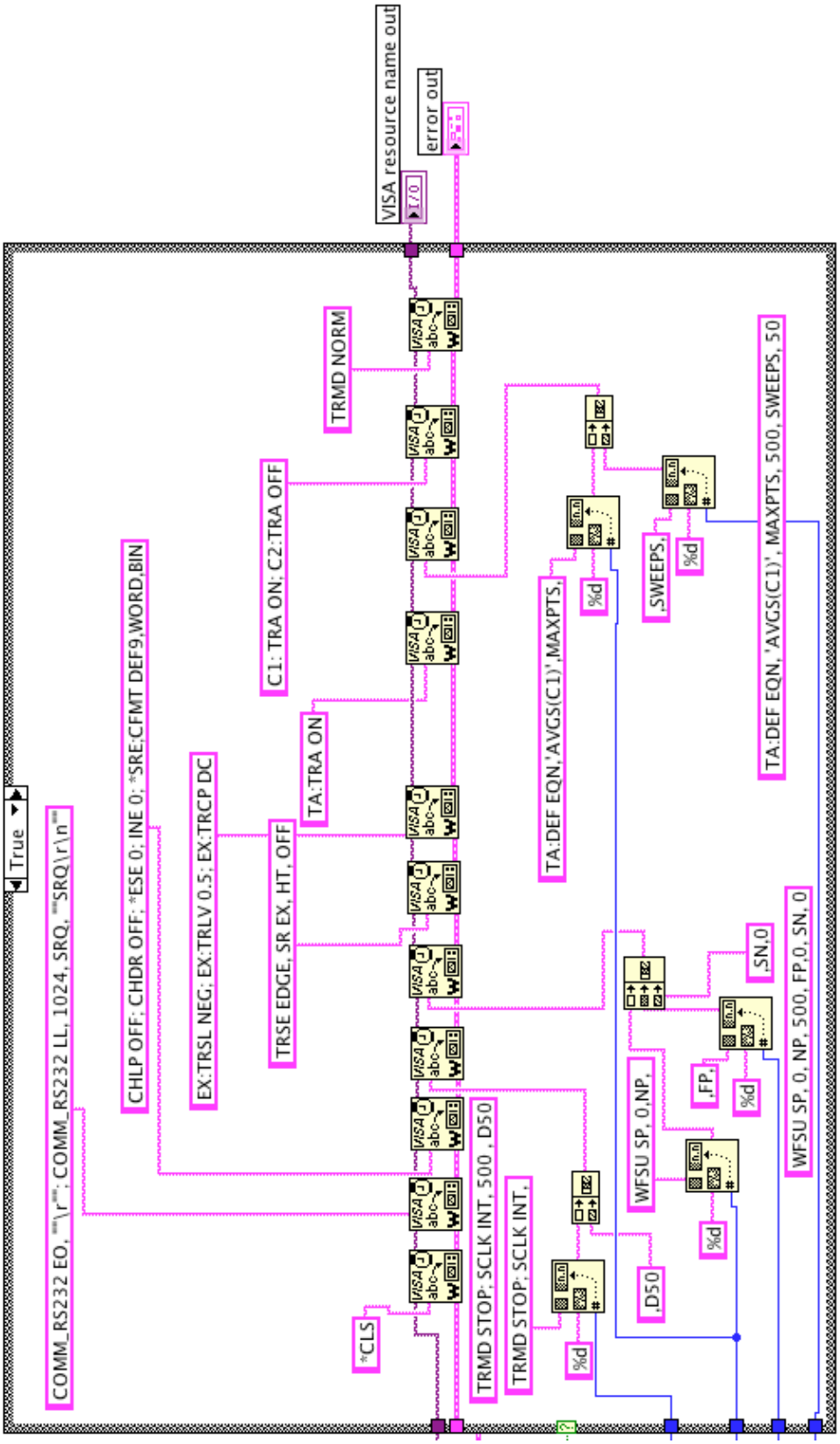


Figure 11: oscilloscope_setup - If true

A.5.2 oscilloscope_read

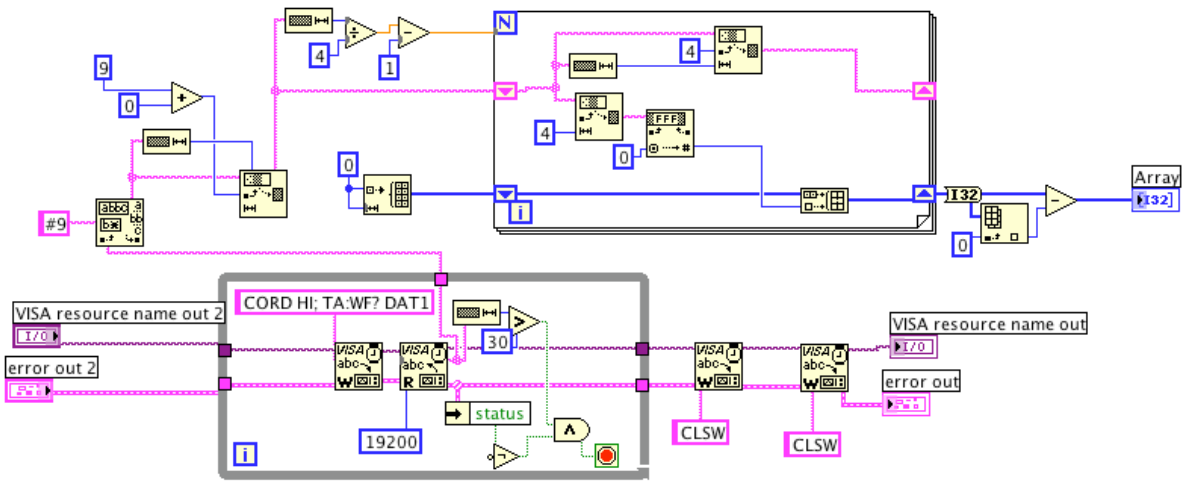


Figure 12: oscilloscope_read

A.5.3 `setup_record`

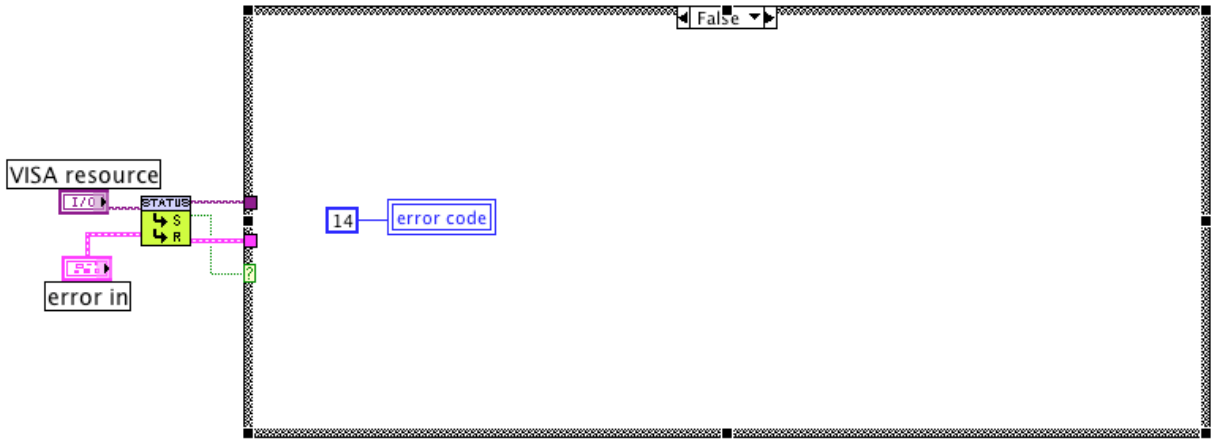


Figure 13: `setup_record` - True

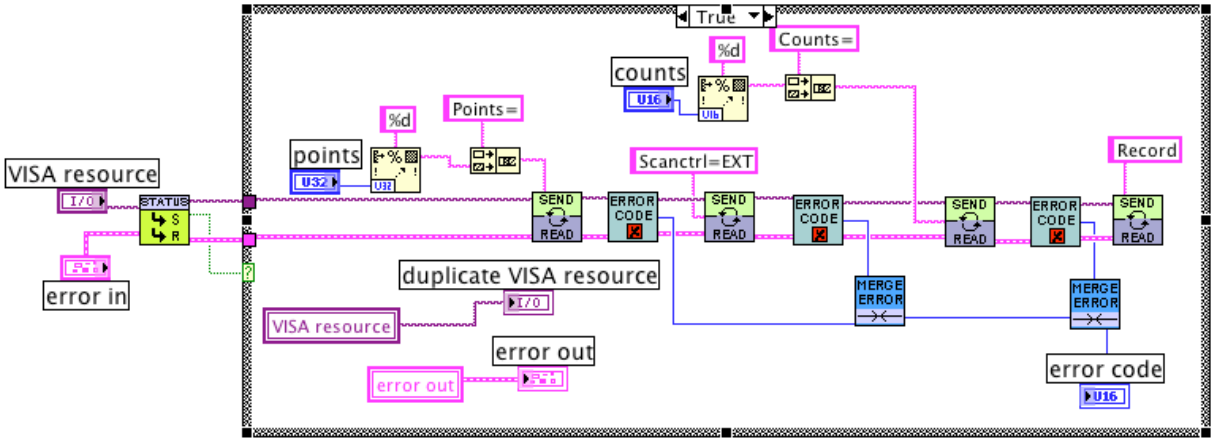


Figure 14: `setup_record` - False

A.5.4 REMPI_Record

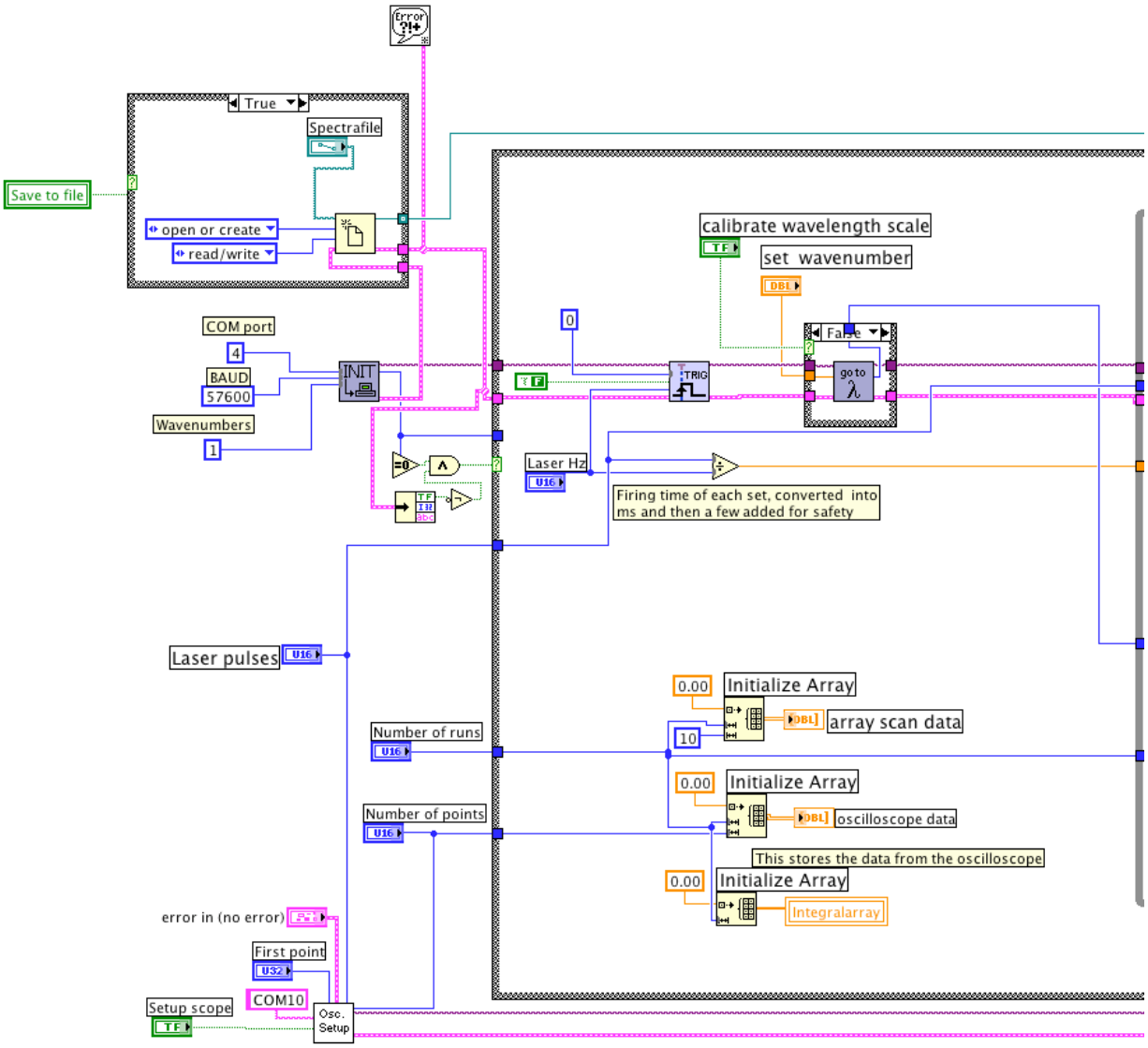


Figure 15: REMPI_record - Beginning

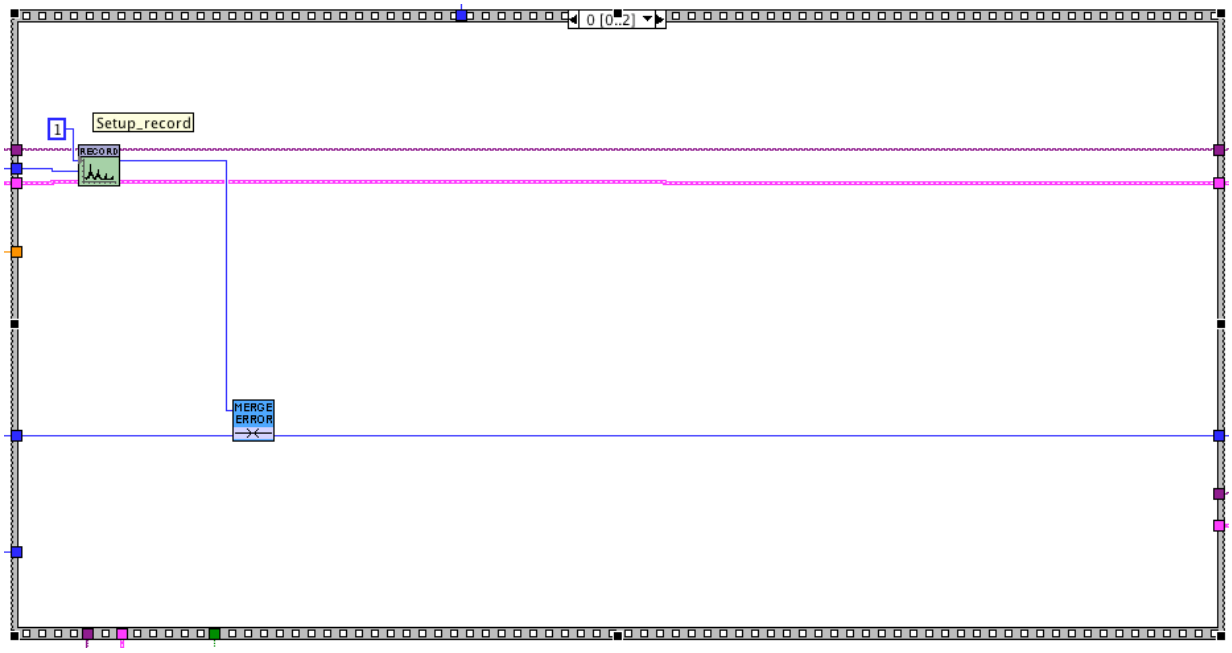


Figure 16: `REMPI_record` - Inner stack - level 0

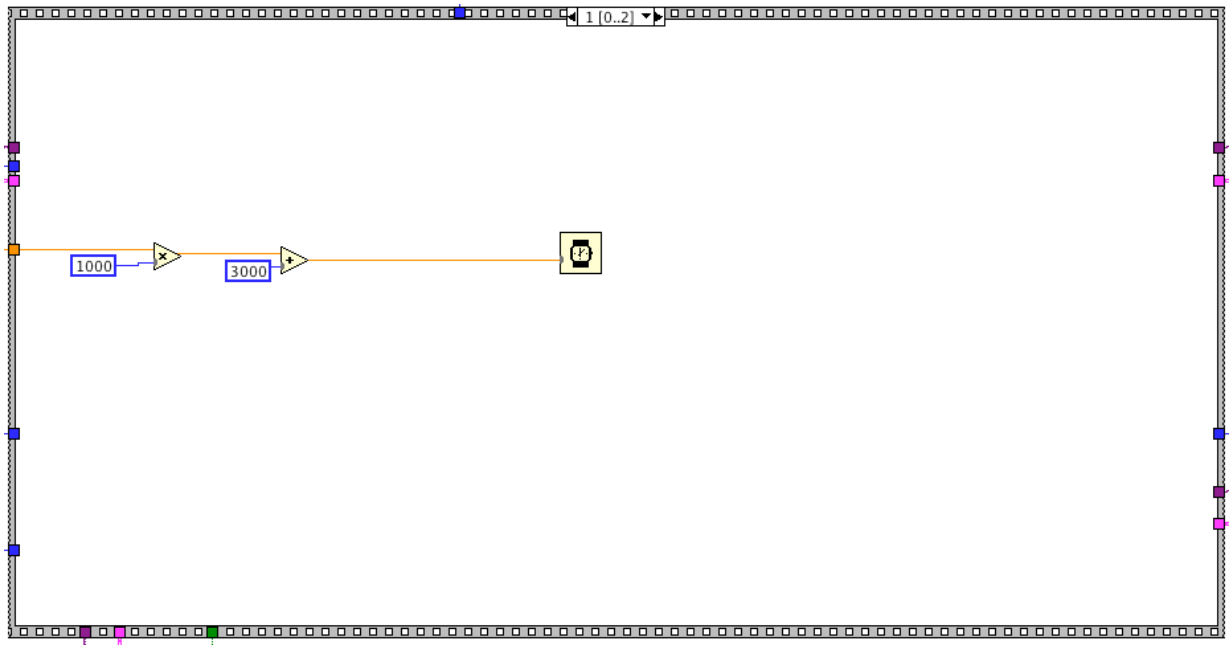


Figure 17: `REMPI_record` - Inner stack - level 1

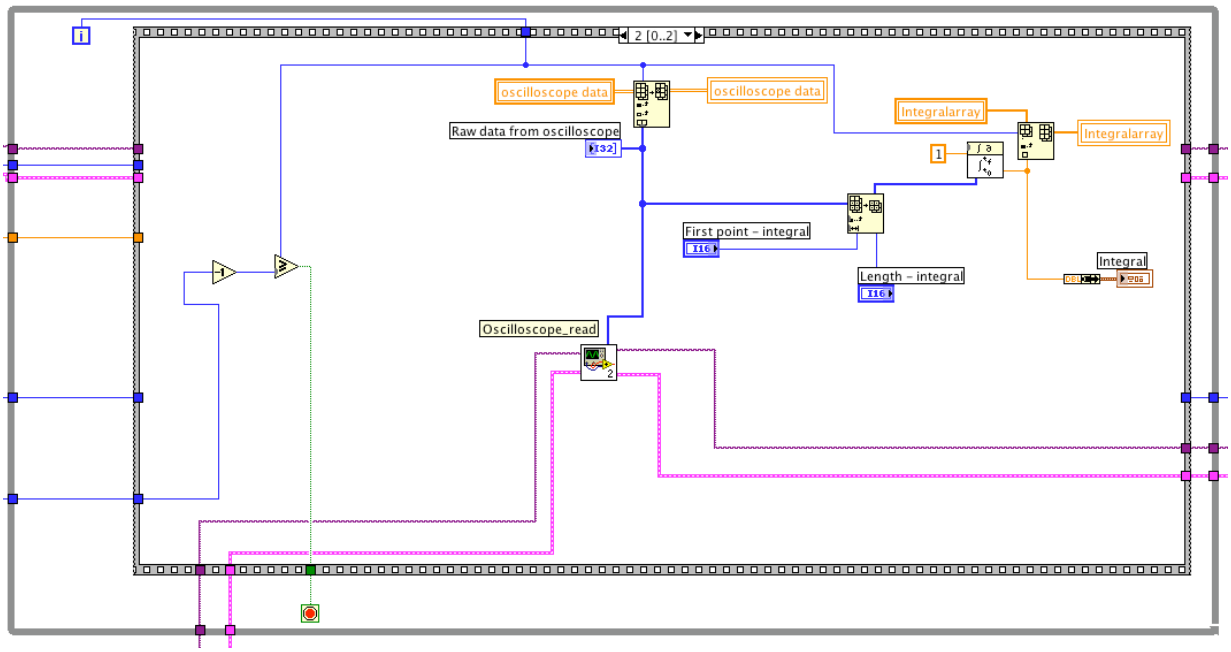


Figure 18: `REMPI_record` - Inner stack - level 2

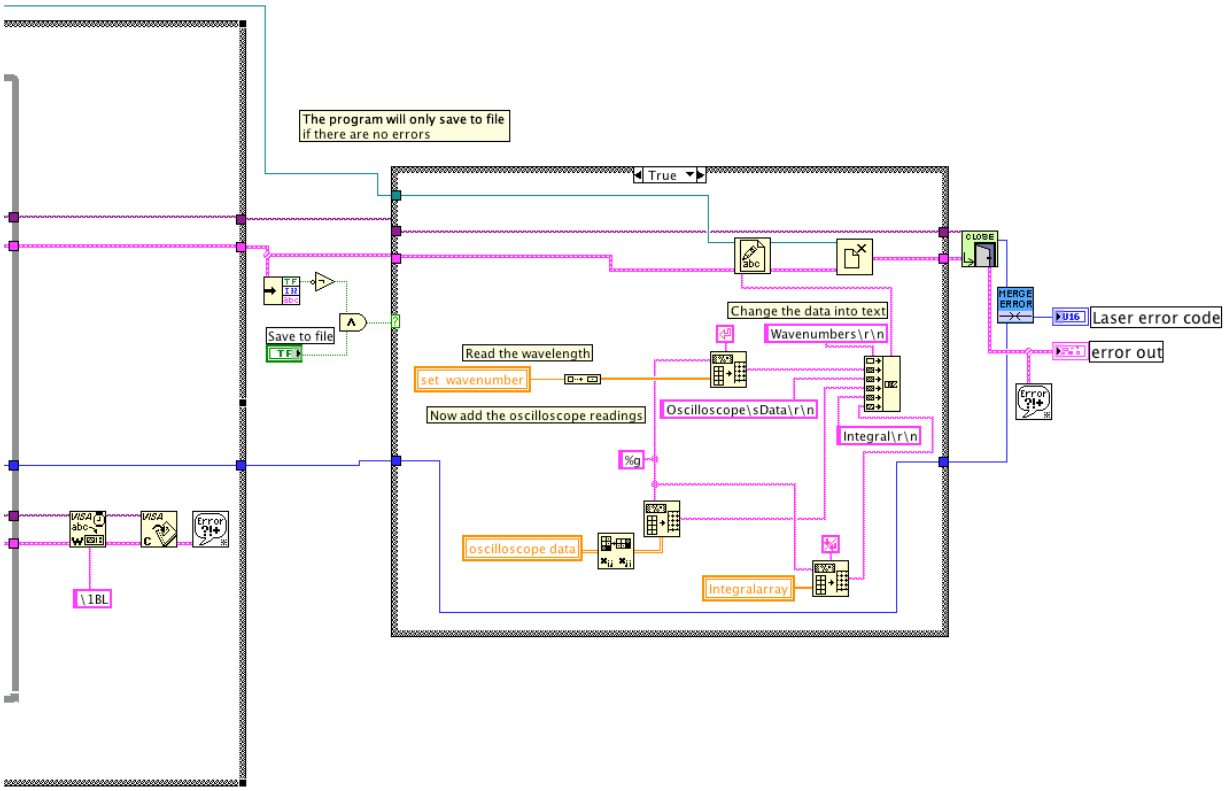


Figure 19: `REMPI_record` - End

A.5.5 REMPI3

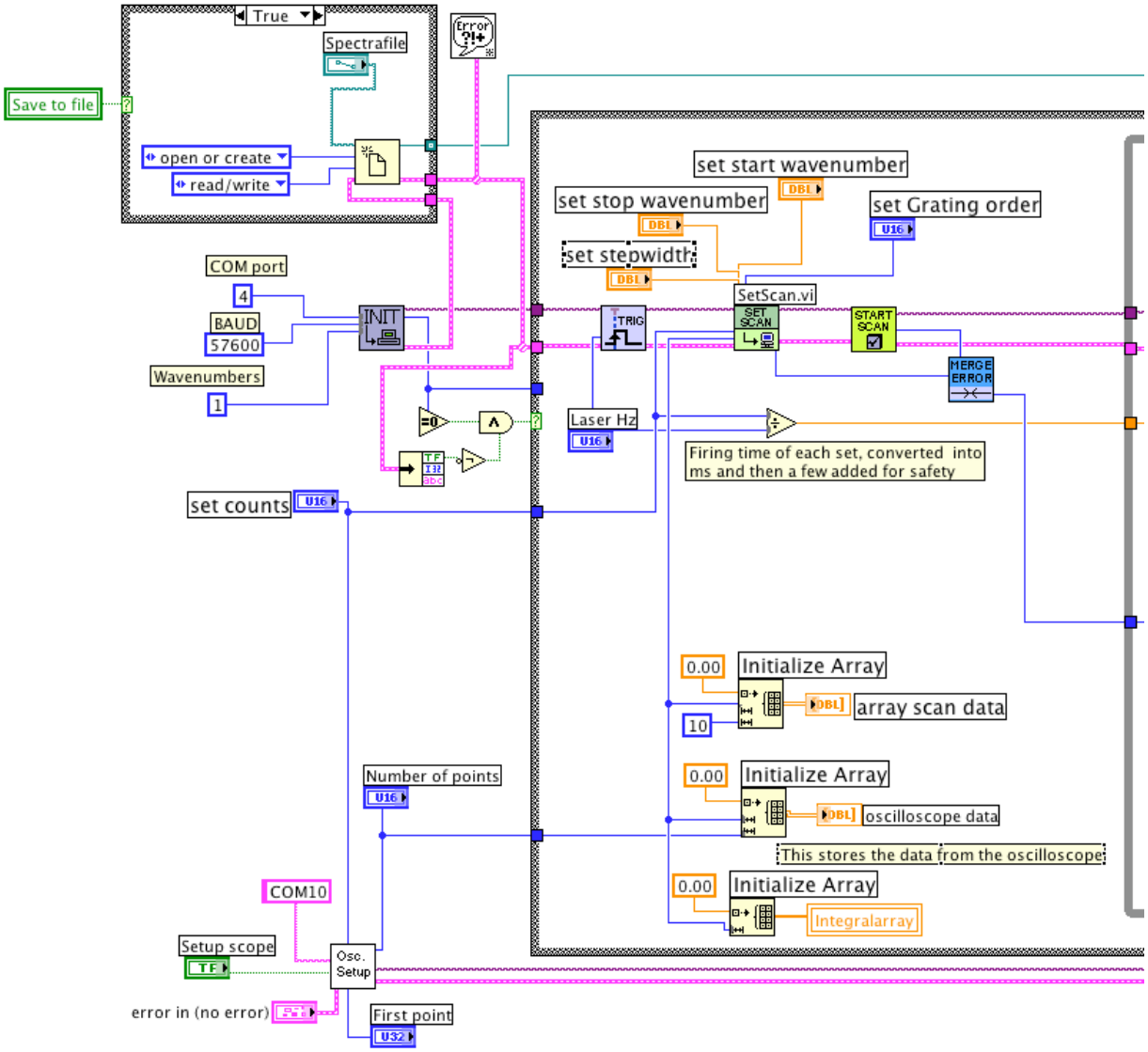


Figure 20: REMPI3 - Beginning

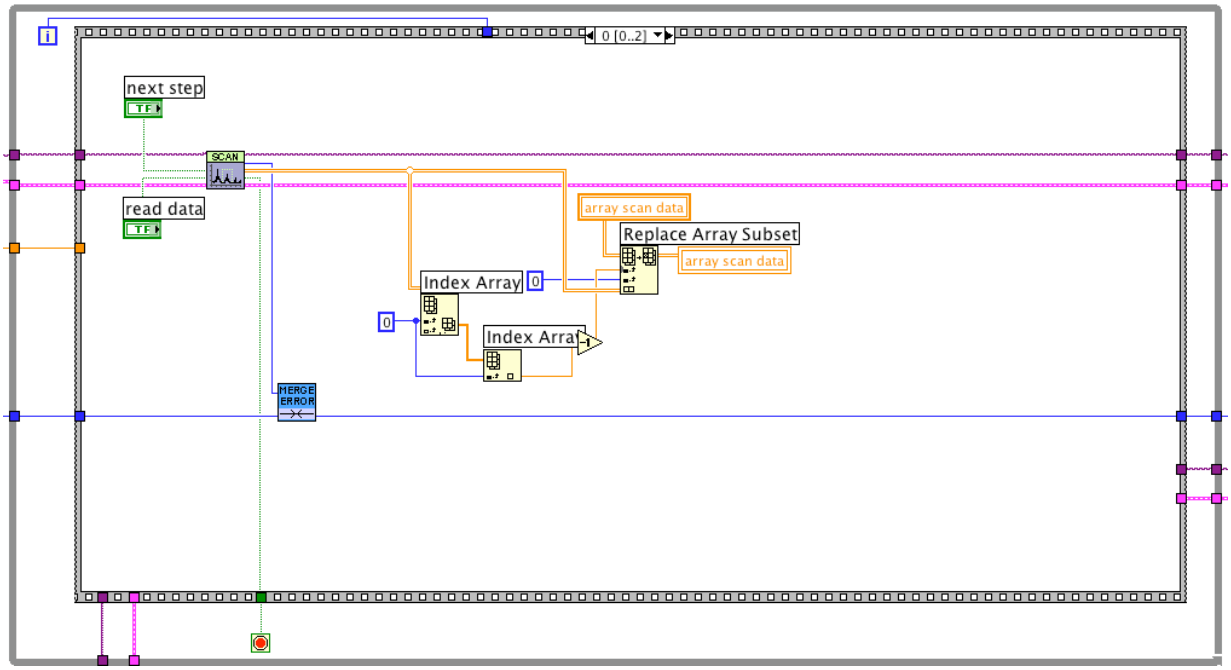


Figure 21: REMPI3 - Inner stack - level 0

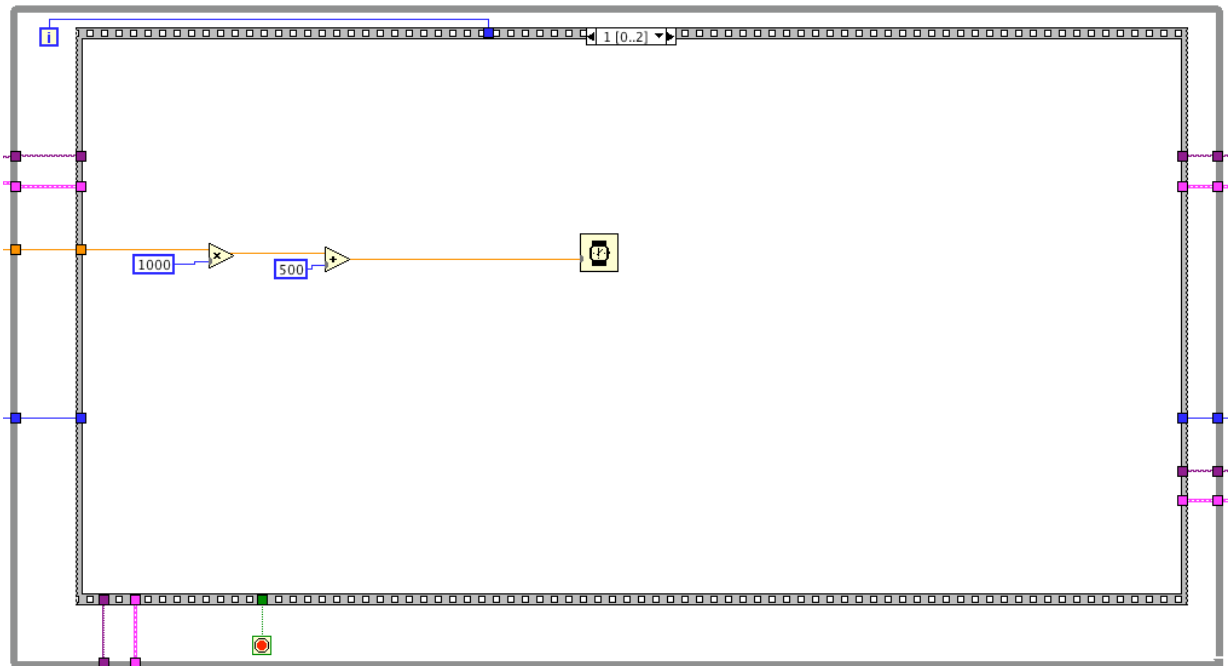


Figure 22: REMPI3 - Inner stack - level 1

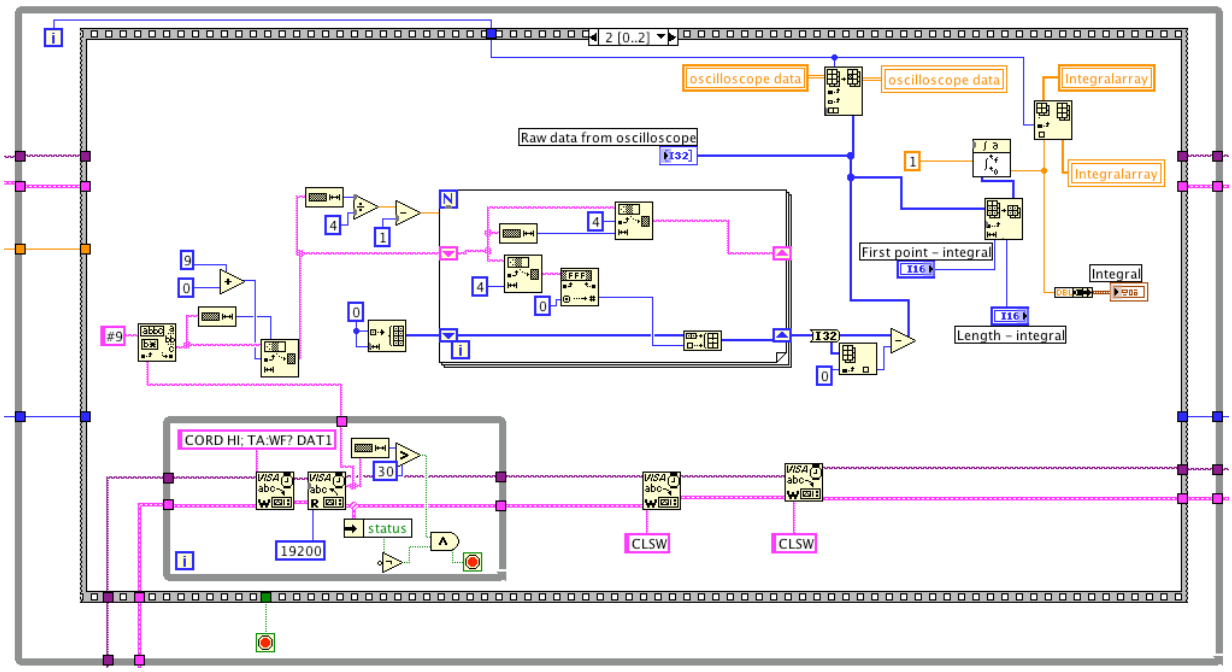


Figure 23: REMPI3 - Inner stack - level 2

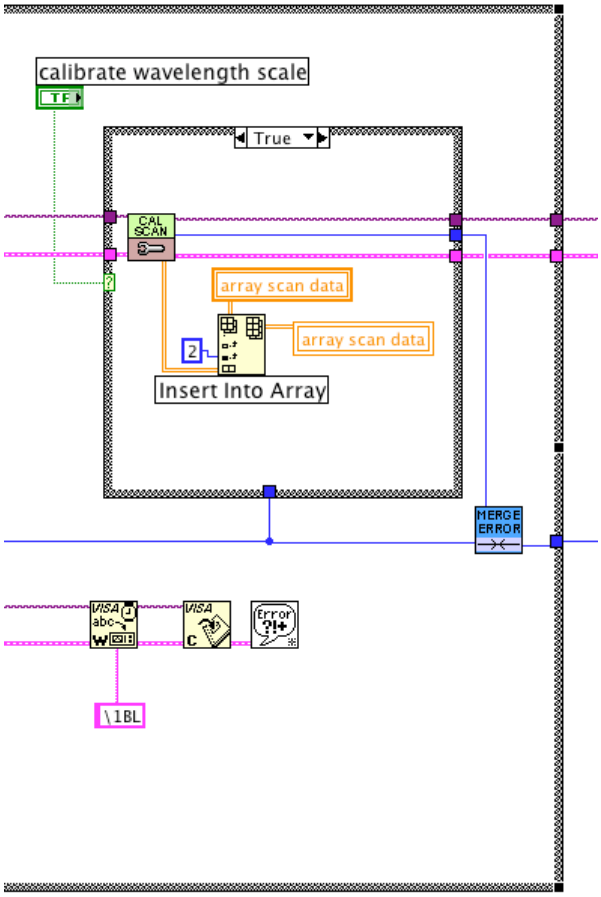


Figure 24: REMPI3 - Wavelength calibration

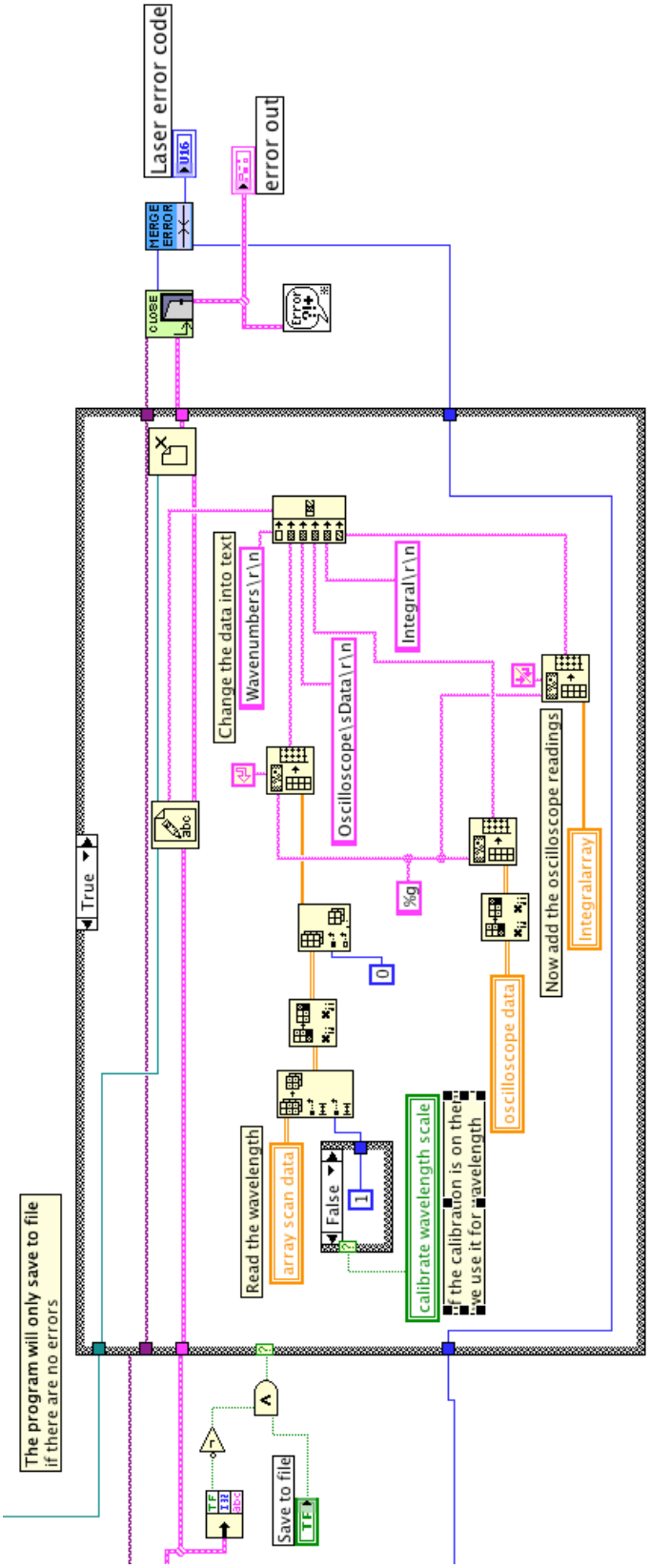


Figure 25: REMPI3 - End

References

[1] Dirac, P. A. M. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **1929**, *123*(792), 714–733.

[2] Kvaran, Á.; Wang, H.; Matthíasson, K.; Bodi, A.; Jónsson, E. *The Journal of Chemical Physics* **2008**, *129*(17), ?

[3] Piecuch, P.; Kucharski, S.; Špirko, V.; Kowalski, K. *The Journal of Chemical Physics* **2001**, *115*, 5796.

[4] Green, D.; Bickel, G.; Wallace, S. *Journal of Molecular Spectroscopy* **1991**, *150*(2), 303–353.

[5] Green, D.; Bickel, G.; Wallace, S. *Journal of Molecular Spectroscopy* **1991**, *150*(2), 354–387.

[6] Green, D.; Bickel, G.; Wallace, S. *Journal of Molecular Spectroscopy* **1991**, *150*(2), 388–469.

[7] Hirst, D.; Guest, M. *Molecular Physics* **1980**, *41*(6), 1483–1491.

[8] Bettendorff, M.; Peyerimhoff, S. D.; Buenker, R. J. *Chemical Physics* **1982**, *66*(3), 261–279.

[9] Bettendorff, M.; Buenker, R.; Peyerimhoff, S.; Römelt, J. *Zeitschrift für Physik A Hadrons and Nuclei* **1982**, *304*(2), 125–135.

[10] Pitarch-Ruiz, J.; Meras, A.; Sanchez-Marín, J.; Velasco, A.; Lavín, C.; Martín, I. *J. Phys. Chem. A* **2008**, *112*(14), 3275–3280.

[11] Dunning Jr, T. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* **2002**, *360*(1795), 1079–1105.

[12] Yasuda, K. *Journal of Computational Chemistry* **2008**, *29*(3), 334–342.

[13] Yasuda, K. *Journal of Chemical Theory and Computation* **2008**, *4*(8), 1230–1236.

[14] Matthíasson, K.; Wang, H.; Kvaran, Á. June **2008**, *458*(1–3), 58–63.

[15] Lee, M.; Wang, K.; McKoy, V.; Machado, L. *Journal of Chemical Physics* **1992**, *97*(6).

[16] Vrakking, M.; Lee, Y.; Gilbert, R.; Child, M. *The Journal of Chemical Physics* **1993**, *98*, 1902.

[17] Kvaran, Á.; Logadóttir, Á.; Wang, H. *The Journal of Chemical Physics* **1998**, *109*, 5856.

[18] Kvaran, Á.; Wang, H.; Logadóttir, Á. *The Journal of Chemical Physics* **2000**, *112*(24), 10811–10820.

[19] Kvaran, Á.; Waage, B.; Wang, H. *The Journal of Chemical Physics* **2000**, *113*, 1755.

[20] Wang, H.; Kvaran, Á. *Journal of Molecular Structure* **2001**, *563*(564), 235–239.

[21] Kvaran, Á.; Wang, H.; Waage, B. *Canadian Journal of Physics* **2001**, *79*(2/3), 197–210.

[22] Kvaran, Á.; Wang, H. *Molecular Physics* **2002**, *100*(22), 3513–3519.

[23] Kvaran, Á.; Wang, H. *Journal of Molecular Spectroscopy* **2004**, *228*(1), 143–151.

[24] Kvaran, Á.; Freyr Sigurbjörnsson, Ó.; Wang, H. *Journal of molecular structure* **2006**, *790*(1-3), 27–30.

[25] Chichinin, A.; Maul, C.; Gericke, K. *The Journal of Chemical Physics* **2006**, *124*, 224324.

[26] Banwell, C. N.; McCash, E. M. *Fundamentals of Molecular Spectroscopy;* McGraw Hill, 4 ed., 1994.

[27] Bray, R.; Hochstrasser, R. *Molecular Physics* **1976**, *31*(4), 1199–1211.

[28] Halpern, J.; Zacharias, H.; Wallenstein, R. *Journal of Molecular Spectroscopy* **1980**, *79*(1), 1–30.

[29] Møller, C.; Plesset, M. *Physical Review* **1934**, *46*(7), 618–622.

[30] Helgaker, T.; Jørgensen, P.; Olsen, J. *Molecular electronic-structure theory;* Wiley, 2000.

[31] Szabo, A.; Ostlund, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory;* Dover, 1996.

[32] Hirata, S.; Bartlett, R. *Chemical Physics Letters* **2000**, *321*(3-4), 216–224.

[33] Foresman, J.; Head-Gordon, M.; Pople, J.; Frisch, M. *The Journal of Physical Chemistry* **1992**, *96*(1), 135–149.

[34] Serrano-Andrés, L.; Merchán, M. *Journal of Molecular Structure: THEOCHEM* **2005**, *729*(1-2), 99–108.

[35] Čižek, J. *The Journal of Chemical Physics* **1966**, *45*, 4256.

[36] Dunning Jr, T. *J. Phys. Chem. A* **2000**, *104*(40), 9062–9080.

[37] Watts, J.; Bartlett, R. *Journal of Chemical Physics* **1998**, *108*(6), 2511–2514.

[38] Handy, N.; Cohen, A. *Molecular Physics* **2001**, *99*(5), 403–412.

[39] Bartlett, R. *Int. J. Mol. Sci* **2002**, *3*, 579–603.

[40] Kowalski, K. *The Journal of Chemical Physics* **2005**, *123*, 014102.

[41] Kucharski, S.; Bartlett, R. *The Journal of Chemical Physics* **1998**, *108*, 9221.

[42] Kowalski, K.; Piecuch, P. *The Journal of Chemical Physics* **2000**, *113*, 18.

[43] Kowalski, K.; Piecuch, P. *The Journal of Chemical Physics* **2005**, *122*, 074107.

[44] Kowalski, K.; Piecuch, P. *The Journal of Chemical Physics* **2000**, *113*, 5644.

[45] Taube, A.; Bartlett, R. *The Journal of Chemical Physics* **2008**, *128*, 044110.

[46] Taube, A.; Bartlett, R. *The Journal of Chemical Physics* **2008**, *128*, 044111.

[47] Flocke, N.; Bartlett, R. *The Journal of Chemical Physics* **2004**, *121*, 10935.

[48] Henderson, T.; Runge, K.; Bartlett, R. *Chemical Physics Letters* **2001**, *337*(1-3), 138–142.

[49] Bartlett, R.; Musiał, M. *Reviews of Modern Physics* **2007**, *79*(1), 291–352.

[50] Davidson, E.; Feller, D. *Chemical Reviews* **1986**, *86*(4), 681–696.

[51] Dunning Jr, T. *The Journal of Chemical Physics* **1989**, *90*, 1007.

[52] Woon, D.; Dunning Jr, T. *The Journal of Chemical Physics* **1993**, *98*, 1358.

[53] Kendall, R.; Dunning Jr, T.; Harrison, R. *The Journal of Chemical Physics* **1992**, *96*, 6796.

[54] Woon, D.; Dunning Jr, T. *The Journal of Chemical Physics* **1994**, *100*, 2975.

[55] Woon, D.; Dunning Jr, T. *The Journal of Chemical Physics* **1995**, *103*, 4572.

[56] Peterson, K.; Dunning Jr, T. *J. Chem. Phys* **2002**, *117*, 10548.

[57] Jensen, F. *The Journal of Chemical Physics* **2001**, *115*, 9113.

[58] Jensen, F. *The Journal of Chemical Physics* **2002**, *116*, 3502.

[59] Jensen, F. *The Journal of Chemical Physics* **2002**, *116*, 7372.

[60] Jensen, F. *The Journal of Chemical Physics* **2002**, *117*, 9234.

[61] Jensen, F. *The Journal of Chemical Physics* **2003**, *118*, 2459.

[62] Jensen, F.; Helgaker, T. *The Journal of Chemical Physics* **2004**, *121*, 3463.

[63] Shahbazian, S.; Zahedi, M. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)* **2005**, *113*(3), 152–160.

[64] Kupka, T.; Lim, C. *J. Phys. Chem. A* **2007**, *111*(10), 1927–1932.

[65] Jensen, F. *J. Chem. Theory Comput* **2006**, *2*(5), 1360–1369.

[66] Hohenberg, P.; Kohn, W. *Physical Review* **1964**, *136*, B864.

[67] Kohn, W.; Sham, L. *Physical Review* **1965**, *140*(4A), 1133–1138.

[68] Stephens, P.; Devlin, F.; Chabalowski, C.; Frisch, M. *The Journal of Physical Chemistry* **1994**, *98*(45), 11623–11627.

[69] Parr, R.; Yang, W. *Density-Functional Theory of Atoms and Molecules;* Oxford University Press, USA, 1989.

[70] Runge, E.; Gross, E. *Physical Review Letters* **1984**, *52*(12), 997–1000.

[71] Burke, K.; Werschnik, J.; Gross, E. *The Journal of Chemical Physics* **2005**, *123*, 062206.

[72] Elliott, P.; Burke, K.; Furche, F. **2007**.

[73] Stanton, J.; Bartlett, R. *The Journal of Chemical Physics* **1993**, *98*, 7029.

[74] Hirata, S.; Nooijen, M.; Bartlett, R. *Chemical Physics Letters* **2000**, *326*(3-4), 255–262.

[75] Hirata, S. *J. Phys. Chem. A* **2003**, *107*(46), 9887–9897.

[76] Hirata, S. *The Journal of Chemical Physics* **2004**, *121*, 51.

[77] Kowalski, K.; Hirata, S.; Włoch, M.; Piecuch, P.; Windus, T. *The Journal of Chemical Physics* **2005**, *123*, 074319.

[78] Larsen, H.; Olsen, J.; Jørgensen, P.; Christiansen, O. *The Journal of Chemical Physics* **2000**, *113*, 6677.

[79] Larsen, H.; Olsen, J.; rgensen, P. J.; Christiansen, O. *The Journal of Chemical Physics* **2001**, *114*(24), 10985–10985.

[80] Kowalski, K.; Piecuch, P. *The Journal of Chemical Physics* **2001**, *115*, 2966.

[81] Kowalski, K.; Piecuch, P. *The Journal of Chemical Physics* **2002**, *116*, 7411.

[82] Kowalski, K.; Piecuch, P. *The Journal of Chemical Physics* **2004**, *120*, 1715.

[83] Kowalski, K. *Chemical Physics Letters* **2005**, *411*(4-6), 306–310.

[84] Włoch, M.; Gour, J.; Kowalski, K.; Piecuch, P. *The Journal of Chemical Physics* **2005**, *122*, 214107.

[85] Kowalski, K. *The Journal of Chemical Physics* **2006**, *125*, 124101.

[86] Marston, C.; Balint-Kurti, G. *The Journal of Chemical Physics* **1989**, *91*, 3571.

[87] Balint-Kurti, G.; Ward, C.; Clay Marston, C. *Computer Physics Communications* **1991**, *67*(2), 285–292.

[88] Aces ii mainz-austin-budapest version. J.F. Stanton; J. Gauss; J.D. Watts; P.G. Szalay; R.J. Bartlett with contributions from A.A. Auer; D.B. Bernholdt; O. Christiansen; M.E. Harding; M. Heckert; O. Heun; C. Huber; D. Jonsson; J. Jusélius; W.J. Lauderdale; T. Metzroth; C. Michauk; D.R. Price; K. Ruud; F. Schiffmann; A. Tajti; M.E. Varner; J. Vázquez and the integral packages: MOLECULE (J. Almlöf and P.R. Taylor); PROPS (P.R. Taylor), and ABACUS (T. Helgaker; H.J. Aa. Jensen; P. Jørgensen; and J. Olsen)..

[89] Nwchem, a computational chemistry package for parallel computers, version 5.0. Bylaska, E. J.; de Jong, W. A.; Kowalski, K.; Straatsma, T. P.; Valiev, M.; Wang, D.; Apra, E.; Windus, T. L.; Hirata, S.; Hackler, M. T.; Zhao, Y.; Fan, P.-D.; Harrison, R. J.; Dupuis, M.; Smith, D. M. A.; Nieplocha, J.; Tipparaju, V.; Krishnan, M.; Auer, A. A.; Nooijen, M.; Brown, E.; Cisneros, G.; Fann, G. I.; Fruchtl, H.; Garza, J.; Hirao, K.; Kendall, R.; Nichols, J. A.; Tsemekhman, K.; Wolinski, K.; Anchell, J.; Bernholdt, D.; Borowski, P.; Clark, T.; Clerc, D.; Dachsel, H.; Deegan, M.; Dyall, K.; Elwood, D.; Glendening, E.; Gutowski, M.; Hess, A.; Jaffe, J.; Johnson, B.; Ju, J.; Kobayashi, R.; Kutteh, R.; Lin, Z.; Littlefield, R.; Long, X.; Meng, B.; Nakajima, T.; Niu, S.; Pollack, L.; Rosing, M.; Sandrone, G.; Stave, M.; Taylor, H.; Thomas, G.; van Lenthe, J.; Wong, A.; Zhang, Z. Pacific Northwest National Laboratory, Richland, Washington 99352-0999, USA, **2006**.

[90] Nwchem, a computational chemistry package for parallel computers, version 5.1. Bylaska, E. J.; de Jong, W. A.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Valiev, M.; Wang, D.; Apra, E.; Windus, T. L.; Hammond, J.; Nichols, P.; Hirata, S.; Hackler, M. T.; Zhao, Y.; Fan, P.-D.; Harrison, R. J.; Dupuis, M.; Smith, D. M. A.; Nieplocha, J.; Tipparaju, V.; Krishnan, M.; Wu, Q.; Voorhis, T. V.; Auer, A. A.; Nooijen, M.; Brown, E.; Cisneros, G.; Fann, G. I.; Fruchtl, H.; Garza, J.; Hirao, K.; Kendall, R.; Nichols, J. A.; Tsemekhman, K.; Wolinski, K.; Anchell, J.; Bernholdt, D.; Borowski, P.; Clark, T.; Clerc, D.; Dachsel, H.; Deegan, M.; Dyall, K.; Elwood, D.; Glendening, E.; Gutowski, M.; Hess, A.; Jaffe, J.; Johnson, B.; Ju, J.; Kobayashi, R.; Kutteh, R.; Lin, Z.; Littlefield, R.; Long, X.; Meng, B.; Nakajima, T.; Niu, S.; Pollack, L.; Rosing, M.; Sandrone, G.; Stave, M.; Taylor, H.; Thomas, G.; van Lenthe, J.; Wong, A.; Zhang, Z. Pacific Northwest National Laboratory, Richland, Washington 99352-0999, USA, **2007**.

[91] Huber, K. P.; Herzberg, G. *Constants of Diatomic Molecules;* Van Nostrand-Reinhold, New York, 1979.

[92] Tilford, S.; Ginter, M.; Vanderslice, J. *Journal of Molecular Spectroscopy* **1970**, *33*(3), 505–519.

[93] di Lonardo, G.; Douglas, A. *Canadian Journal of Physics* **1973**, *51*, 434.

[94] Remote Control Manual LeCroy Digital Oscilloscopes. LeCroy, September , **1996**.

[95] ScanMate*Pro*^{TM}Remote Control. Lambda Physik, November , **2004**.