

Lawrence Berkeley National Laboratory

Recent Work

Title

Studies of Transition States and Radicals by Negative Ion Photodetachment

Permalink

<https://escholarship.org/uc/item/24v4b1vm>

Author

Metz, R.B.

Publication Date

1991-12-01

LBL--31878

DE92 017011

**Studies of Transition States and Radicals by
Negative Ion Photodetachment**

Ricardo Baer Metz
Ph.D. Thesis

Department of Chemistry
University of California

and

Chemical Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

December 1991

This work was supported in part by the Air Force Office of Scientific Research under Contract No. AFOSR-91-0084, and by the Director, Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences Division, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

MASTER

**Studies of Transition States and Radicals by
Negative Ion Photodetachment**

Copyright © 1991

by

Ricardo B. Metz

The U.S. Department of Energy has the right to use this thesis
for any purpose whatsoever including the right to reproduce
all or any part thereof

Studies of Transition States and Radicals by
Negative Ion Photodetachment

by
Ricardo B. Metz

Abstract

Negative ion photodetachment is a versatile tool for the production and study of transient neutral species such as reaction intermediates and free radicals. Photodetachment of the stable XHY^- anion provides a direct spectroscopic probe of the transition state region of the potential energy surface for the neutral hydrogen transfer reaction $\text{X} + \text{HY} \rightarrow \text{XH} + \text{Y}$, where X and Y are halogen atoms. The technique is especially sensitive to resonances, which occur at a specific energy, but the spectra also show features due to direct scattering. We have used collinear adiabatic simulations of the photoelectron spectra to evaluate trial potential energy surfaces for the bimolecular reactions and have extended the adiabatic approach to three dimensions and used it to evaluate empirical potential energy surfaces for the $\text{I} + \text{HI}$ and $\text{Br} + \text{HI}$ reactions. In addition, we have derived an empirical, collinear potential energy surface for the $\text{Br} + \text{HBr}$ reaction that reproduces our experimental results and have extended this

surface to three dimensions.

Photodetachment of a negative ion can also be used to study neutral free radicals. We have studied the vibrational and electronic spectroscopy of CH_2NO_2 by photoelectron spectroscopy of CH_2NO_2^- , determining the electron affinity of CH_2NO_2 , gaining insight on the bonding of the ${}^2\text{B}_1$ ground state and observing the ${}^2\text{A}_2$ excited state for the first time. Negative ion photodetachment also provides a novel and versatile source of mass-selected, jet-cooled free radicals. We have studied the photodissociation of CH_2NO_2 at 270, 235 and 208 nm, obtaining information on the dissociation products by measuring the kinetic energy release in the photodissociation.

Table of Contents

Abstract	1
Dedication	xii
Acknowledgements	xiii
 Chapter 1. Studies of Transition States and Radicals by Negative Ion Photodetachment	
1. Introduction	1
2. Introduction to Photoelectron Spectroscopy	2
3. Photoelectron Studies of the Transition State Region of the Potential Energy Surface for a Neutral Bimolecular Reaction	3
4. Studies of Neutral Free Radicals	6
4.1 Photoelectron Spectra	6
4.2 Photodetachment as a Source of Free Radicals	7
References	9
Figure Captions	11
Figures	12

**Chapter 2. Probing the Transition State with Negative Ion
Photodetachment: The Cl + HCl and Br + HBr
Reactions**

1. Introduction	15
2. Experimental	21
3. Results and Preliminary Discussion	26
4. Analysis and Discussion	29
4.1 General Considerations	29
4.2 One-Dimensional Approach	34
4.3 Two-Dimensional Analysis	39
4.4 Application to Cl + HCl	46
4.5 Br + HBr Potential Surface	48
4.5.1 Contributions from Resonances and Direct Scattering	49
4.5.2 Functional Form of the Potential Energy Surface	51
4.5.3 Results of the Analysis	53
5. Summary and Conclusions	57
References	60
Tables	66
Figure Captions	72
Figures	76

**Chapter 3. Computational Framework for Analyzing
Photoelectron Spectra of the Bihalide Anions**

1. Introduction	91
2. Adiabatic Methods	93
3. Solving the Schrodinger Equation Using Relaxation	96
4. Matrix Approaches to the Schrodinger Equation	98
5. Solving the Schrodinger Equation in the DVR	101
5.1 One-Dimensional DVR Methods	101
5.2 Incorporating Symmetry in the DVR	107
5.3 The DVR in Two Dimensions	109
5.4 Scattering States in the DVR	112
References	116
Figure Caption	119
Figures	120

**Chapter 4. Adiabatic Three-Dimensional Simulations of the IHI^- ,
 BrHI^- , and BrHBr^- Photoelectron Spectra**

1. Introduction	122
2. Method	128
3. Results	140
3.1 IHI^- on the LEPS-A Potential	140
3.2 IDI^- on the LEPS-A Surface	148

3.3 IHI ⁻ on the LEPS-C Surface	154
3.4 IDI ⁻ on the LEPS-C Potential	156
3.5 BrHI ⁻ on the BP LEPS Surface	157
3.6 BrHBr ⁻ on a Fitted Surface	161
3.7 BrDBr ⁻ on a Fitted Surface	165
4. Conclusions	167
References	169
Tables	174
Figure Captions	176
Figures	181

**Chapter 5. Study of the ²B₁ and ²A₂ States of CH₂NO₂ via
Ultraviolet Photoelectron Spectroscopy of the
CH₂NO₂⁻ Anion**

1. Introduction	197
2. Experimental	201
3. Results	202
4. Analysis and Discussion	204
4.1 Qualitative Features	204
4.2 Molecular Orbital Picture	206
4.3 <i>Ab Initio</i> Calculations	207
4.4 Detailed Analysis of Spectrum	211

4.4.1 Torsion Mode	212
4.4.2 Totally Symmetric Modes	216
4.5 Excited Electronic State	220
4.6 Thermodynamics	222
5. Conclusions	223
References	225
Tables	229
Chart Caption	238
Chart	239
Figure Captions	240
Figures	242
6. Photodissociation of CH₂NO₂ at 208, 235 and 270 nm	
1. Introduction	248
2. Experimental	250
3. Results	253
3.1 Time-of-Flight Spectra.	253
3.2 Primary Photodissociation Products	253
4. Analysis and Discussion	255
4.1. Simulation of the 270 nm	
Time-of-Flight Spectrum	255

4.2. Simulation of the 235 and 208 nm Time-of-Flight Spectra	258
4.3. Comparison with CH_3NO_2	260
5. Conclusions	262
References	264
Table	267
Figure Captions	268
Figures	269

**Appendix A. Computer Programs for Collinear and Three-
Dimensional Adiabatic Simulations of XHY^-
Photoelectron Spectra**

1. Collinear Simulation Code ad2d7	273
1.1 Introduction	273
1.2 Running the ad2d7 Program	274
1.2.1 The ad2d.par Input File	274
1.2.2 Keyboard Inputs	276
1.2.3 Using a Fitted Potential	281
1.3 How the ad2d7 Program Works	282
1.4 Source Code Listing	290
1.4.1 makead2d7	290
1.4.2 ad2d7.inc	290

1.4.3	ad2d7.f	291
1.4.4	leps6.f	311
1.4.5	divah.f	316
1.4.6	findmin.f	319
1.4.7	boundrho3.f	320
1.4.8	dofcf.f	329
1.4.9	pot7.f	334
1.4.10	matrix.f	338
2.	Three-Dimensional Simulation Code ad3d5	342
2.1	Introduction	342
2.2	Running the ad3d5 Program	343
2.2.1	The ad3d.par Input File	343
2.2.2	Keyboard Inputs	345
2.2.3	Using a Fitted Potential	351
2.3	How the ad3d5 Program Works	351
2.4	Source Code Listing	358
2.4.1	makead3d5	358
2.4.2	ad3d.inc	358
2.4.3	ad3d5.f	359
2.4.4	leps3d2.f	379
2.4.5	findmin.f	382
2.4.6	boundr3d.f	383

2.4.7 dof3d.f	391
2.4.8 pot6.f	396
2.4.9 bndhbr.f	401
2.4.10 matrix.f	403
2.4.11 pot.par	403
3. The Simulation Program sim1	405
3.1 Running the sim1 Program	405
3.2 The File sim1.f	405

Appendix B. Tenure, The Data Acquisition Program for the Fixed-Frequency Photoelectron Spectrometer

1. Introduction	409
2. Description of Files	418
3. Source Code Listing	420
3.1 mgplt20.pas	420
3.2 plotlib9.pas	432
3.3 rmgrap15.pas	442
3.4 calib1.pas	465
3.5 scale.pas	472
3.6 ftshort.pas	474
3.7 four1.pas	477
3.8 modfile.pas	479

3.9 tp3decl3.pas	479
3.10 graphli2.tgt	481
3.11 expt.par	481
3.12 help1.par	483
3.13 help2.par	483

Appendix C. Publications from Graduate Work

1. Publications	485
-----------------------	-----

This thesis is dedicated to my parents,

Manuel and Margarita Metz

Acknowledgements

Being part of the start of a research group, building the first experiment, and getting the first results is a thrilling experience. I wish to thank Prof. Dan Neumark for making this possible, and my co-explorers Alex Weaver and Theo Kitsopoulos for making it so much fun. The experiments and calculations described in this thesis were made possible by the enthusiasm and patient support of Prof. Dan Neumark, who always managed to come up with a way to make the experiment work better. Working with Dan has been a great thrill and a very educational experience. He has also given me the chance to help construct not one, but two new experiments.

Alex Weaver (aka The Supreme Alex) has taught me a thing or two about diligence and common sense. She made the long nights (and days - it's hard to tell the difference on D-level) in lab more fun with enlightening discussions of life, morals, sailing and (yes) even chemistry. She (and Theo) expanded my experience and appreciation (?) of Greek music and were companions on Frusen Nights and during Thursday Night Raging. Her studies of the NO_2^- photoelectron spectrum provide the background for a great deal of the discussion of the photodissociation of CH_2NO_2 . Theo greatly livened the lab with his outgoing personality, contagious enthusiasm and constant tinkering. He also greatly extended my Greek vocabulary

(little of it printable) and taught me not to punch solid wooden doors.

Steve Bradforth was always around to answer silly "what if?" and spectroscopy questions. He has also kept the computers up and running, guided me on my forays into the wonderful world of *ab initio* calculations, and his two-dimensional wavepacket calculations have provided a very useful comparison with my more approximate adiabatic calculations. Who else would come up with the idea of dropping CH_3OD in a pulsed beam valve at 2 am to make CH_3ODF^- ? (Yes, it worked).

On to the fast radical beam machine - Dr. Bob Continetti introduced me to the wonders of piezoelectric pulsed valves, the FIG, and countless other acronyms. His electronic inventiveness made the experiment possible and is taking it in exciting new directions. He also wrote the Monte Carlo code used in the analysis of CH_2NO_2 photodissociation. Doug Cyr has set a mighty example with his thoroughness and hard work, designed a great deal of the experiment and wrote a very nice data acquisition program for it. His sense of adventure led to our obtaining the photoelectron spectra of CH_2NO_2^- , HCO_2^- , CF_3^- , and CCl_3^- , which has led to two chapters of this thesis and is likely to result in quite a few chapters of other theses. He also managed to scavenge many indispensable pieces of equipment and was always around when some big, heavy piece of equipment had to be lifted.

Although proper acknowledgements to the remaining group members cannot be made here, among other things, Dr. (now Prof.) Irene Waller

showed that it actually is possible to get an academic position, Don Arnold provided some nice machining and design tips, Caroline Chick and Yuexing Zhao (along with Theo and Irene) were instrumental in getting the threshold photodetachment spectra of IHI^- and IDI^- that showed the need for a three-dimensional treatment. Eun Ha Kim is carrying on the HCO_2^- studies and has provided interesting discussions on this system.

I also wish to thank the members of the Lee and Saykally groups, from whom we borrowed so many pieces of equipment (especially that first year) - maybe someday we'll get around to returning some of it. These experiments would not have been possible without the efforts of the Department of Chemistry Machine Shop, Ron Dal-Porto, Andy Anderson, George Weber (who was especially helpful in suggesting ways to construct things), and machinists Eric Granlund, Hans Graetsch, Dave Murai, Scott Bonilla, Chuck Sounders, and Larry Hull, who did the bulk of the very precise machining on the radical spectrometer. The success of the experiments is proof of their care and skill.

The adiabatic three-dimensional approach was inspired by a talk given by Dr. Russ Pack at the 1989 Dynamics of Molecular Collisions Conference. After reading many papers countless times, the DVR finally "clicked" in a discussion with Prof. John Light. Prof. George Schatz has been instrumental in helping to understand the structure seen in the threshold photodetachment spectra of IHI^- and IDI^- . During many

discussions I have benefitted greatly from his insight into the nature of the scattering states formed by photodetachment. His exact three-dimensional scattering results have provided the "right answer" against which the adiabatic results can be gauged. Prof. Joel Bowman has also proven extremely helpful in understanding the collinear simulations and kindly furnished computer code for addition of the zero-point bend energy to a potential.

Dr. Fred Schwarz of NIST (then NBS) gave me my first opportunity to work in a physical chemistry lab, and subsequent work with Drs. Sharon Lias, Marilyn Jacox (who I also wish to thank for discussions on CH_2NO_2) and Tom Buckley (who taught me data acquisition programming) convinced me to pursue the field.

My parents have always been extremely understanding of my efforts and supported my education, even when it meant moving to faraway California. I also wish to thank Mark Paige, Eric Rawlins, and (especially) Susie Wakefield, and for being good friends as well as roommates. Finally, I wish to thank Susan Cobb - without her I would have finished this thesis a bit sooner, and had a lot less fun doing it.

This research was supported in part by the Director, Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences Division of the U.S. Department of Energy under contract No. DE-AC03-76SF00098, and in part by the Air Force Office of Scientific Research contract No. AFOSR-91-0084.

Chapter 1. Studies of Transition States and Radicals by Negative Ion Photodetachment

1. Introduction

Chemistry is the art of transforming stable reagents into the desired products. All chemical reactions have one aspect in common - they involve going through some chemical transition state (where bonds are broken and new bonds are formed) on the way to products; in addition, they may involve the production of transient chemical intermediates (often free radicals) on the road to products. Physical chemists have long been fascinated by these reactive transition states and transient intermediates,¹ as these intermediates control the course of the chemical reaction, but, due to their reactive and ephemeral nature, these elusive species are as difficult to study as they are exciting. We use photoelectron spectroscopy of a negative ion to study these neutral species spectroscopically. If the negative ion is similar in structure to the transient species of interest, the photoelectron spectrum will reveal structure due to the neutral species. In this manner, species with lifetimes as short as 10^{-15} s can be studied. In Section 2 we introduce negative ion photoelectron spectroscopy, in Section 3 we discuss how we apply the technique to study the transition state region of a neutral bimolecular reaction, and in Section 4 we discuss studies of neutral free radicals (stable reaction intermediates). Section 4.1 discusses

studies of neutral free radicals using negative ion photoelectron spectroscopy and Section 4.2 discusses using photodetachment of a negative ion as a source of cold mass-selected free radicals.

2. Introduction to Photoelectron Spectroscopy

Negative ion photoelectron spectroscopy is a versatile tool for studying transient neutral species. The negative ion corresponding to the species of interest is created in a pulsed molecular beam, so it is vibrationally and rotationally cooled after it is formed. The ion is then accelerated, mass selected, detached using a pulsed fixed-frequency laser and the kinetic energy of the detached electrons is measured. The laser timing is adjusted so that only the ion of interest is detached. Negative ion photoelectron spectroscopy of a bound anion \rightarrow bound neutral system is illustrated schematically in Fig. 1-1. The kinetic energy of the detached electron is given by²

$$eKE = h\nu - EA + E_{v''} - E_{v'} \quad (1)$$

where eKE is the kinetic energy of the detached electron, $h\nu$ is the photon energy, EA is the electron affinity of the neutral and $E_{v''}$ and $E_{v'}$ are the vibrational energies of the anion and neutral, respectively. As the vast majority of the anions are formed in the vibrational ground state, the structure in the photoelectron spectrum is due to vibrational states of the

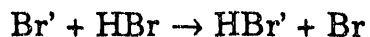
neutral free radical. A schematic photoelectron spectrum based on the potential curves in Fig. 1-1 is given in Fig. 1-2. Note that peaks at *higher* electron kinetic energy correspond to states of the neutral with *lower* internal energy. The positions of the peaks directly give the vibrational frequencies in the neutral, while the intensities, given by the Franck-Condon factors

$$I \propto |\langle \psi_{v'}^- | \psi_{v'}^0 \rangle|^2 \quad (2)$$

can be used to determine the difference in geometry between the anion and neutral.

3. Photoelectron Studies of the Transition State Region of the Potential Energy Surface for a Neutral Bimolecular Reaction

Negative ion photoelectron spectroscopy also provides a unique tool for studying the transition state region of a neutral bimolecular chemical reaction.³ All that is required is an anion similar in structure to the transition state for the reaction of interest. For example, we study the bimolecular hydrogen transfer reaction



by photoelectron spectroscopy of the BrHBr^- anion. The BrHBr^- anion is thought to be linear and centrosymmetric,⁴ and is very strongly hydrogen bonded, with a dissociation energy of 0.91 eV.⁵ The situation can be

visualized with reference to Fig. 1-3, which shows schematic potential curves for BrHBr^- and the $\text{Br} + \text{HBr}$ reaction drawn along the reaction coordinate for the $\text{Br} + \text{HBr}$ reaction. Detachment of BrHBr^- forms the $[\text{BrHBr}]$ complex at the maximum of the upper curve, near the saddle point on the potential energy surface for the reaction. As the $\text{Br} + \text{HBr}$ system has three atoms, it has two coordinates perpendicular to the reaction coordinate. At the saddle point on the neutral surface, the energy along these two coordinates is at a minimum, while it is at a maximum along the reaction coordinate. The photoelectron spectrum of BrHBr^- can thus exhibit structure due to motion along the two bound coordinates,⁶ while the peaks will broaden due to the limited lifetime of the $[\text{BrHBr}]$ complex.

In addition, collinear⁷ and three-dimensional⁸ quantum scattering calculations on Heavy + Light-Heavy systems (such as $\text{Br} + \text{HBr}$ and $\text{I} + \text{HI}$) predict the existence of dynamical resonances, which involve recurring motion along the reaction coordinate. Negative ion photoelectron spectroscopy provides an excellent method of observing these resonances, if they exist. The neutral reactive complex is formed with a well-defined initial geometry and angular momentum, unlike in scattering experiments, where collisions take place over a range of impact parameters and reactant orientations. Resonances would be expected to result in sharp vibrational peaks in the photoelectron spectrum. The photoelectron spectra of IHI^- and BrHBr^- do show structure due to resonances, but much of the structure is

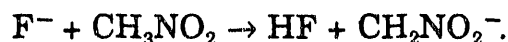
due to transitions to nonresonant states of the neutral complex. In both cases, the photoelectron spectrum gives detailed information on the transition state region of the neutral bimolecular reaction, and this information can be used to refine potential surfaces for the neutral reaction.

We have studied the $\text{Cl} + \text{HCl}$ and $\text{Br} + \text{HBr}$ reactions via photoelectron spectroscopy of ClHCl^- and BrHBr^- and have carried out collinear simulations of the ClHCl^- and BrHBr^- photoelectron spectra using several empirical potential energy surfaces for the $\text{Cl} + \text{HCl}$ and $\text{Br} + \text{HBr}$ reactions. We have also constructed an empirical collinear potential energy surface for the $\text{Br} + \text{HBr}$ reaction that fits our photoelectron results. Our collinear studies are described in Chapter 2. The high resolution photoelectron spectrum of IHI^- ⁹ shows structure attributable to "rotational threshold" states. Theoretical treatment of these states requires a three-dimensional (rather than a collinear) approach. We have used a variety of computational techniques to allow us to extend our collinear calculations to three dimensions. These techniques are described in detail and compared in Chapter 3. Specifically, we have devised an adiabatic three-dimensional method of simulating the photoelectron spectra of Heavy + Light-Heavy systems and applied it to the photoelectron spectra of IHI^- , BrHI^- and BrHBr^- . We have also extended the collinear surface for the $\text{Br} + \text{HBr}$ reaction derived in Chapter 2 to three dimensions. These adiabatic three-dimensional results are discussed in Chapter 4.

4. Studies of Neutral Free Radicals

4.1 Photoelectron Spectra

Negative ion photoelectron spectroscopy has been widely used to study the vibrational and electronic spectroscopy of reactive neutral species such as free radicals.¹⁰ Photoelectron spectroscopy has the advantage that, since the precursor molecule is charged, it can be mass selected and only the radical of interest will be studied, avoiding contamination of the spectrum by other species. We have used negative ion photoelectron spectroscopy to study the nitromethyl free radical CH_2NO_2 , which is thought to be an important intermediate in the combustion of nitromethane, a widely used propellant.¹¹ Thus, to obtain the photoelectron spectrum of CH_2NO_2^- we first form CH_2NO_2^- via proton abstraction by F^-

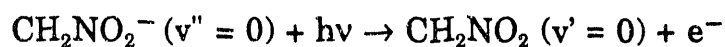


Many other anions (such as $(\text{CH}_3\text{NO}_2)_n \text{CH}_2\text{NO}_2^-$ cluster anions) are also formed in the source. The anions are formed in a pulsed supersonic expansion so that they undergo significant cooling after they are formed (We have measured a rotational temperature of 35 K for N_3 in a similar source).¹² The ions are accelerated (to a high beam velocity) and mass-selected by time of flight and only the ion of interest is detected and its photoelectron spectrum is measured. The photoelectron spectrum of CH_2NO_2^- and its analysis are the topic of Chapter 5.

4.2 Photodetachment as a Source of Free Radicals

One of the greatest difficulties encountered in the study of polyatomic free radicals is in obtaining sufficient quantities of the pure free radical in a single, well-defined quantum state.¹³ Most sources of free radicals produce the species of interest via photodissociation of a stable precursor. However, this can produce molecules other than the radical of interest, and usually will produce the radical of interest in a variety of excited vibrational and rotational states, thus complicating the interpretation of spectroscopic results. Negative ion photoelectron spectroscopy provides a versatile source of cold, mass-selected free radicals.

For example, if we wish to study photodissociation of the nitromethyl free radical CH_2NO_2 , we require a source of (preferably cold) CH_2NO_2 . We produce cold CH_2NO_2^- (see Sec. 4.1) and photodetach it using a pulsed, tunable dye laser. The dye laser is tuned above the electron affinity, but below the first vibrational peak in the photoelectron spectrum. Thus, if all the anions are formed with vibrationally cold, only the process



is energetically possible. Any remaining anions are deflected using an electric field. Thus we have a fast beam of pure, vibrationally and rotationally cold CH_2NO_2 radicals.

The radicals are then photodissociated using a second tunable dye laser. The very high beam velocity allows fragments to be efficiently detected using microchannel plates. Molecules that do not dissociate directly along the beam axis will produce fragments with some velocity perpendicular to the beam axis. We block the beam axis, effectively eliminating signal due to the undissociated radical beam and measure the fragment signal as a function of wavelength to determine the total photodissociation cross section. We can also learn about the kinetic energy release in the photodissociation by measuring the time-of-flight distribution of the fragments. The photodissociation of CH_2NO_2 forms the topic of Chapter 6.

Finally, the computer program used to acquire data on the fixed-frequency photoelectron spectrometer is included and discussed in Appendix A, and computer programs for the adiabatic collinear and adiabatic three-dimensional simulations of photoelectron spectra for Heavy + Light-Heavy systems are included as Appendix B.

1. R. D. Levine and R. B. Bernstein, Molecular Reaction Dynamics and Chemical Reactivity, Oxford University Press, New York, (1987).
2. A good discussion of photoelectron spectroscopy (in the context of neutral \rightarrow positive ion transitions) is given in J. W. Rabalais, Principles of Ultraviolet Photoelectron Spectroscopy, John Wiley & Sons, New York, (1977), Ch. 3.
3. R. B. Metz, S. E. Bradforth and D. M. Neumark, *Adv. Chem. Phys.* (accepted, 1991).
4. A. B. Sannigrahi and S. D. Peyerimhoff, *J. Mol. Struct.* **165**, 55 (1988).
5. G. Caldwell and P. Kebarle, *Can. J. Chem.* **63**, 1399 (1985).
6. R. T. Pack, *J. Chem. Phys.* **65**, 4765 (1976).
7. See Chapter 2 and references therein.
8. See Chapter 4 and references therein.
9. I. M. Waller, T. N. Kitsopoulos and D. M. Neumark, *J. Phys. Chem.* **94**, 2240 (1990).
10. R. D. Mead, A. E. Stevens and W. C. Lineberger in Gas Phase Ion Chemistry, Vol. 3 (Ions and Light), ed. M. T. Bowers, Academic, Orlando, (1984), pp. 213-248.
11. T. L. Cottrell, T. E. Graham and T. J. Reid, *Trans. Faraday Soc.* **47**, 584 (1951); G. M. Nazin, G. B. Manelis and F. I. Dubovitskii, *Russ. Chem. Rev.* **37**, 603 (1968); C. G. Crawford and D. J. Waddington, *Trans.*

- Faraday Soc. **65**, 1334 (1969); F. I. Dubovitskii and B. L. Korsunskii,
Russ. Chem. Rev. **50**, 958 (1981).
12. R. E. Continetti, D. R. Cyr, R. B. Metz and D. M. Neumark, Chem.
Phys. Lett. **182**, 406 (1991).
13. H. Okabe, Photochemistry of Small Molecules, Wiley & Sons, New York,
(1978).

Chapter 1 Figure Captions

Figure 1-1. Schematic illustration of photoelectron spectroscopy.

Downward arrows indicate electron kinetic energies corresponding to each $v'' \rightarrow v'$ transition.

Figure 1-2. Photoelectron spectrum corresponding to potential curves in Fig. 1-1. Note that all the structure in the spectrum is due to vibrations of the neutral, and that neutral states with lowest internal energy appear at highest electron kinetic energy.

Figure 1-3. Schematic view of photoelectron spectroscopy of BrHBr^- .

Detachment of BrHBr^- accesses the transition state region of the $\text{Br} + \text{HBr} \rightarrow [\text{BrHBr}] \rightarrow \text{BrH} + \text{Br}$ reaction.

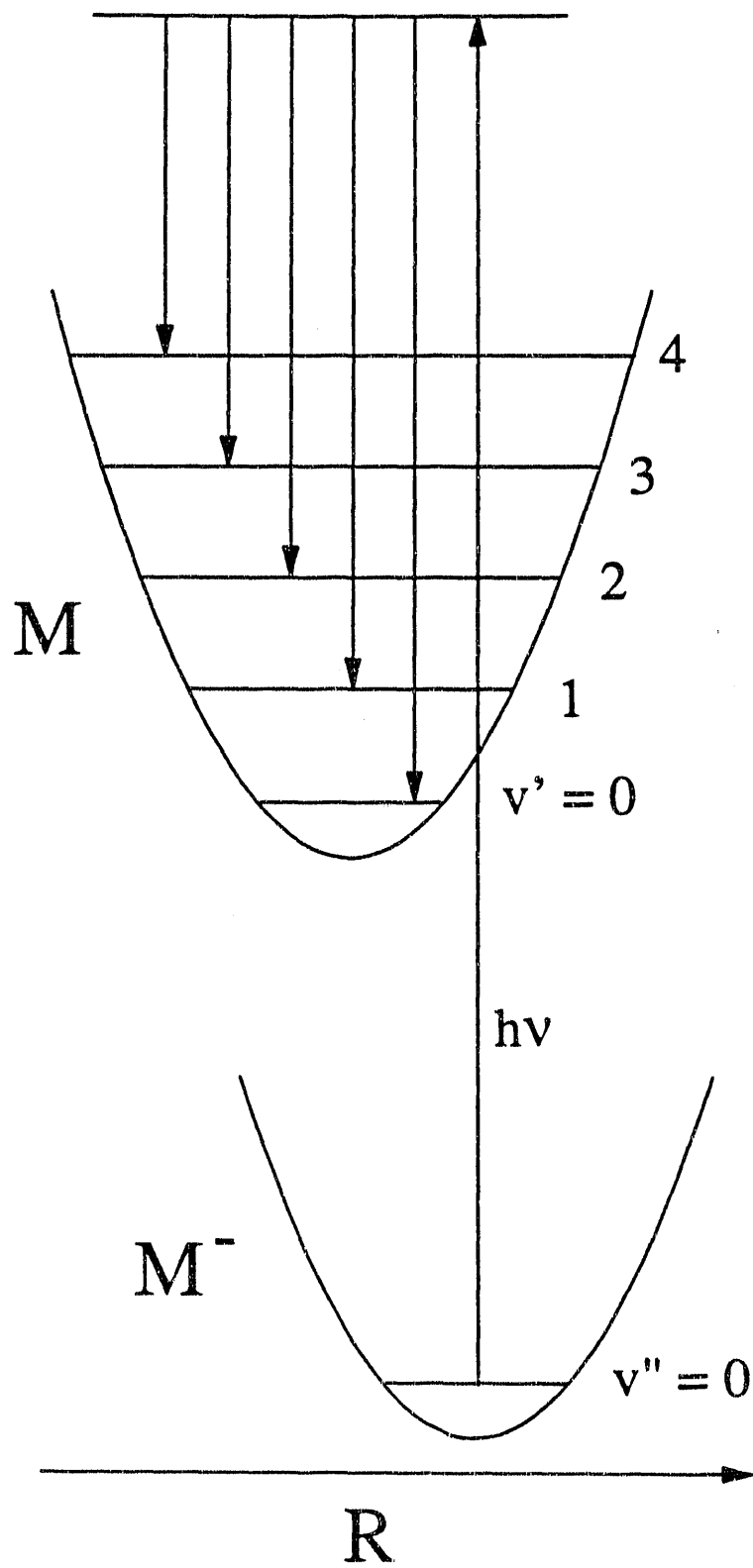


Figure 1-1

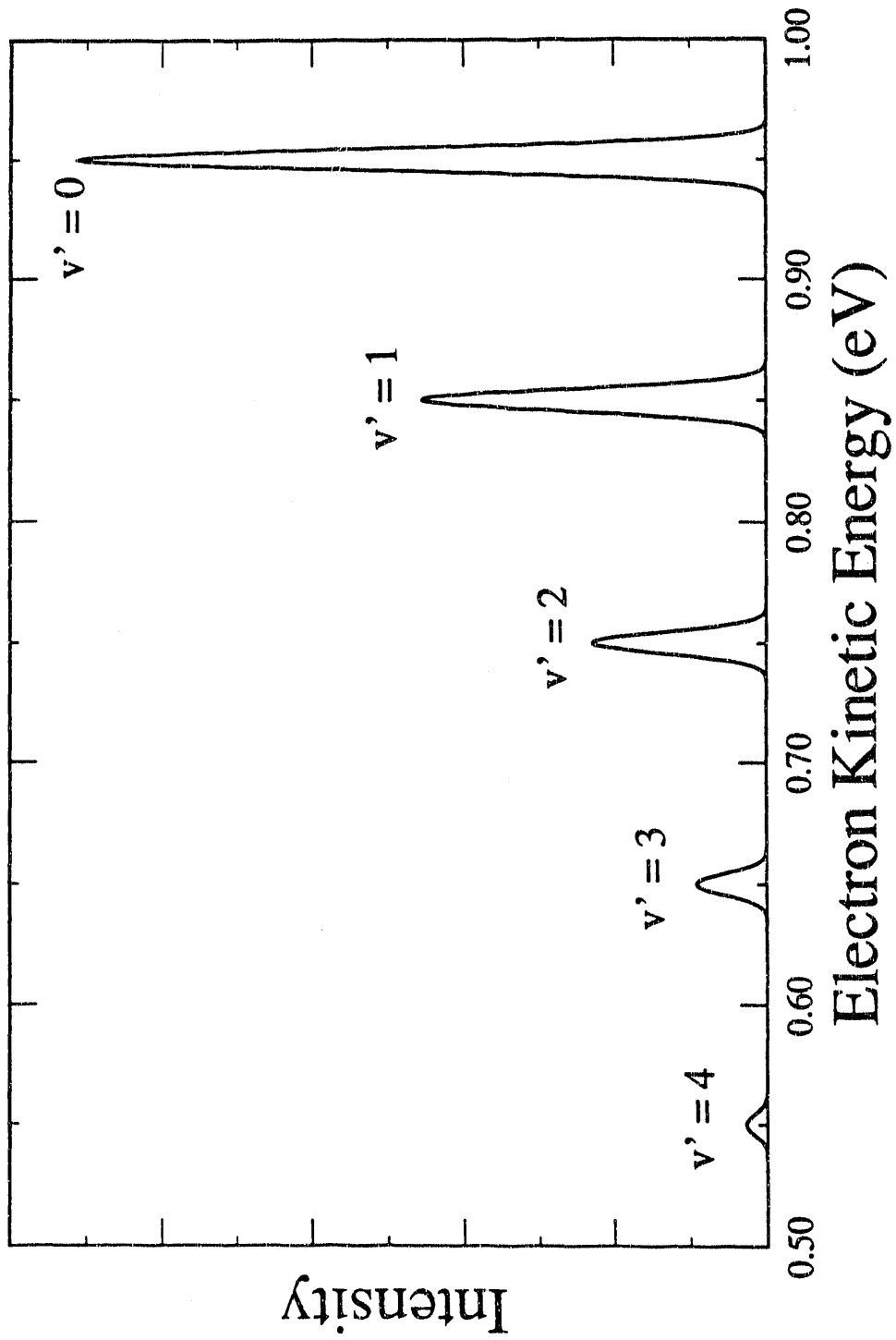


Figure 1-2

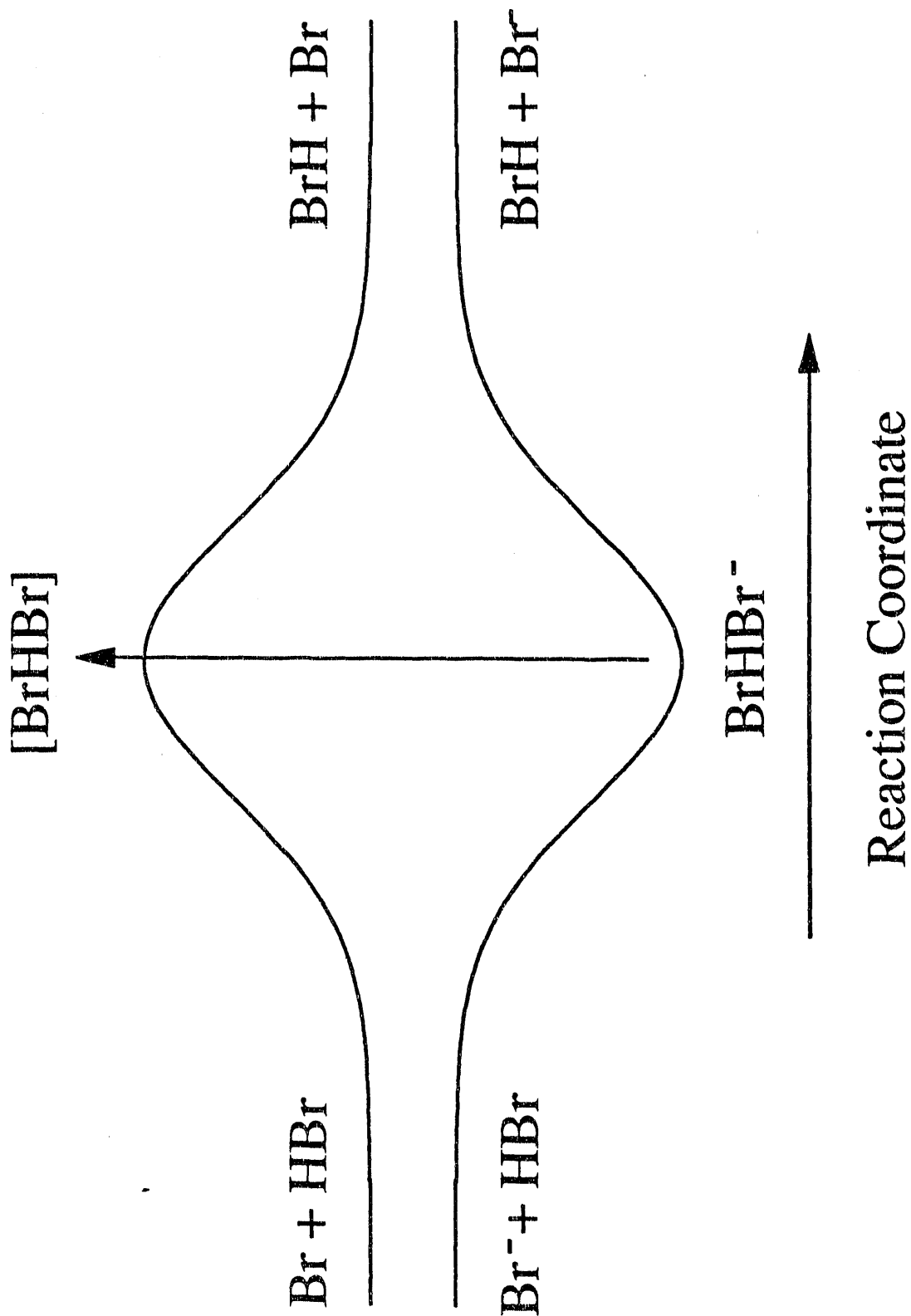


Figure 1-3

Chapter 2. Probing the Transition State with Negative Ion Photodetachment: The Cl + HCl and Br + HBr Reactions

1. Introduction

One of the fundamental issues in the field of reaction dynamics is the nature of the transition state in a chemical reaction. The features of a reactive potential energy surface near the transition state affect nearly all of the important properties of a chemical reaction including the rate constant, the effect of reactant translational and internal energy on reactivity, and the product energy and angular distribution. The aim of much of state-to-state chemistry over the last ten to fifteen years has been to measure these asymptotic properties as accurately and completely as possible and to attempt to use this information to learn about the region of the potential energy surface in the vicinity of the transition state (the transition state region). While state-to-state scattering experiments have led to a qualitative understanding of reactive potential energy surfaces,¹ averaging over reactant orientation and impact parameter in most of these experiments greatly complicates the extraction of transition state properties from the measured asymptotic distributions. The transition state region is, in principle, amenable to study by quantum chemistry calculations, but the

ab initio construction of a potential energy surface is difficult even for the simplest chemical reactions.²

This situation has motivated the development of a series of 'transition state spectroscopy' experiments^{3,4} designed to probe the transition state region itself rather than the asymptotic properties of a reactive encounter. The initial experiments in this field were performed by Polanyi,⁵ Brooks,⁶ and their co-workers. The transition state region of several reactive and dissociative systems has since been studied in a variety of 'full collision'⁷⁻⁹ and 'half collision'¹⁰⁻¹² experiments. Of particular interest are the recent experiments of Nieh and Valentini¹³ on the $H + H_2$ reaction, in which the observation of quasi-bound states of the $[H_3]$ complex was reported, and the real-time studies of photodissociation by Zewail and co-workers.¹⁴

We have devised and implemented a transition state spectroscopy experiment in which negative ion photodetachment is used to probe the transition state region for a neutral bimolecular reaction. Our experiment draws on earlier studies in which photodetachment of a stable negative ion yields an unstable neutral species. These include total photodetachment cross section measurements by Golub and Steiner¹⁵ on $(H_2O)OH^-$ and, more recently, by Brauman and co-workers on a series of ions $ROHF^-$.¹⁶ In both of these studies, photodetachment of a stable, hydrogen-bonded ion produced a neutral complex unstable with respect to dissociation. In another important experiment, Lineberger and co-workers¹⁷ measured the

photoelectron spectrum of the vinylidene anion H_2CC^- . Photodetachment of this anion produces the vinylidene radical H_2CC , which isomerizes rapidly to acetylene HCCH . Nonetheless, the negative ion photoelectron spectrum shows vibrational structure associated with the unstable form of the radical.

In our experiments, the transition state region of the heavy + light-heavy reaction $\text{X} + \text{HY} \rightarrow \text{XH} + \text{Y}$ is probed by laser photoelectron spectroscopy of the stable hydrogen-bonded negative ion XHY^- . This provides a direct, spectroscopic probe of the $[\text{XHY}]$ collision complex, as long as the geometry of the ion is similar to that of the neutral transition state. Figure 2-1 shows a one-dimensional idealization of our experiment for a symmetric exchange reaction ($\text{X} = \text{Y}$). The potential energy of the ion and neutral are shown as a function of an appropriate reaction coordinate. The $[\text{XHX}]$ complex formed at or near the top of the barrier is unstable with respect to motion along the reaction coordinate. However, the barrier along the reaction coordinate is actually a saddle point on a multi-dimensional potential energy surface. The XHX^- photoelectron spectrum can therefore exhibit peaks which correspond to vibrational motion along coordinates 'perpendicular' to the reaction coordinate. The peak spacings yield the frequencies associated with these vibrations, while the peak widths are an indication of motion along the reaction coordinate; the ratio of peak spacing to peak widths measures how many vibrations the complex undergoes as it dissociates. We therefore learn about the spectroscopy and dissociation

dynamics of the collision complex.

X and Y can be atomic or polyatomic species. The simplest systems are symmetric reactions of the type $X + HX \rightarrow XH + X$, where X is a halogen atom. We have also successfully applied this technique to cases where X and Y are two unlike halogen atoms, as well as more complex systems such as $X = F$ and $Y = CH_3O$ and C_2H_5O .¹⁸

Consider the reaction $Cl + HCl \rightarrow HCl + Cl$, which we study by photodetaching $ClHCl^-$. Rate constants have been measured for this reaction,¹⁹ and *ab initio* calculations on the neutral potential energy surface have been performed.²⁰ On the basis of these studies, the reaction is believed to proceed along a collinear or nearly collinear minimum energy path with a barrier of ~ 8 kcal/mole. The calculated interchlorine distance at the saddle point is 2.94 Å. The high resolution infrared spectrum of gas phase $ClHCl^-$,²¹ in conjunction with recent *ab initio* calculations²² and earlier matrix isolation studies²³ indicates that the ion is linear and centrosymmetric with an equilibrium interchlorine distance of 3.1122 Å. Photodetachment of the ion, a vertical process, should therefore access the transition state region of the $Cl + HCl$ potential energy surface. There is no kinetic or scattering data available on the $Br + HBr$ reaction, nor have any high level *ab initio* calculations been performed on it. Nonetheless, we expect that photodetachment of $BrHBr^-$ should allow us to probe the transition state region for the $Br + HBr$ reaction.

Our experiment probes the transition state under well-defined conditions. Photodetaching the ClHCl^- ion initiates the reaction on the $\text{Cl} + \text{HCl}$ surface with all three atoms essentially collinear, and the averaging over reactant orientation that occurs in most scattering experiments is largely eliminated. Because the ejected electron is so light, the ion and neutral complex have approximately the same angular momentum. One can therefore limit the total angular momentum of the complex by performing the experiment with rotationally cold ions. This is in sharp contrast to a scattering experiment in which there is no control over reactant impact parameter and thus the total angular momentum.

A key feature of our experiment is the ability to test an interesting theoretical prediction concerning $\text{X} + \text{HX}$ reactions: the existence of long-lived reactive resonance states.²⁴ Collinear quantum reactive scattering calculations on model potential energy surfaces for $\text{X} + \text{HX}$ reactions^{25,26} show sharp resonance structure in the reaction probability as a function of energy. This is superimposed on a smoothly varying background due to direct scattering. The sharp structure is attributed to long-lived vibrationally excited states of the collision complex localized near the transition state. The resonances have widths of about 1 cm^{-1} (0.12 meV), indicating lifetimes in the range of several picoseconds for these states. Three-dimensional calculations on the $\text{Cl} + \text{HCl}$ reaction also show resonance structure in the total reaction cross section,²⁷ although the

resonances are broader (5 meV) and less pronounced relative to the contribution from direct scattering than in the two-dimensional calculations.

With reference to Figure 2-1, one can think of resonances as arising from states of the collision complex which are not only bound along directions perpendicular to the reaction coordinate, but which are also quasi-bound along the reaction coordinate. Hence, these states can lead to sharp structure in the XH_2^- photoelectron spectrum with peak widths ultimately determined by the resonance lifetimes. The possibility of directly observing these states was a major motivating factor in our investigations. Although state-to-state reactive scattering experiments on the $\text{F} + \text{H}_2$ ²⁸ and $\text{H} + \text{H}_2$ ¹³ reactions show effects attributable to such resonances, the exact interpretation of these results is complicated by the large number of partial waves which contribute to the reaction cross section. This can result in direct scattering obscuring any effects due to resonances. Our experiment suppresses much of the contribution from direct scattering and restricts the total angular momentum available to the reaction, thereby enhancing the contribution from resonances. Aside from the interest in confirming the existence of these long-lived states, their observation would provide detailed information on the nature of the transition state region for these reactions.

We have previously reported preliminary results on the ClHCl^- photoelectron spectrum²⁹ taken with an earlier version of our photoelectron spectrometer. Although the resolution was poor (35 meV), the spectra

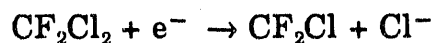
showed resolvable structure which was attributed to the antisymmetric stretch of the ClHCl collision complex. Since then the resolution of the instrument has been improved to 5-8 meV, and results on IHI⁻ were reported³⁰ that strongly suggested the existence of long-lived (>0.1 ps) states of the IHI complex. Here we report the BrHBr⁻ and most recent ClHCl⁻ photoelectron spectra. The BrHBr⁻ spectrum is particularly amenable to detailed analysis, and we have used our spectrum to construct an "effective" collinear potential energy surface for the Br + HBr reaction. The relationship between the potential energy surface and the experimental peak spacings, intensities, and widths is discussed, including the role of reactive resonance states and the possible contribution of transitions to electronically excited Br + HBr potential energy surfaces.

2. Experimental

The negative ion time-of-flight photoelectron spectrometer³¹ used in these studies is shown in Figure 2-2. A gas mixture at a backing pressure of approximately 2 atm is expanded through a pulsed molecular beam valve (1) operating at 20 Hz. A 1 keV, 300 μ A electron beam (2) crosses the gas jet just outside the nozzle, in the continuum flow region of the expansion. The fast electrons produce relatively low-energy secondary electrons via ionization, and these secondary electrons rapidly thermalize due to the high

gas density in this region. Negative ions formed through low-energy electron attachment processes should cool internally as the expansion progresses. We have measured a vibrational temperature of ~150 K for CH_2CN^- in this source and a rotational temperature in the range of 50-75 K for SH^- produced in a similar source.³²

A mixture of 5% HBr in Ar was used to make BrHBr^- , presumably via dissociative electron attachment to HBr to form Br^- and subsequent clustering. In the ClHCl^- experiments, a mixture of 5% HCl seeded in He did not produce sufficient quantities of ions, probably due to the small cross section for dissociative attachment to HCl at low electron energy. Addition of CF_2Cl_2 resulted in a great improvement in ion signal. A mixture of 5% HCl and 25% CF_2Cl_2 seeded in He was used for the experiments. The likely mechanism for ClHCl^- formation is:



The pressure in the source chamber, which is pumped with an Edwards 250M 10 in. diffusion pump, is typically 2×10^{-5} to 1×10^{-4} torr.

The ions are mass selected with a Wiley-McLaren type time-of-flight mass spectrometer.³³ The ion source (including the pulsed valve, electron gun and extraction plates) is floated at -1000 V with respect to ground. The ions are extracted (3) with a 10 μs , 100 V/cm pulse, through a 3 mm orifice into the first differential chamber, which is maintained at a pressure of 2 x

10^{-7} torr by a 6 in. diffusion pump (Edwards 160M). The ions are then accelerated by 1 keV (to ground potential), pass through a second differentially pumped region, maintained at 2×10^{-8} torr by a 150 ℓ/s turbomolecular pump (Leybold-Heraeus) and enter the detection region. This region is also pumped by a 150 ℓ/s turbomolecular pump and is maintained below 1.0×10^{-8} torr. The ions are detected (5) 1.4 m from the extraction region, using a pair of 25 mm diameter chevron-mounted microchannel plates (Galileo). Ion signal at the detector is maximized using two sets of horizontal and vertical deflectors and an Einzel lens (4). The mass resolution of the spectrometer is 250-300. The ions studied in this work are $^{35}\text{ClH}^{35}\text{Cl}^-$ and $^{79}\text{BrH}^{79}\text{Br}^-$ and the deuterated species.

The mass-selected ions are detached (6) 10 cm in front of the ion detector using a pulsed laser. A grid (7) in front of the ion detector can be set at -1400 V to allow only neutrals created by photodetachment to pass. The laser firing delay is varied until optimal temporal overlap is achieved with the ion of interest, as monitored by observing the neutral signal with the grid voltage applied. The fifth harmonic (213 nm, 5.825 eV) of a Nd:YAG laser (Quanta-Ray DCR-3, 20 Hz rep. rate) is used in the photodetachment studies reported here. This is generated by mixing the YAG laser fundamental with the fourth harmonic in a β -barium borate crystal (CSK Corporation). Typical pulse energies are 10-12 mJ. The laser beam is gently focused with a 200 cm lens to a 0.2 cm diameter spot at the

interaction region. The laser beam enters the machine through a MgF_2 window, passes through a series of light baffles, intersects the ion beam at right angles, passes through another set of light baffles, and exits through a quartz window.

The energy of the detached electrons is determined by time of flight. Photoelectrons are detected (8) at the end of a 1 m flight tube orthogonal to the laser and ion beams. The flight tube is magnetically shielded by two concentric cylinders (13 cm and 18 cm diameter) of Hypernom (9). The innermost surface of the shielding is coated with colloidal graphite (to eliminate patch potentials) and degaussed (to limit residual magnetic fields to 1 - 5 milligauss). Electrons are detected by a chevron-mounted pair of 40 mm diameter microchannel plates. The signal is amplified and fed through a discriminator into a 200 MHz transient digitizer (LeCroy TR8828C), which is started by a photodiode triggered by a reflection from the laser. A signal averager (LeCroy 6010) reads the digitizer every laser shot, and the averager is read by an IBM AT computer every 250 shots. The entire timing sequence in the experiment is controlled by a Stanford Research Systems DG535 pulse generator. The electron detector subtends a solid angle of 0.0013 steradians; 0.01% of the detached photoelectrons are detected.

Ultraviolet photons efficiently eject electrons from metal surfaces, so great care is taken to minimize background photoelectrons produced by

stray photons striking metal surfaces in the detector region. The stray photons are largely eliminated by the light baffles between the interaction region and the entrance and exit windows. Additional discrimination against background photoelectrons is provided by four plates (10) (two round plates of 11 cm outer diameter with a 7.5 cm inner diameter hole, alternated with two 7.5 cm square plates with a 2 cm inner diameter hole) placed 5 cm apart, starting 28 cm up the flight tube. With these precautions, the residual background photoelectron contribution is one electron per laser shot at 213 nm. As the kinetic energy distribution of the background photoelectrons is smooth and does not change from day to day, a smooth function is fitted to the background, scaled, and subtracted from the spectrum.

Under typical conditions, the ion density in the laser interaction region ($\sim 0.01 \text{ cm}^3$ volume) is about $10^5/\text{cm}^3$ for the ion of interest. At higher densities, the photoelectron spectrum shifts and broadens due to space charge. Between 10% and 50% of the ions are photodetached, and about one electron is detected per laser shot. A typical spectrum takes 300,000 laser shots. Photoelectron electron flight times (t) are converted to electron kinetic energy (eKE) using

$$eKE = \frac{lm_e}{2(t-t_0)^2} \quad (1)$$

The time offset (t_0) and flight tube length (l) are determined by calibrating

with F^- , Cl^- , Br^- , and I^- , all of which have known electron binding energies.

We have determined the resolution of the spectrometer to be

$$R(\text{meV}) = \sqrt{25 + 180(eKE)^3} \quad (2)$$

where eKE is in eV. The best resolution obtained is ~ 5 meV at low energy, and the 11.5 ns combined time-width of the laser and electronics causes the resolution to degrade at higher electron energy.

3. Results and Preliminary Discussion

The 213 nm photoelectron spectra of $ClHCl^-$ and $ClDCl^-$ are shown in Figure 2-3. These spectra show partially resolved peaks and are similar to the spectra reported earlier²⁹ at considerably lower (35 meV) resolution. In contrast, the 213 nm photoelectron spectra of $BrHBr^-$ and $BrDBr^-$ (Fig. 2-4) show a series of well resolved peaks with widely varying widths. The peak positions and widths are given in Table 2-1. The positions of the highest energy peaks in the $ClHCl^-$ and $BrHBr^-$ spectra are unchanged by deuteration, whereas the remaining peaks shift to higher electron energy. The peak spacings and intensities for the $ClHCl^-$ and $ClDCl^-$ spectra are the same as those previously observed at 193 nm. This shows the spectra result from direct photodetachment to the (neutral + e^-) continuum rather than autodetachment from an excited electronic state of the ion.

We now consider some general features of these results. The first question is whether the peaks in these spectra correspond to transitions to levels of the neutral XHX complex which are bound or unbound with respect to $X + HX$ ($v = 0$). This can be determined with the aid of Figure 2-5, which is drawn for the $BrHBr^-$ system. This figure shows that peaks with electron kinetic energy *less* than $E_b = hv - D_0(BrHBr^-) - EA(Br)$ result from levels of the neutral complex that lie *above* $Br + HBr$ ($v = 0$). Here, hv is the photon energy (5.825 eV), $D_0(BrHBr^-)$ is the energy needed for the dissociation reaction $BrHBr^- \rightarrow Br^- + HBr$, and $EA(Br)$ is the electron affinity of Br. $D_0(BrHBr^-)$ has been determined by high pressure mass spectroscopy to be 0.91 ± 0.05 eV,³⁴ and with $EA(Br) = 3.365$ eV,³⁵ we find $E_b = 1.55 \pm 0.05$ eV. For $ClHCl^-$, $D_0(ClHCl^-) = 1.02 \pm 0.05$ eV,³⁴ $EA(Cl) = 3.617$ eV,³⁵ and $E_b = 1.19 \pm 0.05$ eV. The peak at highest electron kinetic energy appears at 1.353 eV for $BrHBr^-$ and 0.894 eV for $ClHCl^-$, so *all* the observed peaks for both systems result from transitions to unbound states of the neutral complex. This suggests that the potential energy surfaces for the $Cl + HCl$ and $Br + HBr$ reactions have barriers but no wells near the transition state. Although there has been considerable discussion of "vibrationally bound states"³⁶ of IHI and (less so) for $BrHBr$, this would require peaks with electron kinetic energy above E_b . Our results here and those presented earlier³⁰ for IHI^- show no evidence for this.

We next consider the isotope effects observed in the spectra. The

peak shifts show we are observing a vibrational progression in a mode involving H atom motion, and one must determine if this is a progression in the ion ("hot band" transitions) or the neutral. As long as the active vibrational mode is not the dissociation coordinate of the complex (see next section), the electron kinetic energy from a neutral \leftarrow ion transition is given by

$$E = h\nu - [D_0(\text{XHX}^-) + EA(\text{X}) + \Delta E] - E_{v',j'}^{(0)} + E_{v'',j''}^{(-)} \quad (3)$$

Here ΔE is the vibrational zero point and electronic energy of the XHX complex relative to $\text{X} + \text{HX}$ ($v = 0$). For example, $\Delta E = 4.47 - 4.27 = 0.20$ eV for BrHBr (see Figure 2-5). $E_{v',j'}^{(-)}$ and $E_{v'',j''}^{(0)}$ are the rotational and vibrational energy in the active mode for the ion and neutral, respectively. Assuming similar zero-point energies for the ion and neutral, eq. 3 predicts that the $(v' = 0) \leftarrow (v'' = 0)$ transition should not shift upon deuteration, $(v' > 0) \leftarrow (v'' = 0)$ transitions to excited vibrational levels of the neutral should shift to *higher* electron kinetic energies, and "hot band" $(v' = 0) \leftarrow (v'' > 0)$ transitions should shift to *lower* electron energies. Here v'' and v' are vibrational quantum numbers for the active mode in the ion and neutral, respectively. The highest energy peak in the experimental spectrum does not shift upon deuteration, so this peak is the $0 \leftarrow 0$ transition, and the spectra represent a $(v' = n) \leftarrow (v'' = 0)$ progression in a vibrational mode of the *neutral complex*. None of the peaks appear to arise from hot bands.

The active mode could be either the bend (ν_2) or antisymmetric

stretch (ν_3) of the complex. Since the ions are expected to be linear and the minimum energy paths for the reactions are likely to be nearly collinear,²⁰ an extended progression in the bend is unlikely, and we assign the peaks to a progression in the ν_3 mode. Symmetry considerations show that only transitions to even ν_3' levels of the complex are allowed from the $\nu_3'' = 0$ level of the ion.³⁷ The peaks in each spectrum are labelled by their ν_3' quantum number in Table 2-1. As discussed in more detail below, the two broad peaks at lowest energy in the BrHBr^- and BrDBr^- spectra (A' and B') do not appear to belong to the same progression as the higher energy peaks.

The energies of the ν_3' states for the BrHBr complex and the $\text{Br} + \text{HBr}$ asymptotic vibrational energy levels are shown in Figure 2-5. The vibrational spacing in the BrHBr complex is significantly smaller than that in free HBr . The intuitive explanation for this is that the strong HBr bond in free HBr is replaced by two much weaker bonds in the BrHBr complex. This observation is therefore a strong indication that we are probing the transition state region of the potential energy surface where the hydrogen is interacting strongly with both bromine atoms.

4. Analysis and Discussion

4.1. General Considerations

The analysis of the vibrational structure in these spectra centers on

the calculation of Franck-Condon factors for transitions between the ion and the neutral potential energy surface. The goal is to (a) understand the origin of the observed structure in the spectra and (b) use our results to learn about the transition state region of the potential energy surfaces for the Cl + HCl and Br + HBr reactions. The ClHCl⁻/Cl + HCl system is the better characterized of the two. Several potential energy surfaces have been proposed for the Cl + HCl reaction,^{20,25,38} the ion geometry has been experimentally determined,²¹ and simulations of the ClHCl⁻ photoelectron spectrum have been calculated using some of the model surfaces for the reaction.^{39,40} These results will be used to explain the qualitative features of our spectra and to test the validity of the approximations used in our analysis. However, the peaks in the BrHBr⁻ spectra are better resolved, so we are able to extract considerably more information on the Br + HBr reaction.

Our analysis requires knowledge of the geometry and potential functions for the XHX⁻ ions. The ClHCl⁻ results discussed in the Introduction indicate that the ion is linear and centrosymmetric with an interhalogen distance $R_e = 3.1122 \text{ \AA}$.²¹ A MRD-CI calculation⁴¹ on BrHBr⁻ yields a linear, centrosymmetric structure with $R_e = 3.43 \text{ \AA}$. We assumed $R_e = 3.50 \text{ \AA}$ in our analysis; this is within the error of the calculation and greatly facilitated fitting our results with a reasonable potential energy surface (see below).

The potential functions for the ions are inferred from the measured vibrational frequencies. Table 2-2 lists the gas-phase frequencies²¹ for the three vibrational modes of ClHCl⁻ and matrix isolation values for the symmetric (ν_1) and antisymmetric (ν_3) stretch frequencies in ClHCl⁻, ClDCl⁻, BrHBr⁻, and BrDBr⁻.⁴² The similarity between the gas phase and matrix frequencies for ClHCl⁻ (and for FHF⁻⁴³) suggests that the matrix values for BrHBr⁻ and BrDBr⁻ can be used in our analysis, since the gas-phase values are unknown. For both ClHCl⁻ and BrHBr⁻, the matrix results for the antisymmetric stretch yield $\nu_H/\nu_D > 1.40$, the value expected for a harmonic oscillator. This "negative anharmonicity" implies that the potential function for the ν_3 mode is approximately harmonic with a small quartic contribution.^{23c} In our analysis, all the transitions in the spectra are assumed to originate from the ground vibrational state of the ion for which anharmonic effects should be small. We therefore neglected the quartic term and used harmonic oscillator wavefunctions with the frequencies given in Table 2-2 for the vibrational modes in all the ions.

As a first step in analyzing our results, consider the collinear London-Eyring-Polanyi-Sato (LEPS) potential energy surface⁴⁴ for the reaction $\text{Cl} + \text{HCl} \rightarrow \text{HCl} + \text{Cl}$ shown in Figure 2-6. This surface, which has a barrier of 36 kJ/mol, was constructed by Bondi et al.²⁵ on the basis of rate constant measurements by Kneba and Wolfrum.¹⁹ The surface (henceforth referred to as the BCMR LEPS) is plotted using modified hyperspherical coordinates

ρ and z ,⁴⁵ which, for the heavy + light-heavy mass combination, simplify to

$$\rho \approx \sqrt{\frac{m_{Cl}}{2m_H} + \frac{1}{2}} R_{Cl-Cl'} = 4.22 R_{Cl-Cl'} \quad (4)$$

$$z \approx \frac{1}{2}(R_{Cl-H} - R_{H-Cl'})$$

ρ and z represent the size and asymmetry, respectively, of the Cl + HCl system. The coordinate ρ is proportional to the symmetric stretch symmetry coordinate in ClHCl⁻, and $z \approx Q_3/2^{1/2}$, where Q_3 is the antisymmetric stretch symmetry coordinate. The skew angle in the figure is given by

$$\phi_{\max} = \tan^{-1} \left(\sqrt{\frac{m_H(m_H + 2m_{Cl})}{m_{Cl}^2}} \right) = 13.6^\circ \quad (5)$$

where $m_H = 1.008$ amu and $m_{Cl} = 34.969$ amu.

The shaded region in Figure 2-6 indicates the area on the neutral surface that is directly probed by our photodetachment experiment. The center of the shaded region, at $z = 0$, $\rho = 13.15 \text{ \AA}$, corresponds to R_e for the ion. The extent of the shaded region represents the zero point amplitude for the symmetric stretch in the ground vibrational state of the ion. Our spectra are very sensitive to the details of this "Franck-Condon region" of the potential energy surface.

Figure 2-6 shows that the Franck-Condon region does not include the saddle point, which is not a serious limitation. Calculations by Bondi et.

al.^{46a} and Hartke and Manz^{46b} indicate that very few reactive trajectories actually cross the saddle point because of the acute skew angle for a heavy + light-heavy reaction such as Cl + HCl. At energies below the barrier height, the reaction rate is dominated by tunneling through the ridge that bisects the shaded region in the figure. At energies above the barrier height, most reactive trajectories cross from the reactant to product valley before reaching the saddle point. Thus, our experiment probes the "chemically relevant" part of this potential energy surface.

Figure 2-6 illustrates why a progression in the antisymmetric stretch is observed, even though the complex can dissociate. Near the center of the shaded region, motion along the z coordinate corresponds to the antisymmetric stretch of the *neutral complex*. This motion is nearly perpendicular to the minimum energy path which leads to dissociation via the reactant or product valleys. The antisymmetric stretch of the complex is therefore poorly coupled to the reaction coordinate in the region of the potential energy surface probed by our experiment. This is another consequence of the small skew angle for the collinear surface. Even though photodetachment of ClHCl⁻ produces a complex that dissociates, we observe a progression in a vibrational mode not strongly coupled to the dissociation coordinate. Similar effects have been proposed to explain vibrational structure in electronic absorption to dissociative states of neutral molecules.⁴⁷

The discussion so far suggests that we should be able to explain the experimental peak positions and intensities with a one-dimensional model in which only the antisymmetric stretch of the complex is considered. This is discussed in more detail in the following section.

4.2. One-Dimensional Approach

As discussed above, the peaks in the spectra originate from the $\nu_3'' = 0$ level of the anion, and this is assumed to be described by a harmonic oscillator wavefunction. For the LEPS surface in Figure 2-6, one can obtain an approximate potential function for the ν_3' mode in the neutral complex by taking a cut at constant ρ through the center of the shaded region. The resulting double minimum potential is shown in Figure 2-7 (dashed line), above the harmonic oscillator potential assumed for the ion (dotted line). The neutral \leftarrow ion transition involves going from a single to double minimum potential for the antisymmetric stretch; this is why the ν_3 mode is active in our spectra.

We numerically solve for the eigenvalues and eigenfunctions supported by this potential. A simulated "one-dimensional" photoelectron spectrum is obtained by calculating the Franck-Condon overlap between the $\nu_3'' = 0$ wavefunction of the ion and the ν_3' wavefunctions supported by the double minimum potential.⁴⁸ This overlap is zero for all odd ν_3' . The

simulated ClHCl⁻ and ClDCl⁻ spectra from the LEPS surface are shown in Figure 2-7 superimposed on the experimental spectra. The peak spacings and intensities in the stick spectra show some resemblance to the broad peaks in the experimental spectra. In particular, the higher $v_3' = 0/v_3' = 2$ ratio in the experimental ClHCl⁻ spectrum relative to the ClDCl⁻ spectrum appears in the simulation. This is a consequence of the lower energy $v_3' = 0$ wavefunction for the ClDCl complex having less probability near $z = 0$ and hence poorer overlap with the $v_3'' = 0$ wavefunction for the ion.

However, the peak spacing in the simulated spectra is too large, and the overall intensity distribution is incorrect. As a first step towards determining the true potential energy surfaces for the Cl + HCl and Br + HBr reactions, we have constructed empirical double minimum potentials with adjustable parameters to better reproduce the experimental results for these systems. These are functions of the antisymmetric stretch symmetry coordinate $Q_3 = x = (R_{Cl-H} - R_{H-Cl})/2^{1/2}$:

$$V(x) = b + \frac{1}{2}k(x - x_1)^2 + q(x - x_1)^4 \quad \text{for } x > x_1 \text{ and} \quad (6)$$

$$V(x) = b + h - d(x - x_1)^2 + e(x - x_1)^4 + f(x - x_1)^6 \quad \text{for } x \leq x_1$$

Equation 6 is for $x \geq 0$. For $x < 0$, $V(x) = V(-x)$. The complete potential function has two minima at $x = \pm x_1$, separated by a barrier of height h at $x = 0$. The parameters k and q determine the steepness of the outer walls. The other (dependent) parameters are given by $d = \kappa/8 + ex_1^2$, $e = 3h/x_1^4$ -

$k/4x_1^2$, and $f = (k - 8ex_1^2)/24x_1^2$. These conditions ensure that the potential and its first and second derivatives are continuous.

The potential of this form which best fits the ClHCl^- and ClDCl^- results is shown in Figure 2-7, superimposed on the double minimum potential derived from the LEPS surface. The one-dimensional simulations obtained from this potential are also shown in Figure 2-7. The parameters for the optimal double minimum potential are given in Table 2-3. Although the barrier is slightly higher in our potential, the biggest difference is that the outer walls are too steep and the separation between the minima is too small in the LEPS potential. The simulated peak spacings from a LEPS surface therefore tend to be too large, and the simulated $v_3' = 0$ peak is too intense. These discrepancies with experiment were also seen in the $\text{IHI}^-/\text{I} + \text{HI}$ system when the spectrum was simulated with a LEPS surface.³⁰

The positions and intensities of the three highest energy peaks in the BrHBr^- spectrum and the first four peaks in the BrDBr^- spectrum were simulated using the double minimum potential in the lower half of Figure 2-8. The simulated spectrum from this potential is shown with dotted lines in Figure 2-8. Because of the large variation in the experimental peak widths, the width of each simulated peak is set equal to the experimental width to facilitate comparison with the experimental intensities. The area under each simulated peak would give the relative height in a simulated stick spectrum.

The two broad low-energy peaks (A' and B') present more of a problem. These peaks do not seem to belong to the same vibrational progression as the higher energy peaks. It is not clear how a bimodal intensity distribution could arise if all the peaks were due to the same progression. In addition, the spacing between the $\nu_3' = 4$ peak and peak A' in the BrHBr^- spectrum is nearly 0.35 eV, an anomalously large value which is greater than the HBr fundamental frequency. A possible explanation is that peaks A' and B' are part of a second vibrational progression arising from a low-lying excited electronic state of the complex. The large isotope shift of peak A' indicates that it is not the origin of this progression. However, adding the A'-B' splitting to the energy of peak A' gives a value of 0.82 eV in the BrHBr^- spectrum and 0.84 eV in the BrDBr^- spectrum. This suggests that peaks A' and B' are the second and third peaks in a vibrational progression for which the origin is not clearly observed.

The entire spectrum for each isotope can be simulated by assuming peaks A' and B' result from transitions to the $\nu_3' = 2$ and $\nu_3' = 4$ levels of the upper double minimum potential shown in Figure 2-8. The simulated spectrum from this potential is indicated with a dashed line in Figure 2-8. The high barrier between the wells makes the $0 \leftarrow 0$ transition very weak. Note that the simulation suggests that the broad feature under the narrow $\nu_3' = 6$ peak in the BrDBr^- spectrum is the $0 \leftarrow 0$ transition in the second

progression.

The $v_3' = 0$ peaks of the two progressions in Figure 2-8 are separated by 0.49 eV. This is slightly larger than the $^2P_{1/2} - ^2P_{3/2}$ spin-orbit splitting in the Br atom (0.45 eV) and suggests that the second progression is due to an electronically excited state of the complex which asymptotically correlates to $\text{Br}^*(^2P_{1/2}) + \text{HBr}$. Based on diatomics-in-molecules (DIM) calculations⁴⁹ on $\text{F} + \text{HF}$ and $\text{Cl} + \text{HCl}$, this excited state interaction is expected to be considerably more repulsive than the ground state. The high barrier between the wells in Figure 2-8 is consistent with a repulsive interaction. We have observed other examples of transitions to spin-orbit excited surfaces in IHI^- and several asymmetric bihalides¹⁸ which will be reported in detail in the near future. Based on these considerations, it is quite possible that the ClHCl^- and ClDCl^- photoelectron spectra also contain transitions to electronically excited states of the neutral complex. These transitions could give rise broad peaks that interfere with the ground state spectra.

The one-dimensional analysis outlined here provides a useful first-order explanation of our experimental results. The reasonable agreement between experimental and simulated peak spacings and intensities supports the assignment of the structure in our spectra to a progression in the antisymmetric stretch of the complex. However, this analysis does *not* explain the dramatic variation in the peak widths observed in the BrHBr^-

and BrDBr^- spectra. The peak widths are sensitive to the dynamics of the complex *along* the reaction coordinate. The analysis of the peak widths requires a more sophisticated, two-dimensional method which is discussed in detail in the next section.

4.3. Two-Dimensional Analysis

A rigorous simulation of our results requires calculation of the Franck-Condon overlap between the bound ionic vibrational states and the full scattering wavefunctions supported by the three-dimensional reactive potential energy surface. Schatz³⁹ has recently simulated parts of the ClHCl^- and IHI^- photoelectron spectra using three-dimensional scattering wavefunctions with the restriction that the ion is in its ground vibrational state and that total angular momentum $J = 0$ for the ion and the final scattering states. Even with these restrictions on the ion and total angular momentum, the calculations are quite involved and have only been performed on model LEPS surfaces for the $\text{Cl}+\text{HCl}$ and $\text{I}+\text{HI}$ reactions. Gazdy and Bowman^{40b} have also calculated three-dimensional ($J = 0$) simulations of these photoelectron spectra using an L^2 basis set.

Our goal is to use the experimental results to construct better potential energy surfaces for these reactions. We use an iterative approach, since the experimental results cannot be directly inverted to give the

potential energy surface. A flexible functional form is chosen for the potential energy surface, and the parameters are adjusted until simulations reproduce experimental results. This procedure would be far too time-consuming if a full three-dimensional simulation were performed for each iteration. We have therefore used an approximate scheme which is essentially a two-dimensional simulation. The aim here is to use our calculation to devise a surface which reproduces the experimental spectra. Once this is determined, a more sophisticated simulation could be performed to check the accuracy of the resulting potential energy surface.

Our simulations incorporate two significant approximations concerning the scattering wavefunction on the neutral surface. First, the three-dimensional potential energy function $V(\rho, z, \gamma)$ (where ρ and z were defined previously and γ is the X-H-X bending angle) is reduced to an "effective" collinear potential energy surface $V(\rho, z)$ via

$$V(\rho, z) = V(\rho, z, \gamma = \pi) + \epsilon_0(\rho, z) \quad (7)$$

Here $\epsilon_0(\rho, z)$ is the bending zero-point energy for the linear configuration of the nuclei specified by ρ and z . Equation 7 results from the adiabatic bend theory developed by Bowman⁵⁰ as part of his reduced dimensionality model of reactive scattering. This assumes that the bending mode is separable from the other degrees of freedom of the system. Equation 7 includes the change in bending zero-point energy with the size and asymmetry of the complex and therefore incorporates some effects of the bending mode into

our analysis. From the point of view of our experiment, we are assuming that only transitions to the lowest bending level of the neutral complex occur. This theory should be reasonably accurate if the reaction is collinearly dominated, so that photodetachment of a linear ion should not produce substantial bending excitation in the neutral complex. One can also construct effective collinear surfaces for bend-excited states ($v_2' = n$) of the complex by substituting $\epsilon_n(\rho, z)$ for $\epsilon_0(\rho, z)$ in eq. 7.

We now have to find the scattering wavefunction $\psi(\rho, z)$ supported by the effective collinear potential $V(\rho, z)$. In these coordinates, the Schrodinger equation is

$$\left[-\frac{1}{2m} \left(\frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{\partial^2}{\partial z^2} \right) + V(\rho, z) \right] \psi(\rho, z) = E \psi(\rho, z) \quad (8)$$

where m is the reduced mass of HX. We now invoke our second approximation, an adiabatic approximation which has been used by several investigators^{25,36,45} in studies of collinear heavy + light-heavy reactions. We assume the scattering wavefunction can be written as a product of two wavefunctions:

$$\psi(\rho, z) = \theta_{v_3}(z; \rho) R_{v_3}(\rho) \quad (9)$$

This is analogous to the Born-Oppenheimer approximation in diatomic molecules. $\theta_{v_3}(z; \rho)$ is the wavefunction for the fast antisymmetric stretch vibration along the z direction, and depends weakly on ρ . $R_{v_3}(\rho)$ is either a symmetric stretch or translational wavefunction which primarily involves

relatively slow heavy atom motion. θ_{v_3} and R_{v_3} satisfy the one-dimensional equations

$$\left[-\frac{1}{2m} \frac{\partial^2}{\partial z^2} + V(\rho, z) \right] \theta_{v_3}(z; \rho) = \epsilon_{v_3}(\rho) \theta_{v_3}(z; \rho) \quad (10a)$$

and

$$\left[-\frac{1}{2m} \frac{\partial^2}{\partial \rho^2} + U_{v_3}(\rho) \right] R_{v_3}(\rho) = ER_{v_3}(\rho) \quad (10b)$$

where

$$U_{v_3}(\rho) = \epsilon_{v_3}(\rho) - \frac{1}{8m\rho^2} - \frac{1}{2m} Q_{v_3, v_3} \quad (11a)$$

and

$$Q_{v_3, v_3} = \left\langle \theta_{v_3}(\phi; \rho) \left| \frac{\partial^2}{\partial \rho^2} \theta_{v_3}(\phi; \rho) \right. \right\rangle \quad (11b)$$

or, using finite differences,²⁵

$$Q_{v_3, v_3} = \frac{2}{(\Delta\rho)^2} \left[\left\langle \theta_{v_3}(\phi; \rho) \left| \theta_{v_3}(\phi; \rho + \Delta\rho) \right. \right\rangle - 1 \right] \quad (11c)$$

where $\phi = z/\rho + \phi_{\max}/2$ and brackets indicate integration over ϕ .

Equation 10a is essentially the same one-dimensional Schrodinger equation that was solved in the last section to obtain the antisymmetric stretch eigenvalues and eigenfunctions for the neutral complex. By solving eq. 10a at many values of ρ , one generates a set of curves $\epsilon_{v_3}(\rho)$ which map out the energy of the n th antisymmetric stretch level as a function of the

size of the complex, and a set of antisymmetric stretch eigenfunctions $\theta_{v_3}(z;\rho)$ which depend parametrically on ρ . For a symmetric X + HX system, one finds two adiabatic curves which correlate to each X + HX ($v = m$) asymptotic state; these curves have $v_3 = 2m$ and $v_3 = 2m+1$ and correspond to wavefunctions θ_{2m} and θ_{2m+1} which are of even and odd parity, respectively.

Equation 10b for $R_{v_3}(\rho)$ is a one-dimensional Schrodinger equation for a particle with total energy E moving in an effective or adiabatic potential $U_{v_3}(\rho)$. The adiabatic potential is determined by the solutions to eq. 10a according to eq. 11a and 11b. The DIVAH correction⁵¹ $Q_{v_3 v_3}$ (eq. 11b) is very small for a heavy + light-heavy reaction.

Figure 2-9 shows the first few adiabatic curves for the Cl + HCl reaction, obtained from the effective collinear surface derived from the three-dimensional BCMR LEPS surface.²⁵ These curves are similar to those in ref. 25, but they also include the bending zero-point energy. For a purely repulsive adiabatic potential such as $U_0(\rho)$ in Figure 2-9, the solutions $R_0(\rho)$ are one-dimensional continuum scattering wavefunctions. The $U_2(\rho)$ and $U_4(\rho)$ potentials each have a well, separated by a barrier from the Cl + HCl ($v=1$) and Cl + HCl ($v=2$) asymptotes, respectively. In addition to supporting continuum states, these potentials support long-lived *symmetric stretch* levels which lead to the sharp resonance structure seen in collinear reactive scattering calculations on this surface. The U_2 potential supports a

narrow "shape resonance", indicated with a dotted line in Figure 2-9. This is a long-lived state of the complex which can decay either by tunneling through the barrier on the U_2 potential to $\text{Cl} + \text{HCl}$ ($v=1$) or by undergoing vibrational predissociation to form $\text{Cl} + \text{HCl}$ ($v=0$). The U_4 potential supports several shape resonances as well as a state indicated with a solid line which lies below the $\text{Cl} + \text{HCl}$ ($v=2$) asymptote. This is a "Feshbach resonance" which can decay only by vibrational predissociation.

In our adiabatic approximation, coupling between the potentials $U_{v_3}(\rho)$ is neglected and vibrational predissociation is not allowed. Thus, Feshbach resonances cannot decay and shape resonances can decay only by tunneling. Since collinear and three-dimensional scattering calculations show that these resonances are narrower than our experimental resolution, we treat both types of resonances as bound states with widths determined by the experimental resolution. The resonance wavefunctions are assigned a symmetric stretch quantum number v_1 based on the number of nodes inside the well in the adiabatic potential.

The adiabatic potentials are instrumental in simulating the negative ion photoelectron spectrum. We calculate the Franck-Condon overlap between the initial state of the ion and the bound and continuum states supported by the adiabatic potentials $U_{v_3}(\rho)$. The intensity of a transition from an ion in its (v_1'' , v_3'') vibrational level to a resonance state with symmetric stretch wavefunction $R_{v_1 v_3}(\rho)$ is given by

$$I \propto \left| \int d\rho \left[\int dz \theta_{v_3}(z; \rho) \psi_{v_3}^-(z) \right] R_{v_1, v_3}(\rho) \psi_{v_1}^- \right|^2 \quad (12)$$

where $\psi_{v_1}^-$ and $\psi_{v_3}^-(z)$ are harmonic oscillator wavefunctions for the symmetric and antisymmetric stretch, respectively, in the ion. Such a transition yields a sharp peak in the simulated spectrum. A broad peak can arise from transitions to continuum states supported by an adiabatic potential. For example, transitions to the repulsive $U_0(\rho)$ potential in Fig. 2-9 (i.e. transitions to the $v_3 = 0$ level of [ClHCl]) will yield a broad peak with a width determined, to first order, by the slope of the adiabatic potential in the Franck-Condon region.

The simulation of the experimental spectrum requires solving eq. 10 and 11, and calculation of Franck-Condon factors. Equation 10a is solved to obtain $\theta_{v_3}(z; \rho)$ at 60 values of ρ , using a relaxation algorithm⁵² (see Chapter 3.3) that takes advantage of the observation that $\theta_{v_3}(z; \rho)$ varies slowly with ρ . If the adiabatic potential supports resonances, wavefunctions are found variationally.⁵³ Two-dimensional overlap integrals (12) are calculated for each state. For the repulsive region of each adiabatic potential, continuum wavefunctions and Franck-Condon factors are determined at an energy spacing of 1 to 4 meV, depending on the steepness of the adiabatic potential. The continuum Franck-Condon factors are energy-normalized by forcing the asymptotic amplitude of the outgoing wavefunction $R_{v_3}(\rho)$ to be⁵⁴

$$P = 0.27844 \left(\frac{m}{E} \right)^{\frac{1}{4}} \quad (13)$$

where m is as in eq. 8, E is the energy above the asymptote, and P has units $(\text{cm}^{-1} \text{ \AA})^{-1/2}$. Calculation of Franck-Condon factors as a function of energy⁵⁵ then gives Franck-Condon factors in units of $(\text{cm}^{-1})^{-1}$.

Multiplication by the instrumental resolution (in cm^{-1}) gives a dimensionless intensity that can be compared directly with that derived for a bound state.

4.4. Application to Cl + HCl

We have simulated the ClHCl^- photoelectron spectra with our "effective" collinear approach using the ionic equilibrium distance and vibrational frequencies in Table 2-2 and the BCMR LEPS surface. The ions are assumed to be in their ground vibrational state. Figure 2-10 shows the simulated ClHCl^- spectrum, convoluted with our experimental resolution (eq. 2). The dashed lines are the results for $v_2' = 0$ in the neutral complex. The dotted lines represent the contribution from $v_2' = 2$, the lowest bend-excited state of the complex that has non-zero overlap with the ion ground state. The bending corrections in eq. 7, $\epsilon_0(\rho, z)$ and $\epsilon_2(\rho, z)$, are calculated in a harmonic approximation.

The $v_2' = 0$ simulation shows a broad (fwhm = 35 meV) peak at high electron energy due to overlap of the ion vibrational wavefunction with the

continuum wavefunctions supported by the repulsive $U_0(\rho)$ adiabatic curve. The narrow peak at 0.74 eV is due to a transition to the long-lived shape resonance supported by the $v_3' = 2$ adiabatic curve. The $v_3' = 4$ adiabatic curve supports a Feshbach resonance (at 0.489 eV) and two shape resonances (at 0.452 and 0.424 eV). This simulation is essentially identical to one obtained by Bowman⁵⁶ who calculated the Franck-Condon overlap with the exact two-dimensional scattering wavefunctions supported by the same effective collinear surface, again assuming $v_2' = 0$. The considerably smaller contribution from $v_2' = 2$ mimics the $v_2' = 0$ simulation with all the peaks shifted to slightly lower electron kinetic energy.

We can further test our analysis by comparing our results to the full three-dimensional simulations (with $J=0$) recently reported by Schatz.³⁹ In his calculation, Schatz simulated the ClHCl^- photoelectron spectrum between 1.1 and 0.6 eV electron kinetic energy using the BCMR LEPS surface. Our peak positions in this energy range, which includes the broad $v_3' = 0$ peak and the narrow $v_3' = 2$ peaks, agree well with Schatz' calculation. The broad peak is slightly wider (50 meV) in Schatz' simulation. The width of the resonance peaks in both simulations is determined by the experimental resolution. The $v_3' = 0$ peak in our calculation is noticeably larger relative to the resonance peaks. Schatz calculated the intensity of the $v_3' = 0$ peak at only two energies, and with a finer energy grid the appearance of this peak in his simulation might be

different. The three-dimensional simulations by Gazdy and Bowman^{40b} do in fact show a more intense $v_3' = 0$ peak. These comparisons suggest that our simple model yields reasonable results for a three-dimensional potential energy surface with a collinear minimum energy path.

All of the above methods of simulation yield peaks which are much narrower and somewhat more widely spaced than the experimental peaks. Based on our experience in fitting the BrHBr^- and BrDBr^- spectra (see Section 4.5.3 below), these discrepancies can be explained in terms of possible deficiencies in the $\text{Cl} + \text{HCl}$ model potential energy surface. We point out, however, that even the three-dimensional calculations are restricted to $J = 0$ and do not include the possibility of transitions to low-lying electronic states of $[\text{ClHCl}]$; these restrictions may lead to overly narrow peaks in the simulated spectrum.

4.5. Br + HBr Potential Energy Surface

We wish to construct an "effective" collinear potential energy surface for the $\text{Br} + \text{HBr}$ reaction and reproduce the experimental photoelectron spectra. This section discusses the connection between the experimental spectra and qualitative features of the surface, construction of the surface, and an evaluation of the uniqueness of the surface within the assumptions made in our analysis.

4.5.1. Contributions from Resonances and Direct Scattering

The development of an appropriate potential energy surface is aided by knowing which peaks result from transitions to repulsive adiabatic potentials and which come from adiabatic potentials with wells which can support shape or Feshbach resonances. Consider the $v_3' = 0, 2,$ and 4 peaks in the BrHBr^- spectrum and the $v_3' = 0, 2, 4,$ and 6 peaks in the BrDBr^- spectrum. The $v_3' = 4$ BrHBr^- peak and the $v_3' = 6$ BrDBr^- peak are substantially narrower than the other peaks. Based on the peak widths alone, it is tempting to assign the narrow peaks as transitions to resonance states and the broad peaks to direct scattering. This assignment is supported by comparing the energy of each v_3 peak to the energy of the $\text{Br} + \text{HBr}$ ($v_3/2$) asymptotic level. Figure 2-5 and Table 2-1 show that the broad peaks lie above their respective asymptotic levels (at lower electron kinetic energy), while the $v_3' = 4$ peak in the BrHBr^- spectrum and the $v_3' = 6$ peak in the BrDBr^- spectrum result from transitions to states that lie just below the $\text{Br} + \text{HBr}$ ($v=2$) and $\text{Br} + \text{DBr}$ ($v=3$) levels, respectively. Each broad peak then results from a transition to a region of an adiabatic potential curve which lies above the asymptotic limit of the curve. Such a transition will occur if the adiabatic potential is purely repulsive, in which case the broad peak can be attributed to direct scattering. Each narrow peak appears to result from a transition to a region of an adiabatic potential

which lies *below* its asymptotic limit. This means the potential *must* have a well, which is the necessary condition for the existence of Feshbach resonances.

Unfortunately, the 0.05 eV uncertainty in the bond dissociation energy for BrHBr^- is comparable to the energy differences between the narrow peaks and the relevant asymptotic levels. For example, the $v_3' = 6$ BrDBr^- peak could result from a transition to a state that lies 0.04 eV above the $\text{Br} + \text{DBr}$ ($v=3$) level. This peak could therefore arise from a shape resonance or from a transition to a very flat repulsive curve. In any case, the assignment of the broad peaks to direct scattering and the narrow peaks to resonances is supported by the narrowing of peak widths in the energy range where resonances might be expected.

These considerations are useful in constructing a $\text{Br} + \text{HBr}$ potential energy surface. On the correct surface, the U_0 and U_2 (and, for $\text{Br} + \text{DBr}$, U_4) adiabatic potentials should be repulsive in the Franck-Condon accessible range of ρ , while the $\text{Br} + \text{HBr}$ U_4 and $\text{Br} + \text{DBr}$ U_6 potentials should each have a well or be very flat in this region. This information is helpful, because it is easier to visualize the effect of varying the potential energy surface on the adiabatic potentials than on the full simulation of the photoelectron spectrum.

4.5.2. Functional Form of the Potential Energy Surface

We next choose a flexible functional form for the potential energy surface, using adjustable parameters which can be optimized to reproduce the experimental photoelectron spectra. Our one-dimensional analysis indicates that a double minimum potential given by eq. 6 is appropriate for the antisymmetric stretch potential at fixed ρ . A two-dimensional potential energy surface $V(\rho, z)$ is then constructed by smoothly connecting a series of one-dimensional potentials $V(z)$ at various values of ρ . Each one-dimensional potential represents a cut through the effective collinear surface along the z direction at a constant value of ρ . For convenience, we use the mass-independent coordinates $R_{\text{Br-Br}}$ and $x = (R_{\text{Br-H}} - R_{\text{H-Br}})/2^{1/2}$ for the potential energy surface. The resulting two-dimensional function $V(R_{\text{Br-Br}}, x)$ is then converted to the (mass-weighted) hyperspherical coordinates ρ and z so that $V(\rho, z)$ may be substituted in the differential equations 10 a,b. The equations for the one-dimensional potentials are those in eq. 6, except that the potential parameters (b, h, x_1, k, q) depend on the interbromine distance $R_{\text{Br-Br}}$. The potentials have a single minimum for $R_{\text{Br-Br}} < R_{\text{Br-Br}}$ (saddle point) and two minima at higher values of $R_{\text{Br-Br}}$. The potential parameters are defined at 12 values of $R_{\text{Br-Br}}$ (see Table 2-4) and are interpolated using cubic splines to form a smooth potential. The saddle point is specified by the largest value of $R_{\text{Br-Br}}$ for which $x_1 = 0$; the barrier is then $b(R_{\text{Br-Br}} = \text{saddle}$

point) - $b(R_{\text{Br-Br}} = \infty)$. The variation of b with $R_{\text{Br-Br}}$ specifies the energy along the minimum energy path; the steepness of the minimum energy path can easily be varied and strongly affects the simulated peak widths.

This form of the potential energy surface is completely empirical. It does not automatically predict the characteristic frequencies of reactants and products at the asymptotes. In the asymptotic region ($R_{\text{Br-Br}} \geq 4.7 \text{ \AA}$), the location of the minimum in the product valley approaches the limiting value

$$x_1 = \frac{1}{\sqrt{2}}(R_{\text{Br-H}} - R_e(\text{HBr})) = \frac{1}{\sqrt{2}}(R_{\text{Br-Br}} - 2R_e(\text{HBr})) \quad (14)$$

where $R_e(\text{HBr})$ is the bond length in HBr (1.4144 \AA). The remaining parameters are then set to approximate the vibrational frequency and anharmonicity of HBr.

Equation 14 is important in the fitting procedure, since it provides an upper bound to x_1 for any value of ρ (or $R_{\text{Br-Br}}$). A larger value of x_1 in this region means that the HBr bond shortens as the Br atom approaches in a Br + HBr reactive collision. This runs counter to chemical intuition which suggests that the HBr bond weakens and lengthens as the Br approaches. Thus, a physically reasonable potential energy surface should have x_1 below the limit given by eq. 14, and near the saddle point x_1 should be substantially less than the bound given by eq. 14. (Note that $x_1 = 0$ at the saddle point.)

Although the potential energy function given above has a large number of parameters, the goal of this analysis is to establish the existence of an "effective" collinear potential energy surface which reproduces the experimental BrHBr^- and BrDBr^- spectra. It is likely that a simpler functional form employing fewer, more physically meaningful, parameters could be used successfully, and this is a subject for future analysis of this system.

4.5.3. Results of the Analysis

The potential energy surface that provides the best fit to the experimental results is shown in Figure 2-11. Simulated and experimental spectra corresponding to this surface are shown above the potential energy surface. For both BrHBr^- and BrDBr^- , the experimental peak positions, intensities, and widths for the ground state progression are generally reproduced in the simulations. Because of our experience with $\text{Cl} + \text{HCl}$, we expended most effort on fitting the peak positions. The accuracy of the simulated peak widths and intensities is less certain, particularly for the peaks due to direct scattering. The high energy "tail" on the $\nu_3' = 0$ peaks in the experimental spectra is probably due to ions with some excitation in the low frequency ν_1 mode; the simulations assume $\nu_1 = \nu_3 = 0$ in the ion. Transitions to the electronically excited $[\text{BrHBr}]^*$ state are not included in

this analysis.

The simulations in Figure 2-11 assume that the equilibrium interbromine distance R_e in BrHBr^- is 3.50 Å. Attempts to fit the spectra using the *ab initio* R_e of 3.43 Å⁴³ resulted in a potential energy surface which did not satisfy eq. 14. Our assumed value is within the error bars of the *ab initio* calculation and greatly facilitated fitting the data to a reasonable potential energy surface. However, our analysis is quite sensitive to the assumed value of R_e . It is clear that the accurate experimental determination of the anion R_e would provide an important constraint in the construction of the $\text{Br} + \text{HBr}$ potential energy surface. The experimental approach most likely to yield this value is velocity-modulated infrared spectroscopy of the ion,⁵⁷ which should result in a rotationally resolved spectrum from which R_e can be obtained. This method has been used to determine R_e for FHF^- and ClHCl^- .

The parameters for our potential energy surface are given in Table 2-4. This potential energy surface has a saddle point at $R_{\text{Br-Br}} = 3.29$ Å, and a barrier of 46.8 kJ/mol along the minimum energy path. This potential is the effective collinear surface $V(\rho, z)$ in eq. 7, which implicitly includes the bending zero-point energy. Our analysis does *not* divide $V(\rho, z)$ into a collinear and bending contribution, as this would require assuming the form of the bending potential. We estimate the bending zero-point energy at the saddle point to be 4-8 kJ/mol based on model $\text{Cl} + \text{HCl}$ surfaces, so the

saddle point energy for the collinear BrHBr geometry is approximately 40 kJ/mol. This is considerably higher than the barriers on previously proposed model potential energy surfaces for the Br + HBr reaction.^{38b,58} The adiabatic potentials for the first few v_3' levels of Br + HBr and Br + DBr are shown in Figure 2-12. The U_4 (Br + HBr) and U_6 (Br + DBr) curves each have a shallow well which supports a resonance state; the lower curves are purely repulsive.

We next consider the uniqueness of our potential energy surface. We divide the surface into four ranges of ρ . In order of increasing ρ , these are (i) the saddle point region which includes the saddle point and the repulsive wall at small ρ , (ii) the Franck-Condon region, (iii) the intermediate region, between the Franck-Condon region and the asymptotic region, and (iv) the asymptotic region (large ρ). Region iv depends on the asymptotic properties of Br + HBr, which are well characterized. The requirement that the potential energy surface vary smoothly from the Franck-Condon region to the asymptotic region constrains the parameters which specify the intermediate region, although we are not sensitive to features of the surface (such as van der Waals wells) which occur in this range. Our experiment is most sensitive to features of the surface in the Franck-Condon region, and least sensitive to the saddle point region. The saddle point parameters are extrapolations from better-determined regions of the surface, but, as mentioned above, the Franck-Condon region of the surface is likely to be

more important than the saddle point region in the dynamics of this reaction.

To further illustrate the sensitivity of our experiment to the nature of the potential energy surface, the BrHBr^- and BrDBr^- spectra have been simulated using an effective collinear surface derived from a three-dimensional LEPS surface, assuming $v_2' = 0$ in the complex. The barrier height and saddle point geometry on this "effective" collinear surface are similar to our "best fit" surface. The LEPS simulations, shown in Figure 2-13, are clearly inferior. The peaks are shifted to low electron kinetic energy, suggesting that the minimum energy path in the Franck-Condon region lies too far above the asymptote. In addition, the peak spacings are too large, the peaks are too narrow, and the intensities are incorrect. If we take one-dimensional cuts at constant ρ through the Franck-Condon region of each surface, we obtain antisymmetric stretch potentials which are similar to those shown in Figure 2-7 for the $\text{Cl} + \text{HCl}$ reaction; the outer walls are not as steep on our surface, and the two minima in the potential are further apart than those of the LEPS surface. This explains the differences in the simulated peak positions and intensities for the two surfaces. In addition, the minimum energy path on our surface begins to rise later in the course of the reaction (i.e., at smaller values of ρ). For equal barrier heights, the minimum energy path is steeper in the Franck-Condon region on our surface, resulting in adiabatic potentials that are

more repulsive in the Franck-Condon region and peaks that are substantially broader. Since the effective collinear surfaces used in these simulations include the bending zero-point energy, we cannot say whether the later rise in the minimum energy path on our surface is due to differences in the bend potential or in the collinear electronic interaction between the reactants. This comparison shows that reactive surfaces with similar saddle point properties can give very different simulated spectra.

The discrepancies between the simulated and experimental results for ClHCl^- are similar to the differences between the BrHBr^- LEPS simulation and experimental results, and likely result from flaws in the $\text{Cl} + \text{HCl}$ LEPS surface similar to those found for the LEPS surface in Fig. 2-13. The barrier height on the LEPS surface used in the ClHCl^- simulations is consistent with *ab initio* calculations and experimental results on the $\text{Cl} + \text{HCl}$ reaction. However, the BrHBr^- analysis suggests that the $\text{Cl} + \text{HCl}$ LEPS surface is too flat in the Franck-Condon region and that an effective collinear surface with the same barrier height, but sharing more of the features of our $\text{Br} + \text{HBr}$ surface would better reproduce the experimental results.

5. Summary and Conclusions

The photoelectron spectra of the negative ions ClHCl^- , BrHBr^- , and

their deuterated analogues provide a sensitive probe of the transition state region of the Cl + HCl and Br + HBr reactions. The spectra show progressions in the antisymmetric stretch vibration of the neutral [ClHCl] and [BrHBr] complexes. The observed peak widths are an indication of the dissociation dynamics of the complexes. The BrHBr⁻ and BrDBr⁻ spectra exhibit broad peaks assigned to directly dissociating states of the neutral complex, narrow peaks assigned to reactive resonance states, and peaks which appear to originate from electronically excited states of the complex. We have analyzed the spectra by assuming that the reactions under study are governed by "effective" collinear potential energy surfaces. For the Br + HBr reaction, a flexible functional form for such a surface has been developed, and we have found a set of parameters which allows us to simulate the peak positions, widths, and intensities in the experimental BrHBr⁻ and BrDBr⁻ photoelectron spectra.

We plan to study these ions with our high resolution (0.3 meV) threshold photodetachment spectrometer³² to more accurately determine the positions and widths of the resonance peaks. Our results also suggest avenues for further theoretical work on these systems. The validity of the effective collinear surface determined here for the Br + HBr reaction could be tested with more accurate simulations using full three-dimensional potential energy surfaces which yield our surface in the reduced dimensionality model (see Chapter 4). Accurate simulations on potential

energy surfaces with bent transition states, for which the effective collinear approximation is inappropriate, will also be of interest. It will be useful to see if other types of semi-empirical potential energy surfaces (such as a DIM surface) would be more successful than a LEPS surface at reproducing our experimental results. Finally, it is worthwhile to examine how sensitive the asymptotic properties of the $\text{Br} + \text{HBr}$ reaction are to the regions of the potential energy surface which are well characterized by our experiment.

1. R. D. Levine and R. B. Bernstein, Molecular Reaction Dynamics and Chemical Reactivity, Oxford University Press, New York (1987), pp. 396-510.
2. P. Siegbahn and B. Liu, *J. Chem. Phys.* **68**, 2457 (1978); D. G. Truhlar and C. J. Horowitz, *J. Chem. Phys.* **68**, 2468 (1978); H. F. Schaefer III, *J. Phys. Chem.* **89**, 5336 (1985); D. W. Schwenke, R. Steckler, F. B. Brown and D. G. Truhlar, *J. Chem. Phys.* **84**, 5706 (1986).
3. P. R. Brooks, *Chem. Rev.* **88**, 407 (1988).
4. A. H. Zewail, *Science* **242**, 1645 (1988).
5. P. Arrowsmith, F. E. Bartoszek, S. H. P. Bly, T. Carrington, Jr., P. E. Charters and J. C. Polanyi, *J. Chem. Phys.* **73**, 5895 (1980).
6. P. Hering, P. R. Brooks, R. F. Curl, Jr., R. S. Judson and R. S. Lowe, *Phys. Rev. Lett.* **44**, 687 (1980); T. C. Maguire, P. R. Brooks, R. F. Curl, J. H. Spence and S. J. Ulvick, *J. Chem. Phys.* **85**, 844 (1986).
7. H. P. Grieneisen, H. Xue-jing and K. L. Kompa, *Chem. Phys. Lett.* **82**, 421 (1981).
8. P. D. Kleiber, A. M. Lyra, K. M. Sando, V. Zafirooulos and W. C. Stwalley, *J. Chem. Phys.* **85**, 5493 (1986).
9. A. Benz, and H. Morgner, *Molec. Phys.* **57**, 319 (1986).
10. S. Buelow, G. Radhakrishnan, J. Catanzarite, and C. Wittig, *J. Chem. Phys.* **83**, 444 (1985).

11. D. G. Imre, J. L. Kinsey, R. W. Field and D. H. Katayama, *J. Phys. Chem.* **86**, 2564 (1982).
12. W. H. Breckenridge, C. Jouvet and B. Soep, *J. Chem. Phys.* **84**, 1443 (1986).
13. J.-C. Nieh and J. J. Valentini, *Phys. Rev. Lett.* **60**, 519 (1988).
14. M. Dantus, M. J. Rosker and A. H. Zewail, *J. Chem. Phys.* **87**, 2395 (1987); R. M. Bowman, M. Dantus, and A. H. Zewail, *Chem. Phys. Lett.* **156**, 131 (1989).
15. S. Golub and B. Steiner, *J. Chem. Phys.* **49**, 5191 (1968).
16. C. R. Moylan, J. A. Dodd, C. Han and J. I. Brauman, *J. Chem. Phys.* **86**, 5350 (1987).
17. S. M. Burnett, A. E. Stevens, C. S. Feigerle and W. C. Lineberger, *Chem. Phys. Lett.* **100**, 124 (1982); K. M. Ervin, J. Ho and W. C. Lineberger, *J. Chem. Phys.* **91**, 5974 (1989).
18. S. E. Bradforth, A. Weaver, R. B. Metz and D. M. Neumark, Advances in Laser Science - IV Proceedings of the 1988 International Laser Science Conference, American Institute of Physics Conference Proceedings 191; J. L. Gole, D. F. Heller, M. Lapp and W. C. Stwalley, Eds., American Institute of Physics, New York (1989) p. 657; S. E. Bradforth, A. Weaver, D. W. Arnold, R. B. Metz and D. M. Neumark, *J. Chem. Phys.* **92**, 7205 (1990); S. E. Bradforth, D. W. Arnold, R. B. Metz, A. Weaver and D. M. Neumark, *J. Phys. Chem.* xxxxxxxx (1991).

19. M. Kneba and J. Wolfrum, *J. Phys. Chem.* **83**, 69 (1979); F. S. Klein, A. Persky and R. E. Weston, Jr., *J. Chem. Phys.* **41**, 1799 (1964).
20. B. C. Garrett, D. G. Truhlar, A. F. Wagner and T. H. Dunning, Jr., *J. Chem. Phys.* **78**, 4400 (1983); P. Botschwina and W. Meyer, *Chem. Phys. Lett.* **44**, 449 (1976).
21. K. Kawaguchi, *J. Chem. Phys.* **88**, 4186 (1988).
22. P. Botschwina, P. Sebald and R. Durmeister, *J. Chem. Phys.* **88**, 5246 (1988); S. Ikuta, T. Saitoh and O. Nomura, *J. Chem. Phys.* **91**, 3539 (1989).
23. a) D. E. Milligan and M. E. Jacox, *J. Chem. Phys.* **53**, 2034 (1970); b) J. W. Nibler and G. C. Pimentel, Jr., *J. Chem. Phys.* **47**, 710 (1967); c) B. J. Ault, *Acc. Chem. Res.* **15**, 103 (1982).
24. D. G. Truhlar and A. Kuppermann, *J. Chem. Phys.* **52**, 384 (1970); S.-F. Wu and R. D. Levine, *Mol. Phys.* **22**, 991 (1971).
25. D. K. Bondi, J. N. L. Connor, J. Manz and J. Römelt, *Mol. Phys.* **50**, 467 (1983).
26. J. Manz, R. Meyer, and H. H. R. Schor, *J. Chem. Phys.* **80**, 1562 (1984).
27. G. C. Schatz, *Chem. Phys. Lett.* **151**, 409 (1988).
28. D. M. Neumark, A. M. Wodtke, G. N. Robinson, C. C. Hayden and Y. T. Lee, *J. Chem. Phys.* **82**, 3045 (1985).

29. R. B. Metz, T. Kitsopoulos, A. Weaver and D. M. Neumark, *J. Chem. Phys.* **88**, 1463 (1988).
30. A. Weaver, R. B. Metz, S. E. Bradforth and D. M. Neumark, *J. Phys. Chem.* **92**, 5558 (1988).
31. L. A. Posey, M. J. DeLuca and M. A. Johnson, *Chem. Phys. Lett.* **131**, 170 (1986); O. Cheshnovsky, S. H. Yang, C. L. Pettiette, S. J. Craycraft and R. E. Smalley, *Chem. Phys. Lett.* **138**, 119 (1987); G. Gantefor, K. H. Meiwes-Broer and H. O. Lutz, *Phys. Rev. Lett.* **37**, 2716 (1988).
32. T. N. Kitsopoulos, I. M. Waller, J. G. Loeser and D. M. Neumark, *Chem. Phys. Lett.* **159**, 300 (1989).
33. W. C. Wiley and I. H. McLaren, *Rev. Sci. Instrum.* **26**, 1150 (1955).
34. G. Caldwell and P. Kebarle, *Can. J. Chem.* **63**, 1399 (1985).
35. H. Hotop and W. C. Lineberger, *J. Phys. Chem. Ref. Data* **14**, 731 (1985).
36. E. Pollak, *J. Chem. Phys.* **78**, 1228 (1983); J. Manz, R. Meyer, E. Pollak, J. Römelt and H. H. R. Schor, *Chem. Phys.* **83**, 333 (1984); D. C. Clary and J. N. L. Connor, *J. Phys. Chem.* **88**, 2758 (1984).
37. G. Herzberg, Molecular Spectra and Molecular Structure, Vol. III, Van Nostrand, New York (1966), pp. 150-157.
38. a) A. Persky and H. Kornweitz, *J. Phys. Chem.* **91**, 5496 (1987); b) I. Last and M. Baer, *J. Chem. Phys.* **80**, 3246 (1983).

39. G. C. Schatz, *J. Chem. Phys.* **90**, 3582 (1989).
40. a) J. M. Bowman and B. Gazdy, *J. Phys. Chem.* **93**, 5129 (1989); b) B. Gazdy and J. M. Bowman, *J. Chem. Phys.* **91**, 4615 (1989).
41. A. B. Sannigrahi and S. D. Peyerimhoff, *J. Mol. Struct.* **165**, 55 (1988).
42. M. E. Jacox, *J. Phys. Chem. Ref. Data* **13**, 945 (1984).
43. K. Kawaguchi and E. Hirota, *J. Chem. Phys.* **87**, 6838 (1987).
44. S. Sato, *Bull. Chem. Soc. Jpn.* **28**, 450 (1955).
45. G. Hanke, J. Manz and J. Römelt, *J. Chem. Phys.* **73**, 5040 (1980); A. Kuppermann, J. A. Kaye and J. P. Dwyer, *Chem. Phys. Lett* **74**, 257 (1980); V. K. Babamov and R. A. Marcus, *J. Chem. Phys.* **74**, 1790 (1981); V. Aquilanti, S. Cavalli and A. Lagana, *Chem. Phys. Lett.* **93**, 179 (1982).
46. a) D. K. Bondi, J. N. L. Connor, B. C. Garrett and D. G. Truhlar, *J. Chem. Phys.* **78**, 5981 (1983); b) B. Hartke and J. Manz, *J. Am. Chem. Soc.* **110**, 3063 (1988).
47. R. T. Pack, *J. Chem. Phys.* **65**, 4765 (1976).
48. We thank G. Barney Ellison for a copy of his program for calculating Franck-Condon factors.
49. N. C. Firth and R. Grice, *J. Chem. Soc. Faraday Trans. 2* **87**, 1023 (1987).
50. J. M. Bowman, *Adv. Chem. Phys.* **61**, 115 (1985).
51. J. Römelt, *Chem. Phys.* **79**, 197 (1983).

52. W. H. Press, B. P. Flannery, S. A. Teukolsky and V. T. Vetterling,
Numerical Recipes, Cambridge University Press, Cambridge, England
(1986).
53. B. W. Shore, *J. Chem. Phys.* **58**, 3855 (1973).
54. J. Tellinghuisen, *Adv. Chem. Phys.* **60**, 299 (1985).
55. R. J. LeRoy, Univ. of Waterloo Chemical Physics Research Report CP-
329 (1988).
56. J. M. Bowman, private communication.
57. R. J. Saykally, *Science* **239**, 157 (1988).
58. J. M. White and D. L. Thompson, *J. Chem. Phys.* **61**, 719 (1974).

Table 2-1: Peak Positions and Widths in the 213 nm (5.825 eV)

Photoelectron Spectra of ClHCl^- and ClDCl^- ^a and of BrHBr^-
and BrDBr^- .^b

v_3'	eKE (eV)	width (meV)
ClHCl^-		
0	0.894	130
2	0.693	85
4	0.517	-- ^c
ClDCl^-		
0	0.884	-- ^c
2	0.734	70
4	0.611	55
6	0.479	-- ^c

v_3'	eKE (eV)	width (meV)	asymptote (eV)
BrHBr ⁻			
0	1.353	170	1.554
2	1.159	80	1.237
4	0.950	20	0.931
A'	0.608	140	
B'	0.397	140	
BrDBr ⁻			
0	1.344	175	1.560
2	1.190	100	1.332
4	1.061	64	1.109
6	0.895	20	0.892
A'	0.686	140	
B'	0.529	140	

^a Average uncertainty in peak positions is 0.015 eV.

^b The asymptotes correspond to Br + H(D)Br ($v = v_3'/2$). Average uncertainty in these peak positions is 0.010 eV.

^c Accurate widths could not be determined, as these peaks are not sufficiently well resolved.

Table 2-2: Anion Frequencies (in cm^{-1}).

	ν_1''	ν_2''	ν_3''
ClHCl^- (gas) ^a	318	792	723
ClHCl^- (matrix) ^b	260		696
ClDCl^- (matrix) ^b	267		463
BrHBr^- (matrix) ^b	164		728
BrDBr^- (matrix) ^b	170		498

^a Reference 21.

^b Reference 42.

Table 2-3: Potential Parameters for the One-Dimensional Fits.^a

	x_1 (Å)	h (kJ/mol)	k (kJ/mol/Å ²)	q (kJ/mol/Å ⁴)
[ClHCl]	0.49	35	1000	0
[BrHBr]	0.45	30	500	2500
[BrHBr] [*]	0.47	52	450	2500

^a The form of the potential is eq. 6.

Table 2-4: Parameters for the Fitted Br + HBr Potential.^a

$R_{\text{Br-Br}}$ (Å)	x_1 (Å)	b (kJ/mol)	h (kJ/mol)	k (10^3 kJ/mol/Å ²)	q (10^3 kJ/mol/Å ⁴)
2.6	0.0	-50.0	0.0	5.0	1.3
2.8	0.0	-140.0	0.0	4.0	1.6
3.0	0.0	-215.0	0.0	2.9	2.1
3.29	0.0	-330.5	0.0	0.8	2.0
3.4	0.31	-343.5	15.0	1.6	18.0
3.5	0.465	-360.5	33.5	1.1	62.0
3.6	0.541	-370.0	51.0	0.8	22.0
3.7	0.613	-374.9	84.0	0.9	6.0
3.85	0.719	-376.8	135.0	1.0	2.2
4.0	0.826	-377.0	170.0	1.05	1.2
4.2	0.968	-377.1	217.0	1.08	0.9
4.7	1.322	-377.3	300.0	1.23	0.3

^a The form of the potential is eq. 6, for each value of $R_{\text{Br-Br}}$. The global potential is defined by connecting the parameters with natural cubic splines at the values of $R_{\text{Br-Br}}$ given in the table. For x_1 and h, the smallest spline point is $R_{\text{Br-Br}} = 3.29$ Å.

Table 2-5: Parameters for the Br + HBr LEPS Potential.^a

	D_e (kJ/mol)	r_e (Å)	b (Å ⁻¹)	S
HBr	378.3	1.4144	1.810	0.075
Br ₂	192.1	2.281	1.964	0.075

^a The "effective collinear" potential has a barrier of 48.65 kJ/mol at $R_{\text{Br-Br}} = 3.266$ Å.

Chapter 2 Figure Captions

Figure 2-1. Schematic potentials along the reaction coordinate for $X + HX$ and $X^- + HX$.

Figure 2-2. Schematic diagram of the time-of-flight photoelectron spectrometer. See the text for a description.

Figure 2-3. 213 nm photoelectron spectra of $ClHCl^-$ and $ClDCl^-$.

Figure 2-4. 213 nm photoelectron spectra of $BrHBr^-$ and $BrDBr^-$.

Figure 2-5. Energy level diagram for the $BrHBr^-/[BrHBr]$ system. The uncertainty in $D_0(BrHBr^-)$ is 0.05 eV.

Figure 2-6. Contour plot of the BCMR LEPS potential for the collinear $Cl + HCl$ system. The hyperspherical coordinates $z \approx \frac{1}{2}(R_{ClH} - R_{HCl})$ and $\rho \approx 4.22 R_{Cl-Cl}$ are used (eq. 4). Contours are at -300, -350, -400, and -425 kJ/mol with respect to three-atom dissociation. The Franck-Condon region is shaded.

Figure 2-7. (bottom,) Harmonic potential for ν_3 mode of $ClHCl^-$ with the $\nu_3'' = 0$ levels for $ClHCl^-$ (right) and $ClDCl^-$ (left). The abscissa $x = 2^{1/2} z$.

(top and sides) Antisymmetric stretch potentials for $[ClHCl]$ from a cut of the BCMR LEPS potential at $R_e = 3.112 \text{ \AA}$ (- - -) and double minimum potential which gives best fit to experiment (—). For each potential, eigenvalues with even

v_3' are shown, as are the 'stick' spectra obtained from one-dimensional simulations of ClHCl^- and ClDCl^- photoelectron spectra. (States with odd v_3' have zero overlap with the ground state of the anion.) The simulations are superimposed on the experimental spectra. Each tick mark on the vertical axis is 0.1 eV. The parameters for the best fit double minimum potential are given in Table 2-3.

Figure 2-8. Best fit antisymmetric stretch potentials for $[\text{BrHBr}]$ and $[\text{BrHBr}]^*$ and resulting spectra. Energy levels for even v_3' states are shown (.... for $[\text{BrHBr}]$ and - - - for $[\text{BrHBr}]^*$), as are the results of one-dimensional simulations of the BrHBr^- and BrDBr^- photoelectron spectra. For comparison with experiment, the Franck-Condon sticks have been convoluted with Gaussians to match the experimental peak widths (Table 2-1b). Each tick mark on the vertical axis is 0.1 eV. The parameters for the potentials are given in Table 2-3.

Figure 2-9. Adiabatic curves $U_{v_3}(\rho)$ for $\text{Cl} + \text{HCl}$ on the BCMR LEPS potential (ref. 25), including zero-point bend. Shape resonances are denoted by dotted lines and the Feshbach resonance is shown by a solid line. The asymptotic energy levels of $\text{Cl} + \text{HCl}$ ($v = v_3/2$) are denoted by arrows. The

dashed line is drawn at R_e for ClHCl^- ($R_{\text{Cl-Cl}} = 3.112 \text{ \AA}$; $\rho = 13.15 \text{ \AA}$).

Figure 2-10. Two-dimensional simulated photoelectron spectrum of ClHCl^- using the BCMR LEPS potential including zero-point bend for different bending levels of $[\text{ClHCl}]$ (- - -, $v_2' = 0$;, $v_2' = 2$) superimposed on the experimental spectrum (—). The simulated spectrum has been convoluted with the experimental resolution (eq. 2) and assumes the anion is in its ground vibrational state.

Figure 2-11. (a) Contour plot of the best fit effective linear potential energy surface for the $\text{Br} + \text{HBr}$ reaction. For $\text{Br} + \text{HBr}$, $z \approx \frac{1}{2} (R_{\text{BrH}} - R_{\text{HBr}})$ and $\rho \approx 6.30 R_{\text{Br-Br}}$. The saddle point is at $R_{\text{Br-Br}} = 3.29 \text{ \AA}$; $\rho = 20.72 \text{ \AA}$. The Franck-Condon region is shaded. Contours are at -300, -325, -350, and -370 kJ/mol with respect to three atom dissociation. The parameters for the surface are given in Table 2-4.

(b) Simulated photoelectron spectra of BrHBr^- and BrDBr^- using our best fit $\text{Br} + \text{HBr}$ effective collinear surface. The simulated spectra (- - -) have been convoluted with the experimental resolution (eq. 2) and are superimposed on the experimental spectrum.

Figure 2-12. Adiabatic curves $U_{v_3}(\rho)$ for Br + HBr and Br + DBr on the surface shown in Fig. 2-11. Shape resonances are denoted by dotted lines and Feshbach resonances by solid lines. The asymptotic levels of Br + H(D)Br ($v = v_3'/2$) are denoted by arrows. The dashed line is drawn at 3.5 Å, the value of R_e for BrH(D)Br⁻ used in our analysis.

Figure 2-13. (a) Contour plot of LEPS potential energy surface (plus zero-point bend) for Br + HBr with barrier and saddle point very similar to that of the surface in Fig. 2-11a. Contours are at -300, -325, -350, and -370 kJ/mol with respect to three-atom dissociation. The parameters for the surface are given in Table 2-5.

(b) Simulated photoelectron spectra of BrHBr⁻ and BrDBr⁻ (- - -) using the LEPS surface for Br + HBr shown in Fig. 2-13 a). The simulated spectra have been convoluted with the experimental resolution (eq. 2) and are superimposed on the experimental spectrum.

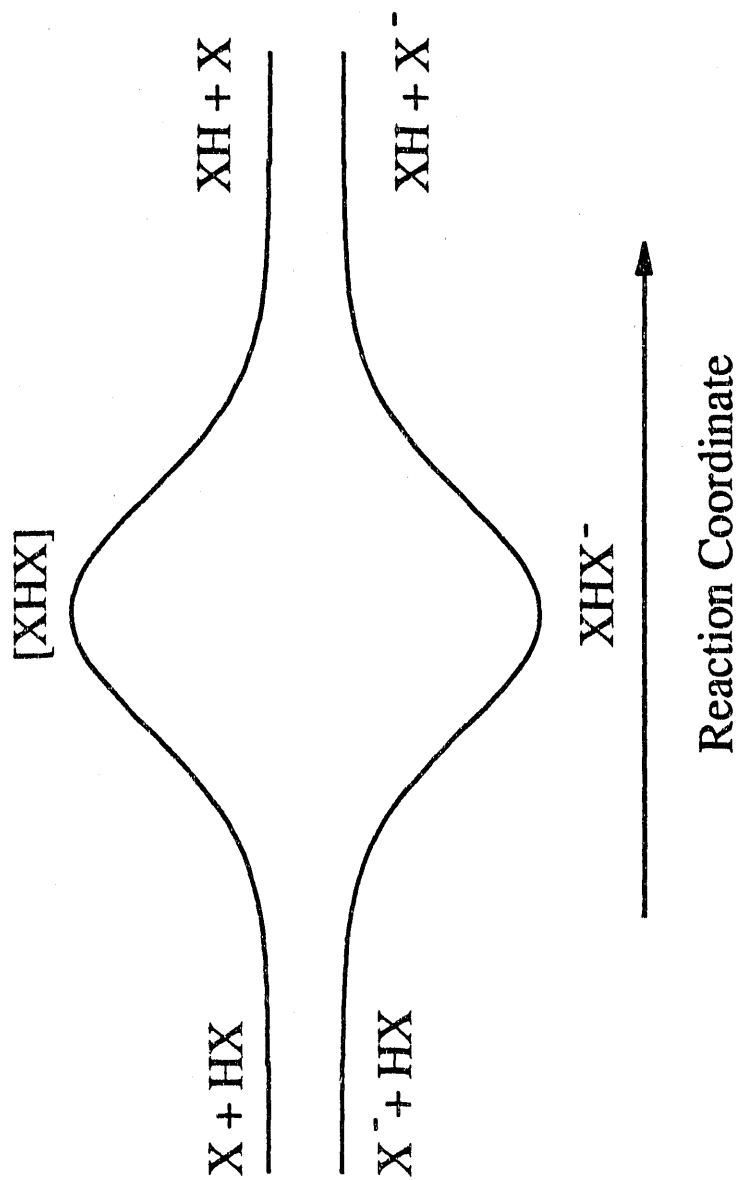


Figure 2-1

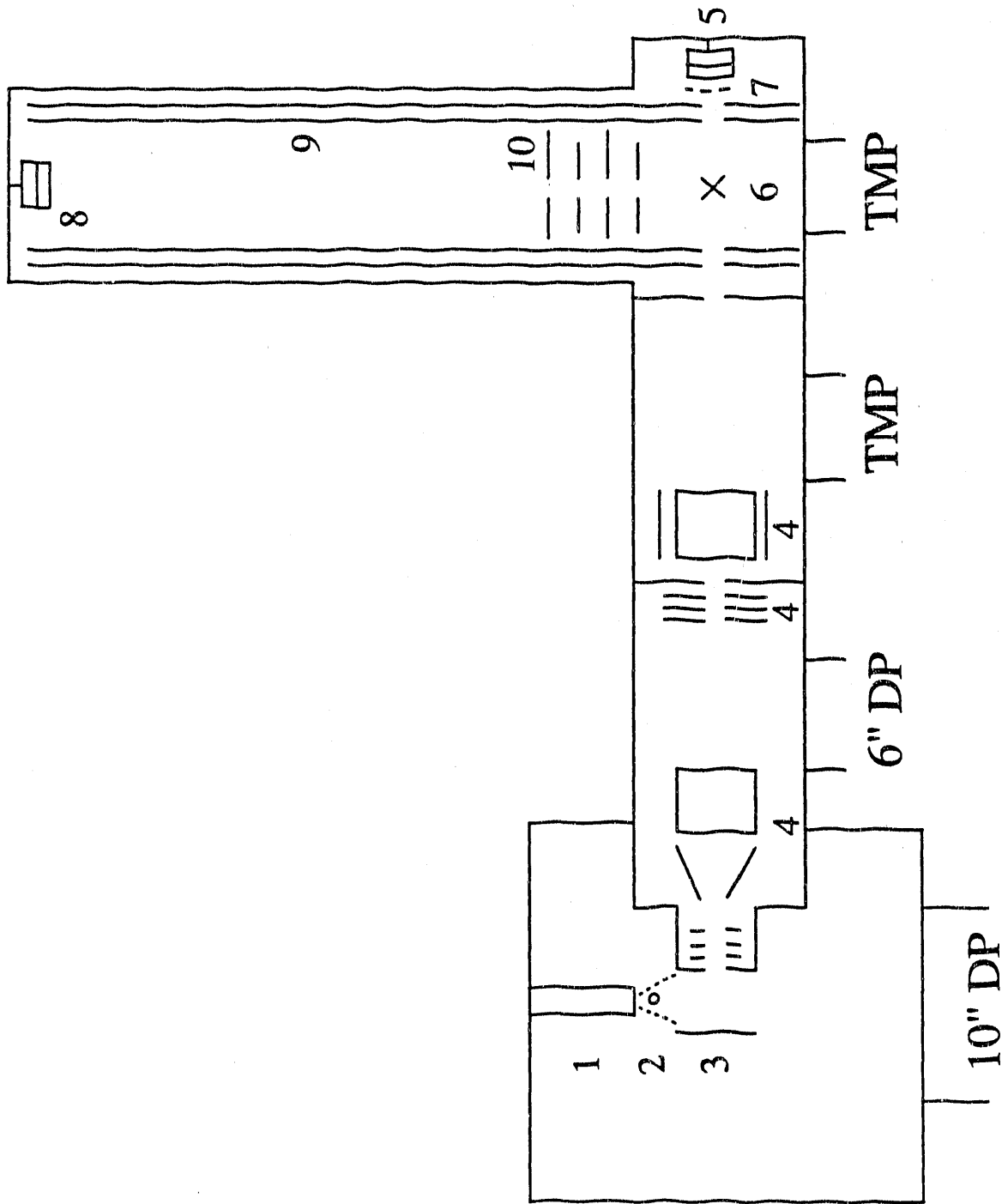


Figure 2-2

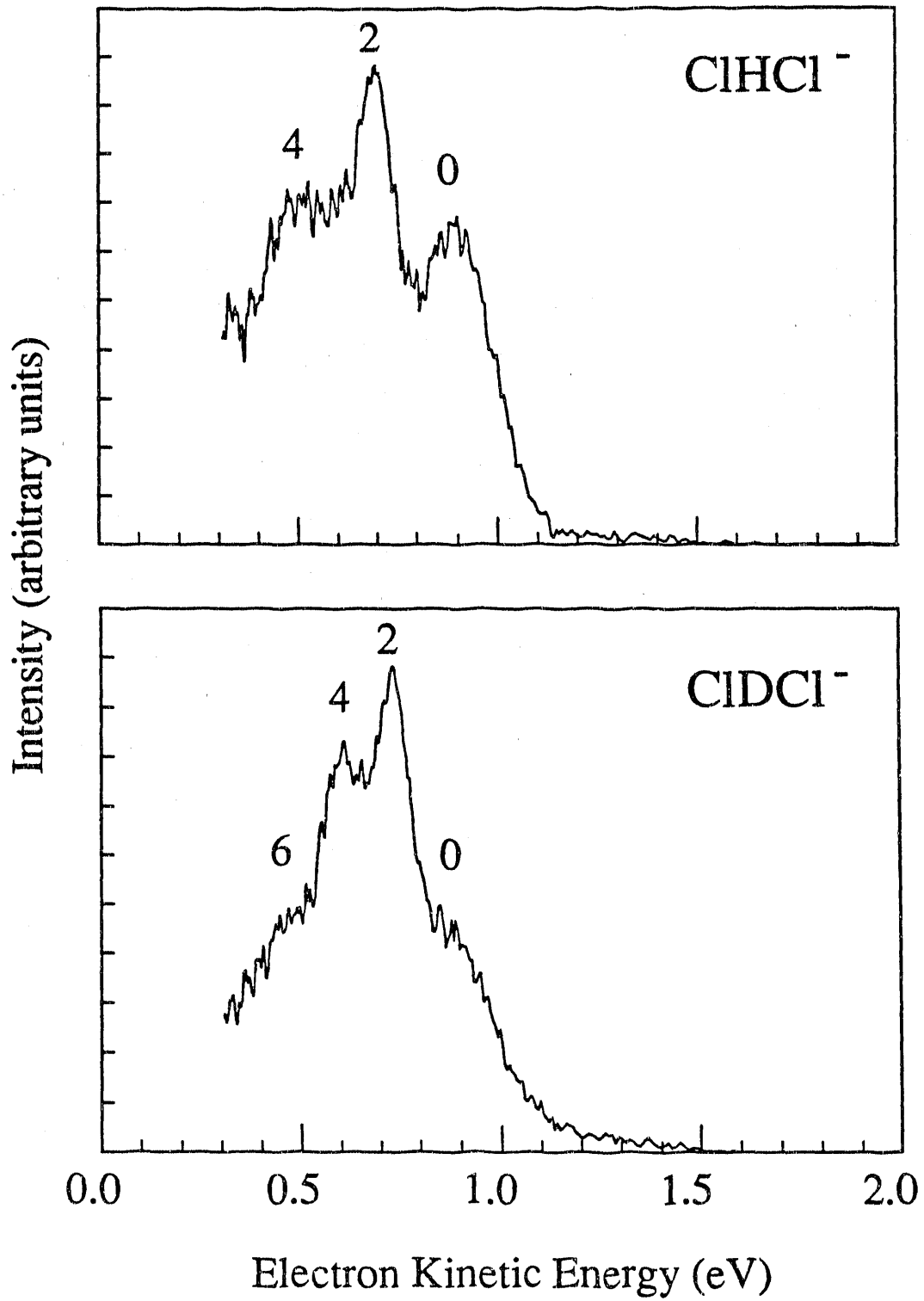


Figure 2-3

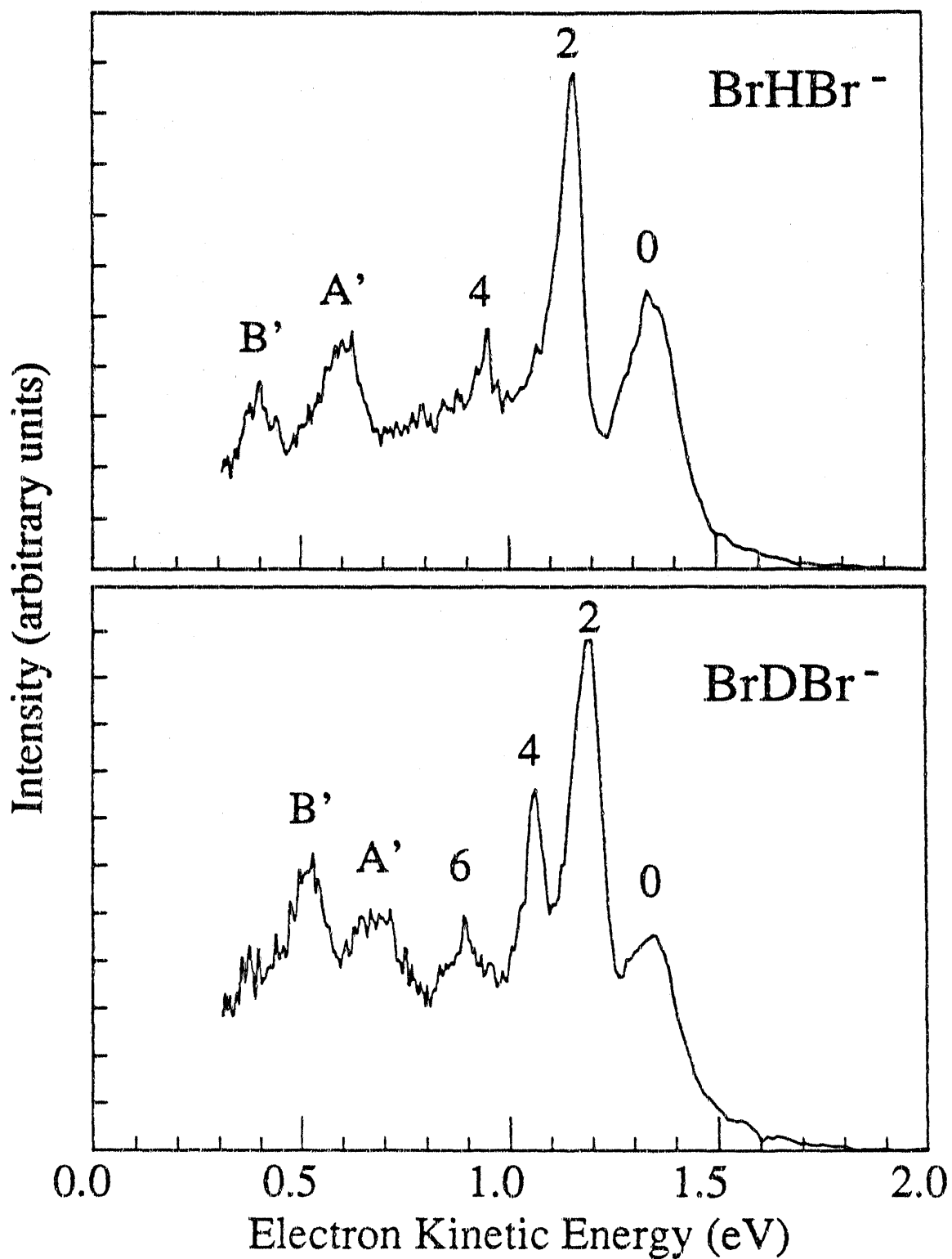


Figure 2-4

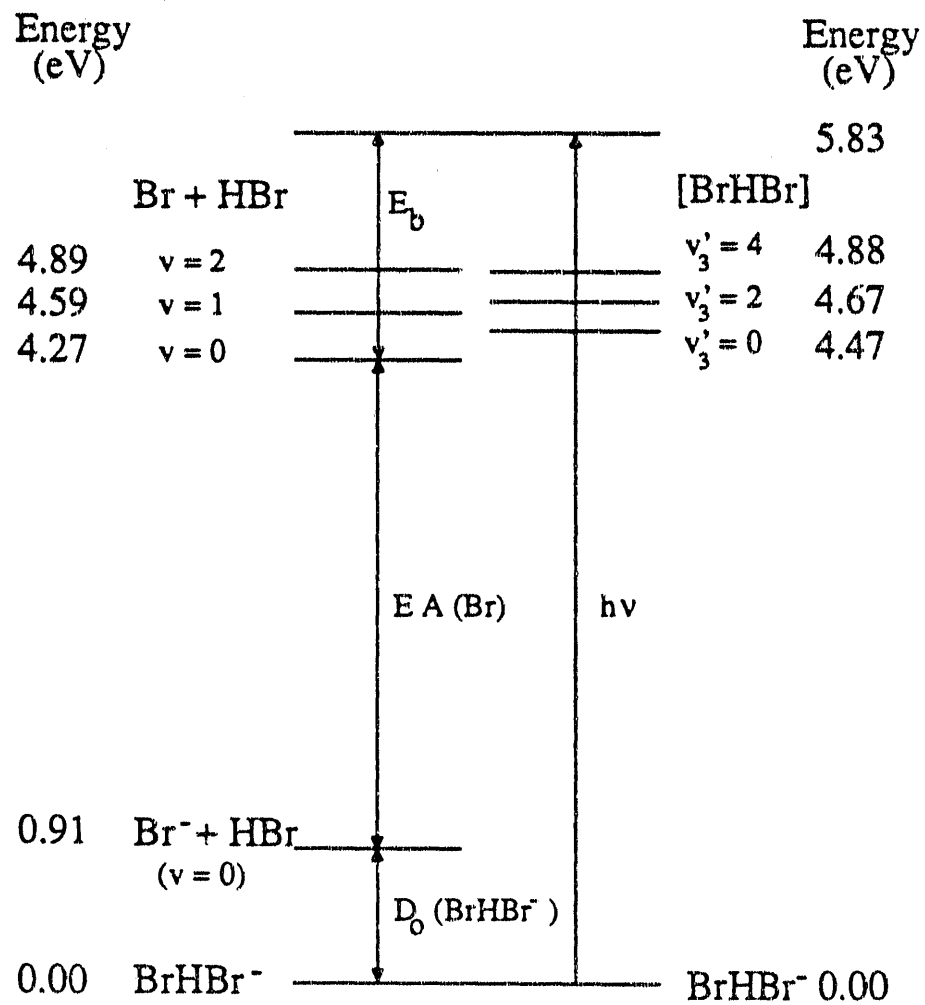


Figure 2-5

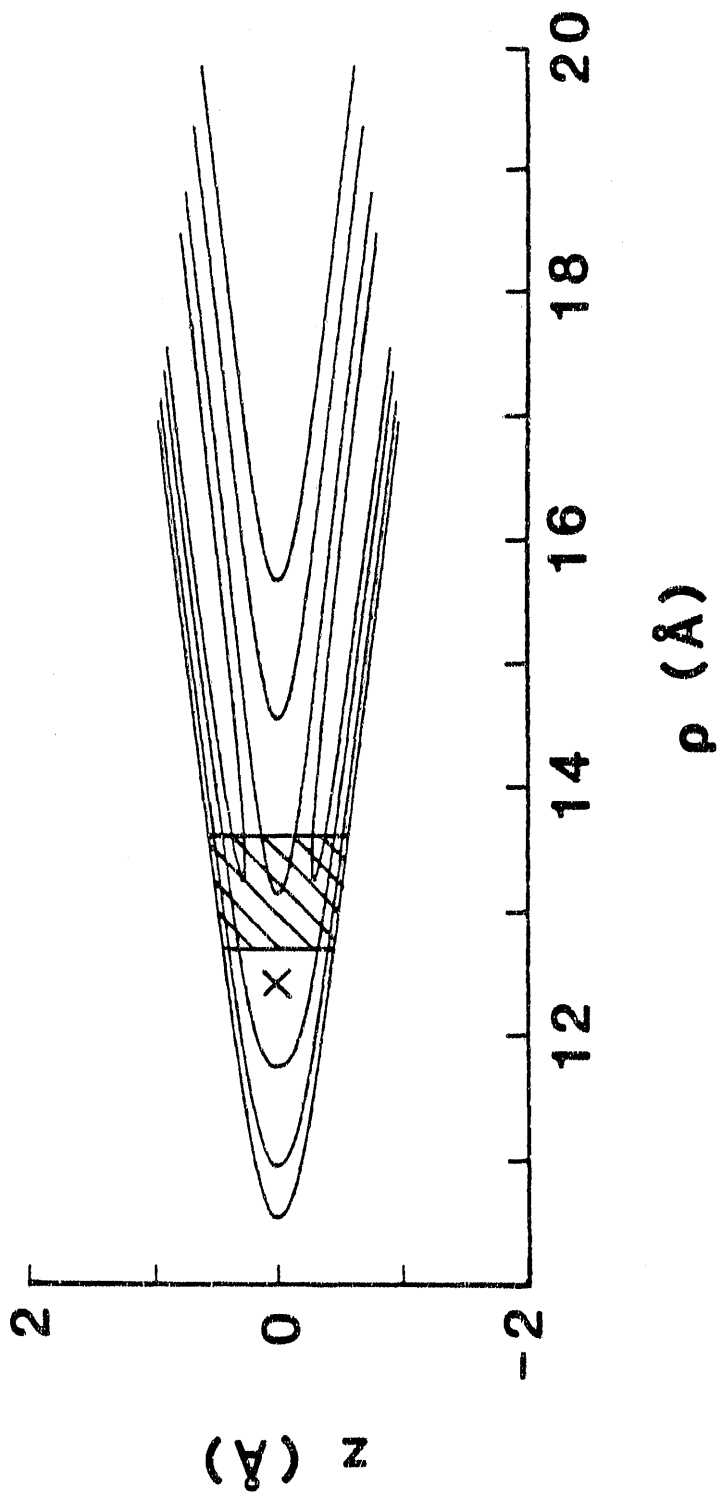


Figure 2-6

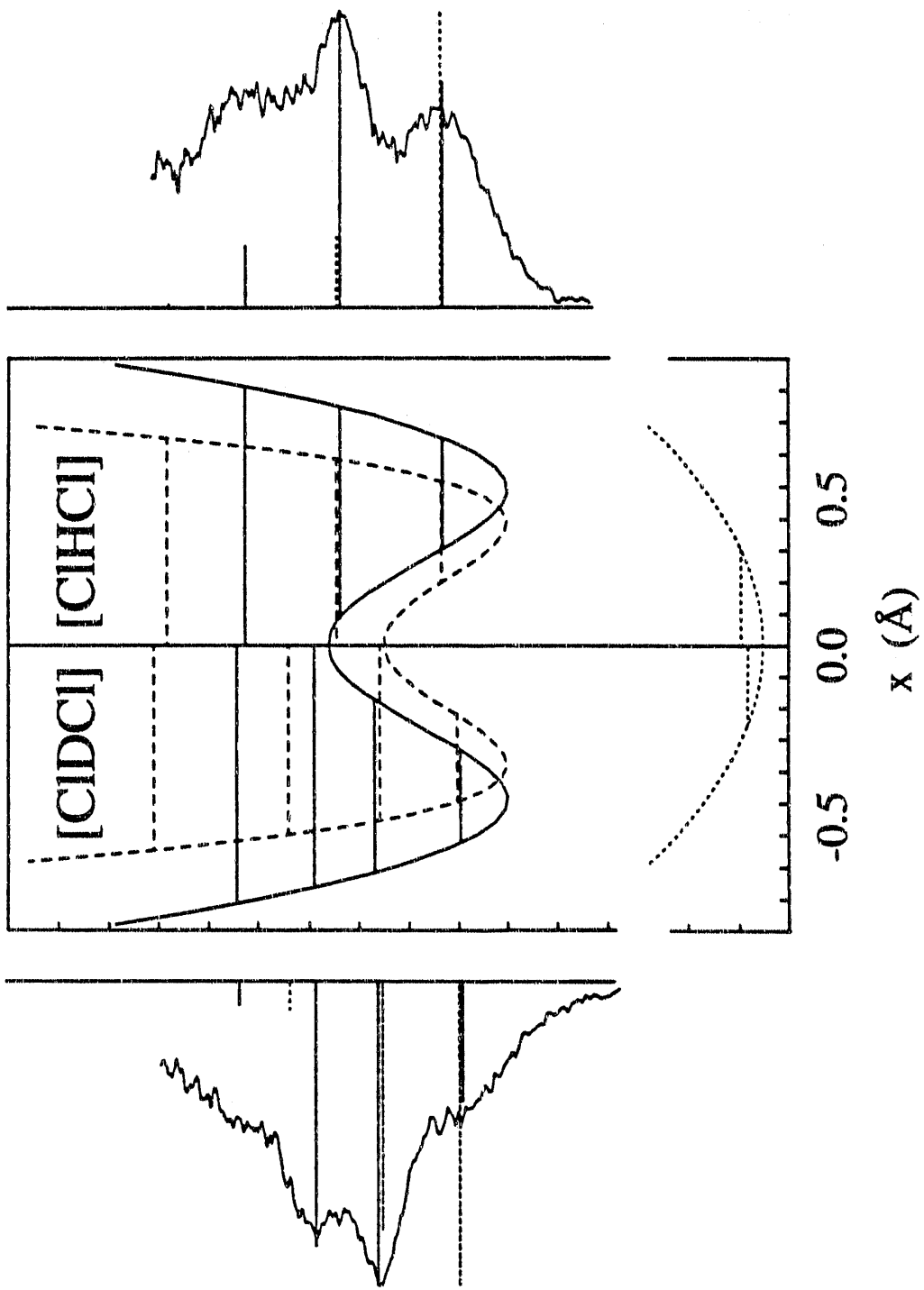


Figure 2-7

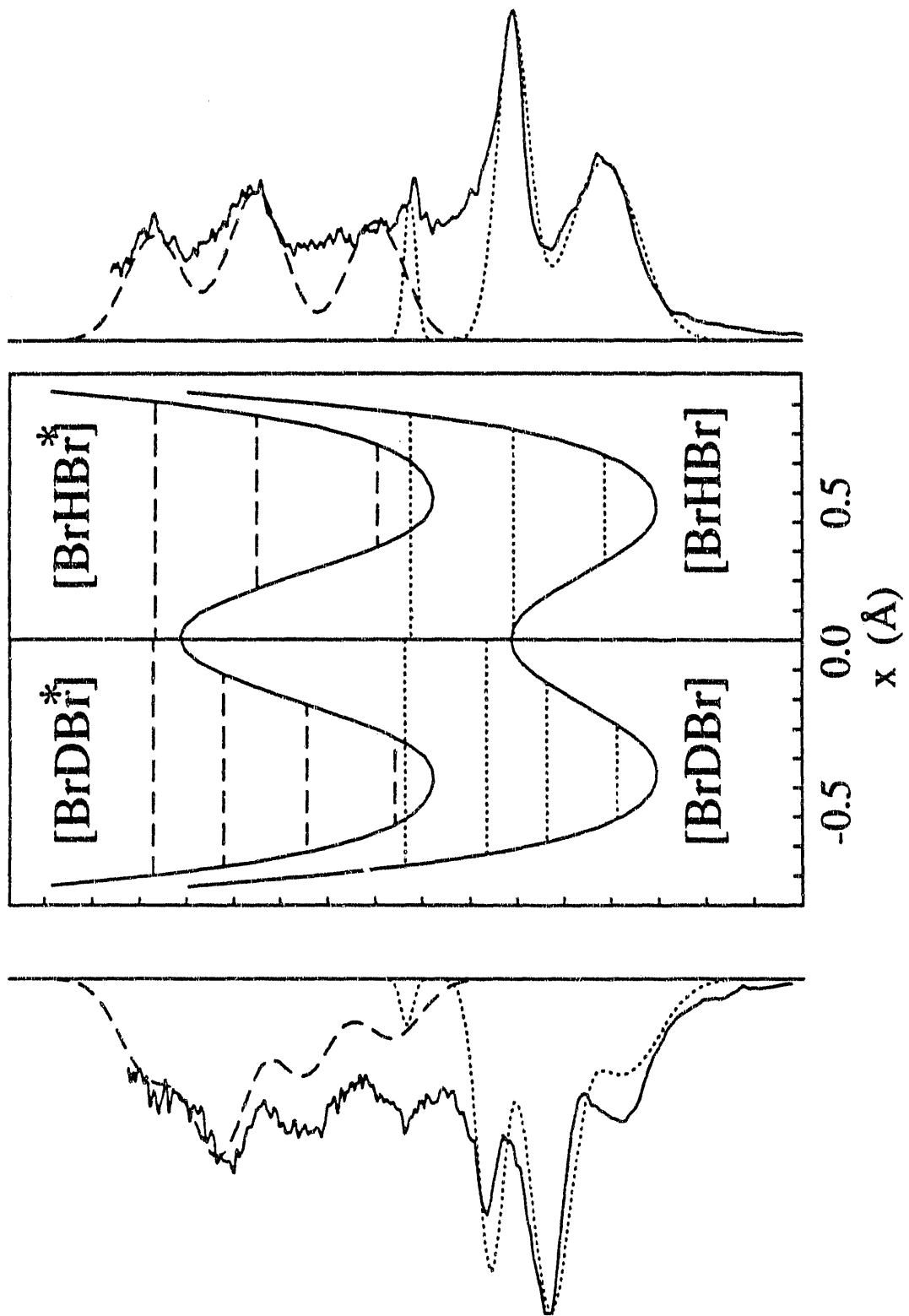


Figure 2-8

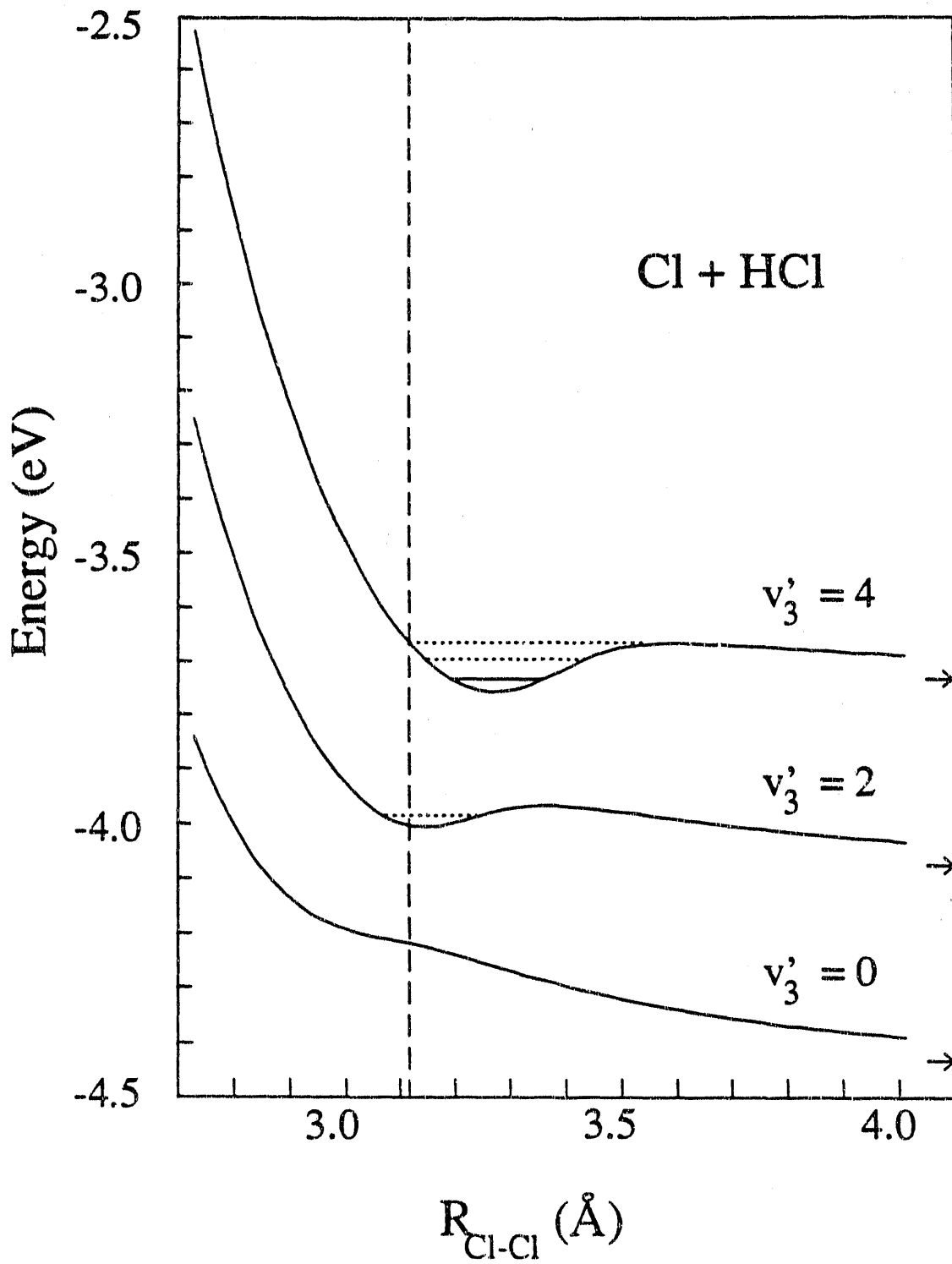


Figure 2-9

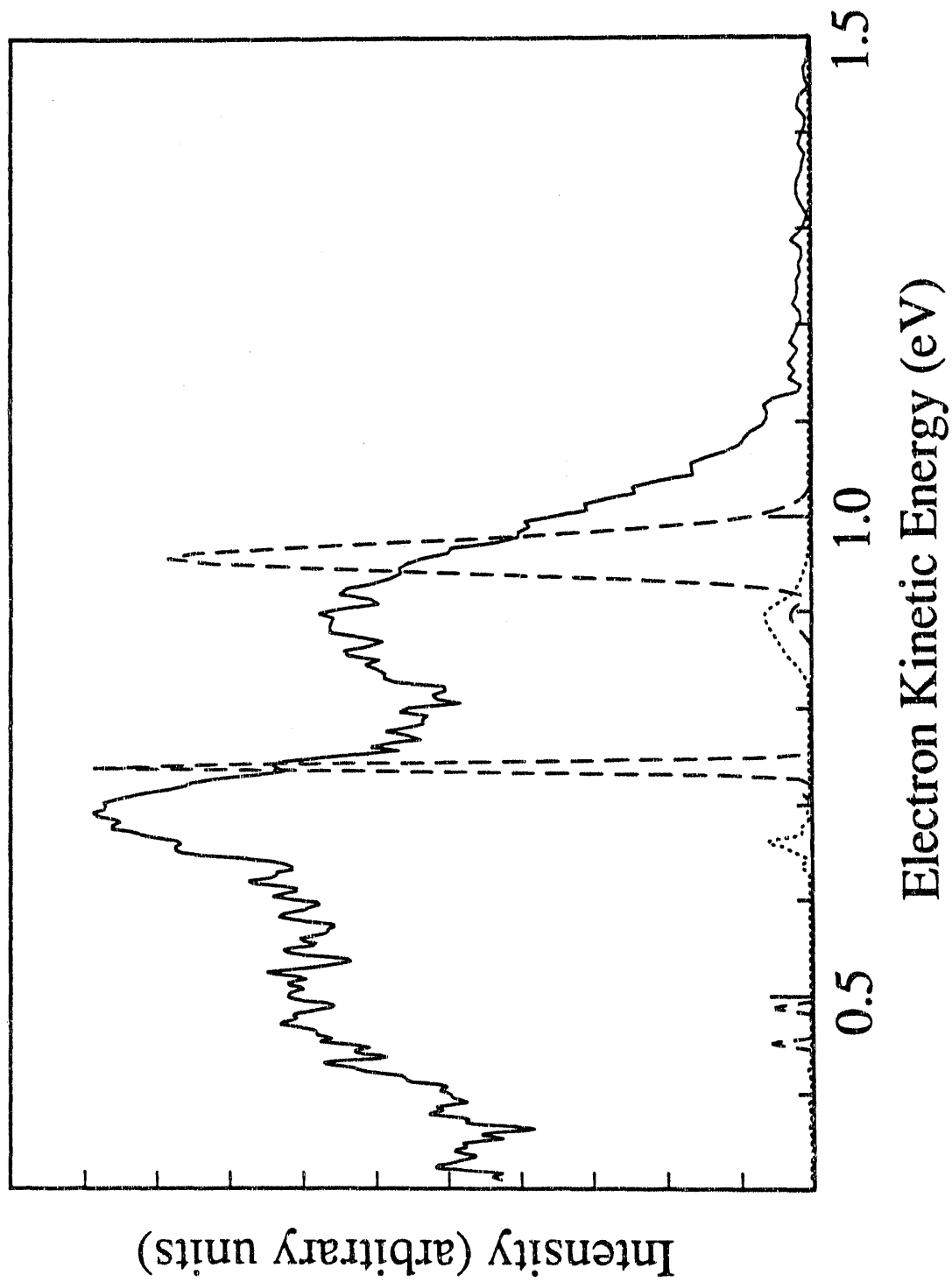


Figure 2-10

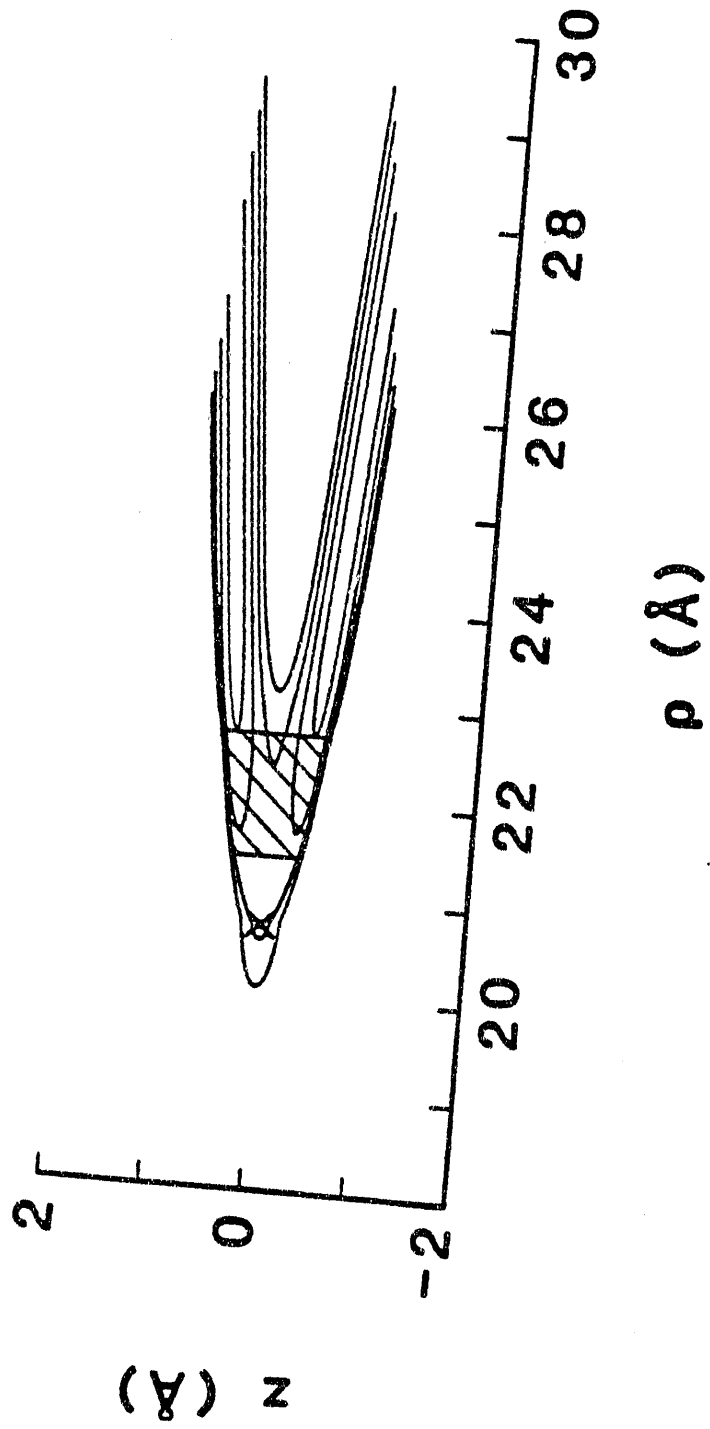


Figure 2-11a

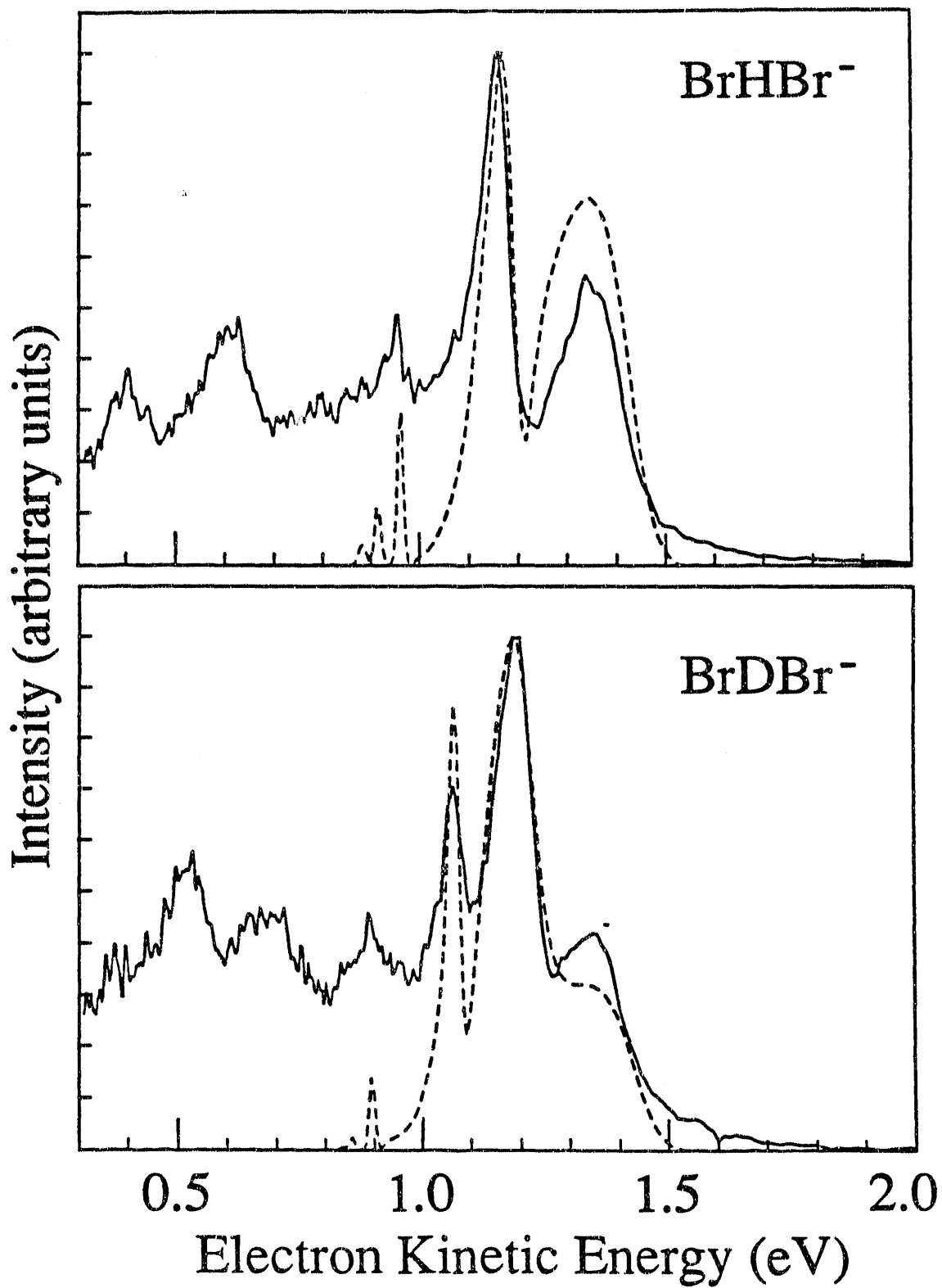


Figure 2-11b

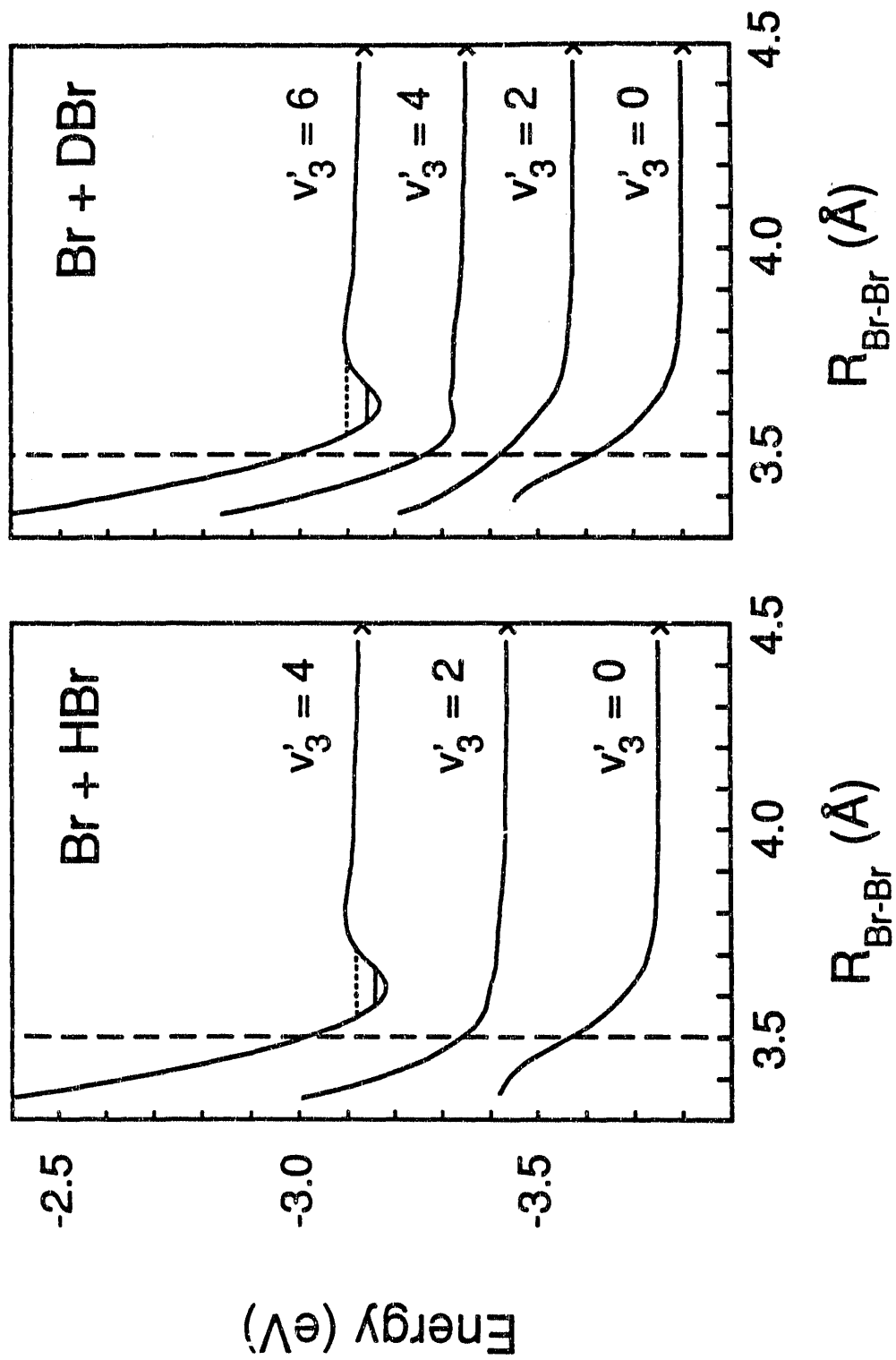


Figure 2-12

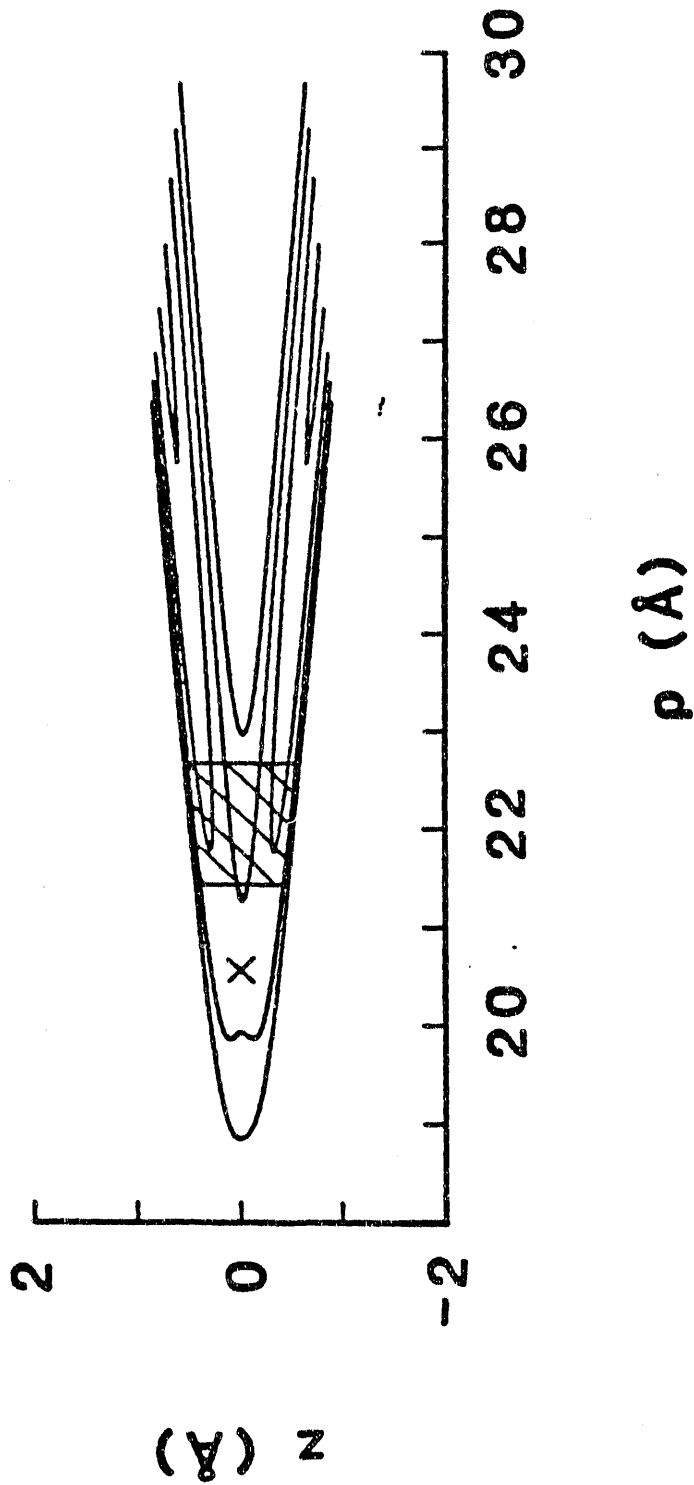


Figure 2-13a

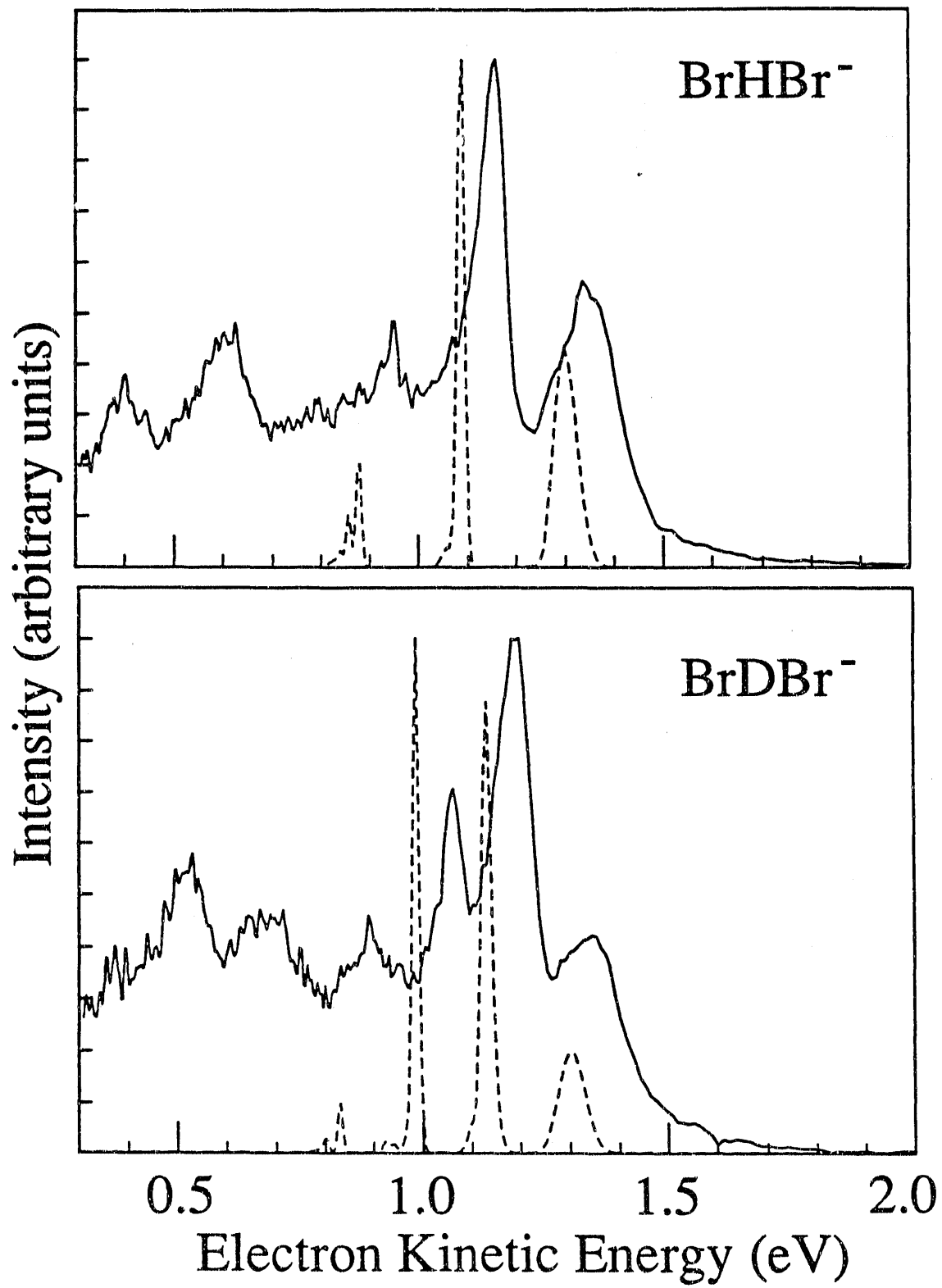


Figure 2-13b

Chapter 3. Computational Framework for Analyzing Photoelectron Spectra of the Bihalide Anions

1. Introduction

The photoelectron spectrum of the M^- anion will exhibit structure due to the potential energy surface for the corresponding neutral M . Within the Franck-Condon approximation, the intensity of a peak observed at an energy E in the photoelectron spectrum is given by

$$I \propto |\langle \Psi^i | \Psi^f(E) \rangle|^2 \quad (1)$$

where Ψ^i is the initial anion $v = 0$ wavefunction and $\Psi^f(E)$ is the wavefunction on the neutral potential energy surface with energy E . Calculating Franck-Condon factors for the case where M^- and M are both bound molecules (and hence Ψ^i and Ψ^f are both bound state wavefunctions) is a very well-studied problem. Analytic formulae for the Franck-Condon factors where Ψ^i and Ψ^f are both one-dimensional harmonic or Morse oscillators have been known since the 1930's.¹ Polyatomic molecules are usually treated within the normal mode approximation.² If the normal modes are assumed to be uncoupled, the vibrational wavefunction can be written as a product of one-dimensional wavefunctions along each normal mode.

$$\Psi^i = \prod_n \psi_{N_n^i}^i(Q_n^i) \quad (2)$$

and

$$\Psi^f = \prod_n \psi_{N_n^f}^f(Q_n^f) \quad (3)$$

where the Q 's denote normal modes and N_n is the number of quanta in mode n . The parallel mode approximation is usually made, in which the form of the normal modes is assumed to be the same for the initial and final states. This leads to a simple expression for the overall Franck-Condon factor in terms of one-dimensional Franck-Condon factors.

$$\langle \Psi^i | \Psi^f \rangle = \prod_n \langle \psi_{N_n^i}^i | \psi_{N_n^f}^f \rangle \quad (4)$$

This treatment is perfectly adequate in the vast majority of cases. For example, in Chapter 5 it is used to treat the progressions in the Q_3 and Q_4 modes in the CH_2NO_2^- photoelectron spectrum.

In some cases, the parallel mode approximation is not invoked. In this case, the normal modes for the anion are written in terms of the normal modes for the neutral

$$Q^i = JQ^f + K^i \quad (5)$$

where Q is a vector of the normal modes, J is the Duschinsky rotation matrix,³ and K^i is the vector of geometry displacements in terms of normal modes of the anion. For example, this approach has been used in the

analysis of the NO_2^- photoelectron spectrum.⁴ Analytical expressions are available for the Franck-Condon factors if harmonic oscillators are used along each normal mode.^{5,6,7}

While there exist analytic expressions for the Franck-Condon factors for bound \rightarrow bound transitions involving harmonic or Morse oscillators, these expressions are not available for more complex potentials and the Schrodinger equation must be explicitly solved to determine the wavefunctions and calculate the overlap integral (1). Two systems discussed at length in later chapters require explicitly solving the Schrodinger Equation. The torsion vibration of CH_2NO_2 involves motion along the periodic potential

$$V(\phi) = \frac{b}{2} [1 - \cos(2\phi)] \quad (6)$$

which is very non-harmonic when b is small. Treatment of this motion will be discussed briefly in Section 4 and in detail in Chapter 5. A more challenging case is furnished by the photoelectron spectra of the bihalide anions XHY^- , where X and Y are halogen atoms.

2. Adiabatic Methods

Photodetachment of XHY^- accesses the potential energy surface for the $\text{X} + \text{HY} \rightarrow \text{XH} + \text{Y}$ reaction. The potential is far from harmonic and is

unbound along the reaction coordinate. The conceptual framework for treating these reactions was described in detail in Chapter 2, so it will only be briefly outlined here. We take advantage of the difference in time scales between the fast hydrogen atom motion (the antisymmetric stretch and bend for systems with $X = Y$) and the slow halogen atom motion (the symmetric stretch for systems with $X = Y$) and invoke a Born-Oppenheimer style separation.^{8,9,10}

$$\Psi(z,R) = \Delta(R) \Gamma(z;R) \quad (7)$$

R , the interhalogen distance, is approximately the reaction coordinate and z is the hydrogen coordinate(s). For collinear simulations (Ch. 2), where the hydrogen atom can only move along the X-Y axis, z is a single coordinate z , which describes hydrogen motion along the X-Y axis. For three-dimensional ($J=0$) simulations (Ch. 4), where the hydrogen atom is free to move anywhere in space, z is two coordinates r and z , where r , the square of the distance from the hydrogen atom to the X-Y axis, describes motion of the hydrogen atom away from the X-Y axis, as in bending or a hindered rotation. The wavefunctions for hydrogen atom motion $\Gamma(z;R)$ are found by solving the Schrodinger Equation for hydrogen motion

$$[T_H(z) + V(z;R)] \Gamma_k(z;R) = U_k(R) \Gamma_k(z;R) \quad (8)$$

at many values of R . As R , not z is the dissociation coordinate, (8) is a bound state Schrodinger Equation. We isolate the scattering coordinate and

treat the remaining, bound, coordinates. Thus, for the collinear (two-dimensional) simulations, (8) is a one-dimensional Schrodinger Equation, while for the three-dimensional case, (8) is a two-dimensional Schrodinger Equation. The eigenvalues $U_k(R)$ for hydrogen atom motion are found at many values of R . They form adiabatic curves that govern the motion of the halogen atoms, much as solution of the electronic Schrodinger Equation within the Born-Oppenheimer approximation forms electronic potential curves that govern motion of the atoms. In the collinear case, each adiabatic curve correlates to a product vibrational state ($X + HY$ (v) or XH (v) + Y), while in the three-dimensional case it correlates to a ro-vibrational state ($X + HY$ (v, j) or XH (v, j) + Y). The halogen atoms obey the halogen atom Schrodinger equation

$$[T_x(R) + U_k(R)] \Delta_{k,h}(R) = E_{k,h} \Delta_{k,h}(R) \quad (9)$$

The adiabatic curves can be purely repulsive or can have one or more minima. Thus, they can support bound states, shape resonances and scattering states.

In order to simulate the photoelectron spectrum we :

- 1) Solve the one- or two-dimensional bound Schrodinger Equation for hydrogen atom motion (8) at several (~ 40) values of R to obtain hydrogen wavefunctions and corresponding eigenvalues, which form adiabatic curves for halogen atom motion.

- 2) Solve the one-dimensional Schrodinger Equation (9) on each adiabatic curve to find halogen atom bound, resonance and scattering states.
- 3) Calculate the Franck-Condon factor with the $v = 0$ state of the anion for each of these states.

We have used two different approaches to step 1, above. We used a relaxation method in our collinear study of the BrHBr^- photoelectron spectrum (Chapter 2) and we have used a matrix method (the Discrete Variable Representation (DVR)) in subsequent collinear and three-dimensional studies of BrHBr^- , BrHI^- and IHI^- (Chapter 4). We have also used two approaches to step 2, above. In the BrHBr^- study we found scattering wavefunctions as a function of energy using a Numerov algorithm and, if the adiabatic curve supported bound or resonance states, found these variationally. In the later study we used L^2 wavefunctions and determined both bound and scattering states using the DVR method. These methods will now be compared and described in detail.

3. Solving the Schrodinger Equation Using Relaxation

The relaxation method takes advantage of the observation that the potential $V(\mathbf{z}, \mathbf{R} + \Delta\mathbf{R}) \approx V(\mathbf{z}, \mathbf{R})$, so the wavefunction $\Gamma_{\mathbf{k}}(\mathbf{z}; \mathbf{R} + \Delta\mathbf{R}) \approx \Gamma_{\mathbf{k}}(\mathbf{z}; \mathbf{R})$.

The method is implemented as follows :

- 1) At the smallest value of R (R_{\min}) the potential $V(z, R_{\min})$ has only a single minimum and a harmonic oscillator is used as the initial guess for the wavefunction $\Gamma_k(z, R_{\min})$.
- 2) Under the relaxation procedure¹¹ the one-dimensional Schrodinger Equation (8) is converted to a finite difference equation on a grid of ~100 equally spaced points and the trial equation "relaxes" to the true wavefunction.
- 3) The solution at this value of R provides an initial guess for Γ_k at the next value of R .
- 4) Steps 2 and 3 are repeated until the adiabatic curve Γ_k has been determined at all desired values of R .
- 5) Steps 1 - 4 are carried out for each desired adiabatic curve.

In the case of BrHBr^- 3 adiabatic curves were generated in this way, corresponding to states of *gerade* symmetry correlating to $\text{Br} + \text{HBr}$ ($v = 0, 1, 2$). There is extremely little Franck-Condon overlap with higher-lying states, and the *gerade* BrHBr^- ($v=0$) wavefunction has no Franck-Condon overlap with *ungerade* states on the $\text{Br} + \text{HBr}$ potential. BrDBr^- requires 4 adiabatic curves.

While the relaxation method is based on a nice physical picture it suffers from a serious deficiency. Adiabatic curves for nonsymmetrical systems (such as $\text{Br} + \text{HI}$) often have avoided crossings. Near an avoided crossing the wavefunctions $\Gamma_k(z, R)$ and $\Gamma_{k+1}(z, R)$ are very similar and often

differ little in energy. At an avoided crossing point the relaxation method will often jump from the k th to the $k+1$ th adiabatic curve. This "feature" makes the relaxation method useless for asymmetric systems. In three dimensions, the adiabatic curves undergo many avoided crossings, even for symmetrical systems. Thus, the relaxation method is limited to collinear simulations of symmetric systems.

Once this limitation of the relaxation method was discovered we turned to matrix methods of solving the Schrodinger Equation for hydrogen motion. These methods offer the advantage that *all* adiabatic curves are found simultaneously at each value of R . This makes matrix methods well suited to systems with avoided crossings, as both curves involved in the avoided crossing will be calculated. We will now briefly review matrix methods and then will describe the DVR method, which is the method we adopted.

4. Matrix Approaches to the Schrodinger Equation

In matrix methods the solutions to the Schrodinger Equation are expressed in terms of the basis functions $\Xi_i(z)$.

$$\Gamma_n(z) = \sum_{i=1}^N c_{n,i} \Xi_i(z) \quad (10)$$

Solving the Schrodinger equation consists of determining the expansion coefficients $c_{n,i}$. Projecting the Schrodinger Equation over the basis set gives the Rayleigh-Ritz-Galerkin equation¹²

$$(H - ES)c = 0 \quad (11)$$

where H is the Hamiltonian matrix

$$H_{i,j} = \langle \Xi_i | (T + V) | \Xi_j \rangle \quad (12)$$

and S is the overlap matrix

$$S_{i,j} = \langle \Xi_i | \Xi_j \rangle \quad (13)$$

If the basis functions are orthogonal and normalized

$$\langle \Xi_i | \Xi_j \rangle = \delta_{i,j} \quad (14)$$

the Schrodinger Equation becomes

$$(H - E)c = 0 \quad (15)$$

Standard subroutines are available to solve (15) (ie, diagonalize the Hamiltonian matrix) to obtain the eigenfunctions and eigenvalues.¹³

The key to matrix methods lies in choosing the basis functions Ξ_i and constructing the Hamiltonian matrix. If we choose our basis to be the solutions to the Schrodinger Equation with the particular reference potential V_{ref}

$$(T + V_{ref}) |\Xi_j\rangle = E_j^0 |\Xi_j\rangle \quad (16)$$

the matrix elements of the Hamiltonian are

$$\begin{aligned} H_{i,j} &= \langle \Xi_i | (T + V) | \Xi_j \rangle \\ &= \langle \Xi_i | (T + V_{ref} + \Delta V) | \Xi_j \rangle \\ &= \langle \Xi_i | E_i^0 | \Xi_j \rangle + \langle \Xi_i | \Delta V | \Xi_j \rangle \\ &= E_i^0 \delta_{i,j} + \Delta V_{i,j} \end{aligned} \quad (17)$$

where

$$\Delta V_{i,j} = \langle \Xi_i | \Delta V | \Xi_j \rangle \quad (18)$$

All of our studies use basis sets that follow these assumptions, although basis sets that do not follow them have been used by others.¹⁴

In some very special cases, the matrix elements $\Delta V_{i,j}$ can be found analytically. The torsion potential of CH_2NO_2 is one such textbook case (see Chapter 5, Section 4.4.1). Unfortunately, textbook cases rarely arise. In the more general case, the integral $\Delta V_{i,j}$ must be calculated numerically. Calculating matrix elements of ΔV in this representation (called the Finite Basis Representation (FBR)) is often the most time-consuming part of the calculation,¹² as $\frac{1}{2} N(N+1)$ integrals have to be computed, where N is the size of the basis. Many techniques have been used to evaluate these integrals. They usually require evaluation of the basis functions and potential over a grid of ≥ 100 points along each coordinate (ie, 10,000 points in the two-dimensional case). This is especially time-consuming in the

three-dimensional simulation of the photoelectron spectrum, where some 500,000 two-dimensional integrals have to be determined.

5. Solving the Schrodinger Equation in the DVR

5.1 One-Dimensional DVR Methods

This shortcoming of the FBR prompts our interest in the Discrete Variable Representation (DVR).¹⁵ Evaluating the Hamiltonian matrix is much faster in the DVR than in the FBR as the ΔV integral is calculated approximately, by evaluating the potential at only a few (N) points. However, these are the best N points for the basis set being used. If the ΔV integral is computed exactly, then the calculation is variational ie, the eigenvalues found will be upper bounds to the true eigenvalues. DVR methods exchange this variational property for a great increase in speed. An early version of the DVR method was proposed by Harris, Engerholm and Gwinn¹⁶ (whose version is often called the HEG method) and has been extended and thoroughly described by Light;¹⁵ a simpler and less formal description will now be given. While the next section contains many equations, it should be emphasized that the DVR is very easy to implement. Most of the equations are used to make the connection between the DVR and the more familiar FBR clear.

The philosophy behind the DVR can be explained in the following, somewhat schematic, manner. The difficulty in the FBR lies in calculating the ΔV integral (18)^{12,15}. We wish to use a unitary transformation to convert the FBR basis E_i to a basis in which the ΔV integral can be easily evaluated. This will make the kinetic energy part of the Hamiltonian more complex than in the FBR, but as we are using a unitary transformation, evaluating the kinetic energy term will still be fairly easy. The key to the DVR method lies in the converting to the position representation. The infinite pointwise (position) representation is defined by

$$\hat{z}|z'\rangle = z'|z'\rangle \quad (19)$$

where $\hat{z} = z$ is the position operator and

$$\langle z'|z''\rangle = \delta(z'' - z') \quad (20)$$

The potential difference ΔV is a function of z , so matrix elements of ΔV are particularly simple to evaluate in the position representation

$$\begin{aligned} \langle z'|\Delta V(z)|z''\rangle &= \langle z'|\Delta V(z')|z''\rangle \\ &= \Delta V(z')\langle z'|z''\rangle \\ &= \Delta V(z')\delta(z'' - z') \end{aligned} \quad (21)$$

The ΔV matrix is thus *extremely* easy to evaluate in this basis - it is a diagonal matrix and only requires evaluating the potential at the point z' .

We now convert from the infinite pointwise representation to a finite pointwise representation that can be implemented on computers with a

finite amount of memory. We define a finite, pointwise representation (the DVR) $|z_\alpha\rangle$ via

$$\hat{z}|z_\alpha\rangle = z_\alpha|z_\alpha\rangle \quad (22)$$

and

$$\langle z_\alpha|z_{\alpha'}\rangle = \delta_{\alpha,\alpha'} \quad (23)$$

The potential difference ΔV is then *approximated* in the DVR by

$$\begin{aligned} \langle z_\alpha|\Delta V(z)|z_{\alpha'}\rangle &\approx \langle z_\alpha|\Delta V(z_\alpha)|z_{\alpha'}\rangle \\ &= \Delta V(z_\alpha)\langle z_\alpha|z_{\alpha'}\rangle \\ &= \Delta V(z_\alpha)\delta_{\alpha,\alpha'} \end{aligned} \quad (24)$$

The ΔV matrix is diagonal and the elements are extremely easily evaluated, so constructing the Hamiltonian in the DVR is very efficient.

We construct the DVR as follows:

- 1) Choose the basis Ξ . The easiest way to do this is to find a exactly-soluble reference problem similar to the problem you wish to solve - the basis will then be the solutions to the reference problem. The greater the similarity between the reference potential and the potential of interest, the smaller the basis that will be required. The collinear calculations use a harmonic oscillator basis. At large R , the actual potential along the z coordinate is a double-minimum potential, which is not well described by a harmonic oscillator, so some 70 basis functions are required. Colbert and Miller have described an alternative construction of the DVR that does not involve an underlying basis.¹⁷

- 2) Find the eigenvalues and eigenvectors of the position matrix \mathbf{X} in the basis set Ξ .

$$X_{i,j} = \langle \Xi_i(x) | x | \Xi_j(x) \rangle \quad (25)$$

For the harmonic oscillator basis, these matrix elements can be found using raising and lowering operators and are¹⁸

$$X_{i,j} = \sqrt{\frac{i}{2}} \delta_{i,j+1} + \sqrt{\frac{i+1}{2}} \delta_{i,j-1} \quad (26)$$

Diagonalize the position matrix \mathbf{X} to obtain \mathbf{z} (the diagonal matrix containing z_α , the eigenvalues of \mathbf{X}) and the eigenvector matrix \mathbf{T} where

$$\mathbf{T}^T \mathbf{X} \mathbf{T} = \mathbf{z} \quad (27)$$

The unitary matrix \mathbf{T} converts from the Finite Basis Representation (FBR) with basis functions $|\Xi_i\rangle$ to the Discrete Variable Representation with basis functions $|z_\alpha\rangle$

$$|z_\alpha\rangle = \sum_i T_{i,\alpha} |\Xi_i\rangle \quad (28)$$

- 3) Evaluate the Hamiltonian matrix in the DVR. We will now derive the expression for matrix elements of the Hamiltonian in the DVR. Recall that in the FBR we had

$$H_{i,j} = \langle \Xi_i | H | \Xi_j \rangle = E_i^0 \delta_{i,j} + \langle \Xi_i | \Delta V | \Xi_j \rangle \quad (29)$$

In the DVR matrix elements of the Hamiltonian are given by

$$\begin{aligned} \langle z_\alpha | H | z_{\alpha'} \rangle &= \sum_{i,j} \langle z_\alpha | \Xi_i \rangle \langle \Xi_i | (T + V_{ref}) | \Xi_j \rangle \langle \Xi_j | z_{\alpha'} \rangle + \langle z_\alpha | \Delta V | z_{\alpha'} \rangle \\ &= \sum_{i,j} \langle z_\alpha | \Xi_i \rangle E_i^0 \delta_{i,j} \langle \Xi_j | z_{\alpha'} \rangle + \langle z_\alpha | \Delta V | z_{\alpha'} \rangle \\ &= \sum_i \langle z_\alpha | \Xi_i \rangle E_i^0 \langle \Xi_i | z_{\alpha'} \rangle + \langle z_\alpha | \Delta V | z_{\alpha'} \rangle \\ &= K_{\alpha,\alpha'} + \langle z_\alpha | \Delta V | z_{\alpha'} \rangle \end{aligned} \quad (30)$$

where

$$K_{\alpha,\alpha'} = \sum_i \langle z_\alpha | \Xi_i \rangle E_i^0 \langle \Xi_i | z_{\alpha'} \rangle \quad (31)$$

or, in matrix notation,

$$K = T^T E^0 T \quad (32)$$

where E^0 is the diagonal matrix of eigenvalues of the reference Hamiltonian. Thus far, we have made no approximations that are not made in the FBR. We now evaluate ΔV in the DVR, so the Hamiltonian matrix in the DVR is

$$\begin{aligned} H_{\alpha,\alpha'} &= \langle z_\alpha | H | z_{\alpha'} \rangle \\ &= K_{\alpha,\alpha'} + \Delta V(z_\alpha) \delta_{\alpha,\alpha'} \end{aligned} \quad (33)$$

The kinetic energy term in the Hamiltonian is slightly more complicated in the DVR than in the FBR, while the potential energy term is much simpler in the DVR. The kinetic energy is found exactly in the DVR, while the potential energy term has been approximated.

Note that the annoying integral over ΔV , which had to be carried out $\frac{1}{2} N(N+1)$ times in the FBR has now been replaced by only evaluating the potential at N points. This, and the fact that the matrix \mathbf{K} need only be found once for a particular type of basis, leads to the great speed of the DVR method. Although the adiabatic method involves solving the Schrodinger Equation for hydrogen motion (8) at some 40 values of R , the \mathbf{X} matrix is only calculated and diagonalized once to obtain the \mathbf{T} matrix. The DVR provides even more advantages in the multi-dimensional case (see below). This great increase in speed is achieved with only a minor drawback - the ΔV integral has been approximated, so the method is no longer variational - the eigenvalues obtained can lie above or below the true eigenvalues. Dickinson and Certain¹⁹ have shown that, if the basis consists of the product of a classical polynomial and its associated weighting function (as, for example, the harmonic oscillator basis consists of the product of a Hermite polynomial and a Gaussian), then evaluating the potential energy difference ΔV at the DVR points z_α is equivalent to approximating the matrix elements of the potential difference using an N point Gaussian quadrature. Thus,

$$T_{i,j} = \sum_{\alpha} (z_{\alpha}) w_{\alpha}^{1/2} \quad (34)$$

the points z_{α} are Gauss-Hermite quadrature points, and w_{α} are the corresponding weights. Note that the quadrature is optimum for the basis set.

- 4) The Hamiltonian matrix is then diagonalized to find the eigenvalues U_k and eigenvectors Γ_k .

As has been pointed out by Light,¹⁶ in the one-dimensional case the DVR method also provides a simple and efficient method to optimize the basis set Ξ_i being used. The procedure is simply to vary the basis set parameters to minimize the trace of the Hamiltonian matrix in the DVR. Note that any change in the basis that is equivalent to scaling the axes (as, for example, changing the oscillator frequency) does not affect \mathbf{T} and simply scales the points z_α , so evaluating the trace of \mathbf{H} does not require any extra matrix diagonalization and takes only a small fraction of the time it would take to diagonalize \mathbf{H} . In the collinear simulations of the XHY^- photoelectron spectra, the basis set frequency is optimized at R_{max} , the largest interhalogen distance at which adiabatic curves will be found (specified by the user) at the beginning of the calculation by varying the frequency until the trace of \mathbf{H} goes through a minimum; this value of the basis set frequency is used at all R .

5.2 Incorporating Symmetry in the DVR

For symmetric systems, XHX^- , the adiabatic curves for the $\text{X} + \text{HX}$ reaction will be either symmetric (*gerade*) or antisymmetric (*ungerade*) with

respect to interchange of the nuclei. As the $v=0$ XHX^- wavefunction has *gerade* symmetry, it will only have Franck-Condon overlap with neutral wavefunctions with *gerade* symmetry. For these symmetric systems it is desirable to incorporate symmetry in the basis, as this halves the size of the basis along the z direction (reducing the time required to calculate adiabatic curves by a factor of about 8, as matrix diagonalization is an N^3 process, for large N) and also giving about half the number of adiabatic curves, speeding the calculation of the halogen atom states by about a factor of two.

Incorporating symmetry in the DVR is quite easy, and we will explain how this is done, using the case where we wish to determine only functions of *gerade* symmetry. Light has discussed the general construction of a symmetry adapted DVR.²⁰

Both the reference eigenvalue matrix \mathbf{E}^0 and the weight matrix \mathbf{T} must be converted to the symmetry-adapted versions \mathbf{E}^{0*} and \mathbf{T}^* . As we are using only even basis functions (in the FBR), we have

$$\begin{aligned} E_i^{0*} &= \langle \Xi_{2i} | (T + V_{\text{ref}}) | \Xi_{2i} \rangle \\ E_i^{0*} &= E_{2i}^0 \end{aligned} \quad (35)$$

The DVR points are symmetrically placed about $z=0$ and, as the basis functions have either *gerade* or *ungerade* parity,

$$z_{N-a} = z_a \quad (36)$$

and

$$T_{l,N-\alpha} = T_{l,\alpha} \quad (37)$$

so the transformation matrix becomes

$$T_{l,\alpha}^* = \langle \Sigma_{2l} | z_{\alpha} \rangle \quad (38)$$

T^* is a square matrix, as α ranges from 1 to $N/2$. Incorporating symmetry in the DVR is thus very simple.

5.3 The DVR in Two Dimensions

Using the DVR to set up the Hamiltonian matrix is especially advantageous in the multi-dimensional case. Three-dimensional simulations of the XHY^- photoelectron spectra involve solving the two-dimensional Schrodinger equation for hydrogen atom motion (8). The DVR in two dimensions is the direct product of two one-dimensional DVRs $|z_{\alpha}\rangle$ along z and $|r_{\beta}\rangle$ along r . Here α and α' are indices in the z direction, while β and β' are indices in the r direction.

$$\hat{r} |r_{\beta}\rangle = r_{\beta} |r_{\beta}\rangle \quad (39)$$

In two dimensions, the ΔV matrix becomes

$$\begin{aligned}
\Delta V_{\alpha,\beta;\alpha',\beta'} &= \langle r_\beta | \langle z_\alpha | \Delta V(r,z) | z_{\alpha'} \rangle | r_{\beta'} \rangle \\
&= \langle r_\beta | \Delta V(r, z_\alpha) | r_{\beta'} \rangle \delta_{\alpha,\alpha'} \\
\Delta V_{\alpha,\beta;\alpha',\beta'} &= \Delta V(r_\beta, z_\alpha) \delta_{\alpha,\alpha'} \delta_{\beta,\beta'}
\end{aligned}
\tag{40}$$

again, the ΔV matrix is diagonal and easily evaluated. The position matrix \mathbf{X} is set up and diagonalized *in each coordinate* (to find T_z and T_r). K_z and K_r are found *via*

$$\begin{aligned}
K_z &= T_z^T E_z^0 T_z \\
K_r &= T_r^T E_r^0 T_r
\end{aligned}
\tag{41}$$

The kinetic energy is evaluated along each coordinate separately, so matrix elements of the Hamiltonian are given by

$$H_{\alpha,\beta;\alpha',\beta'} = K_r \delta_{\beta,\beta'} \delta_{\alpha,\alpha'} + K_z \delta_{\alpha,\alpha'} \delta_{\beta,\beta'} + \Delta V(r_\beta, z_\alpha) \delta_{\alpha,\alpha'} \delta_{\beta,\beta'}
\tag{42}$$

Thus, setting up the Hamiltonian is extremely easy, even in the multi-dimensional case.

In addition to ease in constructing the Hamiltonian matrix, the DVR method offers the advantage that only a few of the elements of \mathbf{H} are nonzero. The Hamiltonian matrix in two dimensions contains only $N_r N_z (N_r + N_z - 1)$ nonzero elements. Using $N_r = 20$ and $N_z = 60$, only 6.6% of the elements of \mathbf{H} are nonzero. As \mathbf{H} is sparse (mostly zeroes), \mathbf{H} can be diagonalized using less memory and CPU time than if it were not sparse. Light has also developed the sequential diagonalization/truncation method of diagonalizing \mathbf{H} , which permits the use of extremely large basis sets (~10,000 basis functions).²¹

Use of the DVR for multi-dimensional problems does have a drawback in that it requires the use of a product basis. To see why this is a limitation, consider the simulation of the photoelectron spectrum of IHI^- in three dimensions, using the adiabatic approach (ie, two-dimensional DVR). At large interiodine distances the wavefunctions look like $\text{I} + \text{HI} (v, j)$, so it makes sense to use ro-vibrational states of HI as the basis. This type of basis has been used by Schatz in full 3D ($J=0$) scattering calculations of the IHI^- photoelectron spectrum^{22,23,24} and by Kubach, et. al.^{25,26,27} to construct adiabatic curves for the $\text{I} + \text{HI}$ reaction. Kubach finds that some 70 basis functions are sufficient to converge the wavefunctions for hydrogen atom motion. However, this basis is not a product basis and they determined matrix elements of the potential using quadrature. We use a product basis consisting of harmonic oscillator wavefunctions along z and the radial portion of a two-dimensional harmonic oscillator (with orbital angular momentum $\ell = 0$) along r . Good convergence at large interiodine distance requires 60 basis functions of *gerade* symmetry in z and 20 basis functions in r , or a 1200 element basis, much larger than the 70 element (non-product) basis used by Kubach. In spite of the large difference in basis set sizes, constructing adiabatic curves using the large basis and DVR is faster than using the small basis and finding the ΔV integral using quadrature (as was done by Kubach).

5.4 Scattering States in the DVR

The adiabatic curves that govern halogen atom motion can be purely repulsive (supporting "direct scattering" states), or can contain global minima (perhaps supporting states that are bound within the adiabatic approximation) or local minima (perhaps supporting shape resonance states). Simulating the photoelectron spectrum requires calculating all of these states, and determining their overlap with the anion ground state. In the early collinear work on the Cl + HCl and Br + HBr reactions (Chapter 2) each adiabatic curve was examined and bound states were determined variationally for adiabatic curves with minima, while scattering states were determined as a function of energy using a Numerov procedure²⁸ for repulsive curves. This is a reasonable procedure for the three adiabatic curves found for collinear Br + HBr; it becomes quite unwieldy when applied to the 50 adiabatic curves that are considered for three-dimensional Br + HBr. Recall that, in the three-dimensional case, each adiabatic curve correlates to Br + HBr (v, j). The increase in the number of adiabatic curves is due to the number of rotational levels associated with each vibrational state. Clearly a procedure that treats resonance and scattering states on an equal footing would be preferable. Such a system, the L^2 method, has been developed by Truhlar²⁹ and Bowman^{30,31}.

The L^2 method uses basis functions that are square-integrable (hence the name)

$$\int_0^{\infty} dR P(R) P^*(R) = 1 \quad (43)$$

where $P(R)$ are the basis functions along the interhalogen distance R . This is standard for bound state basis functions, but asymptotic scattering wavefunctions have the form

$$P_s(R) = A e^{ikR} + B e^{-ikR} \quad (44)$$

and are not square-integrable. The L^2 method is well suited to calculating bound and resonance states, as the wavefunctions for these states are localized. Direct scattering states are *approximated* using the localized L^2 basis. This results in a discrete spectrum of eigenvalues, rather than a continuous spectrum. Convoluting the stick spectrum, the intensities of which are determined by calculating the overlap with the anion wavefunction, with a resolution function (for example, the experimental resolution) results in a continuous spectrum. The result of such a simulation, using a basis of 90 harmonic oscillator wavefunctions along R ($N_R = 90$), and carried out on a fitted surface for the $\text{Br} + \text{HBr}$ reaction, is shown in Figure 3-1a. The series of closely-spaced peaks in the 1.2 - 1.5 eV eKE range correspond to scattering from the purely repulsive adiabatic curve labelled $v_3' = 0$ in Figure 2-12. The simulation leaves much to be

desired when compared with the simulation obtained using the Numerov procedure (Figure 2-11).

We now proceed with the L^2 basis in a different direction than that used by Bowman. As we are approximating scattering wavefunctions using a localized basis, the "eigenvalues" of the scattering wavefunctions will change if we change the size of the basis, while (for fairly large basis sets) the eigenvalues of localized resonance states will change only very slightly. This is analogous to invoking a potential $V(R) = \infty$ for $R > R_*$ and then moving R_* . Resonance wavefunctions have little amplitude at large R and changing the potential in this region has little effect on their eigenvalues, while scattering wavefunctions have significant amplitude even at large R and are strongly affected by the potential in this region. Figure 3-1b shows the result of summing 21 simulations using $N_R = 50, 52, \dots, 88, 90$. The $v_3' = 0$ peak has now filled in, while the resonance peaks at 0.90 - 0.95 eV remain narrow. Collinear simulations of the $\text{Br} + \text{HBr}$ reaction are an extreme case - the lowest adiabatic curve is extremely repulsive, so the L^2 eigenvalues are widely spaced and many different basis sets are required for a smooth simulation. Most collinear simulations require only summing five simulations, and all the adiabatic three-dimensional simulations in Chapter 4 involve summing five or six simulations. Fewer simulations need to be added for the three-dimensional simulations, as there are many more (about a factor of 20) adiabatic curves in the same energy range as in the collinear

calculation, so the individual adiabatic curves are much less steep than in the collinear case.

Using the L^2 basis allows us to calculate the states supported by each adiabatic curve without having to first examine each curve to determine the type of states it will support. Also, for the L^2 basis we use a harmonic oscillator basis, so the DVR is constructed just as in the one-dimensional case (Section 5.1), using the adiabatic potential $U_i(R)$ as the potential. Finding the wavefunctions for halogen atom motion and simulating the photoelectron spectrum just involves solving the heavy atom Schrodinger Equation (9) on each adiabatic curve and calculating the Franck-Condon overlap with the anion $v=0$ wavefunction. In the next chapter we use the DVR to simulate the photoelectron spectra of IHI^- , IDI^- , BrHI^- , BrHBr^- , and BrDBr^- using the adiabatic three-dimensional method.

1. E. Hutchisson, Phys. Rev. **36**, 410 (1930); **37**, 45 (1931).
2. E. B. Wilson, Jr., J. C. Decius and P. C. Cross, Molecular Vibrations, Dover, New York (1980).
3. F. Duschinsky, Acta Physicochim. URSS **7**, 551 (1937).
4. K. M. Ervin, J. Ho and W. C. Lineberger, J. Phys. Chem. **92**, 5405 (1988).
5. T. E. Sharp and H. M. Rosenstock, J. Chem. Phys. **41**, 3453 (1964); and corrections in the Appendix of R. Botter, V. H. Dibeler, J. A. Walker and H. M. Rosenstock, J. Chem. Phys. **44**, 1271 (1966).
6. A. Warshel and M. Karplus, Chem. Phys. Lett. **17**, 7 (1972).
7. H. Kupka and P. H. Cribb, J. Chem. Phys. **85**, 1303 (1986).
8. D. K. Bondi, J. N. L. Connor, J. Manz and J. Römelt, Mol. Phys. **50**, 467 (1983).
9. E. Pollak, J. Chem. Phys. **78**, 1228 (1983); J. Manz, R. Meyer, E. Pollak, J. Römelt and H. H. R. Schor, Chem. Phys. **83**, 333 (1984); D. C. Clary and J. N. L. Connor, J. Phys. Chem. **88**, 2758 (1984).
10. G. Hauke, J. Manz and J. Römelt, J. Chem. Phys. **73**, 5040 (1980); A. Kuppermann, J. A. Kaye and J. P. Dwyer, Chem. Phys. Lett. **74**, 257 (1980); V. K. Babamov and R. A. Marcus, J. Chem. Phys. **74**, 1790 (1981); V. Aquilanti, S. Cavalli and A. Lagana, Chem. Phys. Lett. **93**, 179 (1982).
11. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, Numerical Recipes, Cambridge University Press, Cambridge (1986), Ch. 16.3.
12. B. W. Shore, J. Chem. Phys. **59**, 6450 (1973).

13. Subroutines to solve (15), are available in the NAG, EISPACK and Cray math libraries, among others.
14. Z. Bačić and J. C. Light, *J. Chem. Phys.* **85**, 4594; **86**, 3065.
15. J. C. Light, I. P. Hamilton and J. V. Lill, *J. Chem. Phys.* **82**, 1400 (1985).
16. D. O. Harris, G. G. Engerholm and W. D. Gwinn, *J. Chem. Phys.* **43**, 1515 (1965).
17. D. T. Colbert and W. H. Miller, submitted, ????
18. P. W. Atkins, Molecular Quantum Mechanics, Oxford University Press, Oxford (1983), p. 102.
19. A. S. Dickinson and P. R. Certain, *J. Chem. Phys.* **49**, 4209 (1968).
20. R. M. Whitnell and J. C. Light, *J. Chem. Phys.* **89**, 3674 (1988).
21. R. M. Whitnell and J. C. Light, *J. Chem. Phys.* **90**, 1774 (1989); Z. Bačić, R. M. Whitnell, D. Brown and J. C. Light, *Comp. Phys. Comm.* **51**, 35 (1988)
22. G. C. Schatz, *J. Chem. Phys.* **90**, 4847 (1989).
23. G. C. Schatz, *Isr. J. Chem.* **29**, 361 (1989).
24. G. C. Schatz, *J. Phys. Chem.* **94**, 6157 (1990).
25. C. Kubach, *Chem. Phys. Lett.* **164**, 475 (1989).
26. C. Kubach, G. Nguyen Vien and M. Richard-Viard, *J. Chem. Phys.* **94**, 1929 (1991).
27. G. Nguyen Vien, M. Richard-Viard and C. Kubach, *J. Phys. Chem.* **95**, 6067 (1991).

28. R. J. LeRoy, University of Waterloo Chemical Physics Research Report CP-329 (1988).
29. J. G. Lauderdale and D. G. Truhlar, *J. Chem. Phys.* **84**, 192 (1986); S. C. Tucker and D. G. Truhlar, *J. Chem. Phys.* **86**, 6251 (1987).
30. J. M. Bowman and B. Gazdy, *J. Phys. Chem.* **93**, 5129 (1989).
31. B. Gazdy and J. M. Bowman, *J. Chem. Phys.* **91**, 4615 (1989).

Chapter 3 Figure Captions

Figure 3-1 Simulated Collinear Photoelectron spectra of BrHBr^- .

- a) Using a single basis with 90 basis functions in R.
- b) Using 20 basis sets with 52 - 90 basis functions in R. Both simulations have been convoluted with the instrumental resolution function.

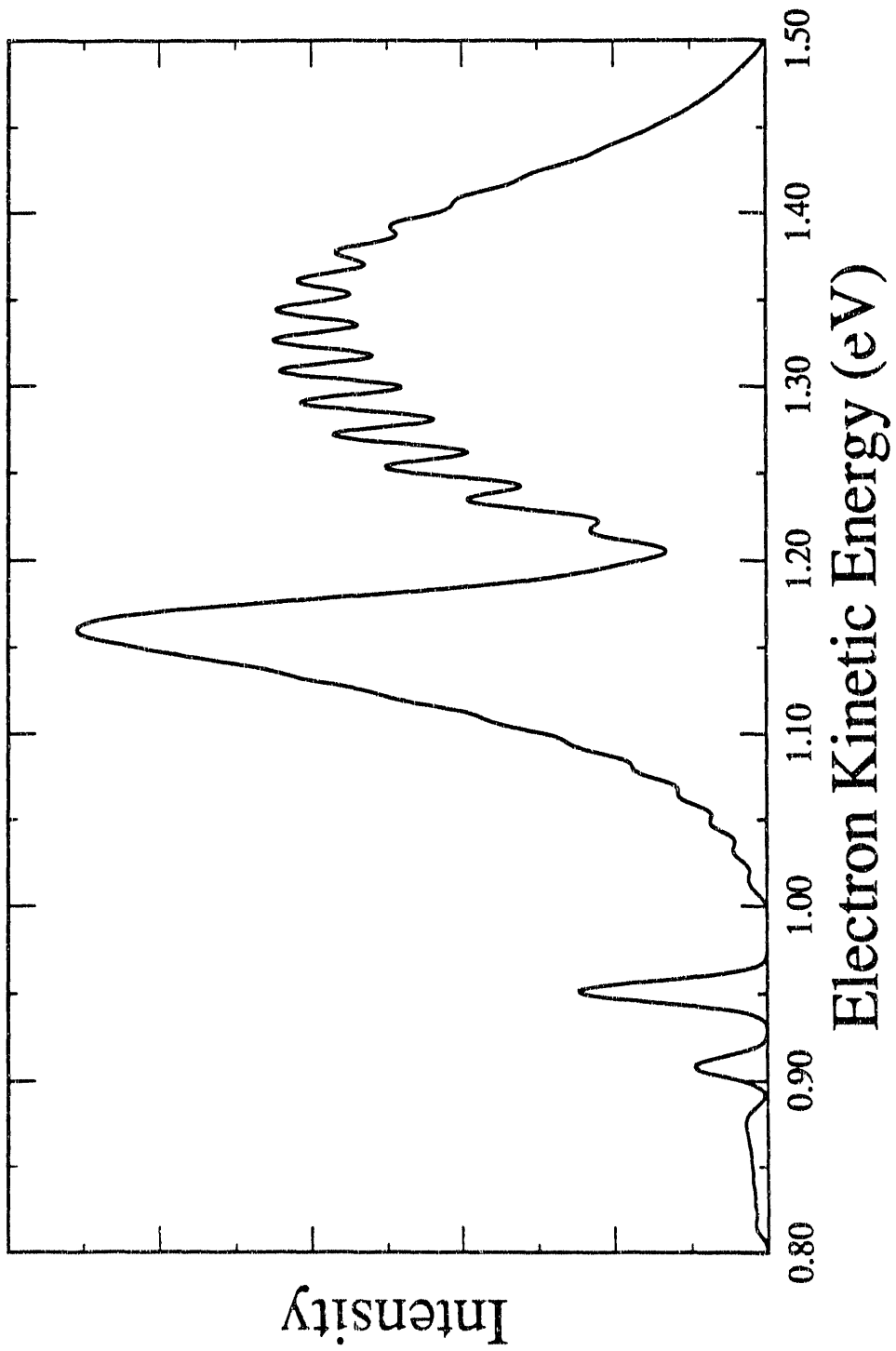


Figure 3-1a

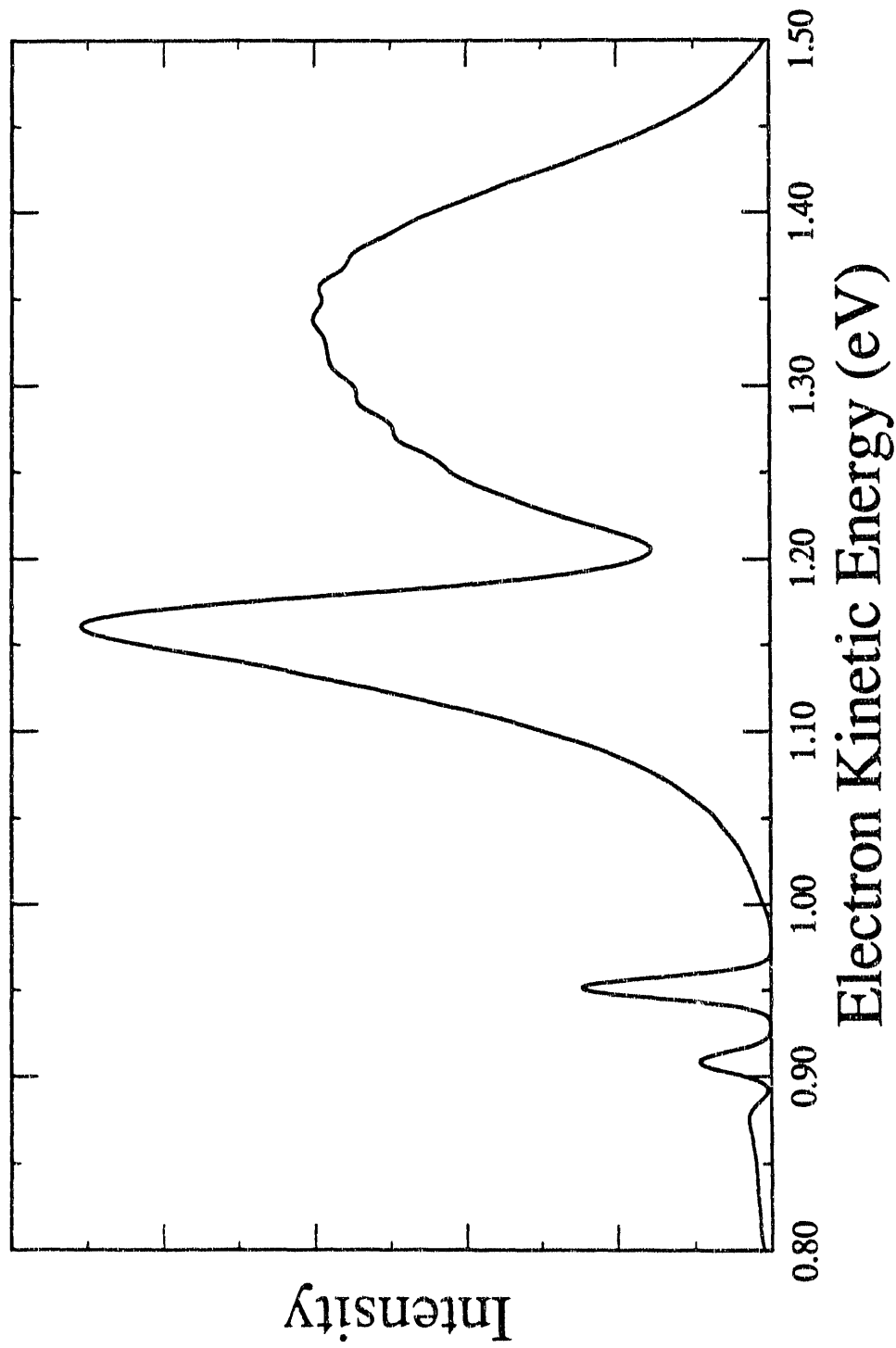


Figure 3-1b

Chapter 4. Adiabatic Three-Dimensional Simulations of the IHI^- , BrHI^- and BrHBr^- Photoelectron Spectra

1. Introduction

The characterization of the transition state region of the potential energy surface for a chemical reaction has been a longstanding goal of chemical dynamics.¹ The transition state region intimately affects the observable features (reaction rate, product energy and angular distributions, etc.) of a chemical reaction. Recently, a variety of experiments² have been performed that seek to directly spectroscopically probe the transition state region of a chemical reaction. In our group we use one of these methods, negative ion photodetachment, to study the transition state region of neutral bimolecular reactions. In these experiments, photodetachment of a stable negative ion is used to access the transition state of a bimolecular reaction in a controlled manner. If the negative ion precursor is similar in structure to the transition state for the neutral reaction, the photoelectron spectrum will contain detailed information on the transition state region of the neutral potential energy surface. In this manner, we have studied the potential energy surface for the hydrogen transfer reactions $\text{X} + \text{HY} \rightarrow \text{XH} + \text{Y}$ by obtaining the photoelectron spectra or the higher-resolution threshold photodetachment spectra of the bihalide anions XHY^- , where X

and Y are like^{3,4,5,6,7} or unlike⁸ halogen atoms. This type of experiment is ideally suited to the detection of long-lived resonance states^{9,10,11} which, by definition, occur only at a particular energy. These states have proven difficult to detect in scattering experiments, where the averaging over impact parameter and, in some cases reactant kinetic energy and orientation tends to obscure sharp resonance structure.^{12,13} The XHY⁻ photoelectron spectra^{5,6} contain structure attributable to resonances, but most of the observed structure consists of broad peaks due to overlap with nonresonant, direct scattering states on the neutral potential energy surface.⁵ Both the resonant and direct scattering features contain information on the potential energy surface for the neutral reaction.

Ideally, one wishes to use the photoelectron spectra to determine the shape of the region of the potential energy surface that has good Franck-Condon overlap with the negative ion. While this problem is simpler than the determination of a full potential energy surface from scattering results, it is by no means easy. The spectroscopic results cannot be directly inverted to give the potential, so we proceed iteratively; we simulate the photoelectron spectrum assuming a particular neutral potential energy surface and anion geometry. The neutral potential is then modified and the procedure is repeated to obtain improved agreement with experimental results. This procedure requires an accurate, efficient method of simulating the photoelectron spectrum. Our experiments have stimulated a number of

theoretical simulations of the photoelectron spectra of the bihalide anions, using a variety of different methods. Of special note are Schatz's 3D scattering simulations of the IHI^- ,^{14,15,16,17} IDI^- ¹⁸ and ClHCl^- ^{19,20,21} photoelectron spectra; Bowman's collinear and 3D L^2 simulations of the IHI^- ^{22,23} and ClHCl^- ^{22,23,24} spectra, and Zhang and Miller's 3D scattering simulations of the FH_2^- photoelectron spectrum.²⁵ These methods have been thoroughly covered in a recent review of the theory of photodetachment spectra,¹⁴ so we will only review the most recent results and those relating directly to this work.

The high-resolution threshold photodetachment spectrum⁶ of IHI^- is shown in Fig. 4-1, along with the lower-resolution photoelectron spectrum.⁴ The peaks are labelled by the antisymmetric stretch ν_3 quantum number of the neutral IHI complex. The threshold photodetachment spectrum shows that the $\nu_3 = 2$ and $\nu_3 = 4$ features consist of progressions of narrow peaks spaced by about 100 cm^{-1} . On the basis of collinear simulations⁷ and Schatz's three-dimensional ($J=0$) scattering simulations^{14,15,16} of the photoelectron spectrum we have assigned these features to resonance states supported by the potential energy surface for the $\text{I} + \text{HI}$ reaction. On the other hand, the $\nu_3 = 0$ manifold consists of a series of fairly broad peaks. On the basis of the peak spacings, these were assigned to hindered rotor states of the $[\text{IHI}]$ complex in which the hydrogen atom orbits each of the iodine atoms. These "rotational threshold" features appear in the three-

dimensional scattering simulations of Schatz, but are *not* reproduced in collinear simulations. On the other hand, the three-dimensional scattering calculations give the exact simulation of the photoelectron spectrum for a given potential energy surface. Any discrepancies between these results and the experimental results are most likely due to deficiencies in the potential energy surface, although they could be due to contributions to the experimental spectrum from states with $J > 0$. While the scattering calculations give the "correct" answer, they are too time-consuming to use in an iterative procedure to fit a potential energy surface for the neutral reaction to the experimental photoelectron spectrum.

What is required is an approximate three-dimensional method that can reproduce all the features observed in the spectrum, yet is much less computationally demanding than the full scattering calculation. One likely candidate is an adiabatic three-dimensional approach. The physical motivation for this method is based on the observation that in the bihalide reactions, the (light) hydrogen atom quickly shuttles between the two (heavy) halogen atoms as the complex falls apart following photodetachment. Motion of the halogen atoms, which is the dissociation coordinate, occurs much more slowly than motion of the hydrogen atom, so photoelectron spectra of the bihalides show a resolved progression in the hydrogen motion (antisymmetric stretch) coordinate.⁷ The observed peaks are often broad, due to the limited lifetime of the transition state complex.

In addition, theoretical calculations on model potential energy surfaces predict the existence of long-lived resonance states that support halogen atom (symmetric stretch) vibrations.^{11,14,19,20,21,24,26,27,28,29,30,31,32,33,34,35,36}

We use an idea pioneered by Manz and co-workers^{29,30,31,32,33} and take advantage of the separation of time scales between the fast hydrogen atom motion and slow halogen atom motion to allow us to use the Born-Oppenheimer style separation in the dynamics of the system to construct vibrationally adiabatic curves for the halogen atom motion by determining hydrogenic stretch states as a function of the interhalogen distance. We used this method in an earlier collinear study and found that for IHI^- the adiabatic calculations are in extremely good agreement with full two-dimensional wavepacket calculations.⁷ Nonadiabatic effects are small and can be approximately included using the DIVAH framework.^{11,37}

Kubach has carried out calculations extending the Born-Oppenheimer type separation of time scales to three dimensions.^{38,39,40} At each value of the scattering coordinate (the interiodine distance) the two-dimensional Schrodinger Equation for hydrogen atom motion is solved to find the bend and antisymmetric stretch eigenvalues. These "rotationally and vibrationally adiabatic curves" are then approximately coupled (fully including the coupling is equivalent to Schatz's CCH approach) to form diabatic curves. He finds two types of curves : "efficient" curves, which

correspond to states of the complex where the hydrogen atom spends most of its time between the two iodine atoms and which support the sharp resonance features seen in the spectrum, and the more common "direct scattering" curves which are smoothly repulsive and, as we shall see, are responsible for the broad rotational threshold features. Resonance positions determined by the diabatic approach are in excellent agreement with the full scattering results of Schatz. Unfortunately, Kubach has not simulated the photoelectron spectrum of IHI^- using his method.

The method we present here is very closely related to the adiabatic method of Kubach : adiabatic potential curves governing the iodine motion are formed by solving the two-dimensional Schrodinger Equation for the hydrogenic motion at many values of the interiodine distance. Scattering wavefunctions are then found on these adiabatic curves and their overlap with the anion wavefunction is determined and used to simulate the photoelectron spectrum of IHI^- . Comparison with the exact results of Schatz¹⁴ shows that the adiabatic three-dimensional method is a significant improvement over the collinear calculations, while requiring significantly less computational effort than the full 3D scattering calculation. A full description of the adiabatic three-dimensional method is given in Section 2. Our results on IHI^- and IDI^- photodetachment on the LEPS-A surface are compared to the exact scattering results of Schatz in Sections 3.1 and 3.2. In the remainder of Section 3 we apply our method to the study of several

less well-characterized systems: IHI^- and IDI^- on the more repulsive LEPS-C potential, which Schatz has studied over a limited energy range,¹⁷ BrHI^- , for which only collinear studies are available,⁸ and BrHBr^- and BrDBr^- on a fitted potential surface, extending our earlier collinear work on this system.⁵ Conclusions are presented in Section 4.

2. Method

In this section we discuss the adiabatic three-dimensional method of calculating the photoelectron spectrum and give details on our implementation of the method. We simulate the spectrum within the Franck-Condon Approximation, so the intensity of a peak at energy E is given by the Franck-Condon Factor

$$I \propto |\langle \Psi_0^i | \Psi^f(E) \rangle|^2 \quad (1)$$

where Ψ_0^i is the initial anion wavefunction (assumed to be the vibrational ground state) and $\Psi^f(E)$ is the scattering wavefunction on the neutral surface with energy E . Simulating the spectrum thus consists of calculating the $v=0$ anion wavefunction and the neutral wavefunction as a function of energy and determining this overlap.

Experimental information on the anions is somewhat limited, as there is no gas phase spectroscopic information on any of the anions in this study, although some vibrational frequencies have been determined in a

matrix.⁴¹ The anion potentials are assumed to be separable along all three directions. For BrH(D)Br⁻ and IH(D)I⁻ we use harmonic potentials with the matrix isolation frequencies for the symmetric stretch (ν_1) and antisymmetric stretch (ν_3), and a harmonic bend (ν_2) potential with a frequency of 700 (495) cm⁻¹ for IHI⁻ (IDI⁻). These are the same parameters as were used in the earlier 3D CCH scattering calculations on IHI⁻^{14,15,16} and IDI⁻¹⁸ and allow comparison of the results. For BrHI⁻ we use a Morse potential fit to the ν_3 matrix isolation frequencies for BrHI⁻ and BrDI⁻;⁴¹ this is the same stretching potential as was used in our earlier collinear study.⁸ We use the same value of ν_2 for BrHI⁻ and BrHBr⁻ as for IHI⁻, in the absence of experimental data. The simulations are very sensitive to the anion geometry. IHI⁻ and BrHBr⁻ are thought to be linear and centrosymmetric, but the interhalogen distance has not been experimentally determined in the gas phase for any of the systems described here. The values used are based on educated guesswork with (in the case of BrHBr⁻) some guidance from *ab initio* calculations. The geometry of BrHI⁻ is the same as was used in our earlier collinear study. The parameters used for each negative ion are given in Table 4-1. We use semiempirical LEPS potential surfaces⁴² for the I + HI and Br + HI reactions. For the Br + HBr reaction we will use a fitted semiempirical potential, described in Sec. 3.5.

We now define the coordinates we use to describe the motions of the

three atoms. The potentials for these systems all have cylindrical symmetry, so cylindrical coordinates (Fig. 4-2) are used to exploit this symmetry. The halogen atoms are taken to lie on the z axis, evenly spaced about $z = 0$; the hydrogen atom is a distance ρ from the z axis, and ϕ is the azimuthal angle. The volume element is $d\tau = \rho d\rho dz d\phi$.⁴³ The Hamiltonian is then

$$H = T_H(\rho, z, \phi) + T_X(R) + V(\rho, z, R) \quad (2)$$

where T_H is the hydrogen kinetic energy, T_X is the kinetic energy of the halogen atoms and V is the potential. We denote the solutions to this Hamiltonian $\Psi(\rho, z, \phi, R)$. We now make the adiabatic approximation^{5,11,38,39,40,44} and write

$$\Psi(\rho, z, \phi, R) = \Delta(R) \Gamma(\rho, z, \phi; R) \quad (3)$$

where the wavefunctions for hydrogen atom motion $\Gamma(\rho, z, \phi; R)$ change slowly with R and are found by solving the Schrodinger Equation with the Hamiltonian for hydrogen atom motion

$$H_H = T_H(\rho, z, \phi) + V(\rho, z, R) \quad (4)$$

at fixed R to find the eigenvalues $U_k(R)$. The hydrogen motion eigenvalues $U_k(R)$ form effective potentials (adiabatic curves) that govern the motion of the halogen atoms, analogous to the nuclear potential curves that are formed by solving the electronic Schrodinger Equation in the Born-Oppenheimer approximation.^{5,11} Once we have the adiabatic curves, the

wavefunctions $\Delta(R)$ for motion of the halogen atoms are found by solving the one-dimensional Schrodinger Equation with the heavy atom Hamiltonian

$$H_x = T_x(R) + U_k(R) \quad (5)$$

Bound, quasibound, and scattering wavefunctions $\Delta(R)$ are found on each of the adiabatic curves and their Franck-Condon overlap with the anion ground state is determined. The method we use to solve the Hamiltonian for hydrogen atom motion will now be described in some detail.

The Schrodinger Equation for hydrogen atom motion at fixed R , in atomic units, is⁴³

$$\left(-\frac{1}{2\mu} \left[\frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2}{\partial \phi^2} + \frac{\partial^2}{\partial z^2} \right] + V(\rho, z) \right) \Gamma_{k,l}(\rho, \phi, z) = U_{k,l} \Gamma_{k,l}(\rho, \phi, z). \quad (6)$$

As the potential is independent of ϕ , we can make the substitution

$$\Gamma_{k,l}(\rho, \phi, z) = \Xi_k(\rho, z) \Phi_l(\phi) \quad (7)$$

where

$$\Phi_l(\phi) = (2\pi)^{-1/2} e^{i l \phi} \quad (8)$$

where $l = 0$ (σ), ± 1 (π),... We need only calculate σ states, as the $v = 0$ XHY⁻ wavefunction has $l = 0$ and so will only have Franck-Condon overlap with $l = 0$ states on the neutral surface. The Schrodinger Equation for $l = 0$ is

We then make the change of variables $r = \frac{1}{2} \rho^2$ to make the volume element $d\tau = dr dz d\phi$. The Schrodinger Equation for hydrogen motion is now

$$\left(-\frac{1}{2\mu} \left[\frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{\partial^2}{\partial z^2} \right] + V(\rho, z) \right) \Xi_k(\rho, z) = U_k \Xi_k(\rho, z). \quad (9)$$

$$\left(-\frac{1}{2\mu} \left[2r \frac{\partial^2}{\partial r^2} + 2 \frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2} \right] + V(r, z) \right) \Xi_k(r, z) = U_k \Xi_k(r, z). \quad (10)$$

The Schrodinger Equation is solved using a product basis consisting of harmonic oscillators: a one-dimensional harmonic oscillator along the hydrogen stretching coordinate z and the radial portion of a two-dimensional harmonic oscillator with $\ell = 0$ along the bending coordinate r . These are the exact solutions of H_H for the potential $V_{\text{ref}}(r, z) = k_r r + \frac{1}{2} k_z z^2$ with energy levels

$$E_{m,n} = \sqrt{\frac{k_r}{\mu}} (m+1) + \sqrt{\frac{k_z}{\mu}} \left(n + \frac{1}{2} \right) \quad (11)$$

The basis set frequencies $(k_r/\mu)^{1/2}$ and $(k_z/\mu)^{1/2}$ are chosen for optimum convergence of the calculation (see below). The eigenfunctions $X_{m,n} = R_m(r)Z_n(z)$ are well known.⁴⁵

As we are using an orthonormal product basis, we can use the Discrete Variable Representation (DVR) to easily and efficiently evaluate matrix elements of the Hamiltonian. DVR methods have been applied to many systems with large-amplitude motions, including reactive scattering problems^{46,47} and vibrations of floppy molecules.^{48,49} The method has been thoroughly described by Light⁵⁰ and our application of it has been reviewed in Chapter 3. Briefly, the Hamiltonian is rewritten as

$$H_H = H_{ref} + \Delta V(r, z) \quad (12)$$

where

$$H_{ref} = T_H(r, z) + V_{ref}(r, z) \quad (13)$$

and

$$\Delta V(r, z) = V(r, z) - V_{ref}(r, z) \quad (14)$$

Setting up the DVR requires the eigenvalues and eigenvectors of the position matrix \mathbf{X} in the $Z_n(z)$ and $R_m(r)$ bases. For the one-dimensional harmonic oscillator, the matrix elements are⁵¹

$$X_{n, n'} = \langle Z_n(z) | z | Z_{n'}(z) \rangle = \sqrt{\frac{n}{2}} \delta_{n, n'+1} + \sqrt{\frac{n+1}{2}} \delta_{n, n'-1} \quad (15)$$

The matrix \mathbf{X} is diagonalized to obtain eigenvalues z_n and the eigenvector matrix \mathbf{T}_z . Similarly, we obtain eigenvalues r_m and eigenvectors \mathbf{T}_r by diagonalizing⁵²

$$X_{m, m'} = \langle R_m(r) | r | R_{m'}(r) \rangle = \frac{m}{2} \delta_{m, m'+1} + \frac{2m+1}{2} \delta_{m, m'} + \frac{m+1}{2} \delta_{m, m'-1} \quad (16)$$

The matrices \mathbf{K}_z and \mathbf{K}_r used to construct the Hamiltonian are given by

$$\begin{aligned} K_z &= T_z^T E_z T_z \\ K_r &= T_r^T E_r T_r \end{aligned} \quad (17)$$

where \mathbf{E}_z and \mathbf{E}_r are the diagonal matrices of eigenvalues to the reference potentials in z and r , respectively (eq. 11). In the DVR, matrix elements of

the Hamiltonian are given by⁴⁸

$$H_{m,n;m',n'} = K_z \delta_{n,n'} \delta_{m,m'} + K_r \delta_{m,m'} \delta_{n,n'} + \Delta V(r_m, z_n) \delta_{m,m'} \delta_{n,n'} \quad (18)$$

The Hamiltonian matrix is then diagonalized to obtain all the adiabatic curves U_x at a particular interhalogen distance. The DVR method evaluates the kinetic energy term in the Hamiltonian exactly. The potential energy terms are found approximately by evaluating the potential difference ΔV at the DVR points (r_m, z_n) . This is equivalent to approximating the matrix elements of the potential difference

$$\langle \Xi_{m,n} | \Delta V | \Xi_{m',n'} \rangle = \int dr \int dz \Xi_{m,n}(r, z) \Delta V(r, z) \Xi_{m',n'}(r, z) \quad (19)$$

using an $N_r \times N_z$ point Gaussian quadrature,⁵³ where N_r is the number of basis functions in r and N_z is the number in z . The method is extremely efficient, as no integrals are actually computed to construct the Hamiltonian matrix. The potential energy is only determined approximately, so the DVR is not variational - the eigenvalues found are not strict upper bounds to the true eigenvalues.⁵⁰ However, eigenvalues found using the DVR are very accurate and quickly converge to the true answer as the size of the basis is increased. The DVR method is very efficient : constructing the Hamiltonian matrix requires diagonalizing a $N_r \times N_r$ matrix, a $N_z \times N_z$ matrix and evaluating the potential at $N_r N_z$ points. This takes 1% of the time it takes to diagonalize the $N_r N_z \times N_r N_z$ Hamiltonian matrix. For symmetric systems, the wavefunctions are either symmetric (*gerade*) or antisymmetric

(*ungerade*) about $z=0$. Only *gerade* states on the neutral potential energy surface will have Franck-Condon overlap with the symmetric $v=0$ anion wavefunction, so the basis set consists of only even functions $Z_n(z)$ for symmetric systems.

The harmonic basis we use is very well suited to geometries where all three atoms are interacting strongly and this is the portion of the potential that the experimental results are most sensitive to. Unfortunately, this basis is not well suited to the asymptotic region, where we have free HX interacting weakly with X. As we wish to determine what product state each adiabatic curve correlates to, we must adequately describe the asymptotic region, and this requires large basis sets. The number of basis functions along each coordinate N_R and N_z and the basis set frequencies $\omega_r = (k_r/\mu)^{1/2}$ and $\omega_z = (k_z/\mu)^{1/2}$ are chosen by solving eq. 10 at large R and varying the number of basis functions and their frequencies until the eigenvalues U_k are insensitive to N_r , N_z , ω_r , and ω_z . The same basis is then used to solve eq. 10 at all R and is also used to solve eq. 10 for the anion. For the IHI^- and IDI^- , good convergence of the first 50 eigenvalues is achieved with $N_r = 24$ and $N_z = 45$ (corresponding to even basis functions in z , up to a quantum number of 90, and 1080 basis functions in all), with $\omega_r = 800 \text{ cm}^{-1}$ and $\omega_z = 180 \text{ cm}^{-1}$ for IHI^- and $\omega_r = 350 \text{ cm}^{-1}$ and $\omega_z = 120 \text{ cm}^{-1}$ for IDI^- while good convergence of the first 30 eigenvalues for the asymmetric BrHI^- system requires $N_r = 18$ and $N_z = 78$ (1404 basis functions) with $\omega_r = 800 \text{ cm}^{-1}$ and

$$\omega_e = 190 \text{ cm}^{-1}.$$

Once the Hamiltonian matrix is constructed, the computational problem consists of determining the eigenvalues and eigenvectors of a large, sparse matrix, which is ideally suited to the use of vector computers. For comparison, Kubach^{38,39} and Schatz¹⁵ use an "asymptotic basis" consisting of slightly modified rovibrational states of HX as their basis and achieve good convergence with only ≤ 75 basis functions. The asymptotic basis is not orthonormal, so it must be orthonormalized, and determining matrix elements of the Hamiltonian requires integrating over the potential. Although the adiabatic DVR method uses a very large basis, the method is very efficient. Constructing adiabatic curves for the Br + HBr system takes 5 CPU minutes on a Cray X-MP/14 and the full simulation takes 7 CPU minutes.

Wavefunctions $\Xi_k(\rho, z; R)$ and eigenvalues $U_k(R)$ for hydrogen atom motion are found at 40 values of the interhalogen distance R to form adiabatic effective potential curves for the halogen motion. The same basis set is used at each value of R . Each adiabatic curve $U_k(R)$ asymptotically correlates to a particular state of $X + HX$ (v, j). Many such curves are responsible for the structure observed in the experimental photoelectron spectrum. For IHI^- some 60 adiabatic curves (correlating to $I + HI$ ($v=0, j=0-27; v=1, j=0-20; v=2, j=0-10$)) must be considered; they are shown in Fig. 4-3a. Once the adiabatic curves have been determined, they could be

coupled exactly (which would be equivalent to Schatz's CCH scattering calculation) or approximately (as has been done by Kubach in his diabatic calculation). In our adiabatic calculation we ignore the coupling and treat each adiabatic curve separately, determining overlap with the bound, quasibound, and scattering states $\Delta(\mathbf{R})$ supported by each curve.

We solve for the halogen atom states $\Delta_n(\mathbf{R})$ supported by each adiabatic curve using a finite, one-dimensional harmonic oscillator basis $P_j(\mathbf{R})$. This means that, rather than a continuous eigenvalue spectrum, we get a series of discrete eigenvalues. We use many basis functions, so the eigenvalues are closely spaced. This approach is quite similar in spirit to that of Bowman, who used an L^2 basis in his two-²² and three-dimensional studies²³ of the photoelectron spectra of IHI^- and ClHCl^- . Our approach is described in more detail in Chapter 3.

Recall that the intensities of the peaks in the photoelectron spectrum are given by eq. 1, and the ground state anion wavefunction Ψ_0^i is given by

$$\Psi_0^i(\rho, z, \mathbf{R}) = \Xi_0^i(\rho, z) \Delta_0^i(\mathbf{R}) \quad (20)$$

$\Xi_0^i(\rho, z)$ is the $v_2 = 0, v_3 = 0$ anion wavefunction obtained by solving eq. 10 at a single value of \mathbf{R} , and $\Delta_0^i(\mathbf{R})$ is the $v_1 = 0$ anion wavefunction. We use separable potentials for the anion and assume the Q_1 normal mode is parallel to the scattering coordinate \mathbf{R} . The wavefunction on the neutral potential is

$$\Psi_{k,h}^f(\rho, z, R) = \Xi_k^f(\rho, z, R) \Delta_h^f(R) \quad (21)$$

where $\Delta_h^f(R)$ is the wavefunction with quantum number h obtained by solving the Schrodinger Equation for halogen motion on the k th adiabatic curve. The wavefunctions can be written in terms of the two-dimensional harmonic oscillator basis functions $X_{m,n}(\rho, z)$ and one-dimensional harmonic oscillator basis functions $P_j(R)$

$$\Xi_o^i(\rho, z) = \sum_{m',n'} c_{m',n',o}^i X_{m',n'}(\rho, z) \quad (22)$$

and

$$\Delta_o^i(R) = \sum_{j=1}^{N_R} d_{j,o}^i P_j(R) \quad (23)$$

and for the neutral,

$$\Xi_k^f(\rho, z, R) = \sum_{m,n} c_{m,n,k}^f(R) X_{m,n}(\rho, z) \quad (24)$$

and

$$\Delta_h^f(R) = \sum_{j=1}^{N_R} d_{j,h}^f P_j(R) \quad (25)$$

The same basis set is used for the anion and neutral. The intensity is then given by

$$\begin{aligned}
I &\propto |\langle \Psi^i | \Psi^f \rangle|^2 \\
&\propto \left| \int dR \int d\rho \int dz \Psi_o^i(\rho, z, R) \Psi_{k,h}^f(\rho, z, R) \right|^2 \\
&\propto \left| \int dR \Delta_o^i(R) \Delta_h^f(R) \int d\rho \int dz \sum_{m',n'} c_{m',n',o}^i X_{m',n}(\rho, z) \sum_{m',n'} c_{m',n,k}^f X_{m,n}(\rho, z) \right|^2
\end{aligned} \tag{26}$$

As we are using the same orthonormal basis set,

$$\int d\rho \int dz X_{m',n}(\rho, z) X_{m,n}(\rho, z) = \delta_{m',m} \delta_{n',n} \tag{27}$$

for the anion and neutral we have

$$I \propto \left| \int dR \Delta_o^i(R) \Delta_h^f(R) \sum_{m,n} c_{m,n,o}^i c_{m,n,k}^f \right|^2 \tag{28}$$

The integral over R is done by quadrature using 400 points in R and interpolating $c_{m,n,k}^f(R)$, as the coefficients change slowly with R within the adiabatic approximation. Note that the hydrogen wavefunction need not be explicitly calculated to determine the overlaps. The stick spectrum is then convoluted with a resolution function (for example, the instrumental resolution) to give a continuous spectrum. Using a basis set of $N_R = 90$ basis functions in R gives eigenvalues spaced by ≤ 10 meV. As the experimental resolution is slightly better than this, the following approach is used to improve the simulated spectrum. Eigenvalues and corresponding overlaps are determined using $N_R = 90, 80, 70, 60$ and the stick spectra are added. For repulsive ("direct scattering" states) changing the size of the basis will change the eigenvalues (see Ch. 3). Thus, a 2500-5000 line stick spectrum is obtained with the sticks spaced by ≤ 2 meV, which is then

convoluted with the experimental resolution function (~ 8 meV resolution) to obtain a smooth spectrum. We have applied the adiabatic three-dimensional method to the photoelectron spectra of IHI^- , IDI^- , BrHBr^- , BrDBr^- , and BrHI^- . These systems will be discussed in the next section.

3. Results

3.1 IHI^- on the LEPS-A potential

The $\text{I} + \text{HI}$ reaction is the prototypical heavy + light-heavy reaction. Thus, there has been a great deal of theoretical interest in this reaction,^{7,14,15,16,17,26,27,28,30,31,32,33,34,35,36,38,39,40} which is predicted to exhibit a number of interesting properties : oscillating reactivity as a function of reactant kinetic energy,³¹ long-lived resonance states, and a bound state on a minimum-free model potential energy surface.^{26,27,30,32,33,34,35} New theoretical methods for treating heavy + light-heavy reactions are often tested on the $\text{I} + \text{HI}$ reaction, usually employing the semiempirical Manz-A LEPS (henceforth LEPS-A) potential.²⁹ This surface has a collinear minimum energy path and a barrier of 0.048 eV. Although recent experimental results^{4,6} have shown that the surface has some significant deficiencies, it has been used for numerous investigations of the reactive cross section of the $\text{I} + \text{HI}$ reaction, vibrationally bound states of the IHI complex, and the photoelectron spectrum of IHI^- . The photoelectron

spectrum of IHI^- has been calculated in three dimensions by Schatz, using an exact (total angular momentum $J = 0$) quantum coupled-channel hyperspherical (CCH) scattering approach.^{14,15,16,19} In addition, it has been calculated by a variety of approximate three-dimensional approaches, including the collinear exact quantum + zero point bend approach⁷ (really a two-dimensional method) and Bowman's 3D L^2 method.²³ Also, Kubach has calculated resonance positions using a diabatic approach,^{38,39,40} although he has not simulated the photoelectron spectrum.

The results we will present are based on three-dimensional adiabatic calculations, but many features of the calculations and results can be better understood by first considering a collinear adiabatic calculation. At large interiodine distance, collinear adiabatic curves correlate to $\text{I} + \text{HI}(\nu)$, while three-dimensional adiabatic curves correlate to $\text{I} + \text{HI}(\nu, j)$, so a collinear simulation of the IHI^- photoelectron spectrum requires the inclusion of only three adiabatic curves of *gerade* symmetry, corresponding to $\text{I} + \text{HI}(\nu=0, 1, 2)$, while a three-dimensional simulation requires the consideration of about 60 states of σ_g symmetry, correlating to $\text{I} + \text{HI}(\nu = 0, j = 0-27; \nu = 1, j = 0-20; \nu = 2, j = 0-10)$. Collinear adiabatic curves⁵⁴ for the $\text{I} + \text{HI}$ reaction are shown in Fig. 4-3b. The three curves are labelled by the antisymmetric stretch quantum number of the IHI complex. Adiabatic curves with $\nu_3 = 2n$ correlate to $\text{I} + \text{HI}(\nu = n)$. The three adiabatic curves in a collinear calculation have no avoided crossings, hence the couplings between the

adiabatic curves are always small - this explains the success of the adiabatic method in collinear calculations on this system. These small couplings can be approximately included using the DIVAH correction.¹¹ All three curves support states that are bound within the adiabatic approximation, several of which are shown by solid lines in Fig. 4-3b. The $v_3 = 0$ adiabatic curve supports two bound states of IHI (with $v_1 = 0$ and 1), while the states indicated on the higher-lying adiabatic curves are Feshbach resonance states that can decay by coupling to lower-lying adiabatic curves. This coupling is ignored in the adiabatic calculation. The shape resonance state supported by the $v_3 = 0$ adiabatic curve is indicated by a dotted line. These resonance states are responsible for the sharp peaks in simulations of the IHI⁻ photoelectron spectrum. The simulated collinear photoelectron spectrum of IHI⁻ is shown in Fig. 4-4. The simulation uses a resolution of 5 meV so that the resonance peaks are clearly resolved. It is indistinguishable from the exact two-dimensional quantum scattering result.⁷ Reduced-dimensionality collinear calculations fairly accurately predict the positions and intensities of resonance states in the photoelectron spectrum of IHI⁻, but do not show the "rotational threshold" features.⁷ This significant failure of the collinear approach prompted our interest in the adiabatic three-dimensional approach.

Three-dimensional adiabatic curves for the I + HI reaction are shown in Fig. 4-3a. Some curves have wells, while many, especially at lower

energy, are purely repulsive. Just as in the collinear simulation, the curves with wells should lead to sharp resonance features in the three-dimensional adiabatic simulation. Repulsive adiabatic curves give rise to the broad rotational threshold features seen in the experimental spectrum, but not reproduced in the collinear simulation.

In contrast to the collinear curves, the adiabatic curves in three dimensions have numerous avoided crossings, especially at higher energy. The reason for this is quite simple : hydrogen antisymmetric stretch vibrations have a frequency of some 700 cm^{-1} , while (iodine) symmetric stretch motion occurs with a 100 cm^{-1} frequency. The separation of time scales is quite distinct. However, in three dimensions, we also have to consider the I-H-I bend, which becomes a hindered rotation at larger R_{HI} , and finally a free HI rotation at the asymptotes. The HI ($v=0$) rotational constant is 6.4 cm^{-1} ,⁵⁶ so, for low rotational levels, rotations occur more slowly than iodine atom motion, violating the separation of time scales assumed in the adiabatic treatment. This causes the large number of avoided crossings in the adiabatic curves. Due to the large number of avoided crossings, the adiabatic approach may not accurately predict scattering results such as reaction rates and product state distributions. However, the adiabatic simulation of the IHI^- photoelectron spectrum gives accurate results when photodetachment occurs to curves with few avoided crossings in the Franck-Condon region. Also, the adiabatic approach is

expected to be better for high-rotor states of the complex, as these involve faster hydrogen motion.

The experimental threshold photodetachment spectrum of IHI^- is shown in Fig. 4-5a and compared to Schatz's exact 3D CCH scattering simulation on the LEPS-A surface (Fig. 4-5b) and our adiabatic three-dimensional simulation (Fig. 4-5c). The calculated spectra have been shifted to lower electron kinetic energy (eKE) by 80 meV and convoluted with 5 meV Gaussians to facilitate comparison with experiment. The neutral scattering energy SE (measured relative to the bottom of the well for the lowest energy diatomic fragment) and the electron kinetic energy in the photoelectron spectrum are, in general, related by

$$\text{SE} = A - \text{eKE}, \text{ with}$$

$$A = h\nu + \text{zpe}(\text{HY}) - D_0(\text{XHY}^-) - \text{EA}(\text{X}),$$

where $h\nu$ is the photon energy of the detachment laser (4.660 eV for IHI^- and IDI^- , and 5.825 eV for BrHI^- , BrHBr^- , and BrDBr^-), zpe is the zero point energy of the HY diatomic fragment,⁵⁶ $D_0(\text{XHY}^-)$ is the dissociation energy of XHY^- ⁵⁶ to $\text{X}^- + \text{HY}$ and EA(X) is the electron affinity of the remaining atom.⁵⁷ The uncertainty in the conversion factor A is 0.05 eV for BrHI^- and BrH(D)Br^- and 0.13 eV for IH(D)I^- and is due almost entirely to the uncertainty in $D_0(\text{XHY}^-)$.⁵⁶ Note that states with *lowest* internal energy occur at the *highest* electron kinetic energy. The two simulated spectra are quite similar and the features in the two simulations

will now be compared in detail.

The lowest adiabatic curve in Fig. 4-3a has a well at $R_{II} = 3.63 \text{ \AA}$, this well supports a state of the IHI complex that lies below $I + HI$ ($v=0, j=0$). This is a "vibrationally bound" state of $IHI^{32,33,34}$ (peak A, Fig. 4-5c) of the type first predicted by Pollak.²⁸ The reasons for the existence of this state on the minimum-free LEPS-A potential have been discussed in detail elsewhere. This state is predicted to lie at a scattering energy of 0.137 eV in our adiabatic calculation. This is the same energy as was found in the full quantum calculations of Clary and Connor³⁴ and the L^2 calculation of Bowman²³ (this is a bound state and therefore does not appear in the CCH scattering calculation). This state is centered at $R_{II} = 3.63 \text{ \AA}$, while the IHI^- anion has been assumed to have $R_{II} = 3.88 \text{ \AA}$, so there is little Franck-Condon overlap with the bound state - it has only 3% the intensity of the large peak at 0.158 eV (Peak C, Fig. 4-5c).

The $v_3 = 0$ adiabatic curves also support long-lived resonance states which, in contrast to the "vibrationally bound" state, lie above $I + HI$ ($v=0, j=0$). The adiabatic simulation has a sharp peak at 0.148 eV (Peak B, Fig. 4-5c) corresponding to a state with one quantum of symmetric stretch excitation. This (100) resonance was found at 0.149 eV in the CCH calculations, but has very low intensity, so it is not obvious in Fig. 4-5b. The intense peak at 0.158 eV (Peak C, Fig. 4-5c) is at the energy of the intense (200) resonance in the CCH calculations (0.159 eV, Peak C, Fig. 4-

5b). Thus the adiabatic method accurately predicts the bound and resonance states with $v_3 = 0$.

In addition to resonance states of the [IHI] complex, the CCH calculation predicts broad "rotational threshold" peaks in the IHI⁻ photoelectron spectrum between 0.18 and 0.33 eV scattering energy. These have been seen in recent high-resolution experiments⁶ (see $v_3 = 0$ peak in Fig. 4-5a). The origin of these features is evident from the three-dimensional adiabatic calculation, which predicts a series of adiabatic curves (correlating to I + HI ($v = 0, j = 4-14$)) that are nearly flat in the Franck-Condon region (Fig. 4-3a). Transitions to these curves are responsible for the rotational threshold features that make a significant contribution to the $v_3 = 0$ peak (Fig. 4-5c). The spacings between the adiabatic peaks are similar to the spacing of the asymptotic HI rotational levels to which they correlate - the adiabatic curves corresponding to these levels are parallel from the Franck-Condon region to the asymptotic region. The adiabatic calculations show an intensity profile that varies smoothly with the rotational level, while the CCH calculations show a nonmonotonic dependence. This variation in intensity may be due to couplings between the adiabatic curves at short R, where there are some avoided crossings, or to coupling between close-lying adiabatic curves at large R. Our adiabatic calculation ignores these couplings.

The agreement between the adiabatic and CCH results is not as good

for the resonances in the $v_3 = 2$ manifold. The adiabatic curves corresponding to the $v_3 = 2$ peak (ie, correlating to I + HI ($v = 1$)) undergo many avoided crossings, so rather than a single diabatic state supporting many resonances (as occurs in Kubach's calculation), we have a series of adiabatic curves, each supporting a single sharp resonance. We find resonance peaks at 0.333, 0.344, 0.354, and 0.365 eV (the series of peaks labelled D in Fig. 4-5c), while resonances in the CCH calculation appear at 0.340, 0.353, 0.368, and 0.380 eV (the series of peaks labelled D in Fig. 4-5b). Kubach's calculation³⁹ gives more accurate eigenvalues for these resonance states. The somewhat worse agreement for the $v_3 = 2$ resonances than the $v_3 = 0$ states is likely due to the number of avoided crossings in the Franck-Condon region for these curves (Fig. 4-3a); these are not treated accurately in our adiabatic approximation. These calculations, as well as the CCH and 3D L^2 calculations, all predict that the rotational threshold contribution will be much smaller for the $v_3 = 2$ peak, where the photoelectron spectrum is dominated by overlap with resonance states. There is only one resonance in the $v_3 = 4$ manifold that has good Franck-Condon overlap with the anion. This is the (004) resonance, which we find at 0.562 eV (Peak F, Fig. 4-5c). It appears at 0.574 eV in the L^2 calculations. The diabatic and CCH calculations did not extend to this energy range.

In summary, results of simulations of the IHI⁻ photoelectron

spectrum on the LEPS-A potential show that the adiabatic three-dimensional approach accurately reproduces the $v_3 = 0$ bound state and resonances from full three-dimensional calculations and does a much better job of reproducing the direct scattering features (the rotational thresholds) of the photoelectron spectrum than do collinear calculations. The adiabatic method only does a fair job of reproducing the $v_3 = 2$ resonances, because of the number of avoided crossings in this region. The efficiency of the method allows us to apply it to a variety of systems for which scattering calculations have not been carried out, or have been carried out over a limited energy range.

3.2 IDI^- on the LEPS-A Surface

As the major progression in the photoelectron spectra of the bihalides is due the hydrogen antisymmetric stretch, there is a large isotope shift upon deuteration. The photoelectron spectra of the deuterated molecules provides additional information on the potential energy surface for the reaction. Schatz has simulated the photoelectron spectrum of IDI^- in three dimensions using the CCH approach.¹⁸ The theoretical treatment of the photoelectron spectrum of deuterides is more computationally demanding than hydrides as the density of ro-vibrational states for deuteride systems is roughly twice that of hydride systems. Due to the difficulty of the

scattering calculation, while the low energy portion of the CCH spectrum is converged, the convergence of the high energy region is less certain. We have simulated the photoelectron spectrum of IDI^- on the LEPS-A potential using the adiabatic approach, obtaining converged (but approximate) results over a wider energy range than was considered in the scattering study.

The experimental photoelectron⁴ and high-resolution photodetachment⁵⁸ spectra of IDI^- are shown in Fig. 4-6a. As in Fig. 4-5a, the peaks are labelled by the ν_3 (antisymmetric stretch) quantum number of the neutral complex. As can be seen, the high-resolution IDI^- spectrum looks quite different from the IHI^- spectrum (Fig. 4-5a). In IDI^- , the $\nu_3 = 0$ peak does not show the discrete "rotational threshold" features seen in IHI^- at comparable resolution. The experimental photodetachment spectrum also shows two features labelled $\nu_3 = 2$ and $2'$ spaced by about 0.035 eV. This is much larger than the -0.010 - 0.015 eV spacing one would expect between resonance states, so the nature of these two features is not clear. As in IHI^- , peaks narrow to higher ν_3 , finally resulting in a narrow $\nu_3 = 4$ resonance peak.

Simulations of the photoelectron spectrum of IDI^- have been carried out to help understand some of the features observed in the experimental spectrum, such as the absence of resolved "rotational threshold" features in the $\nu_3 = 0$ peak and the presence of peak $\nu_3 = 2'$. First we will compare our adiabatic three-dimensional simulation to Schatz's CCH simulation, then we

will compare the simulated spectra to experiment to gain information on deficiencies in the LEPS-A potential for the I + HI reaction.

Schatz's CCH simulation and our adiabatic three-dimensional simulation are shown in Figures 4-6b and c, respectively. The simulated spectra have been shifted 80 meV to lower electron kinetic energy and convoluted with 5 meV Gaussians to facilitate comparison with experiment. The adiabatic results are in better agreement with the exact scattering results at low scattering energy than at high scattering energy. The CCH results show an extremely narrow resonance with extremely low Franck-Condon intensity at 0.110 eV (the (000) state, whose intensity is so low that it cannot be seen in Fig. 4-6b), and another $\nu_3 = 0$ resonance (100) at 0.121 eV (Peak A, Fig. 4-6b). The (100) resonance appears at 0.123 eV in the adiabatic calculation (Peak A, Fig. 4-6c). The $\nu_3 = 2$ features consist of several resonances in the 0.24 - 0.29 eV energy range in both the adiabatic and CCH calculations (Peaks B, Fig. 4-6b,c). In addition, the $\nu_3 = 4$ feature, which is due to a single resonance in the collinear calculations, is composed of three intense resonances (at 0.378, 0.392, and 0.404 eV) in the CCH calculation (Peaks D, Fig. 4-6b). In the adiabatic calculation, we find two sharp $\nu_3 = 4$ resonance peaks at 0.393 and 0.399 eV (Peaks D, Fig. 4-6c). Resonance positions in the adiabatic simulation are in good agreement with the full scattering calculations for the $\nu_3 = 0$ and 2 features. It is difficult to determine if discrepancies in the $\nu_3 = 4$ features are due to failure of the

adiabatic approximation or to lack of convergence of the full three-dimensional scattering calculation at this high energy.

In addition to the resonance features, simulations of the photoelectron spectrum of IDI^- also predict a number of features due to direct scattering. Simulations of the $v_3 = 0$ peak show rotational threshold features, although they are not as pronounced as in the IHI^- spectrum. This is consistent with the smaller spacings between the IDI hindered rotor levels responsible for these states. This helps to explain the experimental finding of resolved threshold features with a width of ~ 20 meV separated by 20-25 meV in IHI^- (Fig. 4-5a), while the IDI^- photoelectron spectrum (Fig. 4-6a) shows no resolved features in the $v_3 = 0$ peak, at the same experimental resolution.⁵⁸

A prominent direct scattering feature is the broad peak at 0.33 eV that dominates the CCH simulation (Peak C, Fig. 4-6b). A smaller version of this peak appears at 0.32 eV in the adiabatic simulation (Peak C, Fig. 4-6c) and a similar feature also appears in the adiabatic IHI^- simulation (Peak E, Fig. 4-5c), though it has lower intensity than in IDI^- . The origin of this feature becomes clear if one looks at the $\text{I} + \text{DI}$ adiabatic curves (Fig. 4-7). This peak is due to overlap with states at the $\text{I} + \text{DI}$ ($v = 1$) threshold (these are the large number of closely spaced adiabatic curves correlating to $\text{I} + \text{DI}$ ($v = 1$, low j) in Fig. 4-7). This peak has much lower intensity in the adiabatic simulation than in the CCH scattering calculation. The reason for

this discrepancy between the CCH and adiabatic results is not clear, although the CCH calculation is likely not well converged at this high energy. Note that collinear calculations will not show a peak at the I + DI ($v = 1$) threshold, as the $v_3 = 2$ adiabatic curve has a minimum and supports several resonance states below I + DI ($v = 1$), so collinear calculations only show the resonance states. The simulations also give an explanation for the nature of the peak labelled 2' in the experimental IDI⁻ photodetachment spectrum. This peak roughly corresponds to peak C, the I + DI ($v = 1$) threshold peak, in the simulations. Although this feature does not occur at the same energy in the simulations and experiment, this discrepancy is much smaller on the LEPS-C potential.

More generally, comparing the simulated and experimental photoelectron spectra of IHI⁻ and IDI⁻ allows us to point out some deficiencies in the LEPS-A potential. The LEPS-A potential supports a bound state of IHI and several long-lived $v_3 = 0$ resonance states of IHI and IDI, which are not observed experimentally. Although this could be due to poor Franck-Condon overlap with these states, calculations predict long-lived resonance states slightly higher in energy than the bound state, and these resonance states have good Franck-Condon overlap with the negative ion. Thus, the vibrationally bound state almost certainly does not exist on the true potential energy surface for the I + HI reaction (recall that the LEPS-A potential was constructed based on chemical intuition, as there

were no experimental results on the I + HI reaction available at the time). This implies that, on the true potential for the I + HI reaction, the lowest adiabatic curve does not have a well and hence does not support a resonance state, so the barrier on the LEPS-A surface is likely too low. The experimental IHI⁻ spectrum shows a smooth rotational contour for the $v_3 = 0$ peak, with a spacing equal to that of the I + HI ($v = 0, j = 12-16$) asymptotic levels.⁶ This suggests that the barrier for the I + HI reaction is sufficiently high that levels below I + HI ($v = 0, j = 11$) do not possess enough energy to reach the Franck-Condon region of the potential. The rotational threshold peaks should be very sensitive to the low-energy portion of the potential energy surface. All of these observations again point to the barrier on the LEPS-A surface being too small. Other discrepancies are that the simulated spectra had to be shifted 80 meV to align them with experiment, the simulated spectra have too much Franck-Condon intensity in the $v_3 = 0$ peak and not enough in the $v_3 = 2$ peak, and the v_3 peaks are spaced too far apart in the LEPS-A simulations, a feature we have also seen with a LEPS potential for the Br + HBr reaction. While the potential used for IHI⁻ does affect the intensity distribution, most of the differences between experiment and the simulations are likely due to discrepancies in the LEPS-A potential, and raising the barrier on the LEPS-A potential is a good first step in alleviating these discrepancies.

3.3 IHI⁻ on the LEPS-C surface

Recently, Schatz has simulated the low-energy portion of the photoelectron spectrum of IHI⁻ in three dimensions on a series of LEPS surfaces with barriers of 0.048 eV (LEPS-A) to 0.243 eV (LEPS-D).¹⁷ He found that the LEPS-C surface, with a 0.161 eV barrier, leads to a simulated spectrum in somewhat better agreement with the experimental results than the LEPS-A surface. In particular, the LEPS-C surface supports no bound or long-lived resonance states in the $v_3 = 0$ manifold - all the observed features are due to rotational thresholds. Also, due to the higher barrier on the LEPS-C surface, only a 40 meV shift is required to align the simulated and experimental spectra, using the known thermodynamics of the IHI⁻/I+HI system, compared to 80 meV for the LEPS-A surface. As Schatz's CCH simulation only covers the $v_3 = 0$ peak, we have calculated the photoelectron spectrum of IHI⁻ on the LEPS-C surface using the adiabatic approach over the entire energy range observed in the experiment.

Adiabatic curves for the I + HI reaction on the LEPS-C potential are shown in Fig. 4-8. The lowest adiabatic curves are purely repulsive, so the LEPS-C potential does not support a bound state or $v_3 = 0$ resonance state of IHI. This is in contrast to the LEPS-A potential, where the lowest adiabatic curve has a well that supports a bound state, and higher-lying

adiabatic curves support $v_3 = 0$ resonance states.

The adiabatic simulation of the IHI^- photoelectron spectrum using the LEPS-C potential (at 5 meV resolution) is shown in Fig. 4-9, along with the high-resolution photodetachment results. The simulated $v_3 = 0$ peak consists of a series of rotational thresholds dominated by peak A at 0.22 eV. This peak is not due to a resonance, as the lowest adiabatic curves on the LEPS-C potential are purely repulsive. The adiabatic results show a series of $v_3 = 2$ resonance peaks at 0.404, 0.416, and 0.425 eV (Peaks B-D, Fig. 4-9) and a sharp $v_3 = 4$ resonance at 0.619 eV (Peak E, Fig. 4-9). Note that, at 5 meV resolution, the resonance peaks are clearly narrower than the rotational threshold peaks. The presence of resonances with $v_3 = 2$ and 4 and absence of resonances with $v_3 = 0$ is in accord with the experimental observations. The LEPS-C simulation shows more intensity in the $v_3 = 2$ and 4 peaks than the LEPS-A simulation, in agreement with experiment, and requires only a 40 meV energy shift to align the peaks with experiment, compared to 80 meV for the LEPS-A; both are within the 130 meV uncertainty in the thermodynamics for the $\text{IHI}^-/\text{I} + \text{HI}$ system.⁵⁶

The LEPS-C surface is certainly an improvement over the LEPS-A potential, although it still does not accurately reproduce the IHI^- photoelectron spectrum. The simulated $v_3 = 0$ manifold is dominated by a single peak, while the experimental spectrum consists of five peaks with similar intensity. Also, the $v_3 = 2$ to $v_3 = 4$ spacing is too large in the

simulation, a common failure with LEPS potentials which is likely due to the repulsive walls of the LEPS surface being too steep. Merely changing the Sato parameter (and hence the barrier) on the LEPS surface may not give enough flexibility to accurately reproduce the experimental results. An alternate approach will be discussed in Section 3.6 with respect to the BrHBr^- spectrum.

3.4 IDI^- on the LEPS-C Potential

The simulated photoelectron spectrum of IDI^- on the LEPS-C potential obtained using the adiabatic three-dimensional method is shown in Fig. 4-10 along with the threshold photodetachment results.⁵⁸ The simulated spectrum is shown at 5 meV resolution and has been shifted 20 meV to lower electron kinetic energy. As with IHI^- , the IDI^- simulation on the LEPS-C potential supports no $\nu_3 = 0$ resonance states, in accord with experiment (Peak A, Fig. 4-10). The simulation of the $\nu_3 = 2$ feature consists of two series of peaks - a group of resonance peaks (Feature B, Fig. 4-10, with the largest peak at 0.339 eV) and a peak due to overlap with states at the $\text{I} + \text{DI}$ ($\nu = 1$) threshold (Feature C, Fig. 4-10, 0.372 eV). The adiabatic simulation of the IDI^- spectrum on the LEPS-A potential also shows these two $\nu_3 = 2$ features (Features B and C, Fig. 4-6c), but the spacing between the resonance and threshold features is smaller on the

LEPS-C surface. These two features appear in the experimental spectrum (peaks 2 and 2' in Fig. 4-6a), and the simulations suggests that, in the experimental spectrum, feature 2 is due to $v_3 = 2$ resonance states and feature 2' is due to states at the I + DI ($v = 1$) threshold. The spacing between these features is sensitive to the repulsiveness of the potential energy surface for the I + DI reaction, with a more repulsive surface (such as the LEPS-C potential) leading to a smaller B-C spacing. The LEPS-C potential supports a single $v_3 = 4$ resonance, at 0.459 eV. The agreement between the experimental and simulated $v_3 = 2$ and 4 features is surprisingly good.

3.5 BrHI⁻ on the BP LEPS Surface

In an earlier study of the BrHI⁻ photoelectron spectrum,⁸ we found that collinear simulations on the Broida-Persky (BP) surface successfully reproduced peak positions and intensities seen experimentally, but that the simulated peaks were much narrower than those observed experimentally. An adiabatic three-dimensional simulation should show whether this discrepancy in peak widths is due to the collinear simulation or discrepancies in the BP surface. While the LEPS-A surface for the I + HI reaction was constructed without the benefit of experimental results, the Br

+ HI reaction is easier to study experimentally and the BP LEPS surface was constructed to reproduce experimental rate constants for the Br + HI reaction at several temperatures, as well as the product HBr $v=2/v=1$ ratio at 300 K in classical trajectory calculations.⁵⁹ The BP surface matches the Br + HI exothermicity (0.704 eV) and has a collinear minimum energy path with a 0.009 eV barrier. While the neutral potential for Br + HI is better characterized than the I + HI potential, even less is known about BrHI⁻ than about IHI⁻. An earlier collinear study of the BrHI⁻ photoelectron spectrum⁸ used physically reasonable values for the anion frequencies and geometry - we will use the same values in this study. The parameters are summarized in Table 4-1.

Neither full nor approximate three-dimensional calculations of the BrHI⁻ photoelectron spectrum have been carried out previously. The difficulty in these calculations lies in the large number of distinguishable products that can be produced. The collinear calculations predict that the photoelectron experiment accesses levels correlating to I + HBr ($v=0,1,2,3$) and Br + HI ($v=0,1$). This is in contrast to the three product vibrational states accessed in IHI⁻ photodetachment. Unlike the other systems we have discussed, the Br + HI system is not symmetrical. Thus, the same level of convergence for Br + HI as for I + HI would require a basis set of about twice the size, raising the computation time by a factor of 8. This is rather prohibitive, so a moderate basis of 1404 basis functions with $N_r = 18$

and $N_z = 78$ was used to calculate the lowest 100 hydrogenic states. This required twice the execution time as for IHI^- . Levels correlating to $\text{I} + \text{HBr}$ ($v=0,1$) are well converged and to $\text{I} + \text{HBr}$ ($v=2$) and $\text{Br} + \text{HI}$ ($v=0$) are fairly well converged. This covers the major features observed experimentally.

The simulated photoelectron spectrum of BrHI^- at the resolution of the fixed-frequency photoelectron spectrometer is shown in Figure 4-11. The simulation consists of two broad, intense peaks at 0.336 eV ($v_3 = 0$) and 0.632 eV ($v_3 = 1$) due to overlap to states correlating to $\text{I} + \text{HBr}$ ($v=0$) and $\text{I} + \text{HBr}$ ($v=1$), respectively, and a slightly smaller $v_3 = 3$ peak at 0.903 eV which corresponds to $\text{I} + \text{HBr}$ ($v=2$). While the collinear calculation predicts a very small $v_3 = 2$ resonance peak due to $\text{Br} + \text{HI}$ ($v=0$) just below the $\text{I} + \text{HBr}$ peak, this feature is too weak to be positively identified in this study. At moderate resolution (the resolution available in the fixed-frequency photoelectron experiment), the broad features are expected to show barely-resolved structure due to rotational thresholds. The positions of the peaks in the present study are in good agreement with those found in our earlier, collinear study.⁸ In both cases, a shift of 62 meV is required to align the experimental and simulated spectra. This is at the level of the uncertainty in the dissociation energy of BrHI^- ,⁵⁶ which is used to determine the conversion from scattering energy to electron kinetic energy.

The peaks in the three-dimensional adiabatic simulation have similar intensities to those found in the collinear simulation. The most striking

difference between the collinear and adiabatic 3D results is in the peak widths : the $v_3 = 0, 1, 3$ peaks have widths of 100, 85, and 62 meV, respectively in the present study and widths of 43, 36, and 15 meV in the collinear study.⁸ The widths of these peaks are due to overlapping rotational thresholds. This resolves some of the discrepancy between the calculated and experimental widths. All three experimental peaks have widths of 160-170 meV and lack resolved rotational threshold features (the resolution of the 213 nm photoelectron spectrum is ~30 meV in this energy range, but spectra taken at 266 nm show a similar lack of structure at 8 meV resolution).

The discrepancies between the three-dimensional adiabatic simulation and the experimental results could be due to errors in the BrHI^- potential, which is not well known. It could, however, point out deficiencies in the BP surface, as it has been found for BrHBr^- that collinear simulations on a LEPS surface give peaks that are much too narrow.⁵ The $\text{Br} + \text{HI}$ reaction could also proceed via a noncollinear minimum energy path, while the minimum energy path on the BP surface is collinear. Alternatively, it has been pointed out⁸ that photodetachment of BrHI^- could access two low-lying excited electronic states of the neutral complex, in addition to the ground state. Transitions to one of these excited states, which correlates to $\text{I}^* (^2\text{P}_{1/2}) + \text{HBr}$, are clearly observed in the photoelectron spectrum below 1.2 eV eKE and obscure any structure due to transitions correlating to $\text{I} + \text{HBr} (v=3)$

and Br + HI ($v=1$) on the ground electronic surface. The other low-lying excited electronic state is due to the splitting caused by the interaction of the open-shelled iodine atom with the HBr dipole. Transitions to this state are expected to lie in the same energy range as the ground state features and therefore could cause the observed peaks to broaden. The observed discrepancies between the simulated and experimental photoelectron spectra of BrHI^- are likely due to a combination of these effects.

3.6. BrHBr^- on a Fitted Surface

In Chapter 2,⁵ the photoelectron spectrum of BrHBr^- was fit using an empirical effective collinear potential energy surface for the Br + HBr reaction. We would like to test this fit by simulating the photoelectron spectrum in three dimensions using the fitted potential. This requires a full three-dimensional surface, so the first step is to extend the collinear surface to three dimensions. As *ab initio* calculations predict that the BrHBr^- anion is linear,^{60,61} and we use a surface for the neutral reaction with a collinear minimum energy path, bent geometries are probably not as important to the simulation as collinear geometries. Thus, we use a simple method to extend the collinear fitted surface to bent geometries : for noncollinear geometries, we use information from the collinear fit, coupled with a reference LEPS potential.

The three-dimensional potential is constructed by combining information from a fitted collinear and a reference LEPS potential. The fitted potential is completely empirical, its functional form is given in Chapter 2 and ref. 5. For the reference LEPS potential we use the LEPS potential used in our earlier study of the Br + HBr reaction.^{5,62} It has the same collinear barrier and saddle point position as the fitted collinear surface. However, as was shown in the earlier paper (and in Ch. 2), simulations using this LEPS surface do not accurately reproduce the experimental spectrum. To distinguish between quantities calculated using collinear and noncollinear geometries we define

$$\Delta = (R_{\text{HBr}} + R_{\text{BrH}}) - R_{\text{BrBr}}$$

$\Delta = 0$ at collinear geometries and $\Delta > 0$ at noncollinear geometries. The fitted surface is an effective collinear surface - at every collinear point it includes the zero-point bend energy. The first step is to approximately remove this energy to form a collinear surface

$$V_{\text{col}}(r_{\text{HBr}}, r_{\text{BrH}}, \Delta=0) = V_{\text{fit}}(r_{\text{HBr}}, r_{\text{BrH}}, \Delta=0) - \text{zpe}(r_{\text{HBr}}, r_{\text{BrH}}, \Delta=0).$$

The zero point bend energy is calculated on the reference LEPS potential at each collinear geometry, within the harmonic approximation, as described by Bowman.⁶³ For the zero point energy calculation *only*, a mass of 1.5 amu is used for the hydrogen atom. This is so that the resulting three-dimensional potential energy surfaces will be the same for Br + HBr and Br + DBr. At collinear geometries, the difference between the reference

LEPS potential and the fitted collinear surface is

$$\Delta V(r_{\text{HBr}}, r_{\text{BrH}}, \Delta=0) = V_{\text{col}}(r_{\text{HBr}}, r_{\text{BrH}}, \Delta=0) - V_{\text{LEPS}}(r_{\text{HBr}}, r_{\text{BrH}}, \Delta=0).$$

The global fitted potential is then defined as

$$V_{\text{3Dfit}}(r_{\text{HBr}}, r_{\text{BrH}}, \Delta) = V_{\text{LEPS}}(r_{\text{HBr}}, r_{\text{BrH}}, \Delta) + \Delta V(r_{\text{HBr}}, r_{\text{BrH}}, \Delta=0).$$

In Chapter 2, the fitted potential V_{fit} was constructed over the range $2.6 \text{ \AA} \leq R_{\text{HBr}} + R_{\text{BrH}} \leq 4.7 \text{ \AA}$. However, in three dimensions, a geometry such as H - Br - Br with $R_{\text{BrH}} = 1.4 \text{ \AA}$, $R_{\text{HBr}} = 4.9 \text{ \AA}$, $R_{\text{BrBr}} = 3.5 \text{ \AA}$ and $R_{\text{HBr}} + R_{\text{BrH}} = 6.3 \text{ \AA}$ can be reached at reasonable energies. This is outside the range where V_{fit} is defined, so we extend the fitted potential to the range $2.6 \text{ \AA} \leq R_{\text{HBr}} + R_{\text{BrH}} \leq 11.0 \text{ \AA}$. The new parameters for $R_{\text{HBr}} + R_{\text{BrH}} > 4.7 \text{ \AA}$ were found by approximately matching the fitted potential to the reference LEPS potential at collinear geometries in this asymptotic region. The additional parameters for the fitted potential are given in Table 4-2.

The experimental photoelectron spectrum of BrHBr^- is shown in Figure 4-12 (solid line) and compared with the collinear simulation (dashed line) and the three-dimensional adiabatic simulation (dotted line). As with the effective collinear calculation, the three-dimensional adiabatic simulation provides an excellent fit to the experimental spectrum. The peak intensities and widths are well reproduced in the fitted spectrum (in fact, the noncollinear simulation does a better job of reproducing the peak widths and shapes than the collinear simulation). Note that the peaks narrow as v_3 increases (to lower electron kinetic energy), finally giving a resonance

peak for $v_3 = 4$. The experimental $v_3 = 4$ peak lies on top of the $v_3 = 0$ peak for transitions to an excited electronic state of the BrHBr complex, corresponding to the $\text{Br}^* (^2P_{1/2}) + \text{HBr}$ reaction, thus it is difficult to compare the experimental and simulated intensities of this peak. Transitions to the excited electronic state were not considered in the present study.

The simulated BrHBr^- photoelectron spectrum does not show the resolved "rotational threshold" peaks found for IHI^- .⁶ The reason for this can clearly be seen by comparing the adiabatic curves for $\text{Br} + \text{HBr}$ (Fig. 4-13) and $\text{I} + \text{HI}$ (Fig. 4-3a). The $v_3 = 0$ $\text{Br} + \text{HBr}$ curves are much more repulsive in the Franck-Condon region than the $\text{I} + \text{HI}$ curves. Thus, each rotational threshold peak is so broad that it merges with neighboring peaks, resulting in a single broad feature. The adiabatic curves for $\text{Br} + \text{HBr}$ are so repulsive because the minimum energy path for the $\text{Br} + \text{HBr}$ reaction on the fitted surface is very steep, much steeper than that on the LEPS-A or C surfaces used for the $\text{I} + \text{HI}$ reaction.

It is interesting to compare the results of the adiabatic collinear simulation with those of the adiabatic three-dimensional simulation. There are several differences between the two simulations : overall peak intensities, positions of peaks corresponding to high v_3 levels, and peak widths. In the three-dimensional simulation, the $v_3 = 0$ peak is noticeably less intense than in the collinear simulation. The positions of the $v_3 = 0$ and $v_3 = 2$ peaks are in good agreement with experiment, while the $v_3 = 4$ peak

has shifted some 20 meV to lower electron kinetic energy from both the collinear simulation and experiment. Similar shifts in the positions of high antisymmetric stretch peaks in going from effective collinear to three-dimensional calculations have been also seen in the IHI^- system. This is probably because the effective bend potential seen by the $v_3 = 4$ wavefunction is somewhat steeper than that seen by the $v_3 = 0$ wavefunction. Unlike BrHI^- , where peaks in the three-dimensional simulation are significantly broader than the collinear peaks, peaks in the three-dimensional BrHBr^- simulation are only slightly broader than the collinear ones.

3.7 BrDBr^- on the Fitted Surface

The photoelectron spectrum of BrDBr^- was simulated using the same potential energy surface as for BrHBr^- . The resulting collinear and adiabatic three-dimensional simulations is shown in Figure 4-14 (dashed and dotted lines, respectively) and compared with experiment (solid line). Again, the simulated photoelectron spectra are in very good agreement with experiment. The peaks for BrDBr^- narrow as the v_3 quantum number increases, dropping from a width of > 140 meV for $v_3 = 0$ to some 12 meV for $v_3 = 6$. The $v_3 = 6$ peak in the experimental spectrum lies on top of the $v_3 = 0$ peak for transitions to an excited electronic state of the $[\text{BrDBr}]$

complex. Transitions to this excited state are not considered in this study. While BrDBr^- does give a sharp resonance peak, it is much less intense than the $\nu_3 = 4$ peak in BrHBr^- , thus the best chance for observing resonances for this system lies in the $\nu_3 = 4$ peak for BrHBr^- .

The BrHBr^- and BrDBr^- studies show that the simulations are likely not very sensitive to the noncollinear portions of the potential, assuming a linear anion and a reaction with a collinear minimum energy path. This study shows a simple way to extend a collinear surface for a bimolecular reaction to three dimensions. This is very useful, as, while there exist several empirical functions for fitting a collinear potential surface, fitting a full three-dimensional potential is quite difficult. For collinearly dominated reactions, where the non-collinear portion of the potential need not be as accurately known as the collinear portion, an approximate three-dimensional surface can be constructed by fitting a empirical collinear surface to experimental results using collinear calculations and extending the surface to three dimensions as above. The surface can then be refined using three-dimensional calculations, or a combination of collinear and three-dimensional calculations. It would clearly be very useful to do an exact three-dimensional simulation of the BrHBr^- photoelectron spectrum on our new surface.

4. Conclusions

The three-dimensional adiabatic method provides a means of efficiently simulating the photoelectron spectra of XHY^- systems in three dimensions. Three-dimensional adiabatic calculations of the IHI^- photoelectron spectrum show rotational threshold features that have been seen experimentally and in exact three-dimensional scattering calculations, but are not reproduced in collinear calculations. The accuracy of the method has been evaluated by comparing adiabatic simulations of the IHI^- and IDI^- photoelectron spectra with the exact ($J=0$) scattering results of Schatz.^{14,18} Adiabatic simulations of the $v_3 = 0$ resonance peaks on the LEPS-A surface are in excellent agreement with the scattering results, while the agreement in the $v_3 = 2$ peaks is not as good, presumably due to the number of avoided crossings these curves undergo in the Franck-Condon region. In general, the adiabatic approach is better suited to simulations of direct scattering states and low-lying resonance states than to higher-lying resonance states, where the adiabatic curves have many avoided crossings. The adiabatic method has also been applied to the LEPS-C potential.¹⁷ Simulations of the IHI^- and IDI^- spectra on the LEPS-C potential are in better agreement with experiment than simulations on the LEPS-A potential. Simulations of the $BrHI^-$ photoelectron spectrum remove some discrepancies found between experiment and the earlier collinear

simulation.⁶ A three-dimensional surface has been constructed for the Br + HBr reaction by extending the earlier fitted collinear surface⁵ to three dimensions. Adiabatic calculations on this surface show that it accurately reproduces the experimental BrHBr⁻ and BrDBr⁻ photoelectron spectra.

1. R. D. Levine and R. B. Bernstein, Molecular Reaction Dynamics and Chemical Reactivity, Oxford University Press, New York, (1987).
2. M. Gruebele and A. H. Zewail, *Physics Today*, May, 1990, p. 24.
3. R. B. Metz, T. Kitsopoulos, A. Weaver and D. M. Neumark, *J. Chem. Phys.* **88**, 1463 (1988).
4. A. Weaver, R. B. Metz, S. E. Bradforth and D. M. Neumark, *J. Phys. Chem.* **92**, 5558 (1988).
5. R. B. Metz, A. Weaver, S. E. Bradforth, T. N. Kitsopoulos and D. M. Neumark, *J. Phys. Chem.* **94**, 1377 (1990).
6. I. M. Waller, T. N. Kitsopoulos and D. M. Neumark, *J. Phys. Chem.* **94**, 2240 (1990).
7. R. B. Metz, S. E. Bradforth and D. M. Neumark, *Adv. Chem. Phys.* (accepted, 1991).
8. S. E. Bradforth, A. Weaver, D. W. Arnold, R. B. Metz and D. M. Neumark, *J. Chem. Phys.* **92**, 7205 (1990).
9. D. G. Truhlar and A. Kuppermann, *J. Chem. Phys.* **52**, 384 (1970).
10. S.-F. Wu and R. D. Levine, *Mol. Phys.* **22**, 991 (1971).
11. D. K. Bondi, J. N. L. Connor, J. Manz and J. Römelt, *Mol. Phys.* **50**, 467 (1983).
12. D. M. Neumark, A. M. Wodtke, G. N. Robinson, C. C. Hayden and Y. T. Lee, *J. Chem. Phys.*, **82**, 3045 (1985).
13. W. H. Miller and J. Z. H. Zhang, *J. Phys. Chem.* **95**, 12 (1991).
14. G. C. Schatz, *J. Phys. Chem.* **94**, 6157 (1990).

15. G. C. Schatz, *Isr. J. Chem.* **29**, 361 (1989).
16. G. C. Schatz, *J. Chem. Phys.* **90**, 4847 (1989).
17. G. C. Schatz, D. Sokolovski and J. N. L. Connor, *Faraday Discuss. Chem. Soc.*,
in press.
18. G. C. Schatz, *J. Chem. Soc. Faraday Trans.* **86**, 1729 (1990).
19. G. C. Schatz, *J. Chem. Phys.* **90**, 3582 (1989).
20. G. C. Schatz, B. Amaee and J. N. L. Connor, *Chem. Phys. Lett.* **132**, 1 (1986).
21. B. Amaee, J. N. L. Connor, J. C. Whitehead, W. Jakubetz and G. C. Schatz,
Faraday Discuss. Chem. Soc. **84**, 387 (1987).
22. J. M. Bowman and B. Gazdy, *J. Phys. Chem.* **93**, 5129 (1989).
23. B. Gazdy and J. M. Bowman, *J. Chem. Phys.* **91**, 4615 (1989).
24. J. M. Bowman, B. Gazdy and Q. Sun, *J. Chem. Soc. Faraday Trans.* **86**, 1737
(1990).
25. J. Z. H. Zhang and W. H. Miller, *J. Chem. Phys.* **92**, 1811 (1990); J. Z. H.
Zhang, W. H. Miller, A. Weaver and D. M. Neumark, *Chem. Phys. Lett.* **182**,
283 (1991).
26. J. A. Kaye and A. Kuppermann, *Chem. Phys. Lett.* **77**, 573 (1981).
27. D. G. Truhlar, B. C. Garrett, P. G. Hipes and A. Kuppermann, *J. Chem. Phys.*
81, 3452 (1984).
28. E. Pollak, *J. Chem. Phys.* **78**, 1228 (1983); *Chem. Phys. Lett.* **94**, 85 (1983).
29. J. Manz and J. Römelt, *Chem. Phys. Lett.* **81**, 179 (1981).
30. J. Manz, R. Meyer, E. Pollak and J. Römelt, *Chem. Phys. Lett.* **93**, 184 (1982);
J. Manz, R. Meyer and J. Römelt, *Chem. Phys. Lett.* **96**, 607 (1983).

31. C. Hiller, J. Manz, W. H. Miller and J. Römelt, *J. Chem. Phys.* **78**, 3850 (1983).
32. J. Manz, R. Meyer and H. H. R. Schor, *J. Chem. Phys.* **80**, 1562 (1984).
33. J. Manz, R. Meyer, E. Pollak, J. Römelt and H. H. R. Schor, *Chem. Phys.* **83**, 333 (1984).
34. D. C. Clary and J. N. L. Connor, *Chem. Phys. Lett.* **94**, 81 (1983).
35. D. C. Clary and J. N. L. Connor, *J. Phys. Chem.* **88**, 2758 (1984).
36. R. T. Skodje and M. J. Davis, *J. Chem. Phys.* **88**, 2429 (1988).
37. J. Römelt, *Chem. Phys.* **79**, 197 (1983).
38. C. Kubach, *Chem. Phys. Lett.* **164**, 475 (1989).
39. C. Kubach, G. Nguyen Vien and M. Richard-Viard, *J. Chem. Phys.* **94**, 1929 (1991).
40. G. Nguyen Vien, M. Richard-Viard and C. Kubach, *J. Phys. Chem.* **95**, 6067 (1991).
41. M. E. Jacox, *J. Phys. Chem. Ref. Data* **13**, 945 (1984).
42. S. Sato, *Bull. Chem. Soc. Jpn.* **28**, 450 (1955).
43. L. Pauling and E. B. Wilson, Jr., Introduction to Quantum Mechanics, Dover, New York, (1985), p. 443.
44. G. Hauke, J. Manz and J. Römelt, *J. Chem. Phys.* **73**, 5040 (1980); A. Kupperman, J. Kaye and J. P. Dwyer, *Chem. Phys. Lett.* **74**, 257 (1980); V. K. Babamov and R. A. Marcus, *J. Chem. Phys.* **74**, 1790 (1981); V. Aquilanti, S. Cavalli and A. Lagana, *Chem. Phys. Lett.* **93**, 179 (1982).
45. W. H. Schaffer, *J. Chem. Phys.* **9**, 607 (1941).
46. Z. Bačić, J. D. Kress, G. A. Parker and R. T. Pack, *J. Chem. Phys.* **92**, 2344

- (1990).
47. D. T. Colbert and W. H. Miller, to be published.
 48. Z. Bačić and J. C. Light, *Ann. Rev. Phys. Chem.* **40**, 469 (1989) and references therein.
 49. S. E. Choi and J. C. Light, *J. Chem. Phys.* **92**, 2129 (1990).
 50. J. C. Light, I. P. Hamilton and J. V. Lill, *J. Chem. Phys.* **82**, 1400 (1985).
 51. P. W. Atkins, Molecular Quantum Mechanics, 2nd Ed., Oxford University Press, Oxford, (1983), p. 102.
 52. S. Bell, R. Davidson and P. A. Warsop, *J. Phys. B* **3**, 113 (1970); *J. Phys. B* **3**, 123.
 53. A. S. Dickinson and P. R. Certain, *J. Chem. Phys.* **49**, 4209 (1968).
 54. The curves shown are calculated on the *effective collinear* surface, which has the zero-point bend energy added at every point. See ref. 5.
 55. K. P. Huber and G. Herzberg, Molecular Spectra and Molecular Structure, IV. Constants of Diatomic Molecules, Van Nostrand, Reinhold & Co., New York, (1979).
 56. G. Caldwell and P. Kebarle, *Can J. Chem.* **63**, 1399 (1985).
 57. H. Hotop and W. C. Lineberger, *J. Phys. Chem. Ref. Data* **14**, 731 (1985).
 58. T. N. Kitsopoulos, C. J. Chick and D. M. Neumark, unpublished results; T. N. Kitsopoulos, Ph. D. Thesis, University of California, Berkeley (1991).
 59. M. Broida and A. Persky, *Chem. Phys.* **133**, 405 (1989).
 60. A. B. Sannigrahi and S. D. Peyerimhoff, *J. Mol. Struct.* **165**, 55 (1988).
 61. S. Ikuta, T. Saitoh and O. Nomura, *J. Chem. Phys.* **93**, 2530 (1990).

62. The value of D_0 (HBr) in Table V of ref. 5 should be 378.3 kJ/mol, not 387.3 kJ/mol.
63. J. M. Bowman, Adv. Chem. Phys. **61**, 115 (1985).

Table 4-1: Negative Ion Frequencies and Geometries.

Molecule (XHY ⁻)	ν_1 (cm ⁻¹)	ν_2 (cm ⁻¹)	ν_3 (cm ⁻¹)	$R_{e\ XY}$ (Å)	$R_{e\ XH}$ (Å)	D_0^a (eV)	A^b (eV)
BrHBr ⁻	164	700	728	3.50	1.75	0.906	1.717
BrDBr ⁻	170	500	498	3.50	1.75	0.898	1.678
BrHI ⁻	100	700	920 ^c	3.88	1.55	0.698	2.231
IHI ⁻	121	700	682	3.88	1.94	0.737	1.006
IDI ⁻	124	495	470	3.88	1.94	0.735	0.967

^a Values of D_0 are from Ref. 56 and have an uncertainty of 0.05 eV. Values of D_0 for deuterides are calculated using D_0 for the hydride and correcting for zero point energy using the above frequencies for XH(D)Y⁻ and the known frequencies of H(D)Y.

^b A is the conversion from scattering energy to electron kinetic energy :

$$S. E. = A - eKE, \text{ see text.}$$

^c A Morse oscillator with $\omega_e = 1276.3 \text{ cm}^{-1}$ and $\omega_e x_e = 178.1 \text{ cm}^{-1}$ is used for BrHI⁻. Harmonic oscillators are used for all other modes.

Table 4-2 : Parameters for the Br + HBr Potential.*

$R_{\text{Br-Br}}$ (Å)	x_1 (Å)	b (kJ/mol)	h (kJ/mol)	k (kJ/mol/Å ²)	q (kJ/mol/Å ⁴)
5.2	1.676	-377.5	360.0	1220	100.0
5.7	2.030	-377.65	370.0	1180	50.0
6.2	2.3835	-377.8	376.0	1150	20.0
7.0	2.949	-378.0	377.0	1130	5.0
8.0	3.656	-378.15	377.8	1130	3.0
9.0	4.363	-378.25	378.0	1130	2.0
10.0	5.070	-378.28	378.1	1130	1.0
11.0	5.778	-378.3	378.2	1130	0.5
12.0	6.485	-378.31	378.3	1130	0.2

* Parameters for $R_{\text{Br-Br}} < 5.2$ Å and the form of the potential are given in Ch. 2 and Ref. 5. Note that k and q should be in units of 10^3 kJ/mol/Å² and 10^3 kJ/mol/Å⁴, respectively in Ref. 5.

Chapter 4 Figure Captions

Figure 4-1. Experimental photoelectron spectra of IHI^- obtained using the fixed frequency photoelectron spectrometer (----), and higher resolution spectra obtained using the zero electron kinetic energy (zeke) spectrometer (—). The peak heights in the zeke spectra have been scaled to those obtained using the fixed frequency spectrometer and the peaks have been shifted by 7 meV.

Figure 4-2. Polar coordinate system. See text for an explanation of the coordinates.

Figure 4-3. (a) Three-dimensional adiabatic curves of σ_g symmetry for the $\text{I} + \text{HI}$ system on the LEPS-A potential. Each curve correlates to $\text{I} + \text{HI}(\nu, j)$.
(b) Collinear adiabatic curves of *gerade* symmetry for the $\text{I} + \text{HI}$ system on the LEPS-A potential, including zero point bend. Each curve correlates to $\text{I} + \text{HI}(\nu)$. States bound within the adiabatic approximation and having significant Franck-Condon overlap with IHI^- are shown with solid line. The dotted line indicates a shape resonance state.

Figure 4-4. Threshold photodetachment (—) and collinear simulated (.....) photoelectron spectra of IHI^- . The collinear simulation has been obtained using the LEPS-A potential, with zero point bend, and has been shifted to lower $e\text{KE}$ by 80 meV and convoluted with 5 meV Gaussians.

Figure 4-5. Experimental and simulated photoelectron spectra of IHI^- .

(a) Experimental threshold photodetachment spectrum of IHI^- (—).

(b) Simulated photoelectron spectrum of IHI^- of Schatz, calculated using the exact 3D CCH scattering method and the LEPS-A potential.

(c) Simulated photoelectron spectrum of IHI^- calculated using the adiabatic 3D method and the LEPS-A potential. The simulated spectra have been convoluted with a uniform resolution of 5 meV and shifted to lower electron kinetic energy by 80 meV to facilitate comparison with experiment.

Figure 4-6. (a) Experimental photoelectron spectrum of IDI^- obtained using the fixed-frequency photoelectron spectrometer (----), and higher-resolution threshold photodetachment spectrum (—). The peak heights in the threshold spectra have been scaled to those obtained using the fixed frequency spectrometer and the peaks have been shifted by 14 meV.

(b) Simulated photoelectron spectrum of IDI^- of Schatz, calculated using the exact 3D CCH scattering method and the LEPS-A potential. Peak C has been clipped at 12% of its true intensity.

(c) Simulated photoelectron spectrum of IDI^- calculated using the adiabatic 3D method and the LEPS-A potential. The simulated spectra have been convoluted with a uniform resolution of 5 meV and shifted to lower electron kinetic energy by 80 meV to facilitate comparison with experiment.

Figure 4-7. Three-dimensional adiabatic curves of σ_g symmetry for the I + DI system on the LEPS-A potential. Each curve correlates to I + DI (v, j).

Figure 4-8. Three-dimensional adiabatic curves of σ_g symmetry for the I + HI system on the LEPS-C potential. Each curve correlates to I + HI (v, j).

Figure 4-9. Threshold photodetachment (.....) and simulated (—) photoelectron spectra of IHI^- on the LEPS-C potential. The simulation is obtained using the adiabatic 3D method and has been shifted 40 meV to lower eKE and convoluted with a uniform 5 meV resolution.

Figure 4-10. Threshold photodetachment (.....) and simulated (—) photoelectron spectra of IDI^- on the LEPS-C potential. The simulation is obtained using the adiabatic 3D method and has been shifted 20 meV to lower eKE and convoluted at 5 meV resolution.

Figure 4-11. Experimental (—) and simulated (.....) photoelectron spectra of BrHI^- . The simulation has been obtained using the adiabatic 3D method and the BP LEPS potential and has been shifted by 62 meV to lower electron kinetic energy and convoluted with the instrumental resolution function. The instrumental resolution is 12 meV at 0.85 eV eKE ($v_3 = 0$) and improves to 7 meV at 0.55 eV eKE ($v_3 = 4$), the actual function is given in ref. 5 and Ch. 2.

Figure 4-12. Experimental (—) and simulated photoelectron spectra of BrHBr^- . Both a collinear (----) and adiabatic 3D (.....) simulation are shown. Both simulations use the fitted potential and have been convoluted with the instrumental resolution function. Parameters used for the fitted potential are given in Table 4-2.

Figure 4-13. Three-dimensional adiabatic curves of σ_g symmetry for the $\text{Br} + \text{HBr}$ system on the fitted potential. Each curve correlates to $\text{Br} + \text{HBr} (v, j)$.

Figure 4-14. Same as Figure 4-12, but for BrDBr^- .

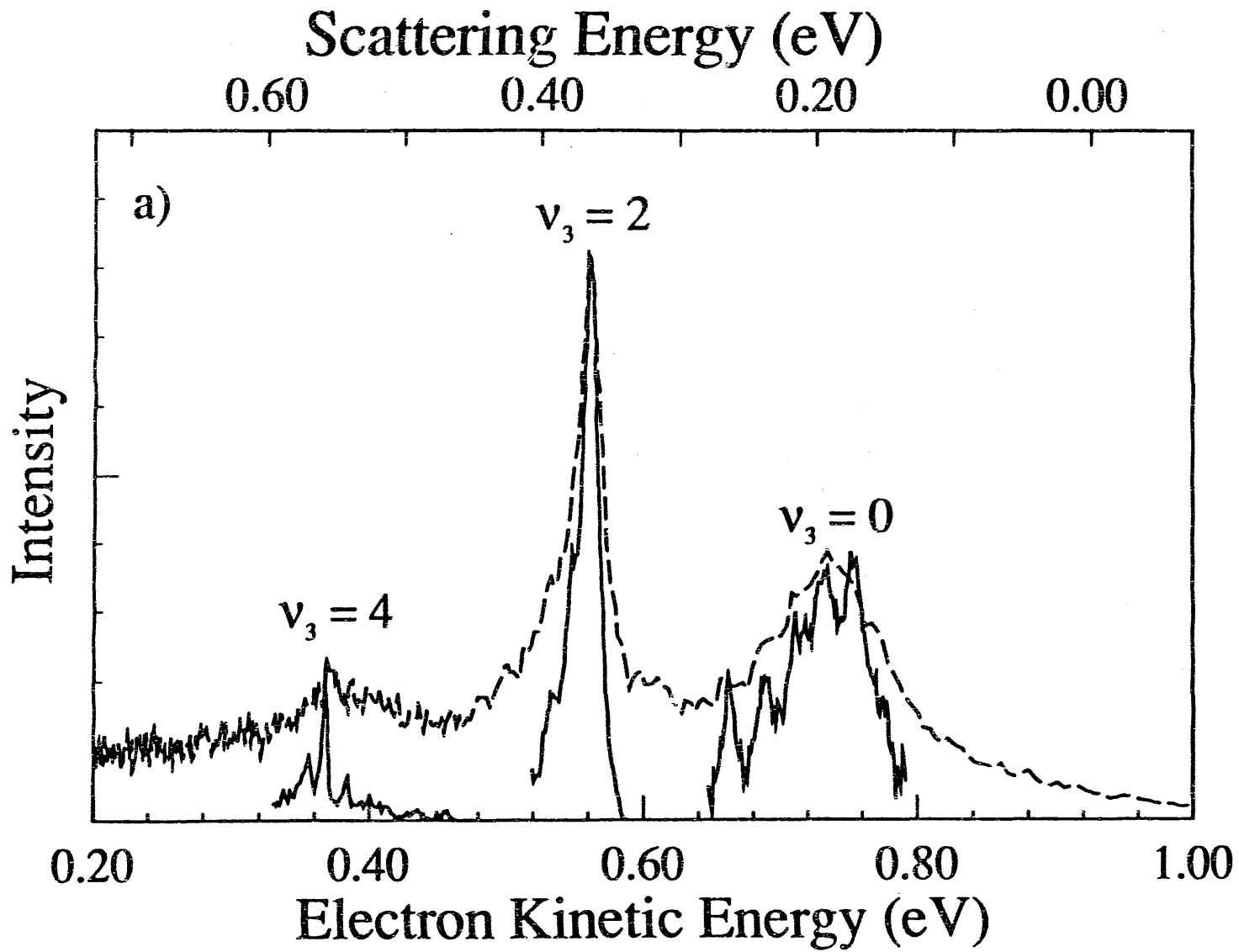


Figure 4-1

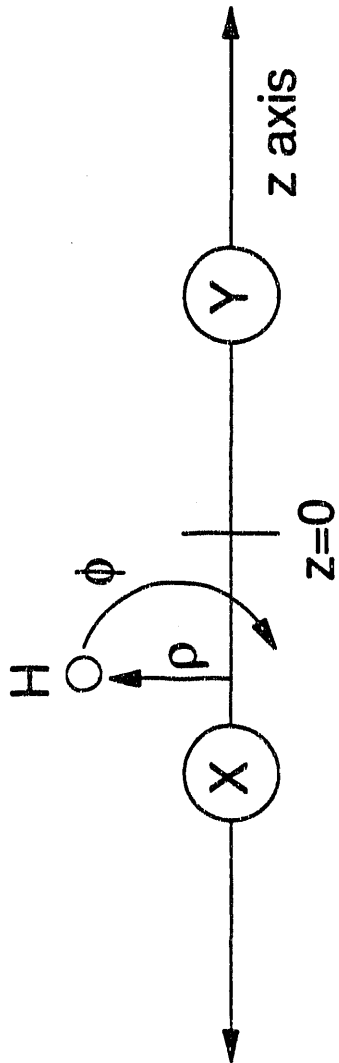


Figure 4-2

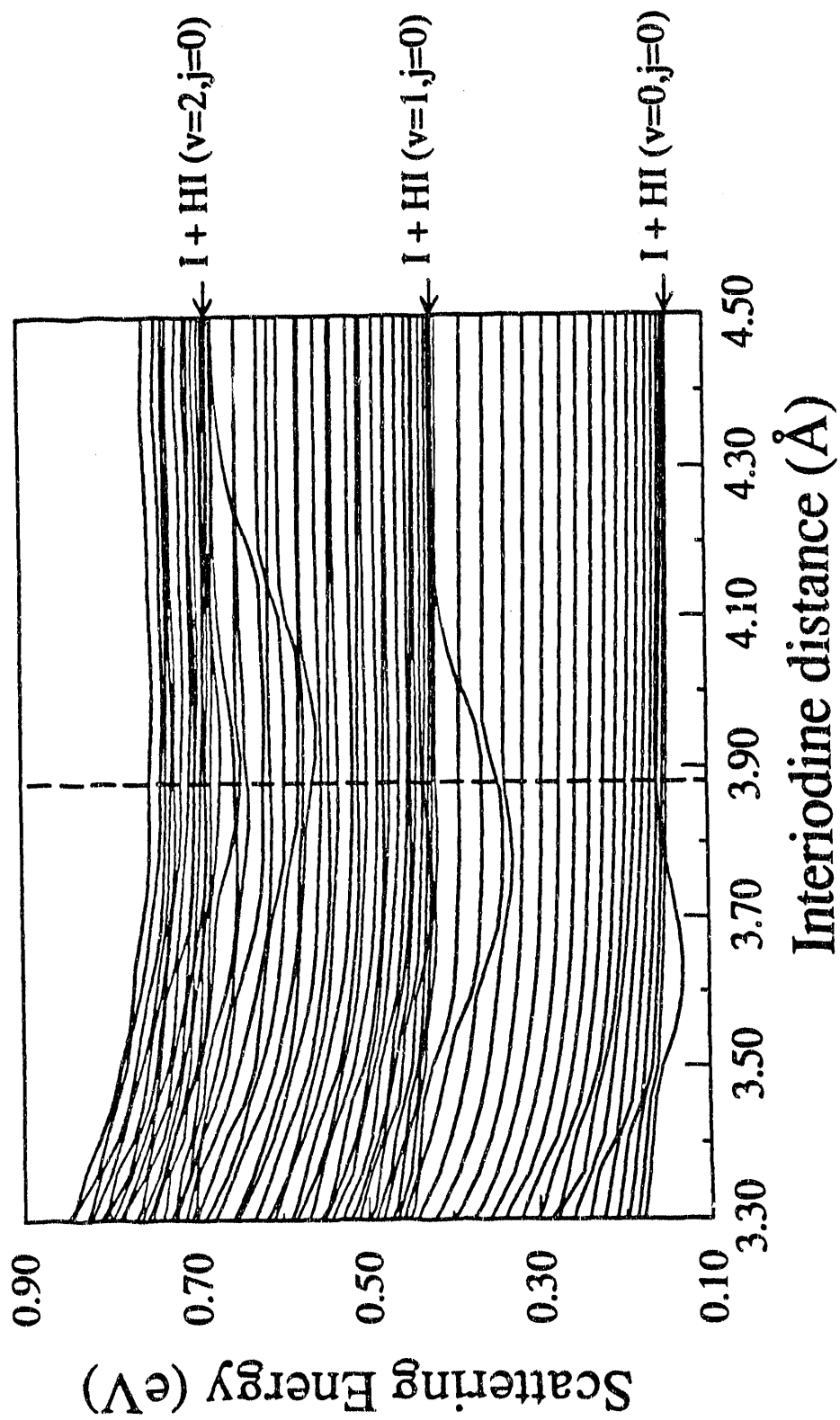


Figure 4-3a

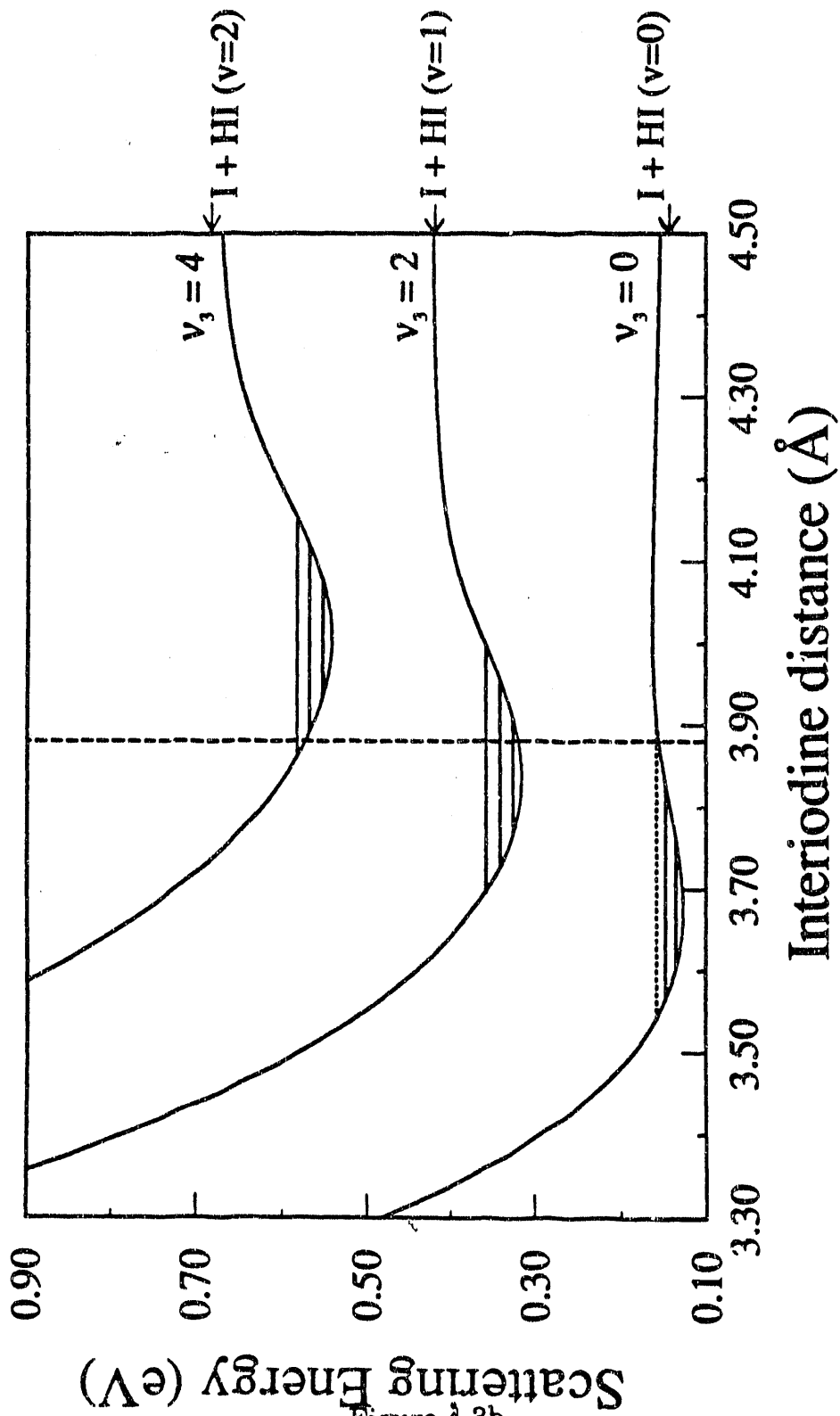


Figure 4-3b

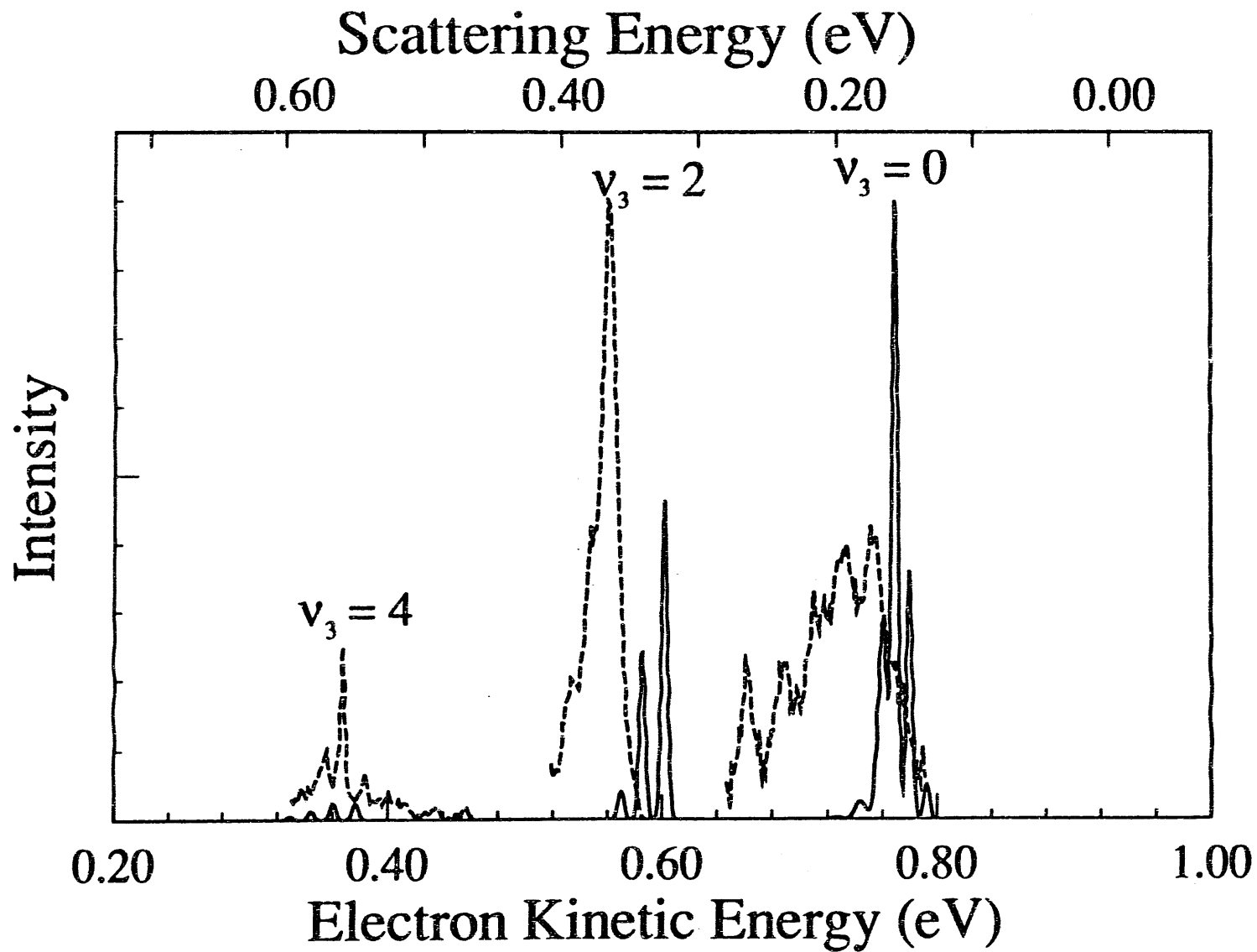


Figure 4-4

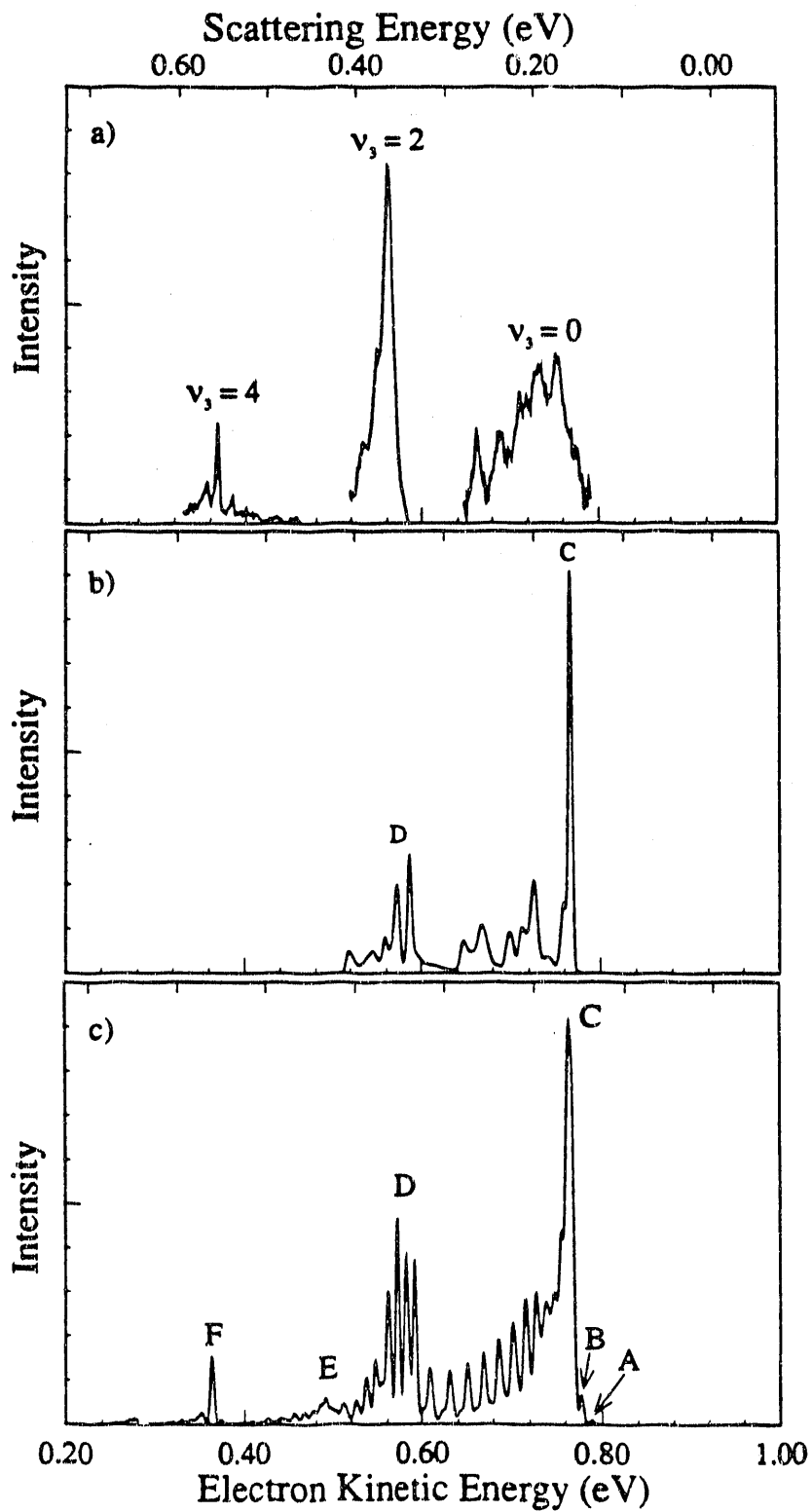


Figure 4-5

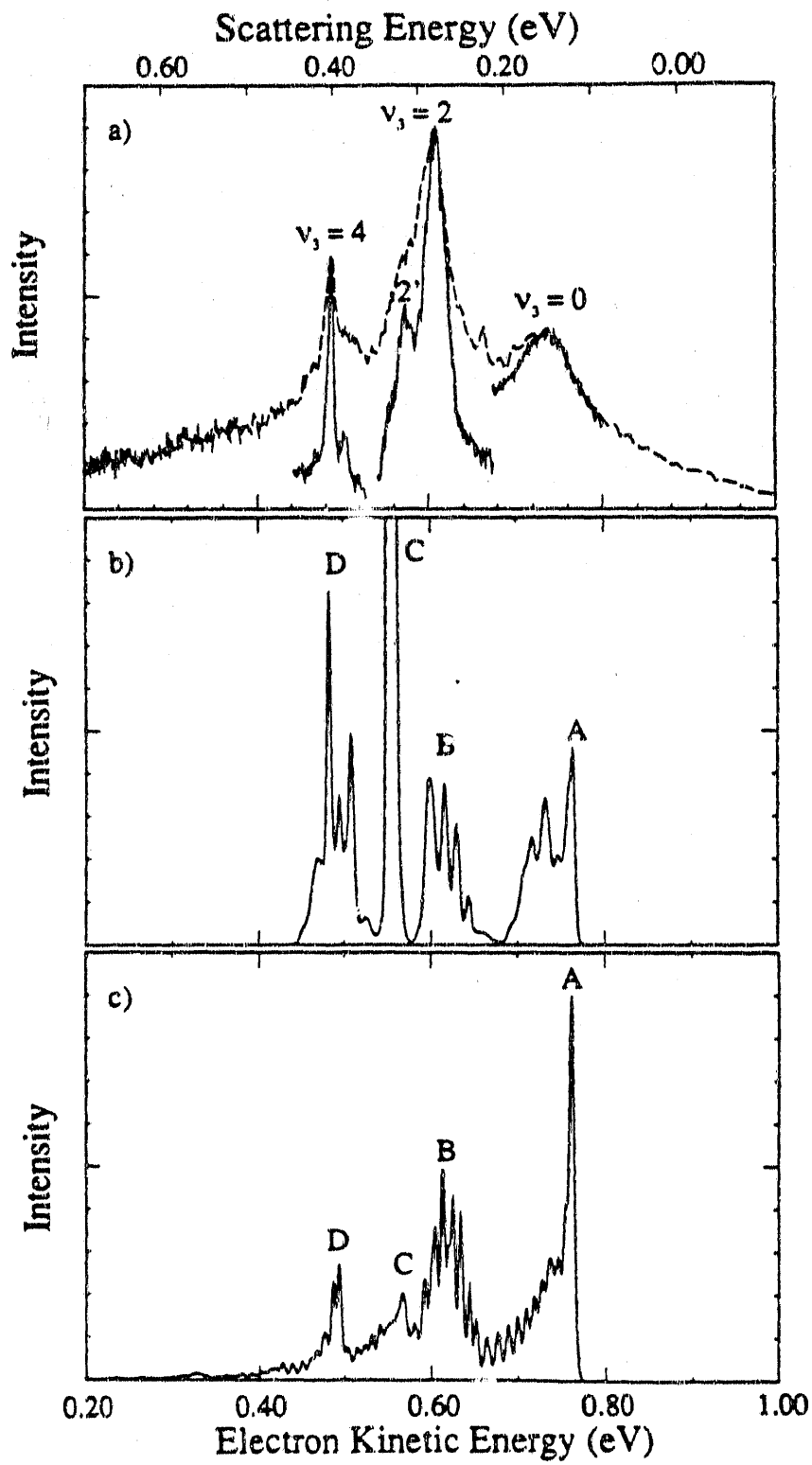


Figure 4-6

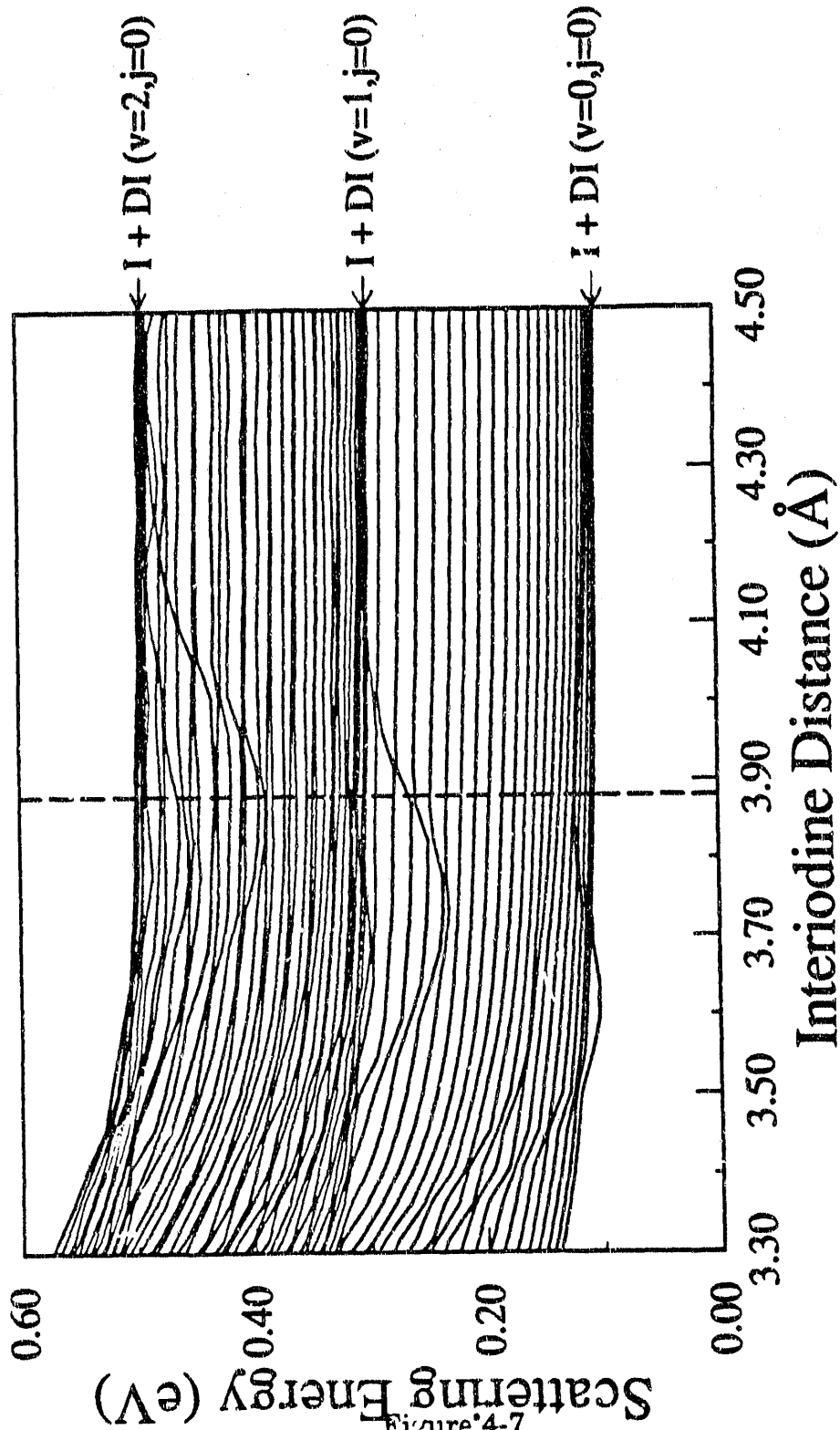


Figure 4-7

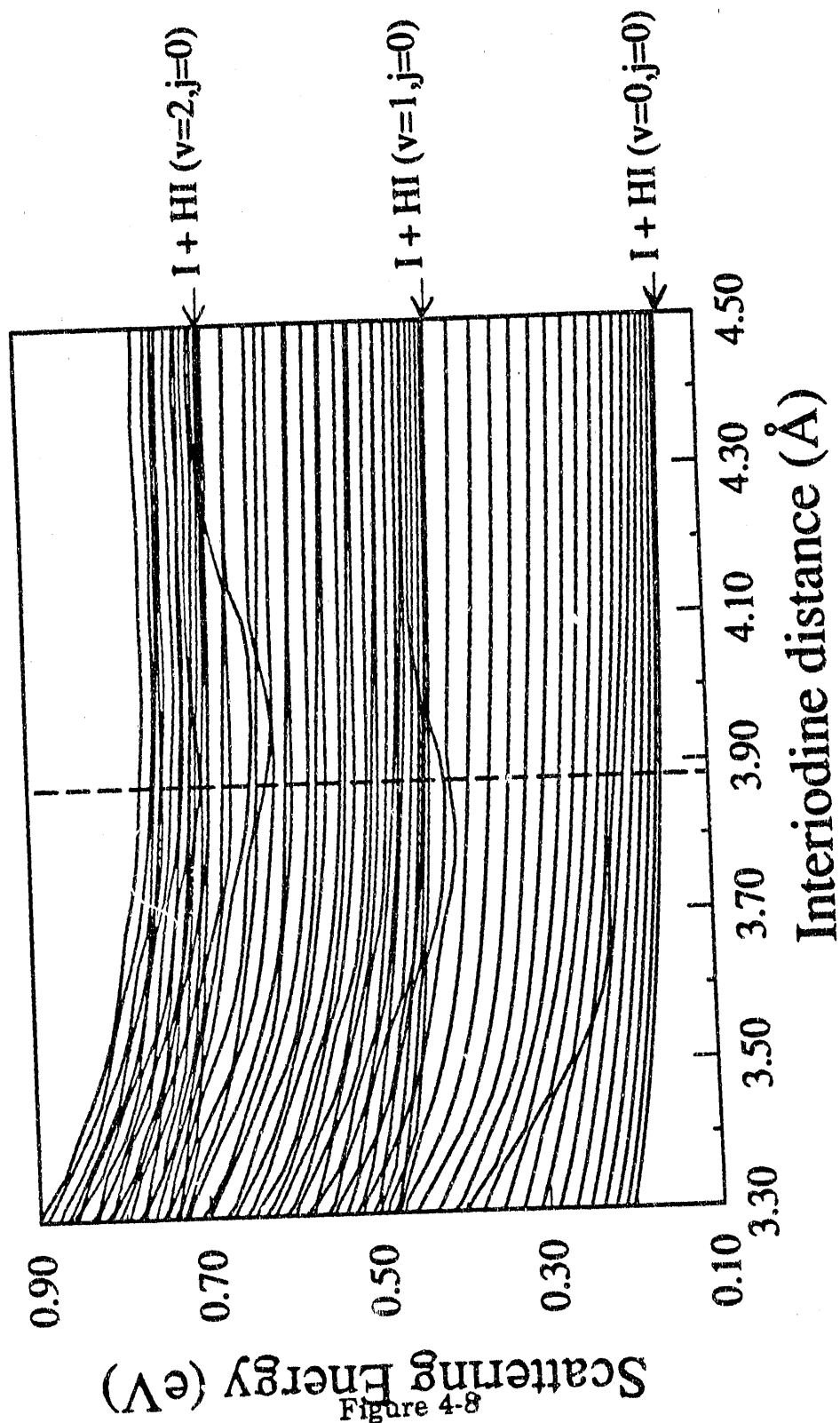


Figure 4-8

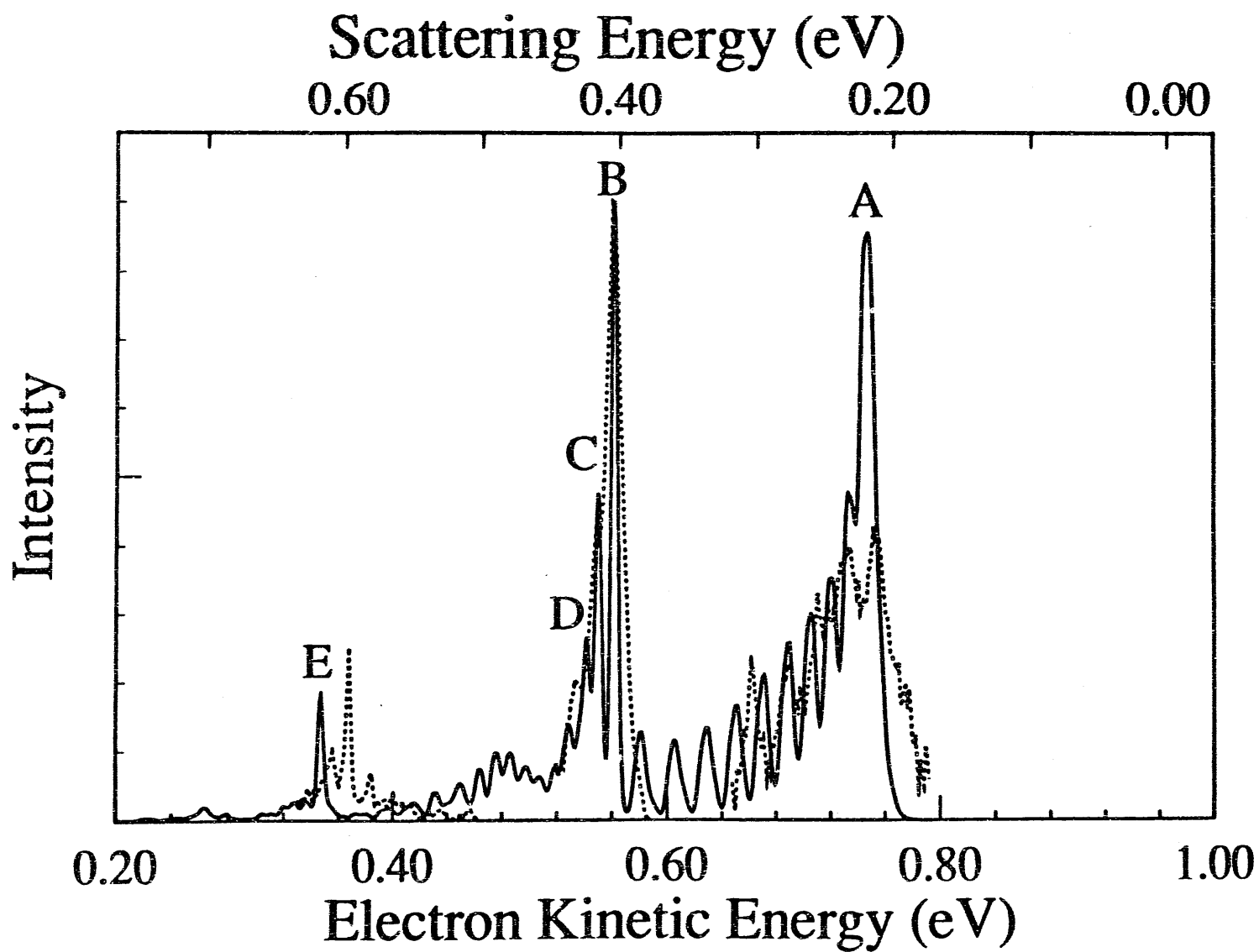


Figure 4-9

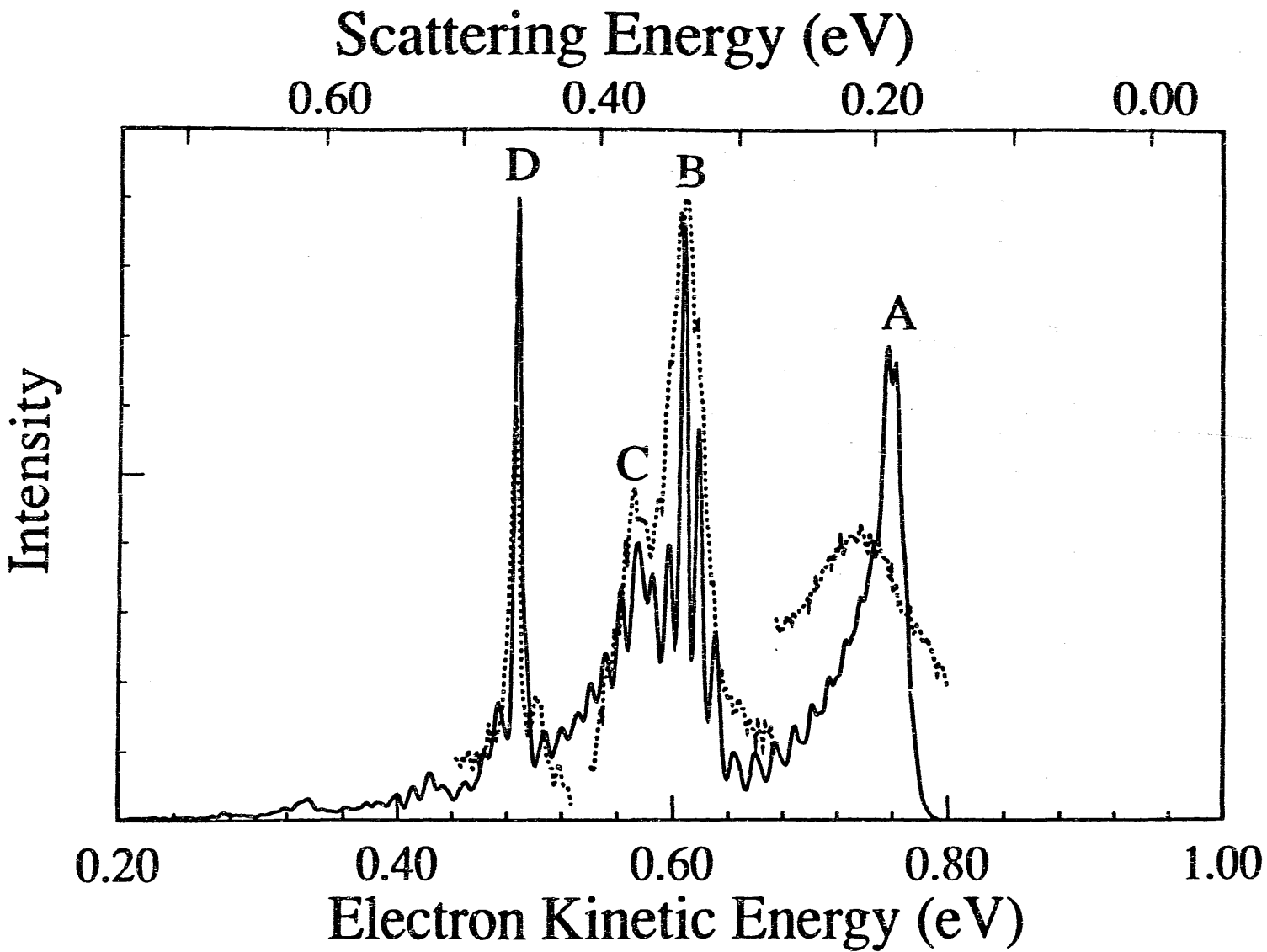


Figure 4-10

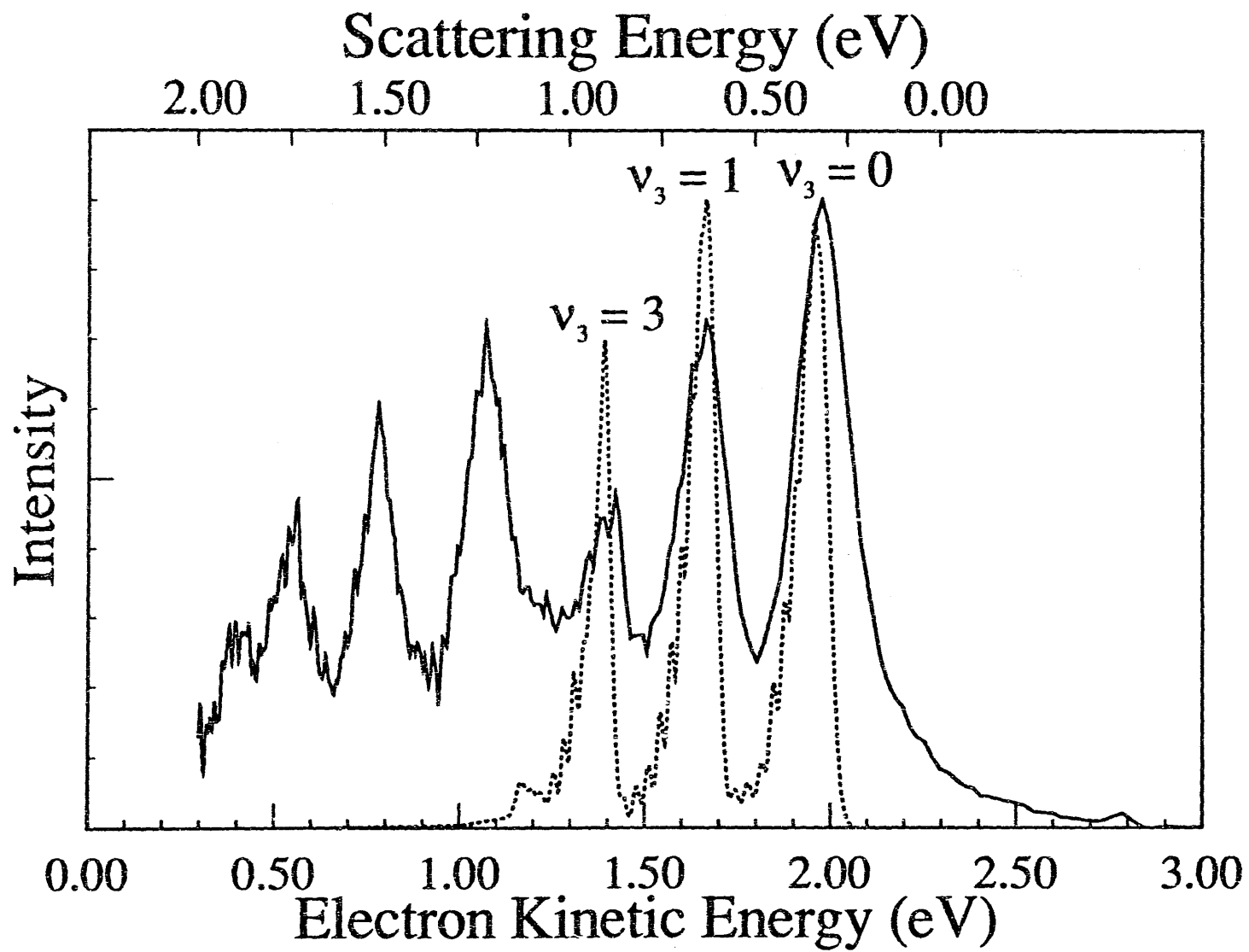


Figure 4-11

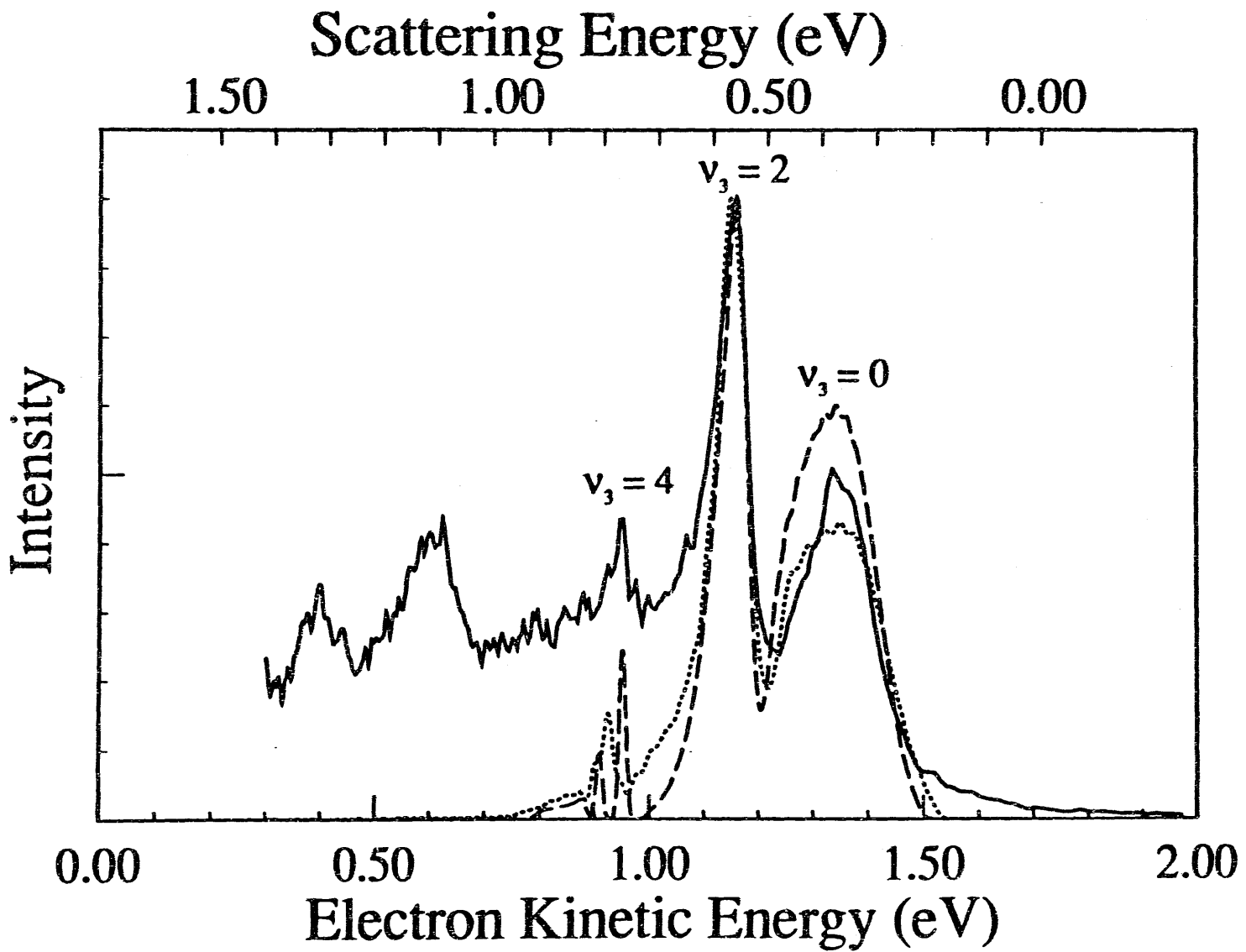


Figure 4-12

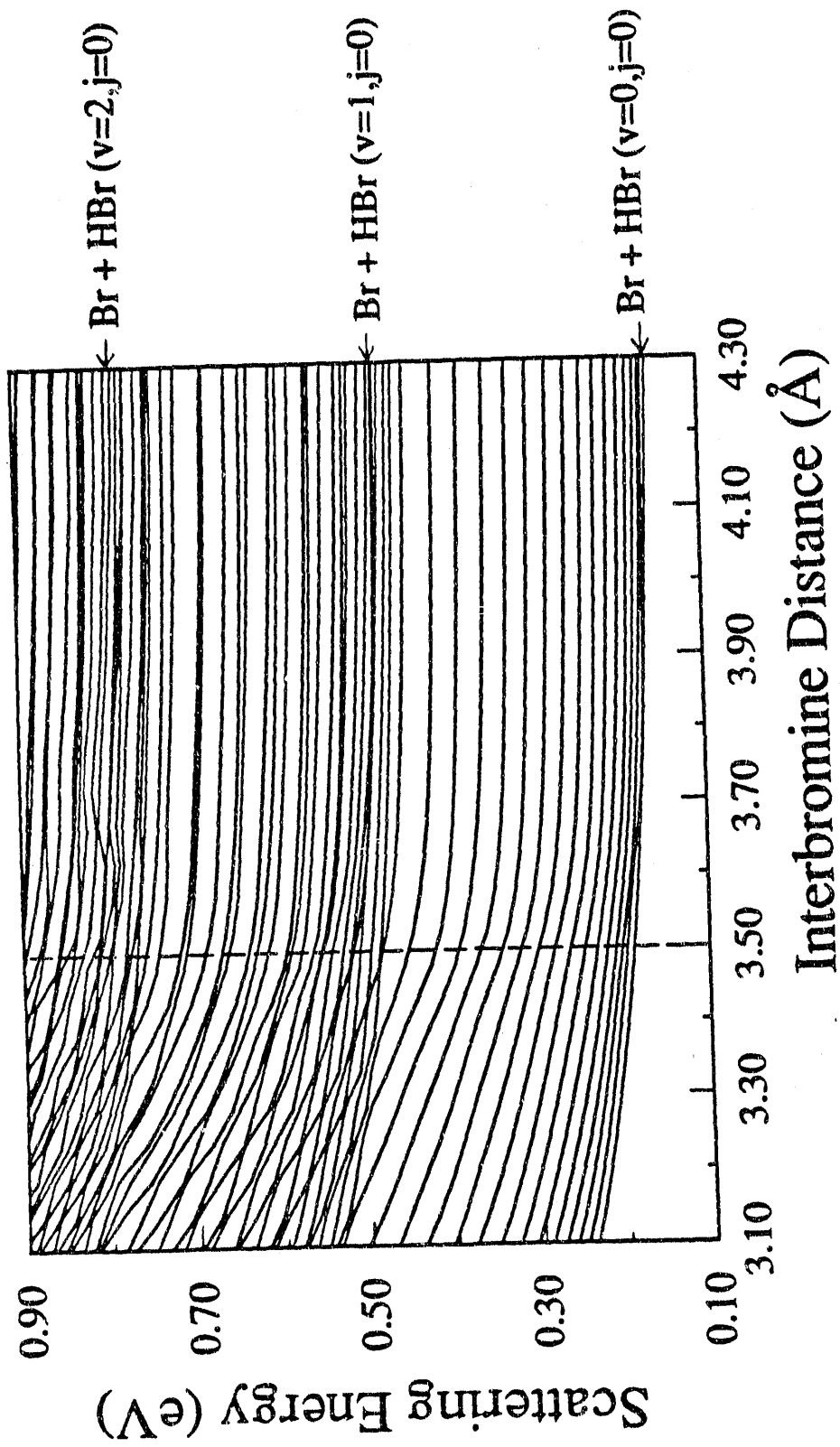


Figure 4-13

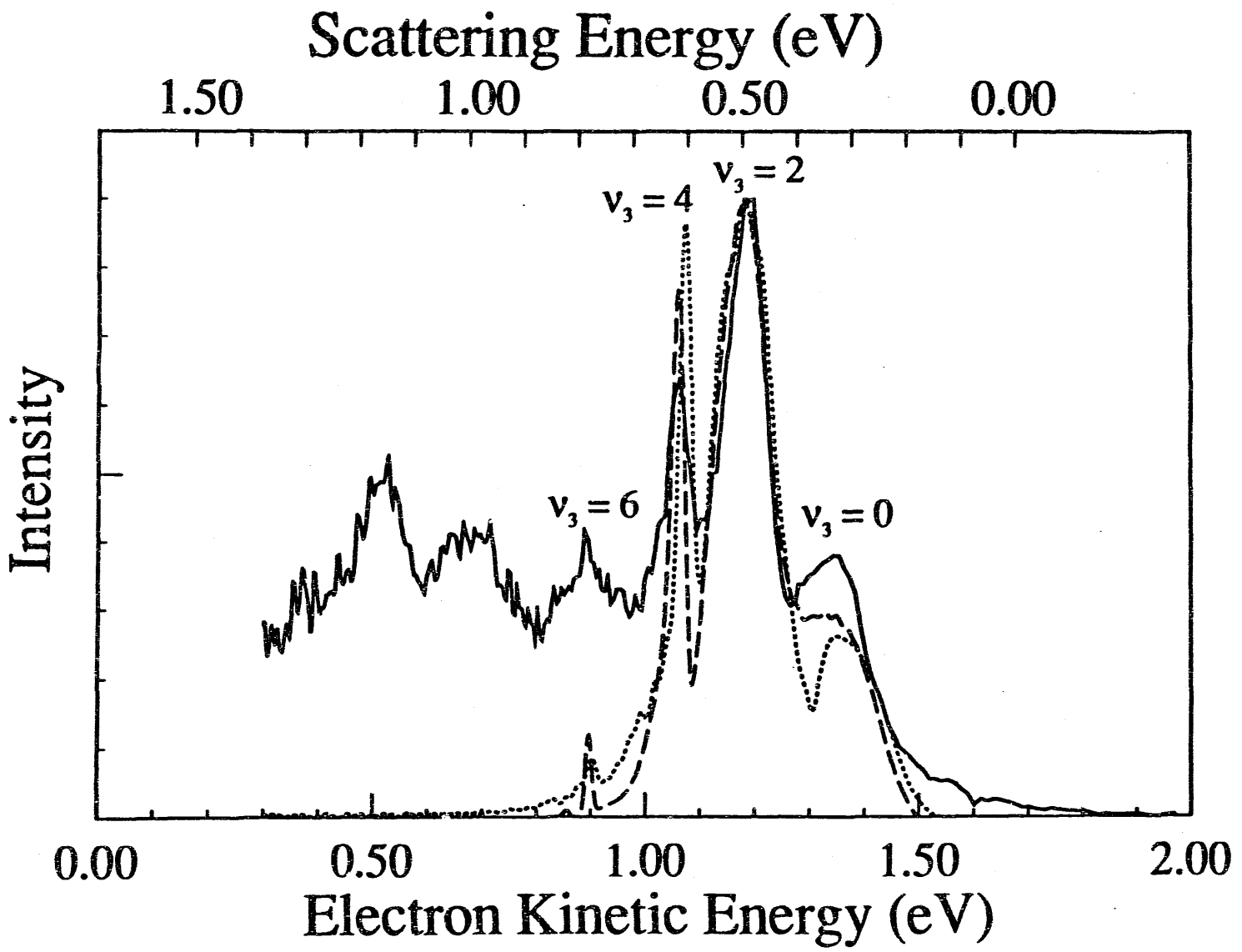


Figure 4-14

Chapter 5. Study of the 2B_1 and 2A_2 states of CH_2NO_2 via ultraviolet photoelectron spectroscopy of the $CH_2NO_2^-$ anion

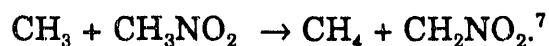
1. Introduction

Negative ion photoelectron spectroscopy is a versatile technique which has been used to determine vibrational frequencies and structural information on negative ions and the neutral species formed by photodetachment.¹ It is particularly suited to the study of neutral free radicals since open-shell systems generally have positive electron affinities. The corresponding anion will therefore be stable with respect to electron detachment. However, when used alone, photoelectron spectroscopy of polyatomic anions suffers from several limitations. Typically, only a small number of vibrational modes are active in the photoelectron spectrum; these are often (but not always) the totally symmetric modes of the neutral. Secondly, while the intensity distribution of the various vibrational progressions in the spectrum is directly related to the normal mode displacements between the anion and neutral, one generally wants to convert these normal mode displacement into changes in molecular geometry. In order to do this, it is necessary to know the form of the normal modes of vibration.² This requires knowing the geometry and the bond stretching and bending force constants for either the anion or the

neutral. Often, neither structure is known, or not enough frequencies have been measured to determine a force constant matrix. In such cases, it is difficult to extract geometric information from the photoelectron spectrum.

Recently, Lineberger^{3,4} and Ellison^{5,6} have presented examples in which the combination of photoelectron spectroscopy and *ab initio* calculations provide a much more detailed picture of bonding in the anion and neutral. We have applied this combination of methods to the $\text{CH}_2\text{NO}_2^-/\text{CH}_2\text{NO}_2$ system, thereby learning about the structure and energetics of the nitromethide anion and nitromethyl radical.

The nitromethyl free radical (CH_2NO_2) is thought to be important in the thermal decomposition of nitromethane via the reaction



The wide use of nitromethane as a propellant, as well as its implication in reactions of the upper troposphere⁸, have provided an impetus to the recent matrix isolation^{8,9} and *ab initio*^{10,11} studies of this radical. There has been some discussion as to the structure of the nitromethyl radical. Among the structures that have been proposed¹⁰ are planar, C_{2v} structures (1, 2); planar, C_s structures (3), and nonplanar structures (4,5) (Chart 5-1).

Jacox has studied the vibrational spectroscopy of the nitromethyl radical suspended in an argon matrix.⁸ Thirty vibrations were found for five isotopomers. The measured frequencies were used to determine⁹ a force constant matrix, based on a C_{2v} reference geometry. The excellent fit

obtained shows that the ground state of CH_2NO_2 is planar and has C_{2v} symmetry. The force constant obtained for the C-N bond is consistent with a single bond, thus favoring structure (1) as the ground state. *Ab initio* calculations on nitromethyl radical have been carried out by McKee, who has studied the radical at the single¹¹ and multi-configuration SCF¹⁰ (MCSCF) levels. At the single configuration SCF level, a C_s ground state (3) is predicted. However, the C_{2v} ground state (1) is lower in energy when correlation is included (UMP2/3-21G). The MCSCF calculations favor a planar, C_s ground state (3), formed by a second-order Jahn-Teller interaction between the 2B_1 (1) and 2A_2 (2) states. At the highest level calculations, the predicted ground state is not yet converged with respect to the basis set used and the amount of correlation included.¹⁰ In our discussion, we will assume a planar, C_{2v} structure for CH_2NO_2 , based on the matrix isolation results.

Bonding in the nitromethide anion, CH_2NO_2^- , can be viewed as a competition between resonance forms 6 and 7. In structure 6, the negative charge is largely localized on the carbon atom, and the C-N single bond is likely bent out of the H-C-H plane. In structure 7, the negative charge is on the oxygen atoms, and the C-N double bond favors a planar structure. Other carbanions of the form CH_2R^- furnish examples ranging between the two extremes.¹² For example, CH_2NC^- can best be described by structure 6', with a C-N single bond and a C-N triple bond, rather than by structure

7' (two C-N double bonds).¹³ This gives rise to a nonplanar structure, and leaves the negative charge substantially localized on the methylene group. Methylene groups are poor electron acceptors, so the electron binding energy of CH_2NC^- is quite low (1.059 eV). CH_2CN^- has substantially more contribution from structure 7', leading to a more nearly planar structure, as well as a higher electron binding energy⁵ (1.543 eV). CH_2CHO^- goes even further, with structure 7' dominating, leading to a planar anion with a C-C double bond and a high binding energy (1.825 eV).¹⁴ We will show that CH_2NO_2^- also has structure 7, and an even higher binding energy (2.475 eV).

In this paper we present the photoelectron spectra of CH_2NO_2^- and CD_2NO_2^- and the results of *ab initio* calculations on CH_2NO_2^- and CH_2NO_2 . We will use these results to show that CH_2NO_2^- is planar, with an electron affinity of 2.475 eV. We will also offer support for Jacox's assignment⁹ of (1) as the ground state of CH_2NO_2 from a measurement of the (infrared inactive) torsion vibration. In addition, we will present experimental results on an excited electronic state of CH_2NO_2 corresponding to structure 2.

2. Experimental

The photoelectron spectrometer used in this work has been described previously,¹⁵ so only a brief description will be presented. Negative ions are generated by crossing a pulsed molecular beam with a 1 keV electron beam. Ions cool in the supersonic expansion; negative ions are extracted from the beam, accelerated to 1 keV, and mass-separated using a Wiley-McLaren time-of-flight mass spectrometer.¹⁶ The mass-selected ion of interest is intersected by the third (355 nm, 3.495 eV) or fourth (266 nm, 4.660 eV) harmonic of a pulsed Nd:YAG laser. A small fraction (0.01 %) of the detached electrons are detected at the end of a 1 m field-free region perpendicular to the ion and laser beams. Photoelectron flight times are then converted to energy. The resolution of the spectrometer is 8 meV (64 cm^{-1}) at 0.65 eV electron kinetic energy (eKE) and degrades as $(\text{eKE})^{3/2}$ at higher eKE. The eKE of a given anion \rightarrow neutral transition is

$$\text{eKE} = h\nu - EA + E^{(-)} - E^{(0)},$$

where EA is the electron affinity, $E^{(-)}$ is the internal energy of the negative ion and $E^{(0)}$ is the internal energy of the neutral. Peaks at *higher* eKE correspond to transitions to states of the radical with *lower* internal energy.

Nitromethide ions are generated by passing 1 atm NF_3 over CH_3NO_2 and expanding the mixture through a pulsed molecular beam valve. The likely mechanism for CH_2NO_2^- formation is as follows. First, dissociative

attachment of low-energy secondary electrons from the 1 keV electron beam produces F^- :



Fluoride ion then reacts with nitromethane entrained in the beam to produce nitromethide by proton abstraction:



The photoelectron spectra shown were obtained by averaging over 150,000 laser shots. All spectra were taken with the laser polarized perpendicular to the direction of electron collection.

3. Results

The 355 nm photoelectron spectra of $CH_2NO_2^-$ and $CD_2NO_2^-$ are shown in Fig. 5-1 and 5-2. The positions of the features in each spectrum are given in Table 5-1. The spectra are surprisingly simple, considering that nitromethide possesses 12 vibrational modes. Peak A, at 1.020 eV, is identified as the 0-0 band, giving a raw electron affinity of 2.475 ± 0.010 eV. Peak a (the small feature at 1.07 eV) is a "hot band" due to detachment of vibrationally excited ions. The remaining features in the spectrum are believed to originate from the ground vibrational state of the ion.

The prominence of a single progression in the $CH_2NO_2^-$ spectrum (peaks A, D, H, L, N) indicates that most of the geometry change between

the ion and neutral is along a single normal mode. The peaks in this progression are spaced by 0.161 eV ($1300 \pm 25 \text{ cm}^{-1}$). The photoelectron spectrum of CD_2NO_2^- (Fig. 5-2) shows a similar progression (peaks A, E, I, L, N) with a nearly identical spacing of $1292 \pm 25 \text{ cm}^{-1}$. Two smaller progressions are also evident. The first (A-B, D-E-F, H-I-J, L-M) is at 409 cm^{-1} in the CH_2NO_2^- spectrum and shifts to 280 cm^{-1} (A-B-C, E-F-G, I-J, L-M) in the CD_2NO_2^- spectrum, indicating that this mode involves hydrogen atom motion. There is also a less intense progression (A-C, D-G, H-K) at $954 \pm 25 \text{ cm}^{-1}$ in the CH_2NO_2^- spectrum. A similar, but less resolved, progression (A-D, E-H, I-K) appears in the CD_2NO_2^- spectrum.

The 266 nm photoelectron spectrum of CH_2NO_2^- (Fig. 5-3) shows a series of peaks at high electron kinetic energy due to detachment to the ground electronic state of CH_2NO_2 , as well as sharp peaks at low eKE due to an excited electronic state of CH_2NO_2 lying 1.591 eV above the ground state. This excited state spectrum is less extended than the ground state spectrum, suggesting that the excited state has a geometry more similar to the anion. In addition, there is a small, very broad feature at $\sim 1.0 \text{ eV}$ electron kinetic energy, between the two sharp bands. The intensity of this band lies just above the noise level in the spectrum. The 266 nm photoelectron spectrum of CD_2NO_2^- is very similar to that of CH_2NO_2^- . The 355 nm spectrum will be discussed first; the excited state results at 266 nm will be briefly discussed at the end of the chapter.

4. Analysis and Discussion

4.1. Qualitative Features

If the ion and neutral have the same symmetry, transitions from the vibrational ground state of the anion to vibrations of any quanta in totally symmetric (a_1 , assuming C_{2v} symmetry) modes and vibrations of *even* quanta in non-totally symmetric modes are allowed.¹⁷ The intensity of the allowed peaks is governed by the Franck-Condon principle. Because of this, the frequency of a non-symmetric mode must be substantially different in the neutral and anion before the vibration will be active and the photoelectron spectrum is typically dominated by progressions in totally symmetric modes.

The peaks in the main progression in the CH_2NO_2^- spectrum are spaced by 1300 cm^{-1} . A comparison with the matrix isolation frequencies of CH_2NO_2 (Table 5-3) shows that this is a progression in the "NO₂ symmetric stretch" (ν_3 , a_1 symmetry), found at 1297 cm^{-1} in the matrix studies.⁸ The form of this normal mode is shown in Fig. 5-4. The C-N bond lengthens as the N-O bonds shorten and the O-N-O angle increases.

The remaining peaks in the spectrum can be identified as combination bands superimposed on the main progression. The small peaks (C, G, K) which appear 0.118 eV ($954 \pm 25\text{ cm}^{-1}$) from the ν_3 peaks are

assigned to the 'C-N stretch' (ν_4), found at 986 cm^{-1} in the matrix studies. This totally symmetric mode (Fig. 5-4) is complementary to the ν_3 mode; as the C-N bond stretches, the N-O bond lengthens and the O-N-O angle increases.

The medium sized peaks (B, E, I, M) at 0.051 eV (409 cm^{-1}) and the smaller features 0.087 eV (700 cm^{-1}) from the main peaks in the spectrum of CH_2NO_2^- are somewhat more puzzling. The lowest-frequency vibration observed in the matrix studies is the NO_2 rock (ν_{12}) at 484 cm^{-1} . If the anion and neutral are both planar, C_{2v} species, we would only observe even quanta of this (non-totally symmetric) mode, leading to a spacing of 968 cm^{-1} . The lowest totally symmetric vibration is the ν_5 (NO_2 scissors) mode at 693 cm^{-1} . The frequency of the mode we observe shifts to 280 cm^{-1} in the spectrum of CD_2NO_2^- . Both ν_5 and ν_{12} were found to have much smaller H-D isotope shifts in the matrix studies. Thus, none of the vibrations observed in the matrix studies appears to be responsible for the 409 cm^{-1} peaks. There is one vibrational mode in the ion that is not infrared active : the torsion mode (ν_6). This is a non-totally symmetric mode with a_2 symmetry. The torsion mode primarily involves hydrogen atom motion, leading to a large frequency shift upon deuteration. We will show that the torsion mode is responsible for this progression. We believe that, for CH_2NO_2 (CD_2NO_2), two quanta of this mode are at 409 (280) cm^{-1} , and four quanta are near 700 (~ 600) cm^{-1} .

It is somewhat surprising that a progression in a non-totally symmetric vibration is observed in the photoelectron spectrum. The reason that the usual propensity rule is not followed will be examined with the aid of molecular orbital theory and *ab initio* calculations which show that the ν_6 frequency is very different in the anion and neutral. In addition, the calculations show the changes in structure and bonding expected upon photodetachment.

4.2. Molecular Orbital Picture

Simple molecular orbital arguments can be used to gain a qualitative feel for the bonding in nitromethyl radical. To construct the molecular orbitals for CH_2NO_2 and CH_2NO_2^- , we use the procedure described by Walsh¹⁸, and treat the CH_2 group as a single atom with 6 valence electrons. Both the anion and radical are assumed to be planar and have C_{2v} symmetry. The relative energies of the molecular orbitals is based on orbital energies found in our *ab initio* calculation on CH_2NO_2^- . The highest occupied molecular orbitals for CH_2NO_2^- have the electron configuration $[8a_1^2 5b_2^2 1a_2^2 2b_1^2]$. The highest occupied molecular orbital ($2b_1$) is C-N π -bonding, N-O anti-bonding, and weakly O-O bonding (Fig. 5-5a). Removal of an electron from the $2b_1$ orbital, which results in the ${}^2\text{B}_1$ ground state of CH_2NO_2 , would cause the C-N bond to lengthen (as the double bond

becomes a single bond), the N-O bonds to shorten, and the O-N-O angle to increase. These changes in geometry should lead to excitation of the ν_3 mode upon detachment, the primary feature observed in the spectrum. In addition, the torsion potential for the 2B_1 state should be very different from that of the anion. Twisting about the single bond in the 2B_1 state should involve only a small barrier, while twisting about the double bond in the anion should have a very large barrier.

The next highest molecular orbital in the ion is the $1a_2$, which consists of out-of-plane π orbitals on the oxygens (Fig. 5-5b). This orbital is weakly O-O antibonding, but is basically nonbonding, so the 2A_2 state of the radical is expected to have a geometry similar to the anion. Note that this implies that the 2A_2 state has a C-N double bond (structure 2), whereas the 2B_1 state has a C-N single bond (structure 1).

4.3. *Ab initio* Calculations

To gain a more quantitative picture than that afforded by simple MO theory, *ab initio* calculations on CH_2NO_2 and $CH_2NO_2^-$ were performed using the Gaussian 88 package.¹⁹ Although several authors have reported calculations on $CH_2NO_2^-$,^{12,20,21} the highest level calculations were carried out at the uncorrelated SCF level by Edgecombe²¹ using the 6-31G* basis set (polarization functions on heavy atoms only). It is well known²² that

diffuse functions are very useful in describing the delocalized extra electron in anions, and it is essential to include correlation effects, if one is to predict electron affinities.²² We have therefore optimized the geometry of CH_2NO_2^- (within the C_{2v} point group) at the HF/6-31++G** (polarization and diffuse functions on all atoms) and MP2/6-31++G** levels (correlation included via second-order Moller-Plesset perturbation theory). Our results are compared to earlier results in Table 5-4. We have also tried allowing the anion to become nonplanar (while maintaining C_s symmetry), but a planar geometry is preferred. Planar, C_s geometries were not explored. Anion geometries significantly distorted from C_{2v} (especially in-plane distortions) would be expected to lead to the excitation of b_2 modes in the radical. This is not observed in the photoelectron spectrum. The planar structure is in contrast to the barrier to planarity seen in CH_2NC^- ,¹³ so structure **6** contributes little. Frequencies for CH_2NO_2^- were calculated at the HF/6-31++G** level (Table 5-5). All the frequencies are real, so the C_{2v} structure is at least a local minimum. The lowest frequency in the ion is the out-of-plane distortion (ν_8) at 535 cm^{-1} . All Hartree-Fock frequencies have been scaled²³ by 0.9.

McKee has done a great deal of *ab initio* work on the CH_2NO_2 radical.^{10,11} A C_{2v} ground state (**1**) is predicted at the single configuration SCF (UMP2/3-21G) level.¹¹ He has also performed multiconfiguration SCF (MCSCF) calculations.¹⁰ In the MCSCF calculations, the geometry was first

optimized using a single configuration, at both C_s and C_{2v} symmetry. MCSCF energies were then determined at these geometries. The MCSCF calculations predict a planar, C_s ground state (3), although the predicted geometry is sensitive to basis set and degree of correlation. In addition, McKee calculated SCF frequencies using several basis sets, including 6-31G*.

On the basis of the matrix spectroscopy work, which indicates a C_{2v} geometry for CH_2NO_2 ,⁹ we calculated optimized C_{2v} geometries at the UHF/6-31+G** and UMP2/6-31+G** levels. Our results are shown and compared to McKee's in Table 5-4. We also investigated small distortions from planarity (still retaining C_s symmetry). At the UHF/3-21G level, a slightly nonplanar geometry is favored, but at the UHF/6-31+G** level a planar geometry is preferred. The electron affinity of CH_2NO_2 is predicted to be 2.274 eV at the UMP2/6-31+G** level, in reasonable agreement with the experimental value of 2.475 eV. Calculations at the MP2 level using basis sets that include diffuse functions have been found to give surprisingly accurate molecular electron affinities.²²

The torsion was explored by calculating energies with the CH_2 group perpendicular to the NO_2 group ($\phi = 90^\circ$, Fig. 5-6). All other bond lengths and angles were left at the planar minimum. Results are summarized in Table 5-6. For $CH_2NO_2^-$, the torsion barrier is 20,724 cm^{-1} (UHF/6-31+G**) and 16,620 cm^{-1} (UMP2/6-31+G**). This barrier is characteristic of a C-N

double bond; for comparison, the torsion barrier in ethylene is 22,750 cm^{-1} ,²⁴ as twisting the molecule breaks the π -bonding, reducing the double bond to a single bond. The torsion barrier is much lower for CH_2NO_2 . It is 3095 cm^{-1} at the UHF/6-31+G** level and 1859 cm^{-1} at the UMP2/6-31+G** level. For comparison, the torsion barrier about the single bond in ethane is 960 cm^{-1} .²⁵ Allowing the other bond lengths to change in the twisted geometry (within the C_{2v} framework) lowered the torsion barrier by only 42 cm^{-1} , to 1817 cm^{-1} . The torsion barrier is very sensitive to the amount of C-N π bonding in the molecule, much more so than the C-N stretching force constant. The small barrier strongly suggests that the C-N bond is a single bond, and, although resonance structures for the radical can be drawn with both C-N single (1) and double bonds (2), only the single bonded structure contributes significantly for the ground state of the radical. McKee¹⁰ reports similar results for the torsion barrier in the ${}^2\text{B}_1$ ground state and a torsion barrier of about 30,000 cm^{-1} in the ${}^2\text{A}_2$ state of the radical, consistent with structure 2.

Our calculations of the anion and neutral geometries at the MP2/6-31+G** level allow us to predict geometry changes upon detachment. Changes in the CH_2 group are predicted to be small (Table 5-7). The C-H bonds lengthen by 0.001 Å, and the H-C-H angle increases by 2.0°, which is consistent with the lack of high-frequency vibrations in the spectrum. Upon detachment, the C-N double bond in the anion becomes a single bond, and

the N-O bonds become stronger. The C-N bond lengthens significantly by 0.083 Å, the N-O bonds shorten by 0.049 Å, and the O-N-O angle increases by 5.9°. These geometry changes are consistent with excitation of the NO₂ symmetric stretching mode (ν_3) (Fig. 5-4), which lengthens the C-N bond, while shortening the N-O bonds and widening the O-N-O angle. The change in the C-N bond leads to a dramatic change in the torsion barrier, and this is manifested in the spectrum as an excitation of the torsion (ν_6) mode.

4.4. Detailed Analysis of Spectrum

In this section, we use the experimental spectra and some of the *ab initio* results to quantitatively determine structural and bonding changes upon photodetachment. These can then be compared to the changes predicted purely on the basis of the *ab initio* calculations. The torsion potential, which is very sensitive to the nature of the C-N bond, will be treated first. The symmetric modes, including a prediction of the geometry change upon photodetachment based on our experimental results, will be treated in the next section.

4.4.1. Torsion

The torsion is the only vibration of a_2 symmetry, and can thus be treated separately from the other 11 modes. The Hamiltonian for this system is²⁶

$$\mathcal{H} = -\frac{\hbar^2}{2I} \frac{\partial^2}{\partial \phi^2} + V(\phi) \quad (1)$$

where ϕ is the torsion angle (the angle between the CH_2 and NO_2 planes). I is the reduced moment of inertia for the system

$$I = \frac{I_{\text{CH}_2} I_{\text{NO}_2}}{I_{\text{CH}_2} + I_{\text{NO}_2}} \quad (2)$$

where

$$I_{\text{CH}_2} = 2M_H[R_{\text{CH}} \sin(\angle \text{HCN})]^2 \quad (3)$$

$$I_{\text{NO}_2} = 2M_O[R_{\text{NO}} \sin(\angle \text{CNO})]^2 \quad (4)$$

Using the UMP2/6-31++G** geometry, $I_{\text{CH}_2} = 1.861 \text{ amu} \cdot \text{\AA}^2$, $I_{\text{CD}_2} = 3.719 \text{ amu} \cdot \text{\AA}^2$, and $I_{\text{NO}_2} = 39.152 \text{ amu} \cdot \text{\AA}^2$, so I is $1.777 \text{ amu} \cdot \text{\AA}^2$ for both CH_2NO_2 and CH_2NO_2^- and $3.396 \text{ amu} \cdot \text{\AA}^2$ for CD_2NO_2 and CD_2NO_2^- . We use a potential²⁵ of the form

$$V(\phi) = \frac{b}{2} [1 - \cos(2\phi)] \quad (5)$$

where b is the torsion barrier. This is a double minimum potential, with minima at $\phi = 0^\circ$ and 180° , where ϕ is 0° for the planar configuration, 90° at the twisted geometry, and 180° for the second (equivalent) planar configuration.

If $b=0$, then $V(\phi) = 0$ and the solutions²⁶ to the Schrodinger Equation are the free rotor wavefunctions $P_j^0(\phi)$ with energy

$$E_j^0 = \frac{j^2 \hbar^2}{2I} \quad (6)$$

For $b \neq 0$, we expand the wavefunction in a free rotor basis

$$P_\nu(\phi) = \sum_{j=-N}^N C_{j,\nu} P_j^0(\phi) \quad (7)$$

Matrix elements of the Hamiltonian can be found analytically and are given by

$$H_{j,k} = \langle P_j^0 | \mathcal{H} | P_k^0 \rangle = (E_j^0 + \frac{b}{2}) \delta_{j,k} - \frac{b}{4} \delta_{j+2,k} - \frac{b}{4} \delta_{j,k+2} \quad (8)$$

The (tridiagonal) Hamiltonian matrix is set up using a basis of 71 free rotor functions ($N = 35$) and diagonalized using standard routines.

Franck-Condon factors are given by

$$FCF = |\langle P_\nu' | P_\nu'' \rangle|^2 = \left[\sum_{j=-N}^N C_{j,\nu}'' C_{j,\nu}' \right]^2 \quad (9)$$

where the primed quantities refer to the neutral wavefunction and the doubly primed to the anion.

The experimental peak spacings and intensities are best fit with $b' = 1250 \pm 250 \text{ cm}^{-1}$ in the radical and $b'' = 25,000 \pm 10,000 \text{ cm}^{-1}$ in the anion. The corresponding potentials are shown in Fig. 5-6, along with several low-lying vibrational levels. The lowest anion levels are far below the barrier. For these levels, the potential looks much like that for two distinct harmonic oscillators, one with a minimum at $\phi = 0^\circ$, the other at 180° . Eigenvalues occur in pairs. For CH_2NO_2^- , the first two are at 485 cm^{-1} and are separated by $\ll 0.01 \text{ cm}^{-1}$. These correspond to two $v=0$ harmonic oscillator wavefunctions: in phase (forming the 0^+ wavefunction) and out of phase (forming the 0^- wavefunction). These nearly degenerate levels are equally populated in our ion beam. The next pair of levels ($1^+/1^-$) are 964 cm^{-1} higher in energy, and are not expected to be populated in our ion beam. Similarly, for the neutral, using $b' = 1250 \text{ cm}^{-1}$, the first few levels occur in nearly degenerate pairs. The $0^+/0^-$ peaks lie 106 cm^{-1} above the minimum, the $1^+/1^-$ peaks are 208 cm^{-1} above the $0^+/0^-$, the $2^+/2^-$ are 405 cm^{-1} above the $0^+/0^-$, etc. From the 0^+ level of the anion, only transitions to the 0^+ , 2^+ , 4^+ , ... levels of the neutral are allowed by symmetry.²⁷ From the 0^- level, transitions to the 0^- , 2^- , 4^- , ... levels are allowed. As both the 0^+ and 0^- levels are equally populated in our source, the peaks in our spectrum are due to transitions from the $0^+/0^-$ states of the ion to the $0^+/0^-$, $2^+/2^-$, and $4^+/4^-$ levels of the neutral. Note that the 4^+ and 4^- states are separated by

only 0.35 cm^{-1} , which is far below our resolution, and the separation between the lower-lying pairs is even smaller.

The peak positions are quite sensitive to b' , while the peak intensities are not very sensitive to b'' , so the value obtained for b'' must be considered approximate. The values from the analysis of the spectrum are in fairly good agreement with our best *ab initio* values ($b' = 1817 \text{ cm}^{-1}$, $b'' = 16,620 \text{ cm}^{-1}$, Table 5-6). A progression in the non-totally symmetric torsion mode occurs because the torsion potentials for the ion and neutral are very different. The torsion "frequency" (the 0^+ to 1^+ spacing) is 964 cm^{-1} in the anion and 208 cm^{-1} in the radical, although they both have minima at $\phi = 0^\circ$ and 180° corresponding to a planar, C_{2v} structure.

The value of b'' is consistent with a C-N double bond in the anion, and b' is consistent with a C-N single bond in the ground state of the radical. This change in bonding is supported by the C-N stretching force constant, which drops from a scaled, *ab initio* value of $7.59 \times 10^2 \text{ N/m}$ in the CH_2NO_2^- anion to $4.89 \times 10^2 \text{ N/m}$ in the CH_2NO_2 radical. In spite of this change in the C-N bond, the frequency of the "CN stretch" mode changes very little upon detachment (it is 1014 cm^{-1} in CH_2NO_2^- and 986 cm^{-1} in CH_2NO_2). This is because the "CN stretch" mode involves a significant amount of N-O stretching, and the N-O bonds strengthen when the C-N bond weakens. The C-N bond length also changes upon detachment. This will be examined in the next section.

4.4.2. Totally Symmetric Modes

The remaining progressions in the spectrum are due to the totally symmetric ν_3 and ν_4 vibrations of the radical. The corresponding Q_3 and Q_4 normal modes are shown in Fig. 5-4. The normal mode displacements upon detachment can be determined by fitting the observed peak intensities for these progressions. We assume the two modes are separable and have the same form in the anion as in the neutral. This parallel mode approximation²⁸ is justified by the nearly equal ν_3 and ν_4 experimental neutral and *ab initio* ion frequencies. The best fit to the spectrum is given by a displacement of $0.26 (\pm 0.03) \text{ \AA amu}^{-1/2}$ in Q_3 and $0.08 (\pm 0.02) \text{ \AA amu}^{-1/2}$ in Q_4 , where the uncertainties include the effect of varying the anion frequency by $\pm 20\%$. The simulated spectrum is shown in Fig. 5-1 (dotted lines). The simulated spectrum includes progressions in the ν_3 , ν_4 , and ν_6 modes and shows good agreement with experiment. The simulated peaks have been given widths of 5 meV to facilitate comparison with experiment. The actual peaks observed are ~ 20 meV wide.

In order to relate these normal mode displacements to changes in internal coordinates, we need the form of the normal modes.² The force constant matrix (and hence the normal modes) were determined by Jacox⁹ by fitting 30 matrix frequencies for five isotopomers of CH_2NO_2 . This was done using a particular reference geometry for CH_2NO_2 ; we have repeated

the calculation using our best *ab initio* reference geometry. The results of the fit (including the form of the normal modes) do not depend on the bond lengths, but are slightly sensitive to changes in bond angles.⁹ The reference geometry used by Jacox had C_{2v} symmetry with $\angle HCH = 122^\circ$ and $\angle ONO = 122^\circ$, compared to our best *ab initio* values of 127.2° and 125.8° . Thus, we have fit the 30 matrix frequencies to the same 16 parameter force constant matrix used by Jacox, using our MP2 reference geometry. The fit (Table 5-8) is slightly better than that obtained by Jacox, the average difference between predicted and observed frequencies in our fit is 1.8 cm^{-1} , versus 2.2 cm^{-1} . The C-H stretching force constant (F_{CH}) is unchanged, F_{CN} increases from 475 to 489 N/m (still consistent with a C-N single bond), and F_{NO} increases from 807 to 851 N/m. The fitted force constants (using the MF'2 reference geometry) and corresponding normal modes are used in the remainder of the analysis. As mentioned above, the neutral modes are used for the anion. We can now use the photoelectron spectrum to determine the change in molecular geometry upon detachment.

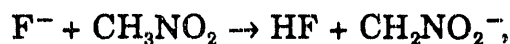
For harmonic modes (as we have here), the photoelectron spectrum gives the geometry change, but not the direction in which it occurs. Here we first turn to the molecular orbital arguments for qualitative insight. Photodetachment to the 2B_1 ground state removes an electron from the $2b_1$ orbital, which is C-N π -bonding and N-O antibonding. We therefore expect the C-N bond to lengthen and the N-O bonds to shorten, so the direction of

the Q_3 displacement between the anion and neutral is that shown in Figure 5-4. The lengthening of the C-N bond should weaken it, and this is also consistent with the large drop in the torsion barrier. The Q_4 mode also chiefly involves C-N stretching, but the displacement in this mode is smaller than the Q_3 displacement. One can infer the sign of the displacement in Q_4 from the *ab initio* calculations, which predict that the C-N bond lengthens by 0.083 Å upon detachment (Table 5-7). The Q_3 displacement accounts for 0.063 Å. If we add the change in C-N bond length due to v_4 (0.022 Å), then the C-N bond is predicted to lengthen by 0.085 Å; if we subtract it we get only 0.041 Å. Adding it is clearly in better agreement with the *ab initio* results, so the Q_4 displacement is also as shown in Figure 5-4. The experimentally predicted and *ab initio* geometry change upon detachment are in excellent agreement (see Table 5-7). The photoelectron spectrum does not directly give the geometry of either the ion or neutral; it gives the change in geometry between the two. Unfortunately, there is no experimental geometry available for either species. If the geometry of the ion or neutral were accurately known, our experimental results would give the geometry of the other species.

Once the geometry change for CH_2NO_2^- detachment has been determined and the force constant matrix is known, we can simulate the photoelectron spectrum of CD_2NO_2^- . This provides a check on our fit. The result is shown in Figure 5-2. The simulated spectrum is also in excellent

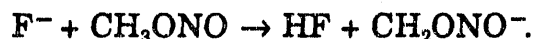
agreement with experiment. The peaks in the photoelectron spectra of CH_2NO_2^- and CD_2NO_2^- are about 15 meV broader than our instrumental resolution. This may be due to the presence of sequence bands (transitions from $v'' = 1$ in the anion to $v' = 1$ in the neutral).

Thus far, we have assumed that all the features we observe are due to CH_2NO_2 . Several studies^{29,30,31} have shown that CH_3NO_2 can isomerize to methyl nitrite, CH_3ONO , which lies 0.1 eV higher in energy, by passing over a 2.4 eV barrier.²⁹ Presumably, nitrite isomers also exist for CH_2NO_2 and CH_2NO_2^- . These isomers are not likely to be responsible for the structure in our spectra. If the $\text{CH}_2\text{NO}_2 \rightarrow \text{CH}_2\text{ONO}$ isomerization barrier is similar to that of CH_3NO_2 , photodetachment of CH_2NO_2^- would access states of CH_2NO_2 , while detachment of CH_2ONO^- would access states of CH_2ONO . The anions in our source are formed by the reaction



which is 0.6 eV exothermic. Even if the entire exothermicity were left in the CH_2NO_2^- , this is not likely to be enough energy to surmount the isomerization barrier and form CH_2ONO^- . The observed ground state frequencies are in excellent agreement with those of CH_2NO_2 , so the dominant isomer in the beam is clearly CH_2NO_2^- . If the CH_2ONO^- isomer were present in the beam, it would be expected to have a lower electron binding energy than CH_2NO_2^- and thus contribute features at higher

electron kinetic energy than peak A. Using methyl nitrite as a precursor could well lead to formation of the CH_2ONO^- anion



The photoelectron spectrum of CH_2ONO^- would then give information on the potential energy surface near the CH_2ONO minimum.

4.5. Excited Electronic State

The 266 nm photoelectron spectrum of CH_2NO_2^- (Fig. 5-3) shows structure due to an excited electronic state of the CH_2NO_2 radical. The progression due to this electronic state is significantly less extended than that due to the ground state, suggesting that this electronic state looks more like the anion than the ground state does. This state is likely formed by the removal of a slightly O-O antibonding $1a_2$ electron from the anion, forming the 2A_2 excited state. This state is predicted to lie 0.35 eV *below* the 2B_1 ground state in McKee's MCSCF + CI calculations¹⁰ and 2.67 eV *above* the ground state in our single configuration (MP2/6-31+G**) calculations. Experimentally, it lies 1.591 eV *above* the 2B_1 state.

The analysis of the photoelectron spectrum of the excited state is somewhat more complicated than that of the ground state, due to the lack of previous experimental information on the 2A_2 state. *Ab initio* calculations on the excited state are difficult, because it is an excited state of a radical

and suffers from significant spin contamination.^{22,32} We have calculated the geometry of the 2A_2 state at the UMP2/6-31+G** level (Table 5-4). This result is expected to be less accurate than the results on $CH_2NO_2^-$ and CH_2NO_2 (2B_1). The removal of a nonbonding electron from $CH_2NO_2^-$ is expected to have little effect on the force constants - the vibrational frequencies of the 2A_2 state are expected to be close to those of the anion, so rather than calculate frequencies for the 2A_2 state, we have used the frequencies calculated for $CH_2NO_2^-$ (Table 5-5). Peak A' (0.594 eV, Table 5-2) is clearly the origin. Peak B' (605 cm^{-1} from the origin) is assigned to one quanta in the ν_6 (NO_2 scissors) mode (646 cm^{-1} in the anion). Peak D' (1218 cm^{-1}) corresponds to two quanta in ν_6 . The observed excitation of the NO_2 scissors mode is reasonable, as the most significant geometry change upon detachment to the 2A_2 state is a 7.2° decrease in the O-N-O angle, consistent with removing an electron from an O-O antibonding orbital.

The remaining small peaks are likely due to a small excitation in the other totally symmetric modes. Thus, peak C' (936 cm^{-1}) could correspond to ν_4 (CN stretch, 1014 cm^{-1} in the ion) and peak E' (1444 cm^{-1}) could be due to one quanta in the ν_3 (NO_2 symmetric stretch) mode (1349 cm^{-1} in the ion). It is not clear what peak F' is due to. The photoelectron spectrum of the 2A_2 state is basically unchanged by deuteration. No progression in the torsion is expected (or seen), as both the anion and 2A_2 state have large calculated barriers to torsion (20,000 - 30,000 cm^{-1}).¹⁰

The broad feature at ~ 1.0 eV electron kinetic energy in Fig. 5-3 could be due to an unresolved extended progression spread over many vibrational modes. This may result from transitions to an electronic state with a geometry significantly distorted from C_{2v} . Thus, this broad feature may be an observation of a state of CH_2NO_2 corresponding to structure 3, 4, or 5. Also, our results do not preclude the possibility that there are other excited states of CH_2NO_2 below the 2A_2 state. Only states formed by removing one electron from the anion will be seen in the photoelectron spectrum¹ (ie, we would not see quartet states, as these would involve removing one electron and promoting one electron). Thus, while the 2A_2 state is certainly a low-lying excited electronic state of CH_2NO_2 , it is not necessarily the lowest excited state.

4.6. Thermodynamics

The electron affinity of the nitromethyl radical can be used to derive several thermodynamic quantities. The C-H bond strength in nitromethane can be determined from³³

$$D(H-CH_2NO_2) = \Delta_{H,acid}(CH_3NO_2) + E. A. (CH_2NO_2) - I. P. (H)$$

where $\Delta_{H,acid}(CH_3NO_2)$ is the gas phase acidity of nitromethane (1493 ± 12 kJ/mol),³⁴ the electron affinity of CH_2NO_2 is 239 ± 1 kJ/mol and the ionization potential of hydrogen is 1312 kJ/mol. This gives a bond

dissociation energy of 420 ± 12 kJ/mol. This value, with the heat of formation of nitromethane (-75 ± 1 kJ/mol)³³ gives a value of 127 ± 12 kJ/mol for the heat of formation of CH_2NO_2 . If we assume the electron to have a heat capacity of zero, then the heat of formation of CH_2NO_2^- is approximately

$$\Delta_f H(\text{CH}_2\text{NO}_2^-) = \Delta_f H(\text{CH}_2\text{NO}_2) - E. A. (\text{CH}_2\text{NO}_2).$$

This gives $\Delta_f H(\text{CH}_2\text{NO}_2^-) = -112 \pm 12$ kJ/mol, in excellent agreement with the previous value (-114 ± 13 kJ/mol).³³

5. Conclusions

From the photoelectron spectra of CH_2NO_2^- and CD_2NO_2^- we find an electron affinity of 2.475 ± 0.010 eV for CH_2NO_2 (2.480 ± 0.010 eV for CD_2NO_2). Our results are consistent with a planar, C_{2v} anion with a large ($25,000$ cm^{-1}) torsion barrier, indicative of a C-N double bond. The planar geometry and high electron affinity suggest that structure **7** is the dominant resonance contribution in CH_2NO_2^- . The major geometry change on detachment of an electron from CH_2NO_2^- is the change of a C-N double bond to a single bond, evidenced by a lengthening of the C-N bond and a large drop in the torsion barrier (to 1250 cm^{-1}) in CH_2NO_2 . This supports Jacox's assignment⁹ of structure **1** with a C-N single bond as the ground (2B_1) state of CH_2NO_2 . The geometry change upon detachment determined

from the photoelectron spectrum is in excellent agreement with our *ab initio* prediction (Table 5-7). We have also observed the excited (2A_2) state 1.591 eV above the ground state. The 2A_2 state has structure **2** with a C-N double bond, giving it a geometry similar to that of CH_2NO_2^- .

1. R. D. Mead, A. E. Stevens and W. C. Lineberger in Gas Phase Ion Chemistry, Vol. 3 (Ions and Light), ed. M. T. Bowers, Academic, Orlando, (1984), pp. 213-248.
2. E. B. Wilson, Jr., J. C. Decius and P. C. Cross P. C., Molecular Vibrations, Dover, New York (1980).
3. K. K. Murray, T. M. Miller, D. G. Leopold and W. C. Lineberger, *J. Chem. Phys.* **84**, 2520 (1986); K. M. Christoffel, J. S. Bittman and J. M. Bowman, *Chem. Phys. Lett.* **133**, 525 (1987).
4. K. M. Ervin, J. Ho and W. C. Lineberger, *J. Chem. Phys.* **91**, 5974 (1989).
5. S. Moran, H. B. Ellis, Jr., D. J. DeFrees, A. D. McLean and G. B. Ellison, *J. Am. Chem. Soc.* **109**, 5996 (1987).
6. J. M. Oakes, L. B. Harding and G. B. Ellison, *J. Chem. Phys.* **83**, 5400 (1985); G. J. Vazquez, R. J. Buenker and S. D. Peyerimhoff, *J. Chem. Phys.* **90**, 7229 (1989).
7. T. L. Cottrell, T. E. Graham and T. J. Reid, *Trans. Faraday Soc.* **47**, 584 (1951); G. M. Nazin, G. B. Manelis and F. I. Dubovitskii, *Rus. Chem. Rev.* **37**, 603 (1968); C. G. Crawford and D. J. Waddington, *Trans. Faraday Soc.* **65**, 1334 (1969); F. I. Dubovitskii and B. L. Korsunskii, *Russ. Chem. Rev.* **50**, 958 (1981).
8. M. E. Jacox, *J. Phys. Chem.* **87**, 3126 (1983).
9. M. E. Jacox, *J. Phys. Chem.* **91**, 5038 (1987).

10. M. L. McKee, *J. Chem. Phys.* **81**, 3580 (1984).
11. M. L. McKee, *J. Am. Chem. Soc.* **107**, 1900 (1985).
12. A. Pross, D. J. DeFrees, B. A. Levi, S. K. Pollack, L. Radom, W. J. Hehre, *J. Org. Chem.* **46**, 1693 (1981).
13. S. Moran, H. B. Ellis, Jr., D. J. DeFrees, A. D. McLean, S. E. Paulson, G. B. Ellison, *J. Am. Chem. Soc.* **109**, 6004 (1987).
14. R. D. Mead, K. R. Lykke, W. C. Lineberger, J. Marks and J. I. Brauman, *J. Chem. Phys.* **81**, 4883 (1984).
15. Chapter 3; R. B. Metz, A. Weaver, S. E. Bradforth, T. N. Kitsopoulos and D. N. Neumark, *J. Phys. Chem.* **94**, 1377 (1990).
16. W. C. Wiley and I. H. McLaren, *Rev. Sci. Instrum.* **26**, 1150 (1955).
17. J. H. D. Eland, Photoelectron Spectroscopy, Butterworth and Co., London (1984).
18. A. D. Walsh, *J. Chem. Soc.* 2260 (1953).
19. M. J. Frisch, M. Head-Gordon, H. B. Schlegel, K. Raghavachari, J. S. Binkley, C. Gonzalez, D. J. DeFrees, D. J. Fox, R. A. Whiteside, R. Seeger, C. F. Melius, J. Baker, R. L. Martin, L. R. Kahn, J. J. P. Stewart, E. M. Fluder, S. Topiol and J. A. Pople, Gaussian Inc., Pittsburgh, PA.
20. B. Bigot, D. Roux and L. Salem, *J. Am. Chem. Soc.* **103**, 5271 (1981); J. P. Ritchie, *J. Am. Chem. Soc.* **107**, 1829 (1985).
21. K. E. Edgecombe and R. J. Boyd, *Can J. Chem.* **61**, 45 (1983).
22. J. Baker, R. H. Nobes and L. Radom, *J. Comp. Chem.* **7**, 349 (1986).

23. J. A. Pople, H. B. Schlegel, R. Krishnan, D. J. DeFrees, J. S. Binkley, M. J. Frisch, R. A. Whiteside, R. F. Hout, Jr. and W. J. Herhe, *Int. J. Quantum Chem., Quantum Chem. Symp.* **15**, 269 (1981); D. J. DeFrees and A. D. McLean, *J. Chem Phys.* **82**, 333 (1985).
24. J. M. Hollas, High Resolution Spectroscopy, Butterworth and Co., London (1982), p. 263.
25. G. Herzberg, Infrared and Raman Spectra, Van Nostrand Rheinhold Co., New York, (1945).
26. P. W. Atkins, Molecular Quantum Mechanics, Oxford University Press, Oxford, (1983), pp. 58-63.
27. G. Herzberg, Electronic Spectra of Polyatomic Molecules, Van Nostrand Reinhold Co., New York, (1966), p. 152.
28. K. R. Ervin, J. Ho and W. C. Lineberger, *J. Phys. Chem.* **92**, 5405 (1988).
29. A. M. Wodtke, E. J. Hintsa and Y. T. Lee, *J. Phys. Chem.* **90**, 3549 (1986).
30. A. M. Wodtke, E. J. Hintsa and Y. T. Lee, *J. Chem. Phys.* **84**, 1044 (1986).
31. M. J. S. Dewar, J. P. Ritchie and J. Alster, *J. Org. Chem.* **50**, 1031 (1985).
32. P. M. W. Gill and L. Radom, *Chem. Phys. Lett.* **132**, 16 (1986).

33. S. G. Lias, J. E. Bartmess, J. F. Liebman, J. L. Holmes, R. D. Levin and W. G. Mallard, *J. Phys. Chem. Ref. Data* **17**, 1 (Supp. 1) (1988).
34. J. E. Bartmess, J. A. Scott and R. T. McIver, Jr., *J. Am. Chem. Soc.* **101**, 6047 (1979); J. B. Cumming and P. Kebarle, *Can. J. Chem.* **56**, 1 (1978).

**Table 5-1a. Peak Positions^a in the 355 nm Photoelectron Spectrum of
CH₂NO₂⁻.**

peak	v ₃	v ₄	v ₆	eKE (eV)
a	-	-	-	1.07
A	0	0	0	1.020
B	0	0	2	0.969
C	0	1	0	0.902
D	1	0	0	0.858
E	1	0	2	0.808
F	1	0	4	0.774
G	1	1	0	0.738
H	2	0	0	0.696
I	2	0	2	0.646
J	2	0	4	0.607
K	2	1	0	0.579
L	3	0	0	0.537
M	3	0	2	0.485
N	4	0	0	0.375

^a The average uncertainty in peak positions is 0.007 eV.

Table 5-1b. Peak positions^a in the 355 nm photoelectron spectrum of CD₂NO₂⁻.

peak	v ₂	v ₄	v ₆	eKE (eV)
a	-	-	-	1.053
A	0	0	0	1.015
B	0	0	2	0.982
C	0	0	4	0.935
D	0	1	0	0.899
E	1	0	0	0.853
F	1	0	2	0.818
G	1	0	4	0.782
H	1	1	0	0.729
I	2	0	0	0.692
J	2	0	2	0.657
K	2	1	0	0.578
L	3	0	0	0.532
M	3	0	2	0.496
N	4	0	0	0.374

^a The average uncertainty in peak positions is 0.007 eV.

Table 5-2. Peak positions^a in the 266 nm photoelectron spectrum of CH_2NO_2^- .

peak	eKE (eV)	Assignment
A'	0.594	0°
B'	0.519	5^1
C'	0.478	(4^1)
D'	0.443	5^2
E'	0.415	3^1
F'	0.396	

^a The average uncertainty in peak positions is 0.007 eV.

Table 5-3. Vibrational frequencies of CH₂NO₂.

mode	symmetry	matrix frequency (cm ⁻¹)	description
1	a ₁	3055	CH ₂ sym. stretch
2	a ₁	1419	CH ₂ scissors
3	a ₁	1297	NO ₂ sym. stretch
4	a ₁	986	CN stretch
5	a ₁	693	NO ₂ scissors
6	a ₂	^a	torsion
7	b ₁	719	CNO ₂ out-of-plane
8	b ₁	606	H ₂ CN out-of-plane
9	b ₂	3200	CH ₂ antisym. stretch
10	b ₂	(1484, 1461) ^b	NO ₂ antisym. stretch
11	b ₂	1095	CH ₂ rock
12	b ₂	(484) ^c	NO ₂ rock

^a This mode is not infrared active and was not seen in the matrix studies.

^b In Fermi resonance with (ν₄ + ν₁₂).

^c Estimated from the position of (ν₄ + ν₁₂) peak.

All values are from the matrix studies of ref. 8 and 9.

Table 5-4. *Ab initio* Results on CH_2NO_2^- and CH_2NO_2 .

$\text{CH}_2\text{NO}_2^- (\tilde{X}, {}^1A_1)$							
R_{CH} (Å)	R_{CN} (Å)	R_{NO} (Å)	$\angle\text{HCN}$ (degrees)	$\angle\text{CNO}$ (degrees)	energy (+240 h)	level	ref
1.066	1.295	1.346	118.7	119.7	-1.66332	HF/3-21G	20
1.066	1.290	1.315	118.6	120.4	-2.69278	HF/4-31G	12
1.071	1.313	1.255	118.1	120.3	-3.05770	HF/6-31G*	21
1.069	1.308	1.258	117.8	120.4	-3.08915	HF/6-31**G**	*
1.074	1.347	1.292	117.4	120.1	-3.80446	MP2/6-31**G**	*
$\text{CH}_2\text{NO}_2 (\tilde{X}, {}^2B_1)$							
1.065	1.408	1.257	116.6	117.1	-1.62566	UHF/3-21G	11
1.077	1.470	1.291	116.3	116.5	-2.09675	UMP2/3-21G	11
1.067	1.418	1.198	116.5	117.1	-3.02981	UHF/6-31G	11
1.073	1.430	1.243	116.4	117.1	-3.72090	UMP2/6-31**G**	*
CH_2NO_2 (twisted, $\phi = 90^\circ, {}^2B_2$)							
1.068	1.449	1.191	116.5	117.0	-3.71262	UMP2/6-31**G**	*
$\text{CH}_2\text{NO}_2 ({}^2A_2)$							
1.072	1.318	1.313	117.2	123.7	-3.62282	UMP2/6-31**G**	*

* This work.

Table 5-5. *Ab initio* Vibrational Frequencies of CH_2NO_2^- at HF/6-31+G**.

mode	symmetry	frequency* (cm^{-1})	description
1	a_1	3019	CH_2 sym. stretch
2	a_1	1452	CH_2 scissors
3	a_1	1349	NO_2 sym. stretch
4	a_1	1014	CN stretch
5	a_1	646	NO_2 scissors
6	a_2	637	torsion
7	b_1	785	CNO_2 out-of-plane
8	b_1	535	H_2CN out-of-plane
9	b_2	3131	CH_2 antisym. stretch
10	b_2	1357	NO_2 antisym. stretch
11	b_2	1054	CH_2 rock
12	b_2	542	NO_2 rock

* The *ab initio* frequencies have been scaled by 0.9.

The description of the modes is only approximate and has been done by analogy with CH_2NO_2 .

Table 5-6. Torsion barriers for CH_2NO_2^- and CH_2NO_2 .

$\text{CH}_2\text{NO}_2^- (\bar{X}, {}^1A_1)$		
barrier (cm^{-1})	method	ref.
20,724	HF/6-31++G**//MP2/6-31++G**	*
16,620	MP2/6-31++G**	*
25,000	Fit to peak intensities	*
$\text{CH}_2\text{NO}_2 (\bar{X}, {}^2B_1)$		
3452	UHF/3-21G	*
2649	UMP2/3-21G	*
2833	UHF/6-31G//UHF/3-21G	10
1504	UMP2/6-31G//UHF/3-21G	10
2343	UMP3/6-31G//UHF/3-21G	10
1859	UMP2/6-31++G**	*
1817	UMP2/6-31++G** ^a	*
1250	Fit to peak positions	*
$\text{CH}_2\text{NO}_2 ({}^2A_2)$		
31,020	UHF/6-31G//UHF/3-21G	10
30,740	UMP2/6-31G//UHF/3-21G	10
30,110	UMP3/6-31G//UHF/3-21G	10

The torsion barrier was calculated as the difference in energy between the molecule at the optimized C_{2v} geometry with the torsion angle $\phi = 0^\circ$ and 90° .

The geometry was not reoptimized at $\phi = 90^\circ$, except for ^a and values from ref.

10.

Table 5-7. Geometry Change on CH_2NO_2^- Detachment.

R_{CH} (Å)	R_{CN} (Å)	R_{NO} (Å)	$\angle\text{HCN}$ (degrees)	$\angle\text{CNO}$ (degrees)	Method
-0.001	0.083	-0.049	-1.0	-3.0	<i>ab initio</i> MP2/6-31**G**
0.002	0.085	-0.047	-0.9	-3.0	experiment
	(0.009)	(0.006)	(0.1)	(0.2)	uncertainty

Table 5-8. Force constants for CH₂NO₂.

force constant	old value	uncertainty	new value	uncertainty
CH	5.34	0.01	5.34	0.01
CN	4.75	0.20	4.89	0.22
NO	8.07	0.32	8.51	0.32
HCH	0.28	0.01	0.27	0.01
HCN	0.70	0.01	0.70	0.01
ONO	1.79	0.18	1.89	0.22
CNO	1.07	0.14	0.96	0.04
NO, NO	2.24	0.39	1.64	0.42
NO, CN	1.15	0.29	1.37	0.26
NO, HCN	-0.10	0.06	-0.01	0.05
NO, ONO	0.29	0.17	0.20	0.16
NO, CNO	-0.28	0.26	0.33	0.32
CN, HCH	-0.27	0.02	-0.27	0.02
CN, ONO	-0.49	0.07	-0.68	0.09
HCN, CNO	0.09	0.02	0.08	0.03
HCN, ONO	-0.06	0.03	-0.03	0.04

Stretching and stretching interaction force constants are in 10^2 N/m; bending and bending interaction force constants are in 10^{-18} N·m; stretch-bending interaction force constants are in 10^{-8} N. Old values are from ref. 9 and use a reference geometry with $\angle\text{HCH} = 122^\circ$ and $\angle\text{ONO} = 122^\circ$. New values use our UMP2/6-31++G** reference geometry, with $\angle\text{HCH} = 127.2^\circ$ and $\angle\text{ONO} = 125.8^\circ$. The fit is insensitive to bond lengths. All values are fit to the 30 frequencies given in ref. 8.

Chapter 5 Chart Caption

Chart 5-1. Structures for CH_2NO_2^- and CH_2NO_2 . See text for a discussion of the various structures.

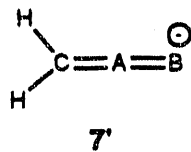
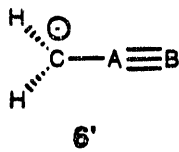
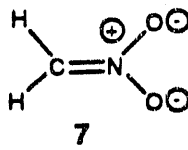
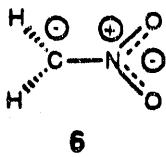
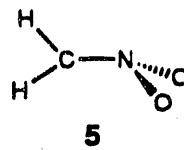
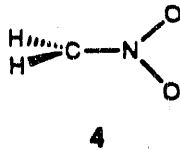
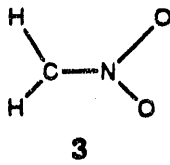
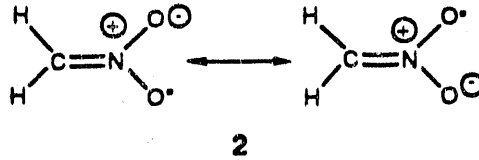
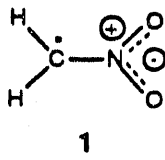


Chart 5-1

Chapter 5 Figure Captions

Figure 5-1. Experimental (—) and simulated (...) 355 nm photoelectron spectra of CH_2NO_2^- . The simulated spectrum has been convoluted with 5 meV Gaussians, for comparison with experiment. The experimental peaks are ~20 meV wide.

Figure 5-2. Experimental (—) and simulated (...) 355 nm photoelectron spectra of CD_2NO_2^- . The simulated spectrum has been convoluted with 5 meV Gaussians, for comparison with experiment. The experimental peaks are ~20 meV wide.

Figure 5-3. 266 nm photoelectron spectrum of CH_2NO_2^- .

Figure 5-4. Form of Q_3 (NO_2 symmetric stretch) and Q_4 (CN stretch) normal modes of the CH_2NO_2 radical.

Figure 5-5. Highest occupied molecular orbitals of the CH_2NO_2^- anion. The $2b_1$ orbital (left) is C-N π bonding and O-O π bonding. The $1a_2$ orbital (right) is weakly O-O π anti-bonding. Removal of an electron from the $2b_1$ orbital of the anion forms the 2B_1 ground state of the CH_2NO_2 radical; removal of an electron from the $1a_2$ orbital forms the low-lying 2A_2 excited state of CH_2NO_2 .

Figure 5-6. Torsion potentials and energy levels for CH_2NO_2 (top) and CH_2NO_2^- (bottom). The torsion angle ϕ is the angle between the NO_2 and CH_2 planes. It is 0° for the planar configuration, 90° at the twisted geometry, and 180° at the second (equivalent) planar configuration.

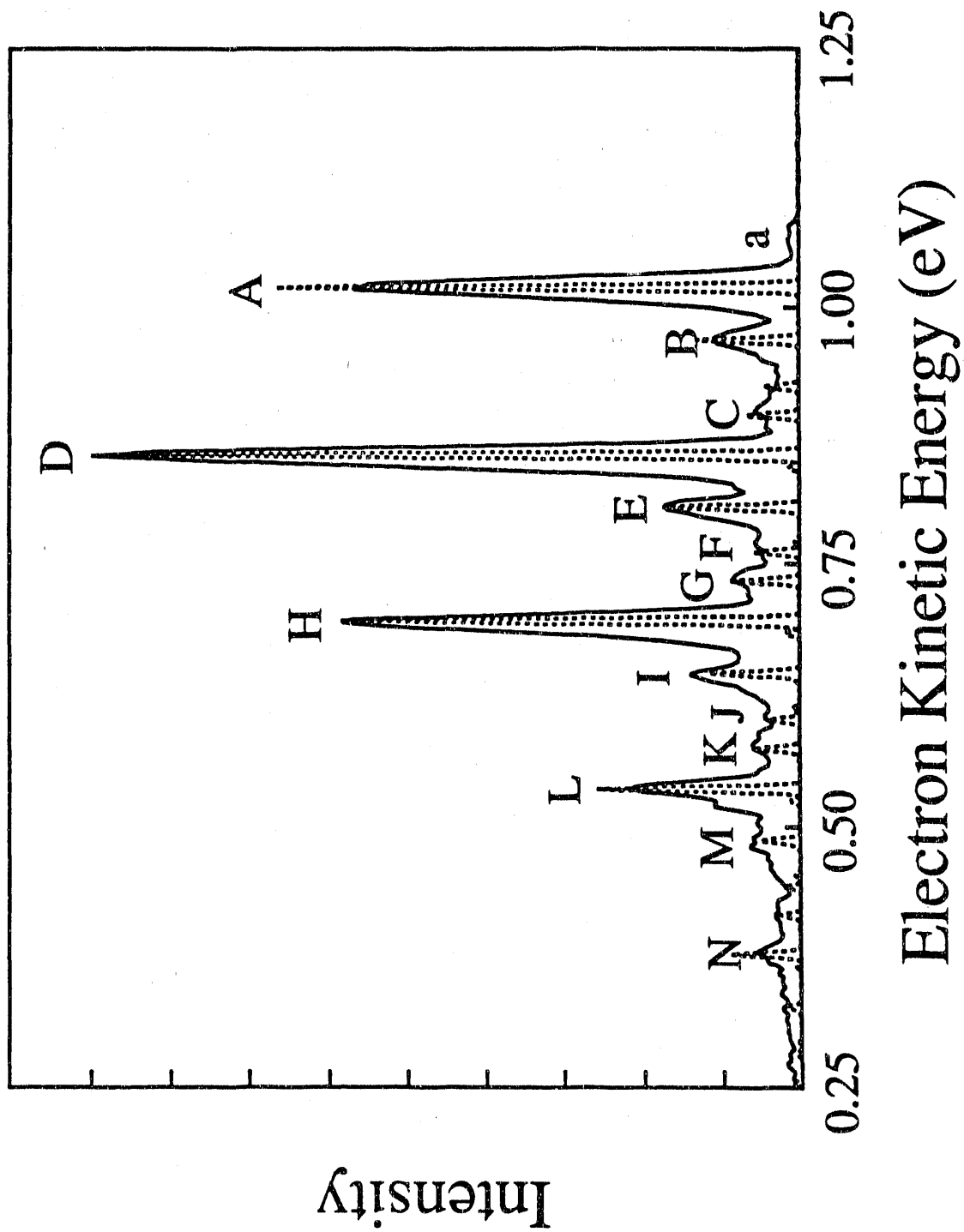


Figure 5-1

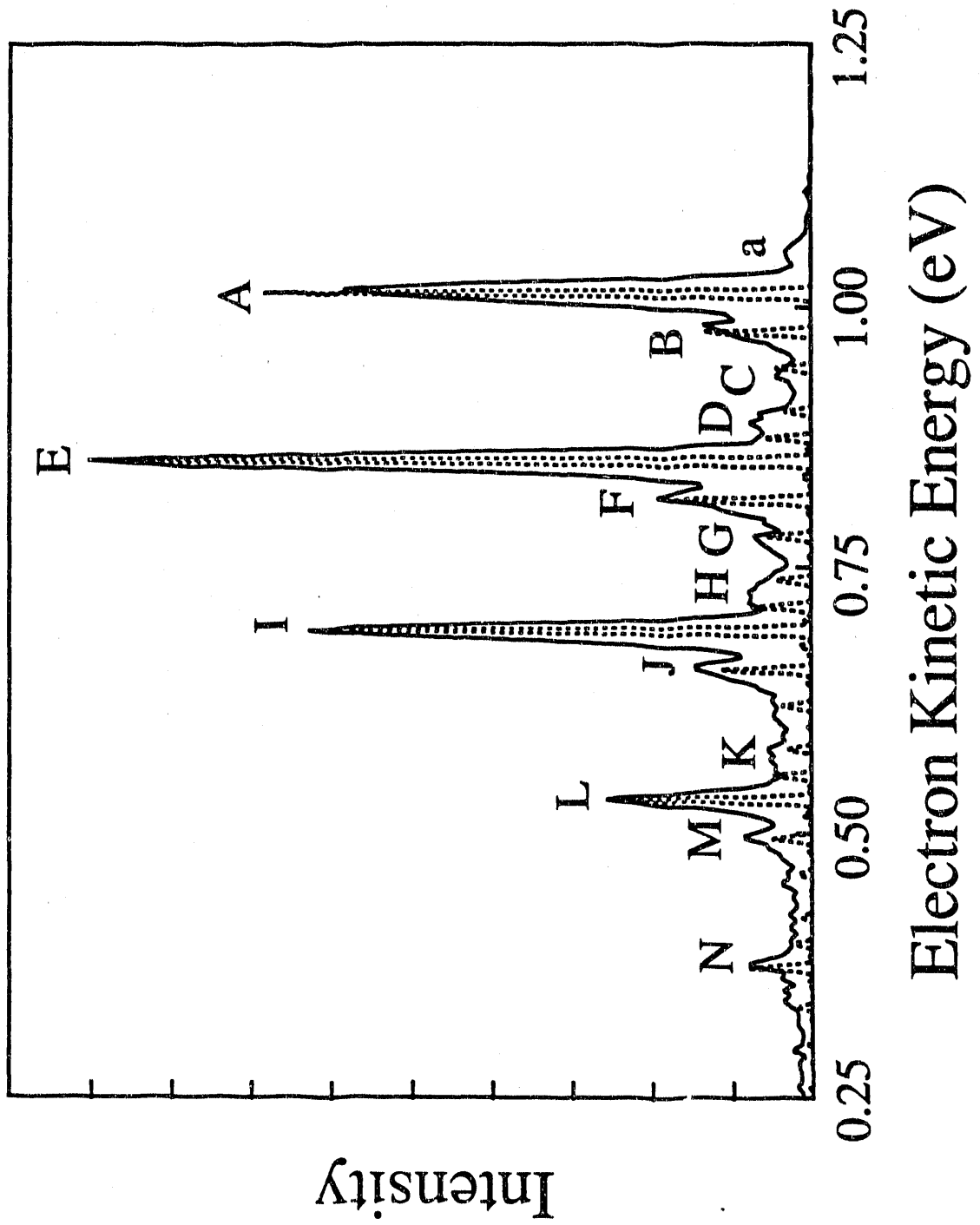


Figure 5-2

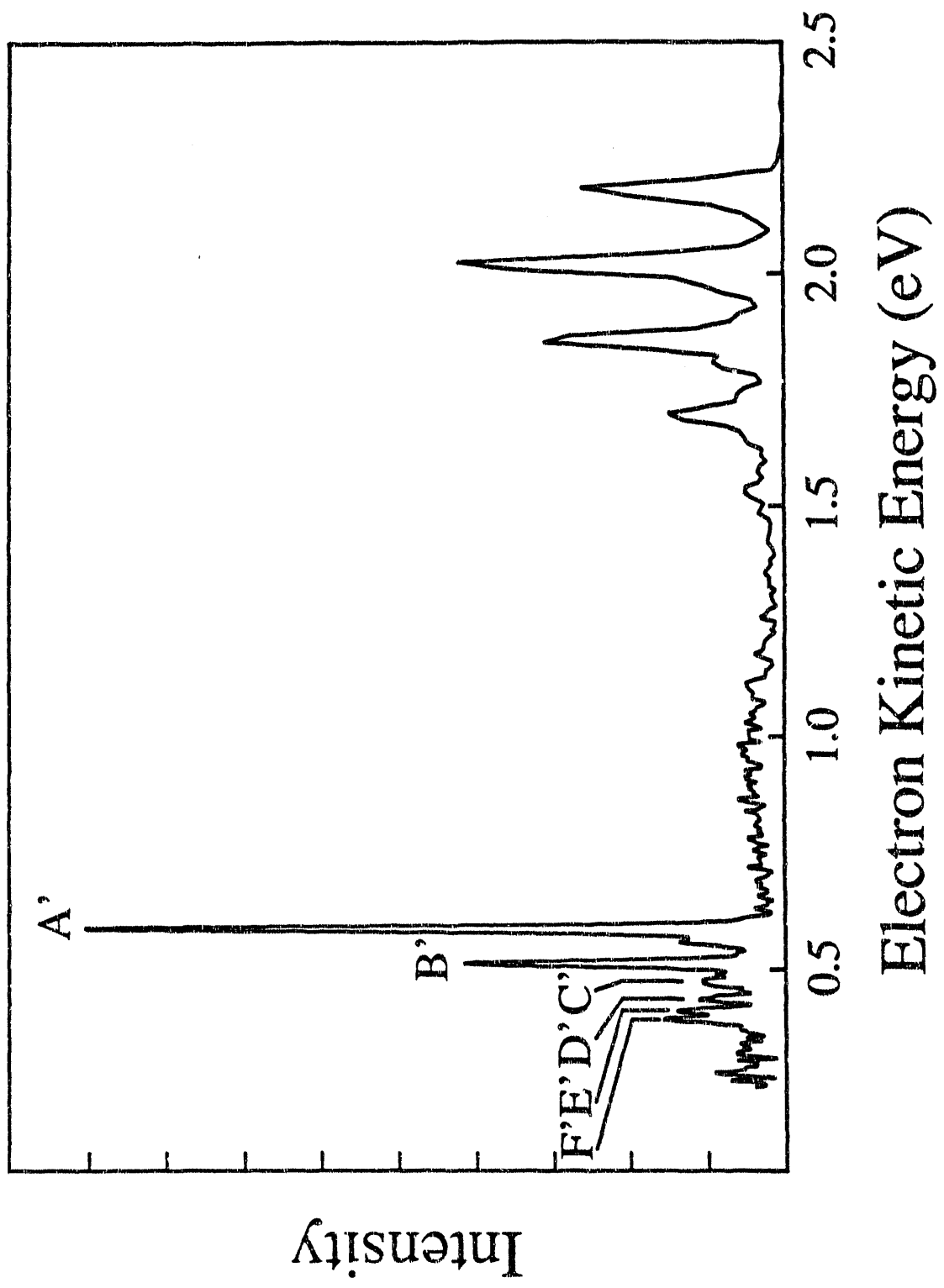
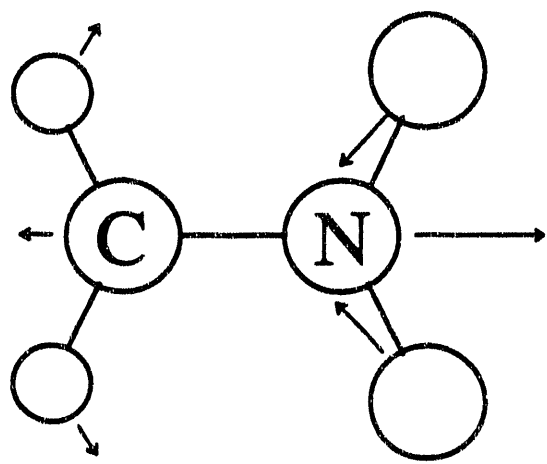
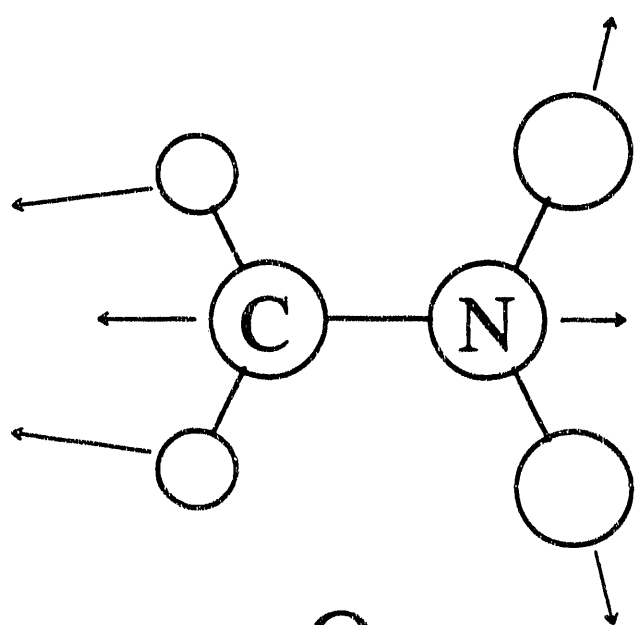


Figure 5-3

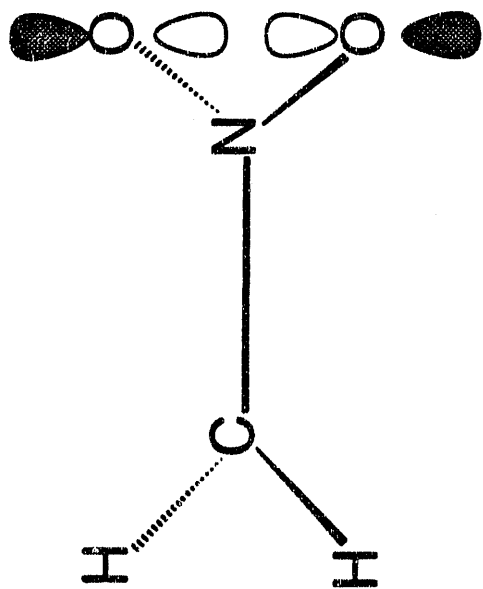


Q₃

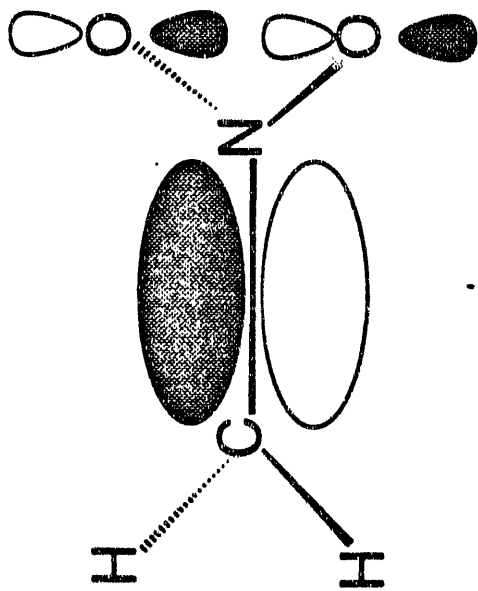


Q₄

Figure 5-4



1a₂



2b₁

Figure 5-5

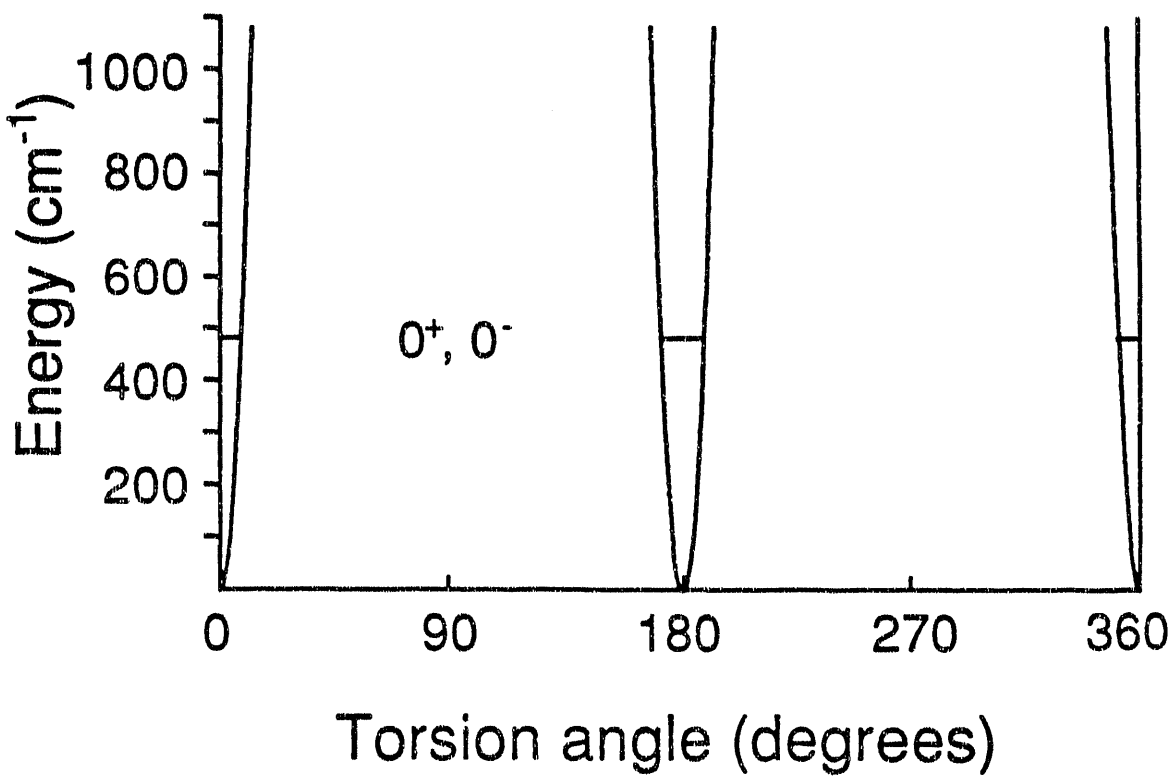
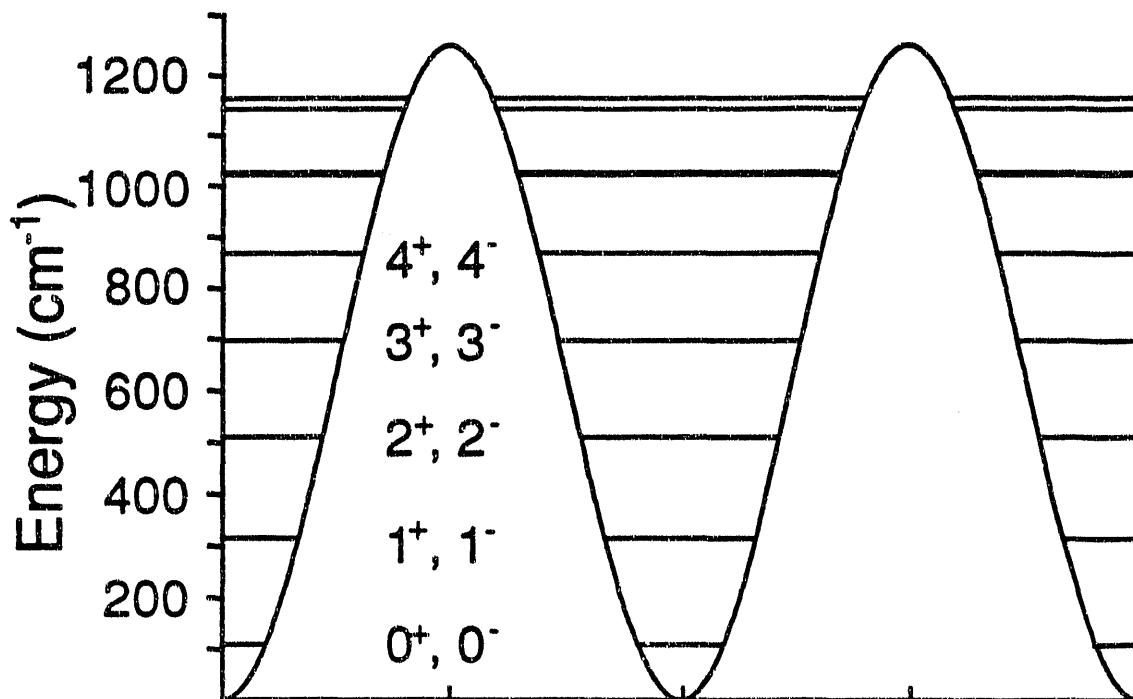


Figure 5-6

Chapter 6. Photodissociation of CH_2NO_2 at 208, 235 and 270 nm

1. Introduction

Studies of excited electronic states of polyatomic free radicals provide a challenging and exciting field for both experiment and theory. Due to their open-shelled nature, free radicals have many low-lying excited electronic states. Free radicals often have weak chemical bonds, so even low-lying electronic states may be dissociative. The challenge to the theorist lies in describing this multitude of states, which can rarely be well described by a single electronic configuration. On the experimental side, while dissociative states of closed-shell molecules have been mapped out using various photodissociation techniques, studies of the dissociative electronic states of polyatomic free radicals are rare. The experimental challenge lies in preparing a pure source of the radical in a well-defined quantum state so as to avoid contamination of the experimental results by other species or vibrationally excited radicals. Only then is it reasonable to excite the radical with a well-defined amount of energy, and measure asymptotic properties of the fragments.

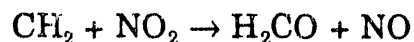
We have developed an apparatus specifically designed to study dissociative excited electronic states of free radicals and have used it to study predissociation of N_3 ¹ and NCO .² This paper describes the extension

of our studies to a larger system, the nitromethyl free radical CH_2NO_2 . The nitromethyl radical is thought to be important in the combustion of nitromethane, a widely used propellant via³



Low-lying electronic states of CH_2NO_2 have been studied theoretically by McKee,^{4,5} while the ground state of the radical has been experimentally studied in a matrix.⁶ The use of isotopic labelling allowed the derivation of a force constant matrix for CH_2NO_2 .^{7,8} Recently, the photoelectron spectrum of CH_2NO_2^- was obtained and used to determine the torsion barrier in the radical.⁸ The photoelectron study also identified an excited $^2\text{A}_2$ electronic state 1.591 eV above the $^2\text{B}_1$ ground state.

In the matrix study it was also discovered that the threshold for CH_2NO_2 photodissociation lies between 300 and 280 nm and H_2CO was detected subsequent to photodissociation.⁶ The lowest energy products are $\text{H}_2\text{CO} + \text{NO}$, which lie 1.50 eV *below* the CH_2NO_2 ground state. However, photodissociation to form H_2CO would require substantial rearrangement. It is also energetically possible to form $\text{CH}_2 + \text{NO}_2$, which lie 3.07 eV *above* the CH_2NO_2 ground state (the energy available at 280 nm is 4.4 eV). The secondary reaction



may explain the observation of H_2CO in the matrix studies.⁶ Thus, the primary photodissociation products could not be determined. The nature of

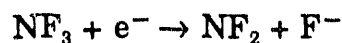
the primary dissociation fragments is of some interest as the closely-related nitromethane molecule CH_3NO_2 primarily forms $\text{CH}_3 + \text{NO}_2$ upon photodissociation using a single UV photon,^{9,10} but rearranges and forms $\text{CH}_3\text{O} + \text{NO}$ upon IR multi-photon photodissociation.^{11,12} Our analysis of the photodissociation of the nitromethyl free radical will draw heavily on comparisons with the photodissociation of nitromethane, which has been widely studied. The questions that arise in the photodissociation of the nitromethyl free radical are the nature of the primary photodissociation products, and how the products and photodissociation cross section vary as a function of wavelength.

2. Experimental

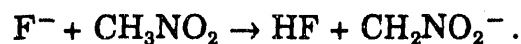
The fast radical beam spectrometer used in our experiments has been described in detail elsewhere,¹ so only a brief description will be given here. Basically, anions corresponding to the radical of interest are created, accelerated to 8 keV and mass-selected, and detached to form the neutral free radicals. The radicals are then photodissociated and fragments are detected.

To study CH_2NO_2 we first produce the nitromethide anion CH_2NO_2^- . Neat NF_3 at 10 psig backing pressure is bubbled through CH_3NO_2 and expanded through a pulsed molecular beam. The supersonic expansion is

crossed by a 1 keV electron beam. Dissociative attachment of slow secondary electrons to NF_3 forms F^-



and nitromethide anions are produced by proton abstraction



The beam is then skimmed, accelerated to 8 kV, and injected into a beam-modulated time-of-flight mass spectrometer.^{13,14} The mass spectrometer uses a 1 mm dia. defining aperture. A pulsed, excimer-pumped tunable dye laser is used to detach the cold nitromethide anion near threshold, producing cold nitromethyl radicals. The detachment wavelength is chosen based on the photoelectron spectrum of CH_2NO_2^- which we have previously determined⁸ (Chapter 5). We detach at 479 nm (2.588 eV), which is above the electron affinity (2.475 eV), and below the strong $v_3' = 1 \leftarrow v_3'' = 0$ transition (2.637 eV). So long as the anions are cold, this ensures that only vibrationally cold CH_2NO_2 radicals are made. Any remaining negative ions are deflected out of the beam, producing a fast beam of cold, pure nitromethyl free radicals.

The radicals pass through a second 1 mm defining aperture and are photodissociated by a second excimer-pumped tunable dye laser. Coumarin 540A dye is used to generate light at 540 nm and doubled in BBO to produce 1.2 mJ of light at 270 nm. Coumarin 480 (470 nm) is doubled to produce 2.4 mJ/pulse at 235 nm, and Exalite 416 dye (416 nm) is doubled to

produce 0.6 mJ/pulse at 208 nm. Fragments are detected at the end of a 1 m flight tube with a 40 mm dual microchannel plate detector.

Undissociated radicals are prevented from hitting the detector by a 3 mm beam block placed across the face of the detector, along the beam axis.

Dissociation fragments will typically have some velocity component perpendicular to the beam axis (except for the small number that dissociate exactly parallel to the radical beam axis); they will miss the beam block and impinge on the detector.

We wish to determine the kinetic energy release of the fragments, as this will give us information on the nature of the photodissociation products. This is done by sending the detector signal into a transient digitizer and measuring the arrival time distribution of the photofragments. These time-of-flight (TOF) spectra are particularly sensitive to the photofragment velocities parallel and antiparallel to the beam direction and are discussed in more detail in the next section. Time-of-flight spectra are also obtained without the dissociation laser to determine the background signal due to incomplete blocking of the undissociated radical beam. The background signal is about two orders of magnitude smaller than the fragment signal and is not sharply peaked in time.

3. Results

3.1 Time-of-Flight Spectra

Time-of-flight spectra are averaged over 100,000 laser shots at 270 and 235 nm and over 150,000 laser shots at 208 nm. The signal due to dissociated radicals is much higher at 235 and 208 nm than at 270 nm. While an accurate photodissociation cross section could not be determined (as the overlap of the radical and laser beams, as well as the number of radicals, varied at each wavelength), the cross section at 235 nm is at least a factor of 2 larger than at 270 nm, and the cross section at 208 nm appears to be at least a factor of 5 larger than at 270 nm.

Fragment time-of-flight spectra recorded at 270, 235 and 208 nm are shown in Fig. 6-1, 6-2 and 6-3, respectively. The maximum extent of the TOF spectrum gives the maximum kinetic energy release in the photodissociation.

3.2 Primary Photodissociation Products

If we ignore spatial and energy spread in the undissociated radical beam, simple kinematics gives the difference in arrival times between the fastest and slowest fragments

$$\Delta\tau = \frac{2l}{v_b^2} \sqrt{\frac{m_2}{m_1} \frac{2 KE}{m_1 + m_2}} \quad (1)$$

where l is the distance the fragments travel to the detector (1 m), v_b is the beam velocity (1.6×10^5 m/s), KE is the maximum kinetic energy release, and m_1 and m_2 are the masses of the light and heavy fragments, respectively. Using these values and converting to more convenient units gives

$$\Delta\tau = 1040 \sqrt{\frac{m_2}{m_1} \frac{KE}{m_1 + m_2}} \quad (2)$$

where $\Delta\tau$ is in ns, masses are in amu and KE is in eV. For CH_2NO_2 , we have

$$KE = 5.55 \times 10^{-5} \frac{m_1}{m_2} (\Delta\tau)^2 \quad (3)$$

The spread $\Delta\tau$ is about 210 ns at 270 nm, 300 ns at 235 nm, and 350 ns at 208 nm.

While we do not measure the photofragment masses directly, we can infer what the primary products are from the maximum kinetic energy release obtained from eq. 3. These differ because $m_1/m_2 = 0.30$ for the $\text{CH}_2 + \text{NO}_2$ channel and $m_1/m_2 = 1.00$ for the $\text{H}_2\text{CO} + \text{NO}$ channel: values are given in Table 6-1. If $\text{CH}_2 + \text{NO}_2$ is the primary photodissociation channel, the measured $\Delta\tau$ values correspond to $\text{CH}_2 + \text{NO}_2$ minimal internal energies

of 0.83 eV (270 nm), 0.80 eV (235 nm), and 0.97 eV (208 nm). These values indicate that no ground state $\text{CH}_2 + \text{NO}_2$ products are produced. As discussed below, we believe these results are consistent with the production of electronically excited NO_2 at all three wavelengths.

On the other hand, if the primary products are $\text{H}_2\text{CO} + \text{NO}$, then the measured $\Delta\tau$ values correspond to minimum fragment internal energies of 3.82 eV (270 nm), 2.16 eV (235 nm), and 1.17 eV (208 nm). Two aspects of these values are counterintuitive. The fragment internal energy *drops* as the dissociation energy is *increased*. Also, when the dissociation photon energy is increased by 1.37 eV from 4.59 eV (270 nm) to 5.96 eV (208 nm), the maximum kinetic energy release increases by 4 eV. From these considerations, while we cannot entirely rule out formation of $\text{H}_2\text{CO} + \text{NO}$, we believe the TOF distributions are from $\text{CH}_2 + \text{NO}_2$ products only.

4. Analysis and Discussion

4.1 Simulation of the 270 nm Time-of-Flight Spectrum

A more accurate estimate of the kinetic energy release at each photodissociation wavelength can be obtained by simulating the experimental TOF spectrum. We use a forward convolution approach - the TOF spectrum is simulated for a particular kinetic energy distribution and anisotropy β . The kinetic energy release and anisotropy are then modified

to obtain improved agreement. The TOF spectrum is simulated using a Monte Carlo approach that correctly treats the spatial overlap of the radical and laser beams, the small spread in the beam energy caused due to the mass spectrometer, the anisotropy in the dissociation, and the finite size of the beam block.

The 270 nm TOF spectrum (Fig. 6-1) shows distinct wings at fast and slow arrival time due to the light CH_2 fragment. This is what would be expected of a kinetic energy release spectrum sharply peaked at a single energy, so we have used the Monte Carlo approach to fit the TOF spectrum, assuming a single value for the kinetic energy release. The kinetic energy release and anisotropy β are varied to obtain the fit shown with a dotted line in Fig. 6-1. The extent of the wings (to high and low TOF) is sensitive to the kinetic energy release, while how well-resolved the wings are and the height of the wings relative to the rest of the TOF spectrum is sensitive to β .

The best fit is obtained with 0.64 ± 0.05 eV kinetic energy release and $\beta = 0.2$. This means that 3.95 ± 0.05 eV goes into breaking the C-N bond and internal excitation of the products. The heat of formation of CH_2NO_2 is 127 ± 12 kJ/mol (1.32 ± 0.12 eV),⁸ based on the gas phase acidity of nitromethane¹⁵ (the primary source of uncertainty), the electron affinity of CH_2NO_2 ⁸ and the heat of formation of nitromethane,¹⁶ so the C-N bond strength in CH_2NO_2 is 3.07 ± 0.12 eV. Thus, the CH_2 and NO_2 fragments

are formed with 0.88 ± 0.13 eV of internal energy. The wings in the TOF spectrum are clearly resolved, so the fragments are formed with a specific internal excitation, not with a broad range of internal energies centered about 0.88 eV. It is unlikely that vibrational or rotational excitation of the fragments would lead to such a narrow distribution of internal energies, so the internal energy likely goes to form electronically excited products.

Of all the possible electronic channels for the $\text{CH}_2 + \text{NO}_2$ products, the closest to 0.88 ± 0.13 eV of electronic excitation is $\text{CH}_2(\tilde{X}, {}^3\text{B}_1) + \text{NO}_2(\tilde{\text{A}}, {}^2\text{B}_2)$; the recent high-level *ab initio* study of Schaefer¹⁷ predicts $T_0 = 0.98$ for $\text{NO}_2(\tilde{\text{A}})$. This agrees with the recent CCSD calculation of Kaldor which gives $T_0 = 1.00$ eV.¹⁸

These *ab initio* values are higher than the experimental value $T_0 \approx 1.2$ eV obtained from $\tilde{\text{A}}-\tilde{X}$ absorption¹⁹ and emission,²⁰ and also from the photoelectron spectrum of NO_2^- .²¹ However, the experimental values are only upper limits - the spectra from which they are obtained show extended progressions in the ν_2 NO_2 bending mode, so it is difficult to assign (or even observe) the origin of this progression. Thus, we believe our translational energy distributions indicate that $\text{CH}_2\text{NO}_2(\tilde{X}, {}^2\text{B}_1)$ dissociates to form $\text{NO}_2(\tilde{\text{A}}, {}^2\text{B}_2)$.

This discussion assumes that the CH_2 product is formed in its $\tilde{X}({}^3\text{B}_1)$ ground state with little or no vibrational excitation. This is because the primary excitation in the photodissociation occurs in the NO_2 group (see

below), and CH_2NO_2 likely dissociates before this initial excitation can be transferred to vibrations of the CH_2 group. Also, from a Franck-Condon standpoint, the difference in geometry between CH_2 (\tilde{X}) ($R_{\text{CH}} = 1.07 \text{ \AA}$, $\alpha = 133.8^\circ$ ²²) and the CH_2 group in CH_2NO_2 ($R_{\text{CH}} = 1.07 \text{ \AA}$, $\alpha = 127.2^\circ$ ⁸) is quite small, so a simple Franck-Condon picture predicts 80% of the CH_2 is formed with $v = 0$. Note that production of ground-state triplet CH_2 from the doublet nitromethyl radical is spin allowed.

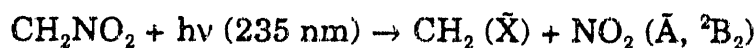
4.2 Simulation of the 235 and 208 nm Time-of-Flight Spectra

We have also simulated the 235 and 208 nm TOF spectra (Fig. 6-2 and 6-3, respectively). At 235 nm, the maximum and minimum extents of the TOF spectrum (which are most sensitive to the maximum kinetic energy release) are fit well with approximately 3.70 eV energy going to break the bond and internal energy of fragments. This is approximately the same value seen at 270 nm, though it is much less precise than the value obtained at 270 nm, due to the increase in available energy. The 235 nm spectrum does not show the sharp wings at the edges of the TOF spectrum that were seen at 270 nm. Thus, it appears that there is a broad kinetic energy release distribution. We model this distribution using a prior distribution,²³ which statistically partitions the excess energy between translational energy and the various internal degrees of freedom of the

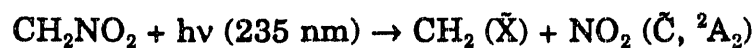
products and vary the anisotropy to obtain a good fit. In the simplest case, the probability of forming products with a fraction f_T of the excess energy in translation is given by

$$P^o(f_T) \propto f_T^{1/2} (1 - f_T)^\alpha \quad (4)$$

where $\alpha = s + r/2 - 1$, s is the number of vibrational degrees of freedom and r is the number of rotations. The 235 nm spectrum also has wings at 0.28 eV kinetic energy release (peaks A and B in Fig. 6-2), suggesting some of the fragments are formed with 1.93 ± 0.13 eV internal excitation. A photoelectron study of NO_2^- found $T_0(\text{NO}_2 \tilde{\text{C}}, {}^2\text{A}_2) = 2.028 \pm 0.009$ eV,²¹ so it seems likely that this feature is due to photodissociation to $\text{CH}_2 + \text{NO}_2(\tilde{\text{C}})$. At 235 nm, the best fits are obtained with 50% of the NO_2 product formed in the $\tilde{\text{A}}$ state, $\beta = 0.4$ and two active vibrational modes (which could be the two symmetric vibrational modes of NO_2) and no active rotational degrees of freedom in the prior distribution, and the remaining 50% of the NO_2 formed in the $\tilde{\text{C}}$ state, vibrationally cold, with $\beta = 0.4$. If more active vibrations or rotations are included in the prior distribution, the wings in simulations of the TOF spectrum are significantly less intense than we observe experimentally. While this fit is not unique, it does give some idea of the degree of vibrational excitation seen in the $\text{NO}_2(\tilde{\text{A}})$ product. Thus, at 235 nm, both



and



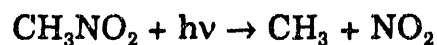
appear to be significant channels.

The TOF distribution at 208 nm shows fewer distinct features than the 270 and 235 nm spectra. We have fit the TOF spectrum assuming 3.95 eV goes to bond breaking and internal excitation of the products and a prior distribution with two active vibrational modes and no active rotational modes. The best fit is obtained with $\beta = 0.3$ and is shown with a dotted line in Fig. 6-3. Photodissociation at 208 nm may also form $\text{NO}_2(\tilde{\text{C}})$, though this cannot be determined from the TOF spectrum. We will now examine why photodissociation of CH_2NO_2 produces electronically excited NO_2 by comparing our results to previous results on the related CH_3NO_2 molecule.

4.3 Comparison with CH_3NO_2

As no previous studies of dissociative states of the nitromethyl free radical exist, and *ab initio* information on these states is lacking, the analysis relies on similarities between the nitromethyl free radical and nitromethane, which has been well studied.²⁴ This is reasonable, as the photodissociation of nitromethane involves exciting electrons based on the NO_2 group,^{25,26} so CH_2NO_2 and CH_3NO_2 should be quite similar in this respect. The photodissociation dynamics of nitromethane will now be reviewed.

The C-N bond in nitromethane is actually slightly weaker than in the nitromethyl radical, 2.60 eV *vs* 3.07 eV.²⁷ Nitromethane exhibits two distinct ultraviolet absorption bands. There is a weak band peaking at 280 nm with a $3.4 \times 10^{-20} \text{ cm}^2$ cross section. The cross section then reaches a minimum of $1 \times 10^{-20} \text{ cm}^2$ near 245 nm and rises to $1.9 \times 10^{-17} \text{ cm}^2$ at 197 nm.²⁸ Absorption in these bands primarily results in the photodissociation²⁹



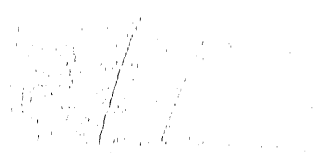
These transitions are due to promoting an electron between two orbitals based on the NO_2 group. The weak, long-wavelength band is due to an $n \rightarrow \pi^*$ transition from a nonbonding oxygen orbital to an N-O π^* orbital.^{25,28,30} This transition is not optically allowed in the local C_{2v} symmetry of the NO_2 group,²⁵ but is weakly allowed in nitromethane, as nitromethane does not have C_{2v} symmetry (due to the hydrogen atoms). This is thought to overlap a weak $\sigma \rightarrow \pi^*$ transition.^{25,31} The intense peak at 197 nm is due to a $\pi \rightarrow \pi^*$ transition from the oxygen lone-pair orbitals to the N-O π anti-bonding orbital.^{25,28,30} This involves a shift in electron density from the oxygen atoms to the nitrogen atom, resulting in a large transition dipole moment and an intense absorption peak. All of these transitions involve exciting an electron to an N-O π^* orbital, so the NO_2 product is likely formed in an excited electronic state. This has been observed by Schoen et al. who observe some direct emission from electronically excited NO_2

product in the 264 nm photodissociation of nitromethane, and more product emission at 237 nm.²⁴ Mijoule et. al. suggest that the even the nonemissive NO₂ product may be formed in the electronically excited "dark" \tilde{C} (²A₂) state.²⁵ Also, in the 193 nm photodissociation of nitromethane, fluorescence detected from the (\tilde{A}) ²B₂ state of NO₂ indicates that most of the NO₂ is formed in this state, while some is formed in higher-lying electronic states.^{27,32} Thus, photodissociation of nitromethyl free radical to form electronically excited NO₂ is consistent with what has been observed in nitromethane.

5. Conclusions

The nitromethyl free radical photodissociates at 270, 235 and 208 nm to produce primarily CH₂ + NO₂. The cross section increases by at least a factor of 5 from 270 to 208 nm. At 270 nm, the measured time-of-flight spectrum is consistent with a kinetic energy release of 0.64 ± 0.05 eV, so the photodissociation products are formed with 0.88 ± 0.13 eV internal excitation. Based on the close proximity of this value to *ab initio* values of T₀ (NO₂, \tilde{A}), we assign the primary photodissociation products as CH₂ (\tilde{X} , ³B₁) and NO₂ (\tilde{A} , ²B₂), with little vibrational excitation. At 235 nm, there appears to be much more product vibrational excitation (most likely in the NO₂ (\tilde{A}) fragment), and the spectrum has been fit using a prior distribution

with two active vibrational modes for the products. The 235 nm spectrum shows evidence of some production of NO_2 ($\dot{\text{C}}, {}^2\text{A}_2$).



1. R. E. Continetti, D. R. Cyr, R. B. Metz and D. M. Neumark, *Chem. Phys. Lett* **182**, 406 (1991).
2. D. R. Cyr, R. B. Metz, R. E. Continetti and D. M. Neumark, to be published.
3. T. L. Cottrell, T. E. Graham and T. J. Reid, *Trans. Faraday Soc.* **47**, 584 (1951); G. M. Nazin, G. B. Manelis and F. I. Dubovitskii, *Russ. Chem. Rev.* **37**, 603 (1968); C. G. Crawford and D. J. Waddington, *Trans. Faraday Soc.* **65**, 1334 (1969); F. I. Dubovitskii and B. L. Korsunskii, *Russ. Chem. Rev.* **50**, 958 (1981).
4. M. L. McKee, *J. Chem. Phys.* **81**, 3580 (1984).
5. M. L. McKee, *J. Am. Chem. Soc.* **107**, 1900 (1985).
6. M. E. Jacox, *J. Phys. Chem.* **87**, 3126 (1983).
7. M. E. Jacox, *J. Phys. Chem.* **91**, 5038 (1987).
8. R. B. Metz, D. R. Cyr and D. M. Neumark, *J. Phys. Chem.* **95**, 2900 (1991).
9. K. Honda, H. Mikuni and M. Takahashi, *Bull. Chem. Soc. Jpn.* **45**, 3534 (1972).
10. I. M. Napier and R. G. W. Norrish, *Proc. R. Soc. London Ser. A* **299**, 317 (1967).
11. A. M. Wodtke, E. Hintscha and Y. T. Lee, *J. Chem. Phys.* **84**, 1044 (1986).
12. A. M. Wodtke, E. Hintscha and Y. T. Lee, *J. Phys. Chem.* **90**, 3549 (1986).

13. J. M. B. Bakker, *J. Phys. E* **6**, 785 (1973); **7**, 364 (1974).
14. T. N. Kitsopoulos, I. M. Waller, J. G. Loeser and D. M. Neumark, *Chem. Phys. Lett.* **159**, 300 (1989).
15. J. E. Bartmess, J. A. Scott and R. T. McIver, Jr., *J. Am. Chem. Soc.* **101**, 6047 (1979); J. B. Cumming and P. Kebarle, *Can. J. Chem.* **56**, 1 (1978).
16. S. G. Lias, J. E. Bartmess, J. F. Liebman, J. L. Holmes, R. D. Levin and W. G. Mallard, *J. Phys. Chem. Ref. Data* **17**, 1 (Suppl. 1) (1988).
17. C. P. Blahous, B. F. Yates, Y. Xie and H. F. Schaefer, *J. Chem. Phys.* **93**, 8105 (1990).
18. U. Kaldor, *Chem. Phys. Lett.* **185**, 131 (1991).
19. A. J. Merer and K.-E. J. Hallin, *Can. J. Phys.* **56**, 838 (1978).
20. A. Delon and R. Jost, *J. Chem. Phys.* **95**, 5686 (1991); A. Delon, R. Jost and M. Lombardi, *J. Chem. Phys.* **95**, 5701 (1991).
21. A. Weaver, R. B. Metz, S. E. Bradforth and D. M. Neumark, *J. Chem. Phys.* **90**, 2070 (1989).
22. P. R. Bunker and P. Jensen, *J. Chem. Phys.* **79**, 1224 (1983).
23. R. D. Levine in Theory of Chemical Reaction Dynamics, Vol. IV, M. Baer, Ed., CRC Press, Boca Raton, Fl, (1984), p. 31.
24. P. E. Schoen, M. J. Marrone, J. M. Schnur and L. S. Goldberg, *NRL Memorial Report 4755*, p. 3 (1981); *Chem. Phys. Lett.* **90**, 272 (1982).
25. G. Mijoule, S. Odiod, S. Fliszar and J. M. Schnur, *J. Mol. Struct.* **149**, 311 (1987).
26. S. Roszak and J. J. Kaufman, *J. Chem. Phys.* **94**, 6030 (1991).

27. L. J. Butler, D. Krajnovich, Y. T. Lee, G. Ondrey and R. Bersohn, J. Chem. Phys. **79**, 1708 (1983).
28. W. D. Taylor, T. D. Allston, M. J. Moscato, G. B. Fazekas, R. Kozlowski and G. A. Takacs, Int. J. Chem. Kinetics **12**, 231 (1980).
29. N. C. Blais, J. Chem. Phys. **79**, 1723 (1983).
30. S. Nagakura, Mol. Phys. **3**, 152 (1960).
31. W. M. Flicker, O. A. Musher and A. Kuppermann, J. Chem. Phys. **72**, 2788 (1980).
32. K. Q. Lao, E. Jensen, P. W. Kash and L. J. Butler, J. Chem. Phys. **93**, 3958 (1990).

Table 6-1: Simple Kinematic Analysis of the TOF Spectrum.

λ (nm)	$h\nu$ (eV)	$\Delta\tau$ (ns)	$\text{CH}_2\text{O} + \text{NO}$		$\text{CH}_2 + \text{NO}_2$	
			max. KE (eV)	min. int. E (eV)	max. KE (eV)	min. int. E (eV)
270	4.59	210	2.27	3.82	0.69	0.83
235	5.28	300	4.62	2.16	1.41	0.80
208	5.96	350	6.29	1.17	1.92	0.97

Chapter 6 Figure Captions

Figure 6-1. Time-of-flight spectrum of CH_2NO_2 photodissociation fragments at 270 nm (4.59 eV) (—) and simulation assuming all but 3.95 of the energy goes to translational energy release with $\beta = 0.2$ (....).

Figure 6-2. Time-of-flight spectrum at 235 nm (5.28 eV) (—) and simulation (....) assuming 50% of the NO_2 products are formed with a translation energy release determined by a prior distribution and a maximum kinetic energy release of 1.58 eV and $\beta = 0.4$ and the remaining 50% of the NO_2 product is formed with 0.28 eV kinetic energy release with $\beta = 0.4$.

Figure 6-3. Time-of-flight spectrum at 208 nm (5.96 eV) (—) and simulation (....) assuming the products are formed with a prior distribution and a maximum kinetic energy release of 2.01 eV and $\beta = 0.3$.

Figure 6-4. Energetics for the photodissociation of CH_2NO_2 and some possible products. The *ab initio* value of Schaefer is used for T_0 ($\text{NO}_2 \tilde{\text{A}}$) (see text).

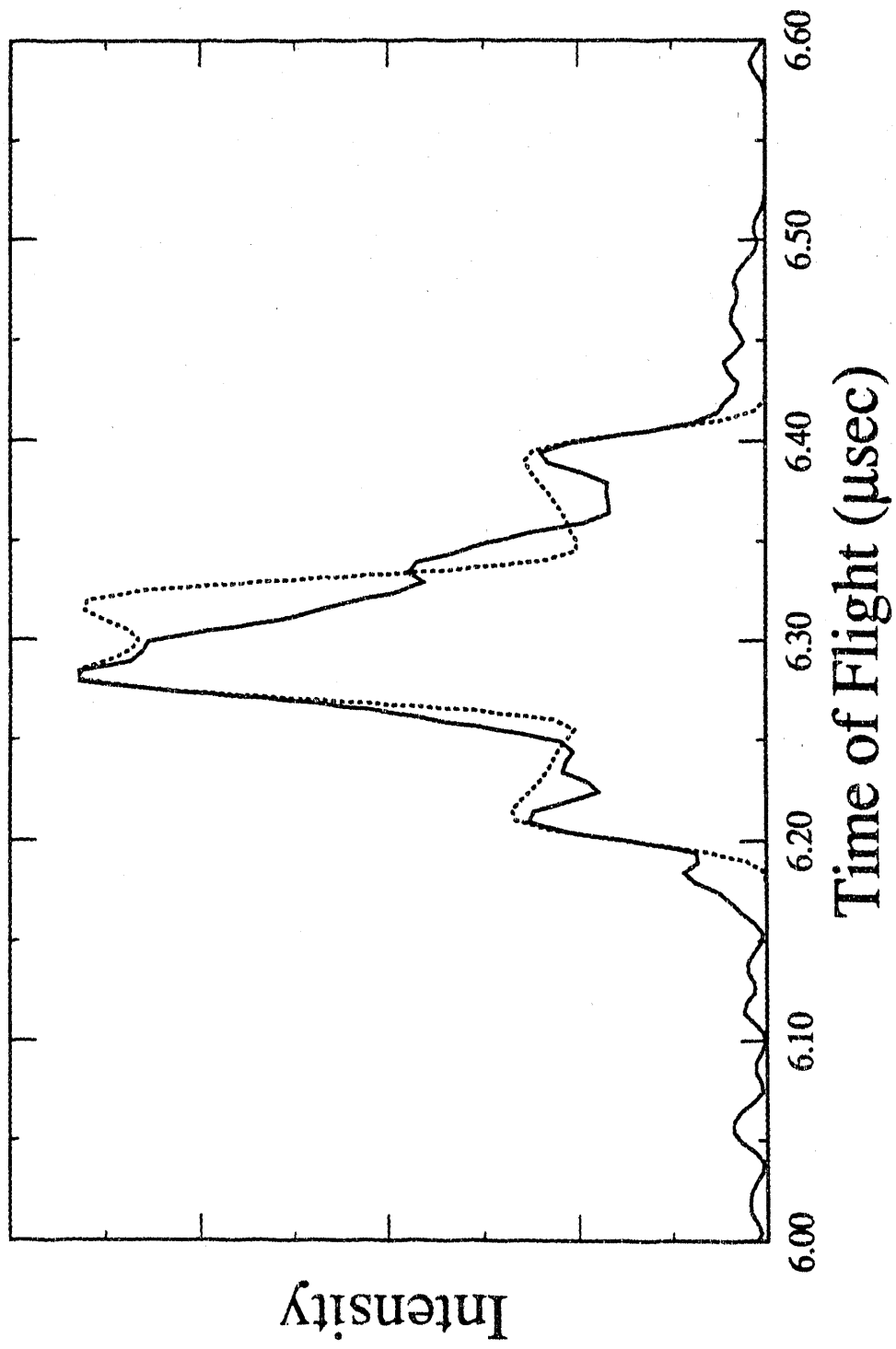


Figure 6-1

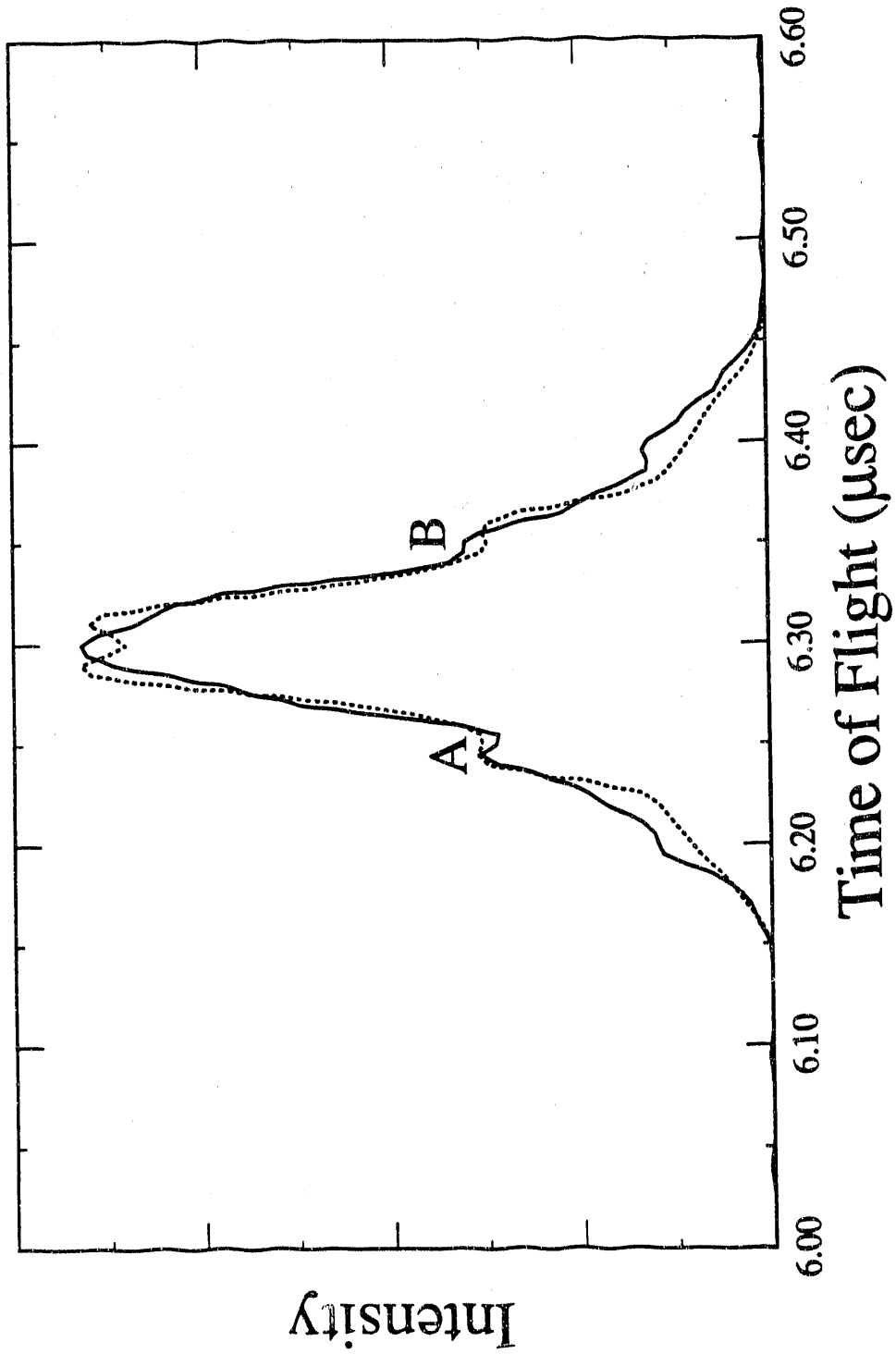


Figure 6-2

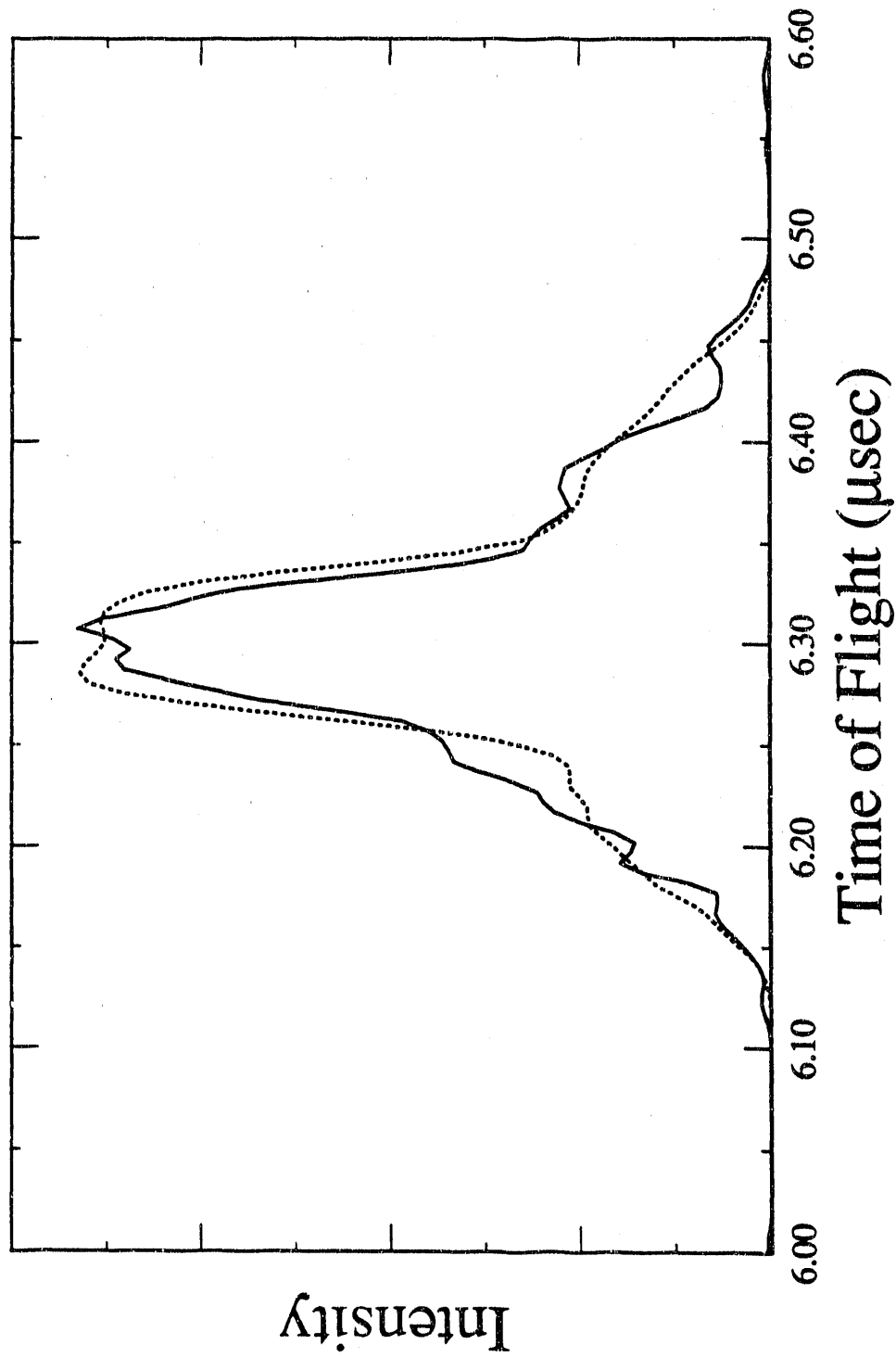


Figure 6-3

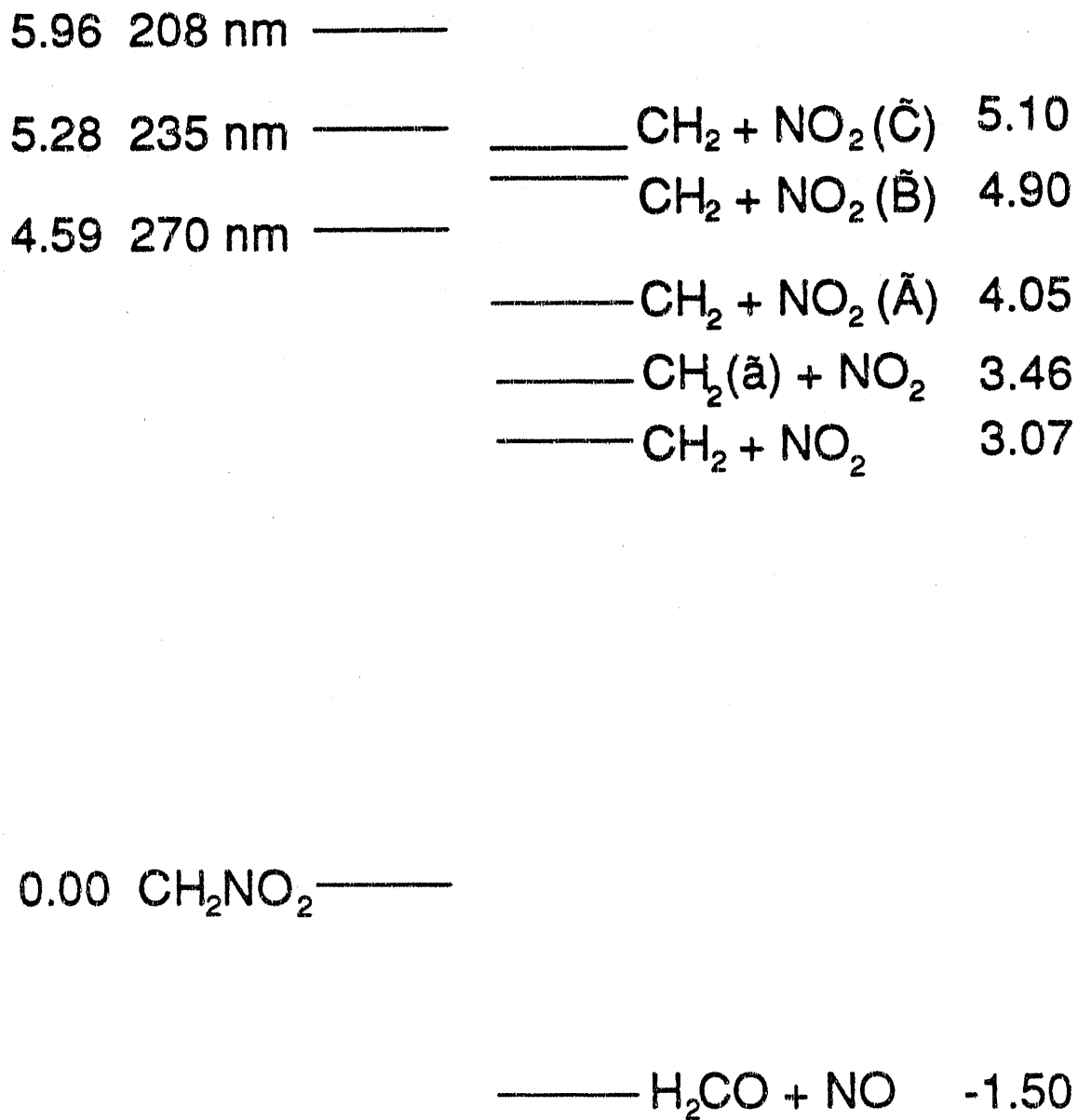


Figure 6-4

Appendix A. Computer Programs for Collinear and Three-Dimensional Adiabatic Simulations of XHY⁻ Photoelectron Spectra

We study hydrogen transfer reactions $X + HY \rightarrow XH + Y$ via photoelectron spectra of the stable XHY⁻ negative ion. The resulting photoelectron spectrum can be simulated using a collinear approach (where the hydrogen is constrained to the X-Y axis) or a full three-dimensional approach. We use an adiabatic treatment and both collinear and three dimensional approaches to analyze the XHY⁻ photoelectron spectra (see Ch 2, 3, 4). The computer codes in this appendix carry out the adiabatic simulations. The collinear simulations are done using the program ad2d7 (sec. 1), the ad3d5 program (sec. 2) carries out three-dimensional simulations, and the sim1 program (sec. 3) convolutes the resulting "stick" photoelectron spectra with a desired resolution function to produce the simulated spectrum.

1. Collinear Simulation Code ad2d7

1.1 Introduction

Simulation of the collinear XHY⁻ photoelectron spectrum requires running the programs ad2d7 and sim1. The ad2d7 program calculates a

"stick" photoelectron spectrum; sim1 then convolutes this stick spectrum with a desired resolution function to produce the simulated spectrum. Section 1.2 discusses how to run the ad2d7 program as a "black box." Section 1.3 discusses how the ad2d7 program works, describing each subroutine as it is used. Section 1.4 is a listing of the ad2d7 program.

1.2 Running the ad2d7 Program

1.2.1 The ad2d.par Input File

The program ad2d7 calculates the photoelectron spectra of XHY^- anions, where the resulting neutral species $[XHY]$ can dissociate and X and Y are (relatively) heavy atoms. Ad2d7 requires one input file (two, if a fitted potential is to be used), and some keyboard input. Three types of output files are generated. Here is a sample ad2d.par input file (note that the line numbers on the left are *not* part of the input file, but are used for ease of reference):

```
1   brhi
2   .d
3   126.9045, 1.00728, 78.918
4   600.0, 0.0
5   0
6   0.1609, 308.4, 17.51, 0.055
7   0.1414, 378.2, 18.10, 0.186
8   0.2469, 177.0, 18.76, 0.22
```

The first two lines define the names of the output files. There are three sets of output files: the first one is a single file (called brhi.d in this

case) that will contain the eigenvalues and corresponding eigenvectors obtained by solving the Schrodinger Equation for hydrogen atom motion at every value of ρ , the hyperspherical radius, which is approximately proportional to R_{XY} . The second set consists of one file for each adiabatic curve, `brhiN.dv`, where N is the adiabatic curve number (0,1,2, etc.), and is used for plotting the adiabatic curves using `plot2d` or an equivalent plotting program. The file has four columns: ρ (in \AA), the approximate value of R_{XY} corresponding to this value of ρ (see Ch. 2, eq. 4), $\epsilon_N(\rho)$, the adiabatic eigenvalue without the DIVAH correction (in eV), and $U_N(\rho)$, the value with the DIVAH correction. The third output file is called `fort.7` and contains the "stick" photoelectron spectrum to be read by the `sim1` convolution program. The third line of `ad2d.par` contains the masses of the three atoms (in amu) in the order X, H, Y. The fourth line defines the harmonic oscillator basis used to solve the Schrodinger Equation for hydrogen motion. The first entry is the harmonic frequency (in cm^{-1}). The value of the harmonic frequency is optimized in the program, so the number used here is unimportant. The second entry defines where the basis functions are centered. The value is given by $c = \frac{1}{2} (R_{HY} - R_{HX})$, in \AA ; this is 0.0 for a symmetric system. For asymmetric systems, the value of c for the anion usually works well (see sec. 1.2.2, line 15), although 0.0 can also be used. You *must* use 0.0 if the DIVAH correction is to be included. The value of c is not optimized in the program. Row 5 specifies the type of potential to be

used: LEPS (0 or 5), LEPS + zero-point bend (4), or a fitted potential (3). If a fitted potential is to be used, the file pot.par, which defines the fitted potential, is required. The format of the pot.par file is given in sec. 1.2.3. Lines 6-8 are the LEPS parameters. These are required (though their values are irrelevant) even if a fitted potential is to be used. Line 6 refers to the diatom HX, line 7 is for HY and line 8 is for XY. The first entry is r_e in nm, the second is D_e in kJ/mol, the third is β in nm^{-1} , and the fourth is the Sato parameter.

1.2.2 Keyboard Inputs

Various parameters determining the type of calculation and anion potential are read from the keyboard. Usually, one creates a file containing these parameters (ie, inbrhi) and then executes the program using this file for keyboard input

```
ad2d7 < inbrhi
```

As the program generates quite a lot of output to the screen, it is useful to also send the output to a file:

```
ad2d7 < inbrhi > out
```

A typical keyboard input file is shown below. Again, line numbers are not part of the file, but are used for reference only. The comments can be part of the file, as the program only reads the first entry in each line.

```

1      1      # 1 = ad. curves, 2 = PES simulation, 3 = 1 and 2
2      40     # quantum number of highest basis function in z
3      8      # number of adiabatic curves
4      2.90   # Rac min (A)
5      4.70   # Rac max (A)
6      60     # points in Rac
7      8      # number of points between each pair of Rac points
8      4      # number of different basis sets to use for rho integral
9      80     # number of basis functions in rho
10     1276.3  # anion a. s. freq.
11     178.1  # anion a. s. anharmonicity
12     100.0  # anion s. s. freq.
13     0.0    # anion s. s. anharmonicity
14     3.88   # Rac in anion (A)
15     -0.39  # (Rbc - Rab)/2 in anion (A) (approx. Rez)
16     1      # 1 = include DIVAH correction, 0 = don't
17     1      # 1 = DVR, 2 = HEG quadrature
18     0      # symmetry in z : 0 = none, 1 = yes
19     1      # 1 : even functions only, 2 : odds only

```

Line 1 selects the type of calculation to be done. A value of 1 means that *only* adiabatic curves should be calculated. The full adiabatic result file `brhi.g` and the adiabatic curves `brhiN.gv` (see section 1.2.1) will be generated. If a value of 2 is used, the full adiabatic result file is read and eigenvalues are calculated on each adiabatic curve and their overlap with the anion is computed to generate a "stick" photoelectron spectrum. A value of 3 will generate a full simulation - it is equivalent to running the code with option 1, then again with option 2. Running the two parts of the code separately is useful as it allows one to obtain a coarse simulation, then refine it (see line 8, below), or to change the anion potential or equilibrium geometry and see how it affects the spectrum. The same basis *must* be used when the full adiabatic output file is read as when it was generated.

Line 2 is the quantum number of the highest basis function used to solve the Schrodinger equation for hydrogen motion. For asymmetric systems, this is also the size of the basis; for symmetric systems, the basis is half this size. Use a value ≤ 100 . The best way to test for convergence of the basis is to calculate a few points at large R_{XY} (see line 5) and increase the basis size until the eigenvalues no longer change. A maximum quantum number of 60 is usually adequate. Line 3 is the number of adiabatic curves to be calculated, up to a maximum of 12. Lines 4 and 5 define the range of interhalogen distances R_{XY} (in Å) over which the adiabatic curves will be calculated. As a rule of thumb, the interhalogen distance in the anion should be at about the midpoint. The adiabatic curves flatten out at large R_{XY} . Do not extend the calculation to excessively large values of R_{XY} , as good convergence in this region requires a large basis. Line 6 is nrhopts, the number of values of R_{XY} at which the Schrodinger Equation for hydrogen motion will be solved. A value of 30 or 40 is fine for simulations; you may wish to use 60 if the adiabatic curves are to be plotted. The maximum value is 100.

Lines 7-9 control the calculation of the heavy atom Schrodinger Equation on each adiabatic curve and the subsequent calculation of integrals. Part of the Franck-Condon integral is calculated using quadrature (see sec. 1.3); line 7 is the number of quadrature points that are taken between each pair of points in R_{XY} . A value of 8 is fine; the maximum

value is 10. Line 9 is `ndvrrho`, the number of basis functions used along R_{XY} , up to a maximum of 100. Do not use values less than 60. Line 8 is `nrhoits`, the number of basis sets that are used along R_{XY} . This is discussed in detail in Ch. 3, sec. 5.4. Use a value of 1 for preliminary calculations, and try 5 or so for later calculations. At high resolution (ie, 5 meV or less), or with particularly repulsive adiabatic curves, the simulation may show sharp structure that will blur out when more simulations are calculated, so you should always try a simulation using `nrhoits = 5`, and, if it looks different from the simulation with `nrhoits = 1`, try again using a value of `nrhoits` greater than 5. If `nrhoits > 1`, the Schrodinger Equation for halogen atom motion will be solved using `nrhoits` different basis sets, ranging in size from `(ndvrrho-40)` to `ndvrrho` in steps of `40/(nrhoits-1)`. There is no maximum value for this parameter, but, with `nrhoits = 1`, solving for heavy atom motion takes about as long as generating the adiabatic curves, so going from `nrhoits = 1` to `nrhoits = 3` will approximately double the time of the entire calculation. The collinear calculation is still very fast - a simulation of the BrHBr^- spectrum using `nrhoits = 20` takes < 1 CPU minute on a Cray XMP.

Lines 10-15 describe the XHY^- anion. Lines 10 and 11 are the frequency and anharmonicity, respectively, for the antisymmetric stretch (or high frequency hydrogen stretch, in asymmetric systems), in cm^{-1} . Morse potentials (unlike harmonic oscillators) do not have a axis of symmetry, so it

is *extremely* important to correctly define the direction of the hydrogen stretching coordinate for asymmetric systems. The hydrogen stretching coordinate is mostly an HY stretch. Thus, the input file ad2d.par places the atoms in the order I - H - Br and the high frequency stretch is the H-Br stretch. Lines 12 and 13 are the symmetric (or interhalogen) stretch frequency and anharmonicity, in cm^{-1} . Lines 14 and 15 describe the geometry of XHY^- . Line 14 is the X - Y distance and line 15 is $\frac{1}{2} (R_{\text{HY}} - R_{\text{HX}})$. Note that this is zero for symmetric systems and is negative for BrHI^- . Both values are in \AA .

Lines 16-19 determine how the calculation of the hydrogen motion wavefunctions is to be carried out. Line 16 is 1 if the DIVAH correction (Ch. 2, eq. 11b) is to be used and is 0 otherwise. Line 17 is 1 if the DVR quadrature is to be used and 2 for the HEG quadrature scheme. If symmetry is to be used, you *must* use the DVR. Also, the DVR is somewhat faster than the HEG scheme, and both give identical answers, so there is little reason to use the HEG method. Line 18 determines if symmetry along the hydrogen stretching coordinate is to be used in calculating the hydrogen stretching wavefunctions (1 for symmetry, 0 for no symmetry). If the anion is symmetric, use symmetry - it will greatly speed the calculation. If symmetry is used, basis functions should be centered at 0.0 (line 4, ad2d.par). Line 19 is used if symmetry is employed. A value of 1 indicates that functions of even (*gerade*) parity are to be calculated, use a value of 2

to calculate odd (*ungerade*) functions. Only even functions have Franck-Condon overlap with the $v_3 = 0$ wavefunction of XHX^- . If odd functions are calculated, their overlap with the $v_3 = 1$ state of XHX^- is calculated.

1.2.3 Using a Fitted Potential

The program allows the use of a very flexible, completely empirical form of the potential energy surface for the neutral reaction. The potential is described in detail in Ch. 2, sec. 4.5.2 and the form of the potential is given in Ch. 2, eq. 6; Chapter 2 describes its use in fitting the BrHBr^- photoelectron spectrum. This type of potential energy surface, called the fitted potential energy surface, can be used by choosing $\text{ipot} = 3$ (line 5 in ad2d.par , see sec. 1.2.1) and including the file pot.par .

The pot.par file defines the fitted potential. A sample file (the fitted $\text{Br} + \text{HBr}$ potential) is given below

```
12
2.6 0.0 -50.0 0.0 5000.0 1300.0
2.8 0.0 -140.0 0.0 4000.0 1600.0
3.0 0.0 -215.0 0.0 2900.0 2100.0
3.29 0.0 -330.5 0.0 800.0 2000.0
3.4 0.31 -343.5 15.0 1600.0 18000.0
3.5 0.465 -360.5 33.5 1100.0 62000.0
3.6 0.541 -370.0 51.0 800.0 22000.0
3.7 0.613 -374.9 84.0 900.0 6000.0
3.85 0.719 -376.8 135.0 1000.0 2200.0
4.0 0.826 -377.0 170.0 1050.0 1200.0
4.2 0.968 -377.1 217.0 1080.0 900.0
```

The first row gives the total number of points in R_{XY} at which the potential is defined (12 in this case). The remainder of the file defines the potential. With reference to Ch. 2, eq. 6, (and see also Table 2-4), the first entry in each row is the value of R_{XY} (in Å), the second value is x_1 (in Å), followed by b (in kJ/mol), h (in kJ/mol), k (in kJ/mol/Å²) and q (in kJ/mol/Å⁴).

1.3 How the ad2d7 Program Works

This section is intended for the user who wishes to modify the ad2d7 program; casual users who wish to use the program as a "black box" should refer to section 1.2. The principles behind the adiabatic method are discussed in Ch. 2. The mathematical framework behind the DVR method and our implementation of the adiabatic method are discussed in detail in Ch. 3. The method is implemented as follows (subroutine names are given in italics):

- 1) The initial parameters for the calculation are read. Names for output files and the type of potential energy surface to be used for the neutral are read from the file ad2d.par (*initpars*). Section 1.2.1 describes the format of the ad2d.par input file. The anion potential and the basis set to be used are read from the keyboard (subroutine *readin*); keyboard inputs are described in section 1.2.2. Also, physical constants are defined (*setconst*), the LEPS potential is initialized (*initleps*, *dojsign*) and, if a fitted

potential is to be used, the parameters for the fitted potential are read from the file *pot.par* and the fitted potential is initialized (*initpot* and *splint*).

The format of the *pot.par* file is described in section 1.2.3.

2) Adiabatic curves are calculated by solving the Schrodinger Equation for hydrogen atom motion at many values of the hyperspherical scattering coordinate ρ (which is roughly proportional to the interhalogen distance R_{XY}). First, the DVR (or HEG quadrature, if that is to be used) is set up (*setdvr*). The position matrix \mathbf{X} is constructed in the harmonic oscillator basis and diagonalized. The matrix \mathbf{K} is calculated. Then, the basis set frequency is optimized (*optimize*). This is done by minimizing the trace of the Hamiltonian matrix (found in *findtrac*) in the HEG quadrature, which is equivalent to minimizing the trace in the DVR, as the trace is invariant to a unitary transformation. The minimum of the trace is found by bisection, searching for the point at which the derivative of the trace with respect to the frequency (*findnup* computes this derivative) is zero. The reference potential is then set up with the optimized frequency (*setv0*), and some basic integrals used for the DIVAH correction are computed (*doover*, described in more detail below).

At each value of ρ (denoted ρ_i) the potential is found along z , the hydrogen stretching coordinate (*setv*) and the Hamiltonian matrix is constructed in the DVR (*setupA2*) or HEG quadrature (*setupA*, *finda*). The Hamiltonian matrix is diagonalized (*findeig*) using the *eispack* library

subroutine *rs*. The DIVAH correction requires the eigenvectors at $\rho_i + \Delta\rho_i$, for each ρ_i , where $\Delta\rho_i = h \rho_i/\rho_{\min}$ and $h = 0.0001$, so the Schrodinger equation for hydrogen motion is solved at both ρ_i and $\rho_i + \Delta\rho_i$, if the DIVAH correction is to be included.

The program can use several functional forms for the potential energy surface for the neutral, based in the *ipot* flag. The subroutine that actually calculates the potential is called from *poten*. If *ipot* = 0 or 5, a LEPS potential is used (*doQJsum*); a LEPS + zero-point bend potential is used if *ipot* = 4. The zero point bend subroutine (*poten3d*, *dpesdr* and *bndh*) was provided by Prof. Joel Bowman (Emory Univ.). If *ipot* = 3, a fitted potential is used. This potential is described in sec. 1.2.3 and Ch. 2. The fitted potential is generated by subroutines *pot6* and *spline*. The *spline* and *splint* subroutines are from Chapter 3.3 of Numerical Recipes (Cambridge Univ. Press, Cambridge, 1986). Modifying the program to use other functional forms for the potential energy surface thus involves modifying *poten* and including a subroutine to calculate the desired potential.

After the Schrodinger Equation for hydrogen atom motion has been solved at all desired values of ρ , the DIVAH correction to the energies (Ch. 2, eq. 11b) is added along each adiabatic curve, if desired (*dodivah*). The DIVAH correction is an integral over the hyperspherical coordinate ϕ . However, the eigenvectors for the hydrogen motion Schrodinger Equation are found as a function of z , where $z = \rho(\phi - \frac{1}{2} \phi_{\max})$, so the coordinate ϕ is

scaled by a factor of $(\rho + \Delta\rho)/\rho = 1 + h/\rho_{\min}$ in going from ρ to $\rho + \Delta\rho$.

Subroutine *doover* calculates the overlap of two harmonic oscillators whose frequency ratio is $(1 + h/\rho_{\min})^2$. Each adiabatic curve is saved (*savewf*, see Sec. 1.2 for the format) as is a file containing the hydrogen motion wavefunctions and the adiabatic curves (*saveall*, see Sec. 1.2). Note that the DIVAH correction only corrects the eigenvalues, not the wavefunctions. Thus, for systems where the DIVAH correction is extremely large (such as F + H₂, where it is often > 100 meV), peak intensities may be wrong.

3) A "stick" photoelectron spectrum is generated by calculating halogen atom wavefunctions on each adiabatic curve and determining their overlap with the anion ground state. The user has the option of only generating the adiabatic curves (parts 1 and 2, above) and later running the program to simulate the spectrum (skipping part 2, above). In this case, the file containing the hydrogen motion wavefunctions and adiabatic curves is read by *readall*. The anion wavefunction is assumed to be a product of wavefunctions along z (taken to be the v_3 coordinate) and along ρ (taken to be the v_1 coordinate). This is a good approximation for Heavy + Light-Heavy systems, but is not good for other mass combinations, such as F + H₂. A more accurate approach could be to assume the symmetric stretch coordinate is proportional to R_{XY} and the asymmetric stretch coordinate is proportional to R_{HY} , although in this case the wavefunction could not be written as the product of a wavefunction in ρ and a second wavefunction in

z and the anion wavefunction would also have to be determined adiabatically. The anion potential is generated by *setva*. The Schrodinger Equation for hydrogen motion for the anion is solved using the same subroutines as were used to solve the corresponding equation for the neutral (step 2, above).

Wavefunctions along the interhalogen distance ρ are found in a fashion analogous to those along z - a harmonic oscillator basis is also used, the potential is linearly interpolated along each adiabatic curve (with the DIVAH correction included, if desired) in *FindV*. However, the HEG quadrature is always used to solve the Schrodinger Equation along ρ , even if the DVR was used to solve the Schrodinger Equation along z . As the Schrodinger Equation along ρ can be solved using nrhoits different basis functions (see 1.2.2, line 8), *initdvr* sets up the matrix **T** for each size basis by calling *setdvr*. Other subroutines are analogous to those used to solve the Schrodinger Equation along z , which are described in step 2. Thus, *setdvr* is analogous to *setdvr*, *setvOrho* : *setv0*, *setvrho* : *setv*, *setArho* : *setupA*, and *findeigr* : *findeig*. The Schrodinger Equation is solved using a 200 cm^{-1} basis set frequency. Subroutines (*optimrho*, *findnupr*, *findtrar*) to optimize the basis set frequency are included, but not used, as optimizing the basis set frequency does not work well on repulsive potentials.

Halogen atom wavefunctions are calculated on each adiabatic curve. First, the adiabatic curve is analyzed as to whether it is purely repulsive or

contains a minimum, and how deep this minimum is (*findmin*). This is carried out so that states can be classified as bound (within the adiabatic approximation) or scattering, but does not affect the results of the calculation. Subroutine *boundrho* then calculates halogen atom wavefunctions and eigenvalues on the adiabatic curve, using the same method as for the anion wavefunction in ρ .

Franck-Condon factors are calculated for each halogen atom wavefunction on each adiabatic curve. This is fairly easy to do, within the adiabatic approximation. The Franck-Condon integral is the collinear version of Ch. 4, eq. 29 :

$$I \propto \left| \int d\rho \Delta_o^i(\rho) \Delta_h^f(\rho) \sum_m c_{m,o}^i c_{m,k}^f(\rho) \right|^2 \quad (1)$$

$c_{m,o}^i$ and $c_{m,k}^f$ are the basis set expansion coefficients for the hydrogen motion wavefunction of the $v_3 = 0$ state of the anion and the $v_3 = k$ state of the neutral, respectively. $\Delta_o^i(\rho)$ and $\Delta_h^f(\rho)$ are the wavefunctions for halogen atom motion for the $v_1 = 0$ state of the anion and wavefunction h on adiabatic curve k for the neutral. The integral is done by quadrature using either the trapezoid rule or Simpson's rule (the program is currently set up to use Simpson's rule, as this is more accurate). The summation over m is done in *zover*, while *findwf* explicitly calculates the basis functions for the harmonic oscillator basis (the product of a Hermite polynomial and a Gaussian), so that *rover* can calculate the neutral wavefunction at the

quadrature points. Subroutine *dofc* calculates the Franck-Condon integral using the appropriate quadrature (provided by *doweight*). The Schrodinger Equation for hydrogen atom motion is solved at *ndvrrho* (typically 30-60) points in ρ . Computing the integral (1) using only *ndvrrho* points in the quadrature gives very inaccurate results. Thus, *nirho* (typically 5-10) quadrature points are used between each pair of points in ρ . Thus, the integral is usually carried out using a ~ 400 point quadrature, ensuring sufficient accuracy. Note that this requires the coefficients $c_{m,o}^i$ and $c_{m,k}^f$ to be interpolated between the points at which the Schrodinger Equation for hydrogen motion is solved.

The cumulative Franck-Condon factors are kept and are printed out. The Franck-Condon factors should sum to 1. A smaller total usually indicates that there is some intensity due to transitions to adiabatic curves lying above the calculated curves. Transitions having Franck-Condon factors greater than 0.001 are saved to the file *fort.7* for plotting by the program *sim1*. The energy saved is $1.0 - (E_i + D_{e_{\max}})$, where E_i is the energy of each transition relative to three-atom dissociation and $D_{e_{\max}}$ is the dissociation energy of the most stable diatomic fragment.

Limitations on the maximum basis set size, number of adiabatic curves and number of points along ρ can all be relaxed by changing the *ad2d7.inc* include file. If this file is modified, the *ad2d7* program should be

recompiled. The ad2d7 program can be compiled using the UNIX make utility via

```
make -f makead2d7
```

1.4 Source Code Listing

1.4.1 makead2d7

```
ad2d7: ad2d7.o leps6.o divah.o findmin.o boundrho3.o dofcf.o pot7.o matrix.o
      segldr -o ad2d7 ad2d7.o leps6.o divah.o findmin.o boundrho3.o dofcf.o pot7.o matrix.o
```

```
ad2d7.o: ad2d7.inc
leps6.o: ad2d7.inc
divah.o: ad2d7.inc
findmin.o: ad2d7.inc
boundrho3.o: ad2d7.inc
dofcf.o: ad2d7.inc
```

1.4.2 ad2d7.inc

```
c
c include file for ad2d7.f
c
c implicit real (a-h,o-z)
c parameter (nm=100, nad=12, maxrpts=100, nXpts=100,maxirho=10)
c
c nm = largest quantum number for basis fcn
c nad = max # ad. curves
c maxrpts = max # pts. in rho
c nXpts = same as above
c maxirho = max # of pts between pairs of pts in rho
c for rho integral (overlaps)
c
c common/units/pi,harev,evwn,hbar,a0,amu,emu,clight,
$ harwn,amass,harkj
c common/gauss/xgs(nm),t(nm,nm),ndvr,ndvrrho
c common/basis/beta,re,rmu,eharm,basisk
c common/debug/iverbose
c common/rand/blank(nm,nm)
c common/arrays/a(nm,nm),s1(nm),s2(nm)
c common/savepot/v0(nm), varray(nm)
c common/inits/nshow,rhomin,rhomin,max,delrho,nrhopts,xmin,xmax
c common/potstuff/ipot,aMa(3),Rel(3),De(3),B(3),S(3),Te(3)
c common/conv/mu,epsilon,gamma,eta,phimax,rtoRxx
c common/results/vec(nad,maxrpts,nm),val(nad,maxrpts)
c common/zcoord/z(nm)
c common/lepstuff/jsign(3,3),cosgam,singam
```

common/anion/veca(maxrpts,nm),rveca(nm)
common/rcoord/rvec(nm,nm)
common/fcfs/fcf(nad,nm)
common/rint/nirho

character*6 molecule, ext
character*1 numbers(10)
common/filestr/numbers,molecule,ext
common/filen0/imollen,iextlen

1.4.3 ad2d7.f

```
c _____  
c file : ad2d7.f  
c Written by RBM, 1991  
c Last modified RBM 9/1991  
c  
c PROGRAM ad2d7  
c  
c Collinear simulations of XHY- photoelectron spectra  
c using DIVAH method and matrix/DVR techniques.  
c  
c Solves 1D S.E. in z at various values of rho -  
c finds all desired wavefunctions at each rho.  
c  
c Contains :  
c main program  
c setconst  
c readin  
c initpars  
c setdvr  
c setupA  
c setupA2  
c Finda  
c findeig  
c setv0  
c setv  
c setva  
c optimize  
c findrup  
c setbeta  
c findtrac  
c savewf  
c saveall  
c readall  
c ShowArr  
c Also uses :
```

```

c   leps6.f, divah.f, findmin.f, boundrho3.f, dofcf.f,
c   pot6.f, matrix.f. (code)
c   makead2d7 (make file)
c   ad2d7.inc (include file with global common blocks)
c   ad2d.par (input file containing output file names,
c           potential type, masses, potential parameters)
c   pot.par (parameters for fitted potential)
c   inbrhbr (sample input file for ad2d7)
c

```

```

include 'ad2d7.inc'

```

```

common/divahin/hrho(maxrpts),dvec(nad,maxrpts,nm)
common/divah/qdiag(maxrpts)
common/divah2/qdiagtot(nad,maxrpts)
common/integ/weight(maxrpts)
common/debug/idebug
common/dvrstuff/ndvr0,ldvr,karr
real karr(nm,nm)
common/sym/isym,istart,istep,ieven
common/saverho/tsave(30,nm,nm),xgssave(30,nm)

```

```

dimension tempval(nm),tempvec(nm,nm)
dimension tempval(nm), Veff(maxrpts)
dimension cstar(maxrpts)
dimension wfx(nm,maxrpts,maxrho),wfint(nm)
dimension rfcf(1000),rval(1000),totfcfad(nad)

```

```

write(*,*) 'Welcome to ad2d7'
call setconst
call initpars

```

```

write(*,*) '1 : Neutral and anion adiabatic curves only'
write(*,*) '2 : Simulation of spectrum using existing ad. curves'
write(*,*) '3 : Both 1 and 2'
write(*,*) 'Pick one : '
read(*,*) isim

```

```

call readin

```

```

write(*,*) 'Number of times rho overlaps are done',
$         ' with different basis'
read(*,*) nrhoits

```

```

write(*,*) 'Number of dvr points in rho : '
read(*,*) ndvrrho

```

```

write(*,*) 'anion a.s. freq : '
read(*,*) aasirq
write(*,*) 'anion a.s. anharmonicity : '
read(*,*) aaswexe
write(*,*) 'anion s.s. freq : '

```

```

read(*,*) assfrq
write(*,*) 'anion s.s. anharmonicity : '
read(*,*) asswexe

write(*,*) 'R (l-l) in anion, in A'
read(*,*) arii
write(*,*) 'z in anion, in A'
read(*,*) reza
Rab = arii/2.0 - reza
Rbc = arii - Rab
write(*,*) 'For anion: Rab = ',Rab,' A and Rbc = ',
$      Rbc,' A'
Rab = Rab/a0
Rbc = Rbc/a0
phiz = atan(gamma*Rbc/(epsilon*Rab + eta*Rbc))
rhoz = Rbc/sin(phiz)
zez = rhoz*(phiz - phimax/2.0)
write(*,*) 'Rho = ',rhoz*a0,' A and z = ',
$      zez*a0,' A'
write(*,*) 'Approx. values are Rho = ',arii*rtorxx,
$      ' A and z = ',reza,' A'

write(*,*) '1 to include DIVAH correction, 0 to ignore'
read(*,*) idivah
write(*,*) '1 for DVR, 2 for HEG quadrature : '
read(*,*) idvr
write(*,*) 'Use symmetry in z ? (1 = yes, 0 = no)'
read(*,*) isym
if (isym .eq. 1) then
  write(*,*) 'Even functions only (1) or odds only (2) : '
  read(*,*) ieven
  ndvr = 1 + (ndvr0 - ieven)/2
  istep = 2
  istart = ieven
else
  istep = 1
  istart = 1
  ndvr = ndvr0
endif

call initleps
if (ipot .eq. 3) call initpot
idebug = 0
if (isim .eq. 2) goto 100

call optimize(anu)
call setdvr
call setv0
hrho0 = 0.0001
if (idivah .eq. 1) call doover(hrho0)
do 10 i = 1,nrhopts

```

```

rho = rho/min + (-1)*delrho
call setv(rho)
if (idvr .eq. 1) then
  call setupA2
else
  call setupA
endif
call findeig(tempval,tempvec)
C
C   save eigenvalues (points on ad. curves)
C
do 5 j = 1, nshow
  val(j,i) = tempval(j)
5  continue
C
C   save eigenvectors (wavefunctions)
C
do 7 j = 1, nshow
  do 6 k = 1, ndvr
    vec(j,i,k) = tempvec(k,j)
6  continue
7  continue

write(*,1010) i,rho,(tempval(j)+harev, j = 1,nshow)
1010 format(i3,' rho : ',f7.3,' a0',10f7.3,2x)
C
C   Solve SE at rho + delta rho, for DIVAH correction
C
if (idivah .eq. 1) then
  hrho(i) = hrho0*rho/rhomin
  call setv(rho+hrho(i))
  if (idvr .eq. 1) then
    call setupA2
  else
    call setupA
  endif
  call findeig(tempval,tempvec)
C
C   save eigenvectors (wavefunctions)
C
do 17 j = 1, nshow
  do 16 k = 1, ndvr
    dvec(j,i,k) = tempvec(k,j)
16  continue
17  continue
endif
10 continue

do 20 j = 1, nshow
  if (idivah .eq. 1) then
    call dodivah(j)

```



```

    else
      do 21 i = 1, ndvr
21      qdiag(i) = 0.0
      endif
      call savewf(j)
      do 22 i = 1,ndvr
        qdiagtot(j,i) = qdiag(i)
22      continue
20    continue
      call saveall(anu)
      if (isim .eq. 1) goto 999
100   if (isim .eq. 2) call readall(anu)
      c
      c calculate anion wavefunction in z -
      c assumes anion potential is separable in rho and z
      c
      call setva(aasfrq,aaswexe,zez,1)
      if (idvr.eq.1) then
        call setupA2
      else
        call setupA
      endif
      call findeig(tempval,tempvec)
      do 24 k = 1, nrhopts
        do 23 i = 1, ndvr
          veca(k,i) = tempvec(i,1)
23        continue
24      continue
      write(*,*) 'Anion a. s. eigenvalues : '
      do 26 i = 1,3
        write(*,*) i,tempval(i)*harwn,' cm-1'
26      continue
      c
      c calculate anion wavefunction in rho
      c
      rera = arii*rtoRxx/a0
      write(*,*) 'Anion is at ',rera,' a0 ',rera*a0,' A'
      anu = 200.0
      call setbeta(anu)
      re = rhomin + (nrhopts/2 - 1) * delrho
      call setdvrr
      call setv0rho
      call setva(assfrq,asswexe,rhoz,2)
      call setArho
      i = 0
      j = 0
      iverbose = 0
      x = 500.0/evwn
      call findeigr(tempval,tempvec,x,i,j)
      iverbose = 0
      do 28 i = 1, ndvrrho

```

```

    rveca(i) = tempvec(i,1)
28  continue
    write(*,*) 'Anion rho eigenvalues : '
    do 29 i = 1,3
        write(*,*) i,tempval(i)*harwn,' cm-1'
29  continue

    call doweight(2)
    call dow
    call findwf(wfx)
    if (iverbose .eq. 1) then
        call testover(wfint,wfx)
        write(*,*) 'Wavefunction overlaps : should all = 1'
        do 31 j = 1, ndvrrho
31     write(*,1090) j, wfint(j)
1090  format(i3,' ',f9.6)
        endif

    totfcf = 0.0
    nline = 0
    if (nrhoits .gt. 1) then
        nsubdel = 40/(nrhoits-1)
    else
        nsubdel = 40
    endif
    re = rhomin + (nrhoits/2 - 1) * delrho

c
c  Set up various basis sets in rho and save them -
c  use same basis sets for each adiabatic curve
c
    anu = 200
    call setbeta(anu)

    do 25 j = 1, nshow
        call findmin(j,lmflag,imin,imax,vmin,vmax,vasy)
        if (lmflag.eq.0) then
            write(*,1100) j
1100  format('Curve # ',i2,' is purely repulsive')
            vmin = val(j,ndvrrho)
        else
            write(*,1110) j,vmin+harev,a0*(rhomin+(imin-1)*delrho),
            $  vmax+harev,a0*(rhomin+(imax-1)*delrho),val(j,nrhoits)+harev
1110  format('Curve # ',i2,' has a min of ',f9.4,' at rho = ',
            $  f7.4,' and a max of ',f9.4,' at rho = ',f7.4,
            $  ' and an asymptote of ',f9.4,' eV')
        endif
        do 30 i = 1, nrhoits
            Veff(i) = val(j,i) + Qdiag(i)
30     continue

        totfcfad(j) = 0.0

```

```

do 35 nit = 1, nrhoits
  nsub = (nit - 1)*nsubdel
  if (j .eq. 1) call initdvr(nit,nsub)
  call boundrho(Veff,0,tempval,rvec,vmax,nbound,nit,nsub)
  if (lmflag .ne. 0) then
    write(*,1120) nbound
1120   format('There are ',i4,' bound levels with energies :')
c     write(*,1130) (i,tempval(i)*harev, i=1,nbound)
1130   format(i3,' ',f12.6)
    endif

    call dofcf(j,ndvrrho,wfx)
    do 40 i = 1, ndvrrho
      if (fcf(j,i) .ge. 0.001) then
        write(*,1140) j,i,tempval(i)*harev,fcf(j,i)
1140   format(i3,' ',i3,' ',f8.5,' ',f5.3)
        nline = nline + 1
        rval(nline) = tempval(i)*harev
        rfcf(nline) = fcf(j,i)
        endif
        if (nit .eq. 1) totfcfad(j) = totfcfad(j) + fcf(j,i)
40    continue
35    continue
        totfcf = totfcf + totfcfad(j)
        write(*,1150) totfcfad(j)
1150   format('FCF (ad curve) = ',f8.6)
25    continue
        write(*,1160) totfcf
1160   format('Total FCF = ',f8.6)
c
c   Save results in fort.7 file for plotting using readfcf15
c
c   Energies are relative to lowest diatomic limit, not
c   3 atom dissociation
c
c   if (De(1) .gt. De(2)) then
      diss = De(1)/96.485
    else
      diss = De(2)/96.485
    endif
    i = 1
    write(7,1170) i
    write(7,1170) nline
1170   format(i3)
    do 200 i = 1,nline
      write(7,1180) rfcf(i)
      write(7,*) 1.0-(diss+rval(i))
200   continue
1180   format(f6.4)
      write(*,*) 'Results in fort.7 - ',nline,' lines saved'
999   stop

```

```

end
C *****
  subroutine setconst
C *****
C
C   define fundamental constants
C
C   include 'ad2d7.inc'

  write(*,*) 'setconst'
  iverbose = 0
  pi = 3.141592654
  hbar = 1.0545887e-34
  harev = 27.211608
  evwn = 8065.479
  a0 = 0.52917706
  amu = 1822.882
  emu = 9.109534e-31
  clight = 2.997925e10
  harwn = harev*evwn
  amass = 1.66056e-27
  harkj = 2.625504e3
  return
end
C *****
  subroutine readin
C *****
C
C   read 'normal' parameters from keyboard :
C
C   ndvr - number of basis functions to be used.
C   nshow - number of wavefunctions to be saved.
C   rhomin - smallest value of rho.
C   delrho - spacing between rho points.
C   nrhopts - number of points in rho.
C   nrho - for overlaps in rho, the number of points BETWEEN
C           each pair of points in rho. Use an even number.
C
C   include 'ad2d7.inc'
  common/dvrstuff/ndvr0,ldvr,karr
  real karr(nm,nm)

  write(*,*) 'readin'
  write(*,1000)
1000 format('Number of basis functions : ', $)
  read(*,*) ndvr0
  write(*,1020)
1020 format('Number of adiabatic curves to calculate : ', $)
  read(*,*) nshow
  write(*,1030)
1030 format('Minimum value of Rac (in A) : ', $)

```

```

read(*,*) racmin
rhomin = racmin*Rtorxx/a0
write(*,1050)
1050 format('Maximum value : ', $)
read(*,*) racmax
rhomax = racmax*Rtorxx/a0
write(*,1060)
1060 format('Number of points in Rac : ', $)
read(*,*) nrhopts
delrho = (rhomax - rhomin)/(nrhopts - 1)
write(*,1070)
1070 format('Number of points between pairs of Rac points for',
$      ' Rac integral : ', $)
read(*,*) nirho
return
end
C *****
c      subroutine initpars
C *****
c
c      Read masses, potential parameters, etc from
c      the file 'ad2d.par' - this avoids having to
c      recompile the code when calculating the effect
c      of isotopic substitution, or when modifying the
c      potential.
c
include 'ad2d7.inc'
common/junk/atomicmu

write(*,*) 'initpars'
numbers(1) = '0'
numbers(2) = '1'
numbers(3) = '2'
numbers(4) = '3'
numbers(5) = '4'
numbers(6) = '5'
numbers(7) = '6'
numbers(8) = '7'
numbers(9) = '8'
numbers(10) = '9'

open(1,file = 'ad2d.par')
read(1,200) molecule
200 format(A6)
imollen = index(molecule, ' ')-1
read(1,200) ext
iextlen = index(ext, ' ')-1
read(1,*) (aMa(i), i = 1,3)
100 format(3f9.5)
read(1,*) anu, re
re = re / a0

```

```

110 format(f10.3,f7.5)
    read(1,120) ipot
120 format(i2)
    if((ipot.eq.0).or.(ipot.eq.3).or.(ipot.eq.4).or.(ipot.eq.5)) then
        do 30 i = 1, 3
            read(1,*) Rel(i), De(i), B(i), S(i)
30    continue
        do 40 i = 1,3
            Te(i) = 1.0/(1.0 + S(i))
40    continue
130 format(4f9.5)
    endif
    close(1)

```

```

    iverbose=1
    if (iverbose.eq.1) then
        write(*,*) 'input file read'
        write(*,*) 'Masses : '
        write(*,100) (aMa(i), i = 1,3)
        write(*,140) anu, re*a0
140    format('Basis set has freq of ',f10.3,' cm-1, '
    $      ',and Re = ',f7.5,' A')
    endif
    iverbose=0

```

- c calculate the parameters needed in the conversion
c from hyperspherical coords

```

    atomicmu = aMa(2)*aMa(3)/(aMa(2) + aMa(3))
    Utemp = aMa(1)*aMa(2)*aMa(3) / (aMa(1) + aMa(2) + aMa(3))
    U = sqrt(Utemp)
    write(*,150) Utemp, U, amu*atomicmu
150    format('utemp : ',f12.8,' u : ',f12.8,' mu : ',f15.8)
    Uabc = aMa(1) * (aMa(2) + aMa(3))/(aMa(1) + aMa(2) + aMa(3))
    gamma = sqrt(aMa(2)*aMa(3)/((aMa(2) + aMa(3)) * U))
    epsilon = sqrt(Uabc/U)
    eta = epsilon * aMa(3)/(aMa(2) + aMa(3))
    rtoRxx = sqrt(aMa(1)*aMa(3)/(aMa(1) + aMa(3))/aMa(2) + 0.5)
    phimax = atan(aMa(2)/U)
    write(*,50) phimax
50    format('Phimax ',f8.5)
    write(*,60) rtoRxx
60    format('rtoRxx = ',f7.5)
    write(*,*) 'gamma = ',gamma,' eta = ',eta,' eps = ',epsilon

    rmu = atomicmu*amu
    call setbeta(anu)
    write(*,160) beta,anu,eharm,eharm*harev*evwn
160    format('beta : ',f9.6,' nu : ',f9.3,' e : ',f9.7
    $      ', h or ',f9.3' cm-1')

```

```

return
end
C *****
  subroutine setdvr
C *****
C
C   Set up for dvr-type quadrature.
C   Use either HEG quadrature or DVR method
C
C   procedure is as follows :
C   1) Find  $\langle Q_n|x|Q_m \rangle = R_{nm}$  and form matrix R.
C   2) Find eigenvalues  $r_i$  and eigenvectors.
C       Use (orthonormalized) eigenvectors to
C       construct T, where
C        $T R T^{-1} = \text{diag}(r_i)$ , and
C        $T^{-1} = T$  (transpose).
C   In HEG quadrature,
C   3) Find  $\langle Q_n|V|Q_m \rangle = A_{nm}$  by sum ( $i=1,n$ ) of  $T_{ni} V(r_i) T_{mi}$ .
C       This is equivalent to a Gaussian Quadrature.
C
C   Notes: For harmonic oscillators  $\langle Q_n|x|Q_m \rangle$  can be
C   found analytically, so the Hermite polynomials need
C   never be evaluated !
C
  include 'ad2d7.inc'

  common/dvrstuff/ndvr0,idvr,karr
  common/debug/idebug
  common/sym/isym,istart,istep,ieven

  dimension wgs(nm,nm),r(nm,nm)
  real karr(nm,nm), epsz(nm,nm), ttrans(nm,nm), et(nm,nm)
C
C    $\langle Q_n|x|Q_{n+1} \rangle = \sqrt{n+1}/(\beta \sqrt{2})$ 
C   all other matrix elements are 0
C   (x can be written in terms of raising/lowering operators..)
C   see Atkins, Molecular QM p. 102
C
  write(*,*) 'setdvr ndvr0 = ',ndvr0
  do 47 i = 1,ndvr0
    r(i,2) = 0.0
    r(i,1) = sqrt(1.0*(i-1))
  47 continue
C
C   Diagonalize R. R is real, tridiagonal, so
C   use eispack routine for banded matrices
C
  call rsb(nm,ndvr0,2,r,xgs,1,wgs,s1,s2,ierr)
C
C   eigenvectors are orthogonal, now normalize
C

```

```

do 70 i=1,ndvr0
  xnorm = 0.0
  do 50 j = 1,ndvr0
    xnorm = xnorm + wgs(j,i)*wgs(j,i)
50  continue
  xnorm = sqrt(xnorm)
  do 60 j = 1,ndvr0
    t(j,i) = wgs(j,i)/xnorm
60  continue
70  continue
  sqrt2 = sqrt(2.0)
  do 80 i = 1,ndvr0
    xgs(i) = xgs(i)/sqrt2
80  continue
c
c  Include symmetry -- use half the points in Tz
c  and mult the rest by sqrt2
c
  if (isym .eq. 1) then
    write(*,*) 'Using symmetry, so chopping T'
    do 58 i = 1, ndvr
      jk = 0
      do 55 j = ieven,ndvr0,2
        jk = jk + 1
        et(jk,i) = t(j,i) * sqrt2
55      continue
58      continue
      do 65 i = 1,ndvr
        do 62 j = 1, ndvr
          t(j,i) = et(j,i)
62        continue
65      continue
    endif

    if(iverbose.eq.1) then
      write(*,*) 'eigenvalues :'
      do 90 i = 1, ndvr0
        write(*,100) xgs(i)
90      continue
      call ShowArr(t,nm,ndvr0)
    endif
100  format(f12.6)
c
c  Set up for DVR method :
c  Set up kinetic energy matrix Karr
c  Karr = Tarr (transpose) Eo Tarr
c  Already found Tarr in setdvr, and Eo is
c  diagonal matrix of eigenvalues to reference potential.
c
  if (idvr .eq. 1) then
    epsz = 0.0

```



```

ik = 0
do 5 i = istart, ndvr0, istep
  ik = ik + 1
  epsz(ik,ik) = eharm*((i-1) + 0.5)
5 continue
if (idebug .eq. 1) then
  write(*,*) 'epsz'
  call ShowArr(epsz,nm,ndvr)
endif
call transpos(nm,t,ndvr,ndvr,ttrans,ndvr,ndvr)
call matmult(nm,epsz,ndvr,ndvr,ttrans,ndvr,ndvr,
#       et,ndvr,ndvr,ierr)
call matmult(nm,t,ndvr,ndvr,et,ndvr,ndvr,
#       karr,ndvr,ndvr,ierr)
endif

return
end

C *****
  subroutine setupA
C *****
C
C Construct the A (Hamiltonian) matrix in HEG quadrature.
C The elements of A (ie, Aij) are
C  $A_{ij} = \langle Q_i | \delta V | Q_j \rangle + e_i \delta_{ij}$ , where
C  $Q_i$  and  $Q_j$  are basis functions (harmonic oscillator,
C in this case), and  $\delta V = V - V_{ref}$ , where
C  $V$  is the potential which we desire to find
C eigenvalues to, and  $V_{ref}$  is the (h.o.) potential
C that  $\{Q_i\}$  are exact solutions to.  $e_i$  is the
C  $i$ th eigenvalue for the reference problem, and
C  $\delta_{ij}$  is 1 if  $i = j$  and 0 otherwise.
C The integral is done using DVR methods – this is
C equivalent to using an ndvr point gauss–hermite
C quadrature, where ndvr is the number of basis functions
C (ie, A is an ndvr by ndvr matrix).
C Note that A is symmetric, as  $\delta V$  contains no operators.
C
  include 'ad2d7.inc'

  if (iverbose.eq.1) then
    write(*,*) 'setupa'
  endif
  do 20 i = 1,ndvr
    do 10 j = 1, i
      A(i,j) = Finda(i,j)
      A(j,i) = A(i,j)
    10 continue
  20 continue
  trace = 0.0
  do 30 i=1,ndvr

```

```

c30  trace = trace + A(i,i)
c   write(*,100) trace
100  format('trace = ',f11.5)
      return
      end
c *****
      subroutine setupA2
c *****
c
c   Find Hamiltonian matrix using dvr methods
c   Includes symmetry !
c
c   Already found kinetic energy Tarr in setdvr
c
c   include 'ad2d7.inc'
c   common/dvrstuff/ndvr0,idvr,karr

      real karr(nnr,nm)

      do 20 i = 1, ndvr
        do 10 j = 1,i
          a(i,j) = karr(i,j)
          if (i .eq. j) a(i,j) = a(i,j) + Varray(i)
          a(j,i) = a(i,j)
10      continue
20      continue
      return
      end
c *****
      function Finda(i,j)
c *****
c
c   Finds  $\langle Q_i | H | Q_j \rangle$ .
c   The  $h_0$  basis set satisfies
c    $h_0 Q_i = e_i Q_i$ , and  $\langle Q_i | Q_j \rangle = \delta(i,j)$ .
c   Let  $H = h_0 + v$ , then
c    $\langle Q_i | H | Q_j \rangle = \langle Q_i | h_0 + v | Q_j \rangle$ 
c    $= \langle Q_i | h_0 | Q_j \rangle + \langle Q_i | v | Q_j \rangle$ 
c    $= e_i * \delta(i,j) + \langle Q_i | v | Q_j \rangle$ , and
c    $e_i = e_{\text{harm}}*(i + 1/2)$ 
c
      include 'ad2d7.inc'

      sum = 0.0
      do 5 k = 1,ndvr
        sum = sum + t(i,k)*Varray(k)*t(j,k)
5      continue
      if (i.eq.j) then
        Finda = sum + eharm*((i-1) + 0.5)
      else
        Finda = sum

```

```

endif
return
end
C *****
  subroutine findeig(tempval,tempvec)
C *****
C
C  uses eispack library routines to diagonalize A,
C  finding the eigenvalues and eigenvectors.
C  The nshow lowest eigenvalues/eigenvectors are
C  saved in tempval(1..nshow), tempvec(1..nshow,1..ndvr),
C  respectively.
C
  include 'ad2d7.inc'
  dimension tempval(nm), tempvec(nm,nm)

  if (iverbose.eq.1) then
    write(*,*) 'findeig'
  endif
  if (iverbose.eq.1) then
    write(*,*) 'A matrix : '
    call ShowArr(a,nm,ndvr)
    write(*,*) 'B matrix : '
    call ShowArr(blank,nm,ndvr)
  endif
C  call rsg(nm,ndvr,a,blank,tempval,1,tempvec,s1,s2,ierr)
  call rs(nm,ndvr,a,tempval,1,tempvec,s1,s1,ierr)
  if (ierr.ne.0) then
    write(*,1000) ierr
    stop
1000 format('ERROR # ',i3,' in diagonalization')
  else
C
C    wavefunctions are ok, so normalize them
C
    do 50 i = 1, nshow
      xnorm = 0.0
      do 30 j = 1, ndvr
        xnorm = xnorm + tempvec(j,i)*tempvec(j,i)
30      continue
      xnorm = sqrt(xnorm)
      do 40 j = 1, ndvr
        tempvec(j,i) = tempvec(j,i)/xnorm
40      continue
50      continue
    endif
    return
  end
C *****
  subroutine setv0
C *****

```

```

c
c  set up reference potential -
c  this is the potential implied by the basis set used...
c  this routine is used once, if the basis set is
c  not modified.
c
  include 'ad2d7.inc'

  if (iverbose.eq.1) then
    write(*,*) 'setv0'
  endif
  do 10 i = 1, ndvr
    z(i) = xgs(i)/beta + re
    dx = z(i) - re
    v0(i) = basisk*dx*dx/2.0
10  continue
  return
  end

c *****
  subroutine setv(rho)
c *****
c
c  set up the difference potential array (varray).
c  varray is the difference between the actual potential
c  and the reference potential (v0).
c
  include 'ad2d7.inc'

  do 10 i = 1, ndvr
    Varray(i) = poten(rho,z(i))
10  continue
c
c  find difference potential by subtracting off
c  reference potential.
c
  do 100 i = 1, ndvr
    varray(i) = varray(i) - v0(i)
100 continue
  return
  end

c *****
  subroutine setva(anwe,anwexe,rea,itpe)
c *****
c
c  For ANION:
c  set up the difference potential array (varray).
c  varray is the difference between the actual potential
c  and the reference potential (v0).
c
  include 'ad2d7.inc'

```

```

dmorse(xx,x0,dee,xalpha)=dee*(1.0-exp(-xalpha*(xx-x0)))**2

if (itype .eq. 1) then
  n = ndvr
else
  n = ndvrrho
endif
if (anwexe .eq. 0) then
  ank = rmu*(anwe/harwn)**2
  do 20 i = 1, n
    rt = z(i) - rea
    Varray(i) = ank*rt*rt/2.0
20  continue
  else
    anweA = anwe/harwn
    anwexeA = anwexe/harwn
    xalphaA=sqrt(2.0*rmu*anwexeA)
    xdeA=anweA**2/(4.0*anwexeA)
    do 30 i = 1,n
      Varray(i) = dmorse(z(i),rea,xdeA,xalphaA)
      if (Varray(i) .gt. 5.0) Varray(i) = 5.0
30  continue
  c   Shelf potential at VERY high values to avoid overflow
  c   in diagonalization.
  endif
  c
  c   find difference potential by subtracting off
  c   reference potential.
  c
  do 100 i = 1, n
    varray(i) = varray(i) - v0(i)
100 continue
  return
  end
  c *****
  c   subroutine optimize(anu)
  c *****
  c   include 'ad2d7.inc'
  c
  c   optimize basis set — only optimize frequency,
  c   leave basis functions centered at value given in ad2d.par
  c   basis set is optimized at largest value of rho
  c   to be calculated, as eigenvalues at large rho are
  c   very sensitive to basis function, while values at small
  c   rho are not very sensitive.
  c   Optimizing the basis set is equivalent to minimizing
  c   the trace of the A matrix, and this is done by finding the
  c   point where (d/danu) trace = 0, where anu is the frequency.
  c
  nit = 0
  anulow = 200.0

```

```

    anuhi = 8000.0
    call setdvr
    anulowp = findnup(anulow)
    anuhip = findnup(anuhi)
5   nit = nit + 1
c   anunew = anulow + anulowp*(anulow - anuhi)/(anuhip - anulowp)
    anunew = (anuhi + anulow)/2.0
    anunewp = findnup(anunew)
    if (anunewp.lt.0.0) then
        anulow = anunew
        anulowp = anunewp
    else
        anuhi = anunew
        anuhip = anunewp
    endif
    if (iverbose.eq.1) then
        write(*,100) nit, anulow, anuhi
    endif
100 format('it : ',i3,' min : ',f8.3,' max : ',f3.3)
    if ((nit.lt.20) .and. ((anuhi-anulow) .gt. 5.0)) goto 5
    anu = (anuhi+anulow)/2.0
    write(*,190) anu
190 format('Optimized frequency is ',f9.4,' cm-1')
    call setbeta(anu)
    return
end

c *****
function findnup(anu)
c *****
c
c   find derivative of trace w.r.t. the frequency anu.
c
c   include 'ad2d7.inc'

    hanu = 0.0001
    call setbeta(anu-hanu)
c   write(*,160) beta,anu,eharm,eharm+harev*evwn
160 format('beta : ',f9.6,' nu : ',f9.3,' e : ',f9.7
    $ ', ' or ',f13.4' cm-1')
    call setv0
    call setv(rhormax)
    call findtrac(trace1)

    call setbeta(anu+hanu)
    call setv0
    call setv(rhormax)
    call findtrac(trace2)
    findnup = (trace2 - trace1)/hanu/2.0
c   write(*,180) anu,findnup
180 format(' nu : ',f13.7,' deriv : ',f16.7)
    return

```

```

end
C *****
  subroutine setbeta(anu)
C *****
C
C   Input basis function frequency and calculate beta in atomic units
C   beta = {(k*mu)/hbar^2}^(1/4)
C   where k is the force constant for a harmonic oscillator
C   of this frequency, and
C       k = [(2*pi*c*freq)^2]*mu
C
  include 'ad2d7.inc'
  common/junk/atomicmu

  eharm = anu/harwn
  beta = sqrt(atomicmu*amu*eharm)
  basisk = atomicmu*amu*eharm*eharm
C   write(*,160) beta,anu,eharm,eharm*harev*ewwn
160  format('setbeta : ',f9.6,' nu : ',f9.3,' e : ',f9.7
  $   ', ' or ',f13.4' cm-1')
  return
end
C *****
  subroutine findtrac(trace)
C *****
  include 'ad2d7.inc'

  trace = 0.0
  do 10 j = 1,ndvr
    trace = trace + FindA(j,j)
10  continue
  return
end
C *****
  subroutine savewf(jcurve)
C *****
  include 'ad2d7.inc'
  common/divah/qdiag(maxrpts)

  character*15 fname

  dimension wfx(nm,nXpts),xnorm(nm)

  write(*,*) 'savewf'

  fname = molecule(1:imollen) // numbers(jcurve)
  $     // ext(1:iextlen)
C
C   save adiabatic curve in plotfile for violet
C
  open(2,file = fname(1:imollen+iextlen+1) // 'v')

```

```

write(2,950) (
$ (rhomin+(i-1)*delrho)*a0,
$ (rhomin+(i-1)*delrho)*a0/rtorxx,
$ val(jcurve,i)*harev,
$ harev*(val(jcurve,i)+qdiag(i)), i = 1, nrhopts)
950 format(f8.4,2x,f8.4,2x,f10.6,2x,f10.6)
close(2)
return
end
C *****
subroutine saveall(anu)
C *****
include 'ad2d7.inc'
common/divah/qdiag(maxrpts)
common/divah2/qdiagtot(nad,maxrpts)

character*15 fname

write(*,*) 'saveall'
fname = molecule(1:imollen) // ext(1:iextlen)

open(2,file = fname)
write(2,900) nshow,ndvr,nrhopts,rhomin,rhomas,anu
900 format(3i5,3f12.5)
do 10 i = 1,nshow
10 write(2,910) (val(i,j)+qdiagtot(i,j),j=1,nrhopts)
c write(2,910) (val(i,j)+qdiagtot(i,j),j=1,nrhopts)
910 format(f9.6)
do 30 i = 1,nshow
do 20 j = 1,nrhopts
write(2,920) (vec(i,j,k), k=1,ndvr)
20 continue
30 continue
920 format(f9.6)
close(2)
return
end
C *****
subroutine readall(anu)
C *****
include 'ad2d7.inc'

character*15 fname

write(*,*) 'readall'

fname = molecule(1:imollen) // ext(1:iextlen)
open(2,file = fname)
read(2,*) nshow,ndvr,nrhopts,rhomin,rhomas,anu

do 10 i = 1,nshow

```



```

10  read(2,*) (val(i,j),j=1,nrhopts)
    do 30 i = 1,nshow
      do 20 j = 1,nrhopts
        read(2,*) (vec(i,j,k), k=1,ndvr)
20  continue
30  continue

    close(2)
    call setbeta(anu)
    write(*,*) 'anu : ',anu,' beta : ',beta
    call setdvr
    call setv0
    return
    end
C *****
C      subroutine ShowArr(a,nm,n)
C *****
C      implicit real (a-h,o-z)
C
C NOTE : change to 'real*8' on anything other than the Cray !!!
C
    dimension a(nm,nm)

    write(6,50)
50  format(' ')
    do 10 i = 1,n
      write(6,100) (a(j,i), j=1,n)
10  continue
100 format(10f10.5,/)
    return
    end

```

1.4.4 leps6.f

```

C -----
C leps6.f - LEPS potential for A + BC system,
C can also include zero-point bend.
C Written by RBM 1988
C Latest version by RBM 9/91
C See Ian Smith, Kinetic and Dynamics of Elementary Gas Reactions
C Sec. 2.3.3.
C zero-point-energy subroutines dpesdr and bndh
C from Prof. Joel Bowman (Emory U)
C
C Includes :
C  initleps

```

```

c doQJsum
c dojsign
c dpesdr
c bndh
c poten
c poten3d
c
c Indices :
c     1 = ab
c     2 = bc
c     3 = ca
c
c Variables:
c     R[1] = ab distance (nm)
c     S[1] = Sato Parameter
c     De[1] = dissociation limit of ab (kJ/mol)
c     Rel[1] = equilibrium ab bond length (nm)
c     B[1] = width of potential for ab (nm-1)
c     M[1] = mass of atom a (amu)
c
c Functions:
c     V(R[1], R[2]) = LEPS potential (kJ/mol)
c     Q(1,R[1]) = Q for ab and Rab
c     aJ(1,R[1]) = J for ab and Rab
c     VM(1,R[1]) = Morse potential for ab and Rab
c     VaM(1,R[1]) = anti-Morse potential for ab and Rab
c
c *****
c     subroutine initleps
c *****
c     include 'ad2d7.inc'
c     call dojsign
c
c     set constants for bndh
c
c     h = 0.01
c     cosgam = cos(pi+h)
c     singam = sin(pi+h)
c     return
c     end
c *****
c     subroutine doQJsum(R,Qsum,aJsum)
c *****
c     include 'ad2d7.inc'
c     dimension R(3), taJ(3)
c
c     Qsum = 0.0
c     aJsum = 0.0
c     do 10 i = 1,3
c       x = -B(i)*(R(i)-Rel(i))
c       aVM = De(i)*(exp(2.0*x)-2.0*exp(x))

```

```

aVaM = De(i)*(exp(2.0*x) + 2.0*exp(x))/2.0
taJ(i) = ((1.0 + S(i)) * aVM - (1.0 - S(i))*aVaM)/2.0
Qsum = Qsum + Te(i)*((1.0+S(i))*aVM +
$      (1.0-S(i))*aVaM)/2.0
10 continue

do 30 i = 3,1,-1
do 20 k = 1,i
aJsum = aJsum + taJ(i)*jsign(i,k)*Te(i)*taJ(k)*Te(k)
20 continue
30 continue
return
end
C *****
C subroutine dojsign
C *****
C include 'ad2d7.inc'

do 10 i = 1,3
do 5 k = 1,3
if (i.eq.k) then
jsign(i,k) = 1
else
jsign(i,k) = -1
endif
5 continue
10 continue
return
end
C *****
C function dpesdr(Rab, Rbc, Rac)
C *****
C
C Obtain numerical first derivative
C of potential surface wrt r(3)
C
C include 'ad2d7.inc'

h = 0.001
C
C this is a sample call to the potential
C which returns the energy in au. user supplies this
C in whatever form is handy. Just remember the convention
C for the array r(3) given in bndh
C
e1 = poten3d(Rab, Rbc, Rac - h)
e2 = poten3d(Rab, Rbc, Rac + h)

dpesdr = (e2 - e1)/h/2.0
return
end

```

```

c *****
  function bndh(Rab,Rbc)
c *****
c
c...program to calculate the bending eigenvalue for an abc triatomic
c...in the harmonic approximation.   written by j.m. bowman
c
c...the bend energy bndh is calculated in the harmonic approximation
c...given by: bndh = hbar*omega*(1.0+nb) (in atomic units).  this
c...is specific for a triatom with a linear minimum, i.e., bend is doubly
c...degenerate.  for a discussion of this and a test of the method see
c...j.m. bowman, chem. phys. lett. 124 (1986) 263.
c
c...rr(1) is rab, rr(2) is rbc and rr(3) is rac=rab+rbc for linear geom.
c...gamma is abc bond angle.  the array rr is input
c...assume that potential is calculated in au
c
  include 'ad2d7.inc'

  h = 0.01
  amasa = aMa(1) * amu
  amb = aMa(2) * amu
  amc = aMa(3) * amu
  anb = 0.0
  Rac = Rab + Rbc
c
c...calculate the g-matrix element
c
  g = 1.0/(amasa*Rab*Rab) + 1.0/(amc*Rbc*Rbc) +
  $   (1.0/Rab + 1.0/Rbc)**2/amb
c
c...calculate the second derivative of the potential wrt gamma using
c...first derivative of pot wrt rr(3) holding rr(1) and rr(2) fixed.
c...calculated in dpesdr and finite difference approx and chain rule.
c
  Ractemp = sqrt(Rab*Rab + Rbc*Rbc - 2.0*Rab*Rbc*cosgam)
c
c now get the first derivative of the potential wrt rr(3)
c
  deriv = dpesdr(Rab, Rbc, Ractemp)
  dpg1 = deriv*singam*Rab*Rbc/Rac
  fgam = dpg1/h
  if (fgam.ge.0.0) then
    bndh = sqrt(g*fgam)*(1.0+anb)
  else
    bndh = 0.0
  endif

  return
end
c

```

```

c some results you should get in tests:
c
c for clhl bcmr leps surface--rhcl=2.77 au at saddle point and
c ground bend energy should be 509 cm-1 (plus or minus 2 cm-1)
c
c for h+h2 on lsth surface at saddle point rhh = 1.757 au
c ground bend should be 906 cm-1.
c
c *****
c      function poten(rho, zt)
c *****
c
c Return potential at (rho, zt) using a LEPS potential (ipot = 0,5)
c a LEPS + bend potential (ipot = 4) or
c a fitted  $x^2 \cdot x^4 / x^6$  potential (ipot = 3).
c rho and phi are in bohrs, and the potential is in hartrees
c
c Includes SMALL bend angles. The formulas don't work for large
c bend angles (angle is the bend angle in radians)
c
c      include 'ad2d7.inc'

      dimension R(3)

      phi = zt/rho + phimax/2.0
      rho2 = gamma*rho

      Rab = (rho2*cos(phi)-(eta/gamma)*rho2*sin(phi))/epsilon
      Rbc = rho2*sin(phi)/gamma

      angle = 0.0
      if ((ipot.eq.0).or.(ipot.eq.5)) then
c      LEPS potential
      Rac = sqrt(Rab*Rab + Rbc*Rbc - 2.0*Rab*Rbc*cos(pi+angle))
      R(1) = a0 * Rab / 10.0
      R(2) = a0 * Rbc / 10.0
      R(3) = a0 * Rac / 10.0

      call doQJsum(R,Qsum,aJsum)

      poten = (Qsum - sqrt(aJsum))/harkj
      endif
      if (ipot .eq. 3) then
c      {  $x^2/x^4/x^6$  potential }
      poten = pot6(a0*Rab, a0*Rbc)
      endif
      if (ipot.eq.4) then
c      LEPS + bend
      R(1) = a0 * Rab / 10.0
      R(2) = a0 * Rbc / 10.0
      R(3) = R(1) + R(2)

```

```

    call doQJsum(R,Qsum,aJsum)

    poten = (Qsum - sqrt(aJsum))/harkj + bndh(Rab,Rbc)
c
c   add bend and convert to hartrees
c
    endif
    return
end

```

```

c *****
  function poten3d(Rab,Rbc,Rac)
c *****
  include 'ad2d7.inc'

  dimension R(3)

  R(1) = a0 * Rab / 10.0
  R(2) = a0 * Rbc / 10.0
  R(3) = a0 * Rac / 10.0

  call doQJsum(R,Qsum,aJsum)

  poten3d = (Qsum - sqrt(aJsum))/harkj
  return
end

```

1.4.5 divah.f

```

c -----
c   file : divah.f
c   Written by RBM 1990
c
c   DIVAH correction for collinear systems.
c   See RBM, thesis, Ch. 2 (and references therein)
c   for a full explanation
c
c   Includes :
c   dodivah
c   doover
c   fac
c *****
  subroutine dodivah(jcurve)
c *****
  include 'ad2d7.inc'

```

```

common/divahin/hrho(maxrpts),dvec(nad,maxrpts,nm)
common/divah/qdiag(maxrpts)
common/overlap/qqint(nm,nm)
write(*,*) 'dodivah'

ii = 1
do 20 i = 1, nrhopts
  sum = 0.0
  do 10 k = 1,ndvr
    do 10 l = 1,ndvr
      sum = sum + dvec(jcurve,i,l)*vec(jcurve,i,k)*qqint(k,l)
10  continue
  qdiag(i) = (1.0 - abs(sum))/hrho(i)/hrho(i)/rmu
20  continue
return
end
c *****
c  subroutine doover(hrho0)
c *****
c
c  Calculate overlap of two harmonic oscillators
c  with same ro and different frequencies. Adapted
c  from Hutchisson, Phys Rev 36, 410 (1930).
c  Note that the square of qqint(mm,nn) is the FCF
c  of the mm->nn transition. However, the sign of
c  qqint is important!
c  By the way, alf = sqrt(wepp/wep), where wepp is the
c  frequency of one oscillator and wep is the other.
c
c  Would be used if different basis sets were
c  used for neutral and anion
c
c  include 'ad2d7.inc'
common/overlap/qqint(nm,nm)
common/divahin/hrho(maxrpts),dvec(nad,maxrpts,nm)
common/dvrstuff/ndvr0,idvr,karr
real karr(nm,nm)
dimension qdint(nm,nm)

alf = 1.0 + hrho0/rhomin
den = 1.0 + alf*alf
xnum = 1.0 - alf*alf
nppmax = ndvr
npmax = ndvr
do 20 nn = 1,nppmax
  npp = nn-1
  do 20 mm = 1,npmax
    np = mm-1
    sum = 0.0
    lmax = min0(np,npp)+1
    do 10 ll = 1,lmax

```

```

l=l-1
prod = 0.0
if(amod(float(np-1),2.0).eq.0.0) then
  i=(np-1)/2
  if(amod(float(npp-1),2.0).eq.0.0) then
    j=(npp-1)/2
    prod = 1.0
    if(l.ne.0) prod = prod*(1./fac(l))*(4.*alf/den)**l
    if(i.ne.0) prod = prod*(1./fac(i))*(xnum/den)**i
    if(j.ne.0) prod = prod*(1./fac(j))*(-xnum/den)**j
  endif
endif
10  sum = sum + prod
   qqint(mm,nn) = sum*sqrt((fac(np)*fac(npp))/2.**(np+npp))
20  continue
c
c  Normalize overlaps so that sum of FCFs = 1.
c  the normalization doesn't depend on nn or mm,
c  just on alf (the mm and nn dependence is taken
c  care of above) -- so calculate overlaps using
c  just the zero wavefcn.
c
tot = 0.0
do 25 mm = 1,npmax
  tot = tot + qqint(mm,1)**2
25  continue
c  write(*,105) tot
tot = sqrt(tot)
do 27 nn = 1,nppmax
  do 27 mm = 1,npmax
27  qqint(mm,nn)=qqint(mm,nn)/tot
c  do 30 nn = 1,nppmax
c  write(*,100) (qqint(mm,nn), mm=1,npmax)
c  write(*,*) '-----'
c30 continue
c100 format(15f10.5,/)
c105 format('- ',f10.5,' -')
c
c  calculate overlap matrix for DVR, using overlap matrix
c  in FBR.
c
if (idvr .eq. 1) then
  do 70 ibeta = 1, ndvr
    do 60 ialf = 1, ndvr
      sum = 0.0
      do 50 j = 1, ndvr
        do 40 i = 1, ndvr
          sum = sum + t(i,ialf)*t(j,ibeta)*qqint(i,j)
40        continue
50      continue
      qdint(ialf,ibeta) = sum

```



```

60   continue
70   continue

      do 90 j = 1,ndvr
        do 80 i = 1,ndvr
          qqint(i,j) = qdint(i,j)
80     continue
90     continue
      endif

      return
      end
c *****
c   function fac(i)
c *****
c
c   Find i!
c
      tot = 1.0
      do 10 k = 1,i
        tot = tot*k
10    continue
      fac = tot
      return
      end

```

1.4.6 findmin.f

```

c -----
c   file : findmin.f
c
c   written by RBM 1990
c
c *****
c   subroutine findmin(jcurve,lmflag,imin,imax,vmin,vmax,vasy)
c *****
c
c   Analyze adiabatic curve to determine if it
c   is purely repulsive or contains local minima.
c   If it has local minima, find the deepest, and
c   also find the highest local maximum.
c
c   include 'ad2d7.inc'
c
c   jcurve : adiabatic curve number
c   lmflag : 0 if purely repulsive curve, 1 if it has a local minimum

```

```

c   imin : pt # for deepest local minimum
c   imax : pt # for local max
c   vmin : potential at deepest local minimum
c   vmax : potential at local maximum
c   vasy : potential at last point (nrhopts)
c

```

```

      lmflag = 1

```

```

c
c   find first local minimum
c
      i = 0
5     i = i + 1
      if (val(jcurve,i+1).gt.val(jcurve,i)) then
          ismall = i
          k = nrhopts + 1
7     k = k - 1
          if (val(jcurve,k-1).gt.val(jcurve,k)) goto 7
          ilarge = k
          vmin = val(jcurve,ismall)
          vmax = val(jcurve,ilarge)
          imin = ismall
          imax = ilarge
          do 8 k = ismall, ilarge
              if (val(jcurve,k) .lt. vmin) then
                  vmin = val(jcurve,k)
                  imin = k
              endif
              if (val(jcurve,k) .gt. vmax) then
                  vmax = val(jcurve,k)
                  imax = k
              endif
          8     continue
      else
          if (i .lt. nrhopts-1) goto 5
          lmflag = 0
      endif

999  vasy = val(jcurve,nrhopts)
      return
      end

```

1.4.7 boundrho3.f

```

c -----
c   file : boundrho3.f

```

```

c   written by RBM 1991
c
c   Solves 1D Schrodinger equation for heavy atom motion on
c   a given adiabatic curve
c
c   Includes :
c   initdvrr
c   boundrho
c   FindV
c   FindAV
c   setvOrho
c   setvrho
c   setArho
c   Findarho
c   optimrho
c   findnupr
c   findtrar
c   setdvrr
c   findeigr
c *****
c   subroutine initdvrr(n,nsub)
c *****
c
c   Set up dvr matrices for this size basis
c   Don't optimize basis frequency
c
c   include 'ad2d7.inc'
c   common/saverho/tsave(30,nm,nm),xgssave(30,nm)
c
c   ndvrsave = ndvrrho
c   ndvrrho = ndvrrho - nsub
c
c   write(*,*) 'ndvrrho = ',ndvrrho
c   call setdvrr
c   do 20 i = 1, ndvrrho
c     do 10 j = 1,ndvrrho
c       tsave(n,i,j) = t(i,j)
10    continue
c     xgssave(n,i) = xgs(i)
20    continue
c   ndvrrho = ndvrsave
c
c   return
c   end
c *****
c   subroutine boundrho(Veff,iaflag,tempval,tempvec,
c   $   vtop, neigen, n, nsub)
c *****
c
c   Find shape and Feshbach resonances on the adiabatic
c   curve Veff. (if iaflag = 0) or anion wavefunctions

```

```

c (if iaflag = 1). vtop is the maximum energy at which
c wavefunction are considered bound. On return, neigen
c is the number of 'bound' states; tempval() is the
c eigenvalues, and tempvec() is the wavefunctions.
c
include 'ad2d7.inc'
common/saverho/tsave(30,nm,nm),xgssave(30,nm)

dimension Veff(*),tempval(nm),tempvec(nm,nm)

c
c set up A matrix
c
ndvrsave = ndvrrho
ndvrrho = ndvrrho - nsub
do 20 i = 1, ndvrrho
do 10 j = 1, ndvrrho
t(i,j) = tsave(n,i,j)
10 continue
xgs(i) = xgssave(n,i)
20 continue

iverbose = 0
call setv0rho
call setvrho(Veff, iaflag)
call setArho

c
c get the eigenvalues
c
call findeigr(tempval, tempvec, vtop, neigen, iaflag)
ndvrrho = ndvrsave

return
end
c *****
function FindV(Veff, rho2)
c *****
c
c Find adiabatic potential at point rho2 on adiabatic
c curve Veff(). Use linear interpolation if necessary.
c
include 'ad2d7.inc'

dimension Veff(*)

rhomax = rhomin + (nrhopts-1)*delrho
if (rho2.le.rhomin) then
FindV = Veff(1) + (Veff(1)-Veff(2))*(rhomin-rho2)/delrho
endif
if (rho2.ge.rhomax) then
FindV = Veff(nrhopts)

```

```

endif
if ((rho2.gt.rhomin).and.(rho2.lt.rhomin)) then
pt = (rho2 - rhomin)/delrho + 1
ipt = int(pt)
p = pt - ipt
FindV = (1.0 - p) * Veff(ipt) + p * Veff(ipt + 1)
endif
c10 if (iverbose.eq.1) write(*,1000) rho2, ipt, p, Veff(ipt)*harev,
c $ Veff(ipt+1)*harev,FindV*harev
1000 format('rho :',f8.3,' pt ',i3,' p ',f6.5,' Ve ',f8.3,
$ ' ',f8.3,' V ',f8.3)
return
end
c *****
function FindAV(rho)
c *****
c
c Find anion potential in rho - unused !
c
FindAV = 0.0
return
end
c *****
subroutine setv0rho
c *****
c
c set up reference potential -
c this is the potential implied by the basis set used...
c this routine is used once, if the basis set is
c not modified.
c
include 'ad2d7.inc'

if (iverbose.eq.1) then
write(*,*) 'setv0rho'
endif
do 10 i = 1, ndvrrho
z(i) = xgs(i)/beta + re
dx = z(i) - re
v0(i) = basisk*dx*dx/2.0
10 continue
return
end
c *****
subroutine setvrho(Veff,iaflag)
c *****
c
c Set up the difference potential array (Varray).
c Varray is the difference between the actual potential
c and the reference potential (v0).
c

```

```

include 'ad2d7.inc'
dimension Veff(*)

do 10 i = 1, ndvrrho
  if (iaflag.eq.0) Varray(i) = FindV(Veff,z(i))
c   if (iaflag.eq.0) Varray(i) = 0.01*(z(i)-re)**2
  if (iaflag.eq.1) Varray(i) = FindAV(z(i))
  if (iverbose.eq.1) write(*,1010) z(i),Varray(i)*harev
1010 format('z : ',f9.4,' : ',f12.4,' eV')
10 continue
c
c   find difference potential by subtracting off
c   reference potential.
c
  if (iverbose .eq. 1) write(*,1000) (varray(i),i=1,ndvrrho)
1000 format(10f9.6,/)
  do 100 i = 1, ndvrrho
    varray(i) = varray(i) - v0(i)
100 continue
  if (iverbose .eq. 1) write(*,1000) (varray(i),i=1,ndvrrho)
  return
end
c *****
subroutine setArho
c *****
c
c   Construct the A (Hamiltonian) matrix.
c
include 'ad2d7.inc'

if (iverbose.eq.1) then
  write(*,*) 'setArho'
endif
do 20 i = 1,ndvrrho
  do 10 j = 1, i
    A(i,j) = Findarho(i,j)
    A(j,i) = A(i,j)
10 continue
20 continue
return
end
c *****
function Findarho(i,j)
c *****
c
c   Find an element of the Hamiltonian matrix.
c   Method : HEG quadrature.
c
include 'ad2d7.inc'

sum = 0.0

```

```

do 5 k = 1,ndvrrho
  sum = sum + t(i,k)*Varray(k)*t(j,k)
5 continue
if (i.eq.j) then
  Findarho = sum + eharm*((i-1) + 0.5)
else
  Findarho = sum
endif
return
end
c *****
c subroutine optimrho(Veff, iaflag)
c *****
include 'ad2d7.inc'

dimension Veff(*)

c
c optimize basis set in rho -- only optimize frequency,
c leave basis functions centered at z = Vmin(rho).
c optimizing the basis set is equivalent to minimizing
c the trace of the A matrix, and this is done by finding the
c point where (d/danu) trace = 0, where anu is the frequency.
c
call setdvrr
nit = 0
anulow = 20.0
anuhi = 4000.0
anulowp = findnupr(Veff,iaflag,anulow)
anuhip = findnupr(Veff,iaflag,anuhi)
5 nit = nit + 1
anunew = (anuhi + anulow)/2.0
anunewp = findnupr(Veff,iaflag,anunew)
if (anunewp.lt.0.0) then
  anulow = aunew
  anulowp = aunewp
else
  anuhi = aunew
  anuhip = aunewp
endif
if (iverbose.eq.1) then
  write(*,100) nit, anulow, anuhi
endif
100 format('it : ',i3,' min : ',f8.3,' max : ',f8.3)
if ((nit.lt.20) .and. ((anuhi-anulow) .gt. 25.0)) goto 5
anu = (anuhi+anulow)/2.0
write(*,190) anu
anu = 200.0
write(*,190) anu
190 format('Optimized frequency is ',f9.4,' cm-1')
call setbeta(anu)
return

```

```

end
c *****
function findnupr(Veff,iaflag,anu)
c *****
c
c find derivative of trace w.r.t. the frequency anu.
c
include 'ad2d7.inc'

dimension Veff(*)

hanu = 0.0001
call setbeta(anu-hanu)
c write(*,160) beta,anu,eharm,eharm*harev*evwn
160 format('beta :',f9.6,' nu :',f9.3,' e :',f9.7
$ , or ',f13.4' cm-1')
call setv0rho
call setvrho(Veff, iaflag)
call findtrar(trace1)

call setbeta(anu+hanu)
call setv0rho
call setvrho(Veff, iaflag)
call findtrar(trace2)
findnupr = (trace2 - trace1)/hanu/2.0
c write(*,180) anu,findnupr
180 format(' nu :',f13.7,' deriv :',f16.7)
return
end
c *****
subroutine findtrar(trace)
c *****
c
c Find trace of Hamiltonian matrix (used to optimize basis)
c
include 'ad2d7.inc'

trace = 0.0
do 10 j = 1,ndvrrho
trace = trace + FindArho(j,j)
10 continue
return
end
c *****
subroutine setdvr
c *****
c
c set up for dvr-type quadrature.
c see setdvr for comments
c
include 'ad2d7.inc'

```



```

dimension wgs(nm,nm),r(nm,nm)
c
c <Qn|x|Qn+1> = sqrt(n+1)/(beta*sqrt(2))
c all other matrix elements are 0
c (x can be written in terms of raising/lowering operators..)
c see Atkins, Molecular QM p. 102
c
write(*,*) 'setdvr'
do 47 i = 1,ndvrrho
  r(i,2) = 0.0
  r(i,1) = sqrt(1.0*(i-1))
47 continue
c
c Diagonalize R. R is real, tridiagonal, so
c use eispack routine for banded matrices
c
call rsb(nm,ndvrrho,2,r,xgs,1,wgs,s1,s2,ier)
c
c eigenvectors are orthogonal, now normalize
c
do 70 i=1,ndvrrho
  xnorm = 0.0
  do 50 j = 1,ndvrrho
    xnorm = xnorm + wgs(j,i)*wgs(j,i)
50 continue
  xnorm = sqrt(xnorm)
  do 60 j = 1,ndvrrho
    t(j,i) = wgs(j,i)/xnorm
60 continue
70 continue
  sqrt2 = sqrt(2.0)
  do 80 i = 1,ndvrrho
    xgs(i) = xgs(i)/sqrt2
80 continue
  if(iverbose.eq.1) then
    write(*,*) 'eigenvalues : '
    do 90 i = 1, ndvrrho
      write(*,100) xgs(i)
90 continue
    call ShowArr(t,nm,ndvrrho)
  endif
100 format(f12.6)
  return
end
c *****
c subroutine findeigr(tempval,tempvec,vtop,neigen,iaflag)
c *****
c
c Uses eispack library routines to diagonalize A, (Hamiltonian)
c finding the eigenvalues and eigenvectors.
c The nshow lowest eigenvalues/eigenvectors are

```

```

c   saved in tempval(1..nshow), tempvec(1..nshow,1..ndvr),
c   respectively.
c
c   include 'ad2d7.inc'
c   dimension tempval(nm), tempvec(nm,nm)

c   if (iverbose.eq.1) then
c     write(*,*) 'findeigr'
c   endif
c   if (iverbose.eq.1) then
c     write(*,*) 'A matrix : '
c     call ShowArr(a,nm,ndvrrho)
c   endif
c   call rs(nm,ndvrrho,a,tempval,1,tempvec,s1,s1,ierr)
c   if (ierr.ne.0) then
c     write(*,1000) ierr
c     stop
1000  format('ERROR # ',i3,' in diagonalization in findeigr')
c   endif

c
c   wavefunctions are ok, so normalize them
c
c
1030  format('iaflag: ',i7)
c   if (iaflag.eq.1) then
c
c     anion - only return wavefunction w/ quantum # neigen
c
c     i = neigen
c     xnorm = 0.0
c     do 25 j = 1, ndvrrho
c       xnorm = xnorm + tempvec(j,i)*tempvec(j,i)
25    continue
c     xnorm = sqrt(xnorm)
c     do 28 j = 1, ndvrrho
c       tempvec(j,i) = tempvec(j,i)/xnorm
28    continue
c   endif
c   if (iaflag.eq.0) then
c
c     neutral -- find number of 'bound' states (states
c     with energies below vtop) and return this number
c     (neigen) and eigenvalues/vectors.
c
c     write(*,1020) (tempval(i)*harev, i = 1,4)
1020  format('eigenvalues: ',4f15.4)
c     i = 0
29    i = i + 1
c     if ((tempval(i) .le. vtop).and.(i .lt. ndvrrho)) goto 29
c     neigen = i - 1
c     write(*,1010) neigen
1010  format(i3,' bound states')

```

```

do 50 i = 1, neigen
  xnorm = 0.0
  do 30 j = 1, ndvrrho
    xnorm = xnorm + tempvec(j,i)*tempvec(j,i)
30  continue
  xnorm = sqrt(xnorm)
  do 40 j = 1, ndvrrho
    tempvec(j,i) = tempvec(j,i)/xnorm
40  continue
50  continue
endif
write(*,1100) neigen
1100 format('at end, neigen = ',i4)
return
end

```

1.4.8 dofcf.f

```

c
c File : dofcf.f
c
c Find FCFs of transitions in photoelectron spectrum.
c Written by RBM.
c Last modification RBM 9/91
c
c Includes :
c   dofcf
c   doweight
c   zover
c   rover
c   findwf
c   dorw
c   testover
c *****
c   subroutine dofcf(iad,issmax,wfx)
c *****
c
c Finds FCFs from v=0 of anion to all 1..issmax rho levels
c supported by adiabatic curve iad.
c
c include 'ad2d7.inc'
c
c common/integ/weight
c
c dimension cstar(maxrpts), wfx(nm,maxrpts,maxirho)
c dimension dstar(maxrpts),weight(maxrpts)

```

```

call zover(iad,cstar)
sumrtot = 0.0
do 90 iss = 1, issmax
  sumr = 0.0
  call rover(iss,wfx,dstar)
  sum = 0.0
  do 80 k = 1, nrhopts
c    if (iss .eq. 1) write(*,1100) k, iad, cstar(k)
1100  format('pt # ',i3,' C*(',i3,') = ',f9.6)
      sum = sum + weight(k)*cstar(k)*dstar(k)
      sumr = sumr + weight(k)*dstar(k)
c    write(*,1000) iss,k,weight(k),cstar(k),dstar(k)
1000  format(i3,' ',i3,' ',f5.3,' ',f10.7,' ',f10.7)
80    continue
      fcf(iad,iss) = sum*sum
      sumrtot = sumrtot + sumr*sumr
c    write(*,1200) iad, iss, sumr*sumr
1200  format('Curve # ',i3,' ss # ',i3,' fcf(r) = ',f9.6)
c    if (sum .gt. 0.0) write(*,*) iad, iss, sum
90    continue
c    write(*,*) 'Tot fcf(r) = ',sumrtot
      return
      end
c *****
      subroutine doweight(itype)
c *****
c
c    Find weighing factors for
c    Trapezoid rule integration (itype = 1)
c    Simpson's rule integration (itype = 2)
c
      include 'ad2d7.inc'

      common /integ/weight(maxrpts)

      rstep = (rhomax - rhomin)/(nrhopts - 1)
      if (itype .eq. 1) then
        weight(1) = rstep/2.0
        weight(nrhopts) = rstep/2.0
        do 10 i = 2, nrhopts-1
10          weight(i) = rstep
        endif
      if (itype .eq. 2) then
        nr = nrhopts
        weight(1) = rstep*17.0/48.0
        weight(2) = rstep*59.0/48.0
        weight(3) = rstep*43.0/48.0
        weight(4) = rstep*49.0/48.0
        weight(nr) = weight(1)
        weight(nr-1) = weight(2)
        weight(nr-2) = weight(3)

```

```

        weight(nr-3) = weight(4)
        do 20 i = 5, nr-4
20      weight(i) = rstep
        endif
c      write(*,*) 'WEIGHTS'
c      do 40 i = 1, nrhopts
c40    write(*,1000) i, weight(i)
1000  format(i3,' : ', f5.3)
        return
        end
c *****
        subroutine zover(iad,cstar)
c *****
c
c      Calculate asym. stretch integral over z,
c      at fixed rho, for all values of rho.
c
        include 'ad2d7.inc'
        dimension cstar(maxrpts)

        do 20 k = 1, nrhopts
            sum = 0.0
            do 10 i = 1, ndvr
                sum = sum + veca(k,i)*vec(iad,k,i)
10      continue
            cstar(k) = abs(sum)
            write(*,1000) k, sum
1000  format('——— ',i4,' : ',f9.5)
20      continue
        return
        end
c *****
        subroutine rover(iss,wfx,dstar)
c *****
c
c      Calculate product of sym. stretch neutral and anion
c      wavefunctions at every value of rho.
c
        include 'ad2d7.inc'
        common /rinteg/rweight(maxirho)

        dimension dstar(maxrpts),wfx(nm,maxrpts,maxirho)

        dstar = 0.0
        do 30 k = 1, nrhopts
            do 25 i = 1, nirho
c
c      find anion wavefunction da at current value of r.
c
                da = 0.0
                do 10 j = 1, ndvrrho

```

```

    da = da + rveca(j)*wfx(j,k,i)
10  continue
c
c  find neutral wavefunction dn at current value of r.
c
    dn = 0.0
    do 20 j = 1, ndvrrho
        dn = dn + rvec(j,iss)*wfx(j,k,i)
20  continue
    dstar(k) = dstar(k) + da * dn * rweight(i)
25  continue
30  continue
c  write(*,*) 'DSTAR'
c  do 40 i = 1,nrhopts
c40  write(*,1000) i, dstar(i)
1000 format(i3,' ',f10.6)
    return
    end

c *****
c  subroutine findwf(wfx)
c *****
c
c  Find value of wavefunctions.
c  ie, calculate Hermite polynomials explicitly -
c  do this using recursion relation in
c  Handbook of Mathematical Functions
c  Abramowitz + Stegun Sec. 22.7
c
c  include 'ad2d7.inc'

    dimension wfx(nm,maxrpts,maxirho),xnorm(nm)
    dimension xval(maxrpts,maxirho)

    dx = (rhomax - rhomin)/(nrhopts - 1)
    dxi = dx/(nirho-1)
c  divide by nirho, not (nirho-1)
    do 7 j = 1, nirho
        do 5 i = 1, nrhopts
            5  xval(i,j) = rhomin + (i-1)*dx + (j-1)*dxi - dx/2.0 - re
            7  continue
c
c  first set h0 and h1
c
c
    do 15 j = 1, nirho
        do 10 i = 1,nrhopts
            wfx(1,i,j) = 1.0
            wfx(2,i,j) = 2.0*beta*xval(i,j)
10  continue
15  continue

```

```

c
c   Now get the rest of the hermite polynomials
c   recursively, using
c        $h[n+1] = 2*y*h[n] - 2*n*h[n-1]$ ,
c   where  $y = \text{beta}*(x_{\text{min}} + i*x_h - \text{re})$ 
c
c   do 30 k = 3,ndvrrho
c       do 25 j = 1, nirho
c           do 20 i = 1,nrhopts
c               wfx(k,i,j) = 2.0*beta*xval(i,j)*
1           wfx(k-1,i,j) - 2.0*(k-2)*wfx(k-2,i,j)
20 continue
25 continue
30 continue
c
c   Include normalization and  $\exp(-y*y/2)$ 
c
c   xnorm(1) = sqrt(beta/sqrt(pi))
c   do 40 k = 2,ndvrrho
c       xnorm(k) = xnorm(k-1)/sqrt(2.0*(k-1))
40 continue
c
c   do 65 j = 1, nirho
c       do 60 i = 1,nrhopts
c           y = beta*xval(i,j)
c           yexp = exp(-y*y/2.0)
c           do 50 k = 1,ndvrrho
c               wtx(k,i,j) = wfx(k,i,j) * xnorm(k) * yexp
50 continue
60 continue
65 continue
c
c   return
c   end
c *****
c   subroutine dorw
c *****
c   include 'ad2d7.inc'
c   common /rinteg/rweight(maxirho)
c
c   rweight(1) = 0.5/(nirho-1)
c   rweight(nirho) = rweight(1)
c   do 10 i = 2, nirho-1
10  rweight(i) = 1.0/(nirho-1)
c
c   return
c   end
c *****
c   subroutine testover(wfint,wfx)
c *****
c

```

```

c Test overlap integrals along rho -
c All of these should give 1. Numbers smaller than 1
c usually mean that the limits of integration are too narrow
c (as integral is nominally from -inf to +inf)
c Numbers > 1 usually mean that the integration step size is
c too small. Current parameters give 1 up to the 70th basis
c function or so, when it becomes > 1.

```

```

c
include 'ad2d7.inc'

dimension wfint(nm),wfx(nm,maxrpts,maxirho)

common /integ/weight(maxrpts)
common /rinteg/rweight(maxirho)

do 20 k = 1, ndvrrho
sum = 0.0
do 15 j = 1, nirho
do 10 i = 1, nrhopts
sum = sum + wfx(k,i,j)*wfx(k,i,j)*weight(i)*rweight(j)
10 continue
15 continue
wfint(k) = sum
20 continue

return
end

```

1.4.9 pot7.f

```

c -----
c file pot7.f
c
c FORTRAN version of pot6.p (PASCAL). (written by RBM 1989)
c does  $x^2/x^4/x^6$  potential
c Application : fitted surface for Br + HBr reaction.
c
c Converted by RBM 4/9/91
c
c Includes :
c pot6
c Initpot
c spline
c splint
c
c *****

```



```

function pot6(rq1,rq2)
c *****
c
c Given r1 (r Br - H) and r2 (r H - Br) in angstroms, returns
c the potential for that (collinear) arrangement in hartree
c using x^2/x^4/x^6 potential
c
c Call Initpot (once) before calling pot6
c
  implicit real (a-h,o-z)
  parameter(npn = 21)
c   npn is number of nodes for cubic spline interpolation
  common/splpot/pX1(npn),pal(npn),pA(npn),pB(npn),pR(npn),px4(npn),
$   pX12(npn),pal2(npn),pA2(npn),pB2(npn),px42(npn),
$   pR2(npn),pAA(npn),pXX1(npn),pAA2(npn),pXX12(npn)

  real curX1, cural, curA, curB, Rxx, curX, curx4
  real x2, x12, pola, polb, polc

  Rxx = rq1 + rq2
c   find R Br-Br

  if ((Rxx .le. pR(npn)) .and. (Rxx .ge. pR(1))) then
    if (Rxx .le. pR(4)) then
      curX1 = 0.0
      curA = 0.0
    else
      call splint(pR2,pXX1,pXX12,npn-3,Rxx,curX1)
      call splint(pR2,pAA,pAA2,npn-3,Rxx,curA)
    endif
    call splint(pR,pal,pal2,npn,Rxx,cural)
    call splint(pR,pB,pB2,npn,Rxx,curB)
    call splint(pR,px4,px42,npn,Rxx,curx4)

    curX = abs(rq1 - rq2)/sqrt(2.0)
    if (curX .lt. curX1) then
      x12 = curX1 * curX1
      pola = 3*curA/(x12*x12) - curB/(4*x12)
      polb = curB/8 + pola*x12
      polc = (curB - 8*pola*x12)/(24*x12*x12)
      x2 = curX * curX
      pot6 = cural + curA - polb*x2 + pola*x2*x2 + polc*x2**3
    else
      pot6 = cural + 0.5*curB*(curX - curX1)**2 +
$       curx4*(curX-curX1)**4
    endif
    pot6 = pot6/96.485/27.211608
  else
    pot6 = 0.0
  endif
  return

```

```

end
c *****
  subroutine Initpot
c *****
c
c   Call before calling pot6 !!!
c   Reads parameters for fitted potential and
c   initializes cubic splines.
c
  implicit real (a-h,o-z)
  parameter(npn = 21)
c   npn is number of nodes for cubic spline interpolation
  common/splpot/pX1(npn),pal(npn),pA(npn),pB(npn),pR(npn),px4(npn),
$ pX12(npn),pal2(npn),pA2(npn),pB2(npn),px42(npn),
$ pR2(npn),pAA(npn),pXX1(npn),pAA2(npn),pXX12(npn)

  real py1, pyn

  write(*,*) 'Reading potential parameters'

  open(2,file = 'pot.par')
  read(2,*) n
  do 10 i = 1, n
10   read(2,*) pR(i),pX1(i),pal(i),pA(i),pB(i),px4(i)

  py1 = 1.0e40
  pyn = py1
c   use 'natural' splines - 2nd deriv = 0 at endpoints

  call spline(pR,pX1,npn,py1,pyn,pX12)
  call spline(pR,pal,npn,py1,pyn,pal2)
  call spline(pR,pA,npn,py1,pyn,pA2)
  call spline(pR,pB,npn,py1,pyn,pB2)
  call spline(pR,px4,npn,py1,pyn,px42)

  do 20 i = 1, npn-3
    pAA(i) = pA(i+3)
    pXX1(i) = pX1(i+3)
    pR2(i) = pR(i+3)
20  continue

  call spline(pR2,pAA,npn-3,py1,pyn,pAA2)
  call spline(pR2,pXX1,npn-3,py1,pyn,pXX12)
  return
  end
c *****
  subroutine spline(x,y, n, yp1,ynpn,y2)
c *****
c
c   procedure translated to FORTRAN from
c   spline.p (Pascal)

```

```

c   Cubic spline interpolation.
c   From Numerical Recipes
c
parameter(npn = 21)
implicit real (a-h,o-z)
real x(npn),y(npn),y2(npn),yp1,ynpn
integer n

real p,qn,sig,un
real u(npn)

if (yp1 .gt. 0.99e30) then
  y2(1) = 0.0
  u(1) = 0.0
else
  y2(1) = -0.5
  u(1) = (3.0/(x(2)-x(1)))*((y(2)-y(1))/(x(2)-x(1))-yp1)
endif

do 10 i = 2, n-1
  sig = (x(i)-x(i-1))/(x(i+1)-x(i-1))
  p = sig*y2(i-1)+2.0
  y2(i) = (sig-1.0)/p
  u(i) = (y(i+1)-y(i))/(x(i+1)-x(i))
  $   -(y(i)-y(i-1))/(x(i)-x(i-1))
10  u(i) = (6.0*u(i)/(x(i+1)-x(i-1))-sig*u(i-1))/p
   continue

if (ynpn .gt. 0.99e30) then
  qn = 0.0
  un = 0.0
else
  qn = 0.5
  un = (3.0/(x(n)-x(n-1)))*(ynpn-(y(n)-y(n-1))/(x(n)-x(n-1)))
endif

y2(n) = (un-qn*u(n-1))/(qn*y2(n-1)+1.0)

do 20 k = n-1, 1, -1
20  y2(k) = y2(k)*y2(k+1)+u(k)

return
end

```

```

c *****
c   subroutine splint(xa,ya,y2a,n,x,y)
c *****
c
c   FORTRAN version of subroutine splint
c   Pascal version is splint.p
c   Set up cubic splines for interpolation by spline.

```

```

c   From Numerical Recipes
c
implicit real (a-h,o-z)
parameter(npn = 21)
real xa(npn),ya(npn),y2a(npn),x,y

real h,b,a

klo = 1
khi = n
10  if ((khi-klo) .gt. 1) then
    k = (khi+klo)/2
    if (xa(k) .gt. x) then
        khi = k
    else
        klo = k
    endif
    goto 10
endif

h = xa(khi)-xa(klo)
if (h .eq. 0.0) then
    write(*,*) 'pause in routine splint'
    write(*,*) ' ... bad xa input'
    stop
endif

a = (xa(khi)-x)/h
b = (x-xa(klo))/h
y = a*ya(klo) + b*ya(khi) +
$   ((a**3-a)*y2a(klo) + (b**3-b)*y2a(khi))*(h*h)/6.0
return
end

```

1.4.10 matrix.f

```

c -----
c   Matrix.f : matrix subroutines:
c
c   Written by RBM 1989
c
c   Includes :
c     matmult : maultiply matrices
c     transpose : find the transpose of a matrix
c     showarr1, showarr2 : print out a matrix
c     showarrs : print out a symmetric matrix

```

```

c   zero : set all elements of a matrix to be 0
c
c   Note: The Cray has more efficient subroutines for
c   transpose, matmult, and zero, but these are used anyhow.
c   Showarrs has a nice way of printing out a symmetric matrix.
c

```

```

subroutine matmult(ndim,a,nra,nca,b,nrb,ncb,c,nrc,ncc,ierr)

```

```

c
c   Finds C = AB, where A and B are matrices.
c   all matrices are dimensioned to ndim x ndim
c   The actual size of A is nra x nca, B is
c   nrb x ncb, etc.
c   Note that nca must be equal to nrb, or the matrices
c   cannot be multiplied. If they are not equal, the
c   routine will exit with ierr = 1. If they are
c   equal, the routine will exit with ierr = 0.
c

```

```

implicit real (a-h,o-z)
dimension a(ndim,ndim), b(ndim,ndim), c(ndim,ndim)

```

```

ierr = 0
if (nca.ne.nrb) then
  ierr = 1
  goto 999
endif

```

```

do 30 ir = 1, nra
  do 20 icb = 1, ncb
    sum = 0.0
    do 10 ic = 1, nca
      sum = sum + a(ic,ir)*b(icb,ic)
10    continue
      c(icb,ir) = sum
20    continue
30  continue
  nrc = nra
  ncc = ncb
999 return
end

```

```

c
c   subroutine transpos(ndim,a,nra,nca,at,nrat,ncat)

```

```

c
c   Finds At = A (transpose)
c
implicit real (a-n,o-z)
dimension a(ndim,ndim), at(ndim,ndim)

```

```

do 20 ir = 1, nra
  do 10 ic = 1, nca

```

```

    at(ir,ic) = a(ic,ir)
10  continue
20  continue
    nrat = nca
    ncat = nra
    return
    end
c -----
c  subroutine zero(ndim,a,nra,nca)
c
c  zeroes the matrix A
c
c  dimension a(ndim,ndim)
c
c  do 20 ir = 1,nra
c    do 10 ic = 1,nca
c      a(ic,ir) = 0.0
10  continue
20  continue
    return
    end
c -----
c  subroutine showarr1(ndim,a,nra,nca)
c
c  displays the matrix A
c
c  implicit real (a-h,o-z)
c  dimension a(ndim,ndim)
c
c  do 10 ir = 1, nra
c    write(*,100) (a(ic,ir), ic = 1,nca)
100  format(15(e12.6,1x),/)
10  continue
    return
    end
c -----
c  subroutine showarr2(ndim,a,nra,nca)
c
c
c
c
c  implicit real (a-h,o-z)
c  dimension a(ndim,ndim)
c
c  do 30 ig = 1,nca/5
c    write(*,150) (ic+5*(ig-1), ic=1,5)
150  format(5(8x,i3,3x))
c    do 20 ir = 1,nra
c      write(*,100) ir,(a(ic+5*(ig-1),ir), ic = 1,5)
100  format(i3,2x,5(e12.6,2x))
20  continue
30  continue

```

```

ileft = nca - 5*(nca/5)
if (ileft.gt.0) then
  write(*,150) (ic+5*(nca/5), ic=1,ileft)
  do 40 ir = 1, nra
    write(*,100) ir,(a(ic+5*(nca/5),ir), ic = 1,ileft)
40  continue
endif
return
end

c -----
c subroutine showarrs(ndim,a,nra,nca)
c
c Display the symmetric matrix A.
c Nice format - only shows upper right triangle.
c
implicit real (a-h,o-z)
dimension a(ndim,ndim)

do 30 ig = 1,nca/5
  write(*,150) (ic+5*(ig-1), ic=1,5)
150  format(5(8x,i3,3x))
  do 20 ir = 5*(ig-1)+1,nra
    imax = ir - 5*(ig-1)
    if (imax.gt.5) imax=5
    write(*,100) ir,(a(ic+5*(ig-1),ir), ic = 1,imax)
100  format(i3,2x,5(e12.6,2x))
20  continue
30  continue
ileft = nca - 5*(nca/5)
if (ileft.gt.0) then
  write(*,150) (ic+5*(nca/5), ic=1,ileft)
  do 40 ir = 5*(nca/5)+1, nra
    imax = ir - 5*(nca/5)
    write(*,100) ir,(a(ic+5*(nca/5),ir), ic = 1,imax)
40  continue
endif
return
end

```

2. The Three-Dimensional Simulation Code ad3d5

2.1 Introduction

The ad3d5 program is used for three-dimensional adiabatic simulations of XHY^- photoelectron spectra, where photodetachment of XHY^- accesses the transition state region of the $X + HY \rightarrow XH + Y$ reaction. Some of the computational considerations behind the method are described in Chapter 3, while the application of the method to the photoelectron spectra of IHI^- , IDI^- , $BrHI^-$, $BrHBr^-$ and $BrDBr^-$ is discussed in Chapter 4. Three-dimensional adiabatic simulations are carried out in a fashion very similar to the collinear simulations. The programs are similar, as are the input and output files, and some of the parameters used for the collinear calculation are transferrable to the three-dimensional calculation. Collinear adiabatic curves take ~10 s CPU time to calculate on a Cray XMP, while three-dimensional adiabatic curves take about 300 s, so collinear calculations *should always* be carried out before attempting three-dimensional simulations on a new system. Also, the results of collinear and three-dimensional simulations are fairly similar, so the collinear calculation serves as a coarse check on the three-dimensional simulation. One significant difference between the collinear and three dimensional calculations is that non-adiabatic effects can be approximately included in the collinear calculation via the DIVAH correction. The three-

dimensional code does not include a similar correction, and nonadiabatic effects are generally much more pronounced in three dimensions than in two. Thus, the three-dimensional program should only be used for Heavy + Light-Heavy systems, as otherwise the adiabatic approximation is poor and the results will be inaccurate.

2.2 Running the ad3d5 Program

2.2.1 The ad3d.par Input File

Ad3d5 requires one input file (two, if a fitted potential is to be used) and some keyboard input. As with the collinear code, three types of output file are generated. The ad3d.par input file is very similar to the ad2d.par input file. A sample file is given below (note that line numbers are not part of the file, but are included for ease of reference):

```
1   brhi
2   .d
3   126.9045, 1.00728, 78.918
4   4
5   0.1609, 308.4, 17.51, 0.055
6   0.1414, 378.2, 18.10, 0.186
7   0.2469, 177.0, 18.76, 0.22
```

The first two lines define the names of the output files. There are three sets of output files: the first one is a single file (called brhi.d in this case) that will contain the eigenvalues obtained by solving the Schrodinger Equation for hydrogen atom motion at every value of R_{xy} . Unlike the

collinear calculation, this file *does not* contain the wavefunctions for hydrogen motion, as saving the wavefunctions would require a prohibitive amount of storage (some 2-5 Mbytes, depending on the system). Instead, the overlap between the anion $v_2 = 0, v_3 = 0$ wavefunction and the hydrogen motion wavefunctions of the neutral is calculated and stored at every value of R_{XY} . The second set consists of one file for each *ten* adiabatic curves, $brhiN.dv$, where N is 1 for the first 10 adiabatic curves, 2 for the next ten, etc. These files are used for plotting the adiabatic curves using `plot2d` or an equivalent plotting program. The file has 11 columns: R_{XY} (in Å) and ten adiabatic eigenvalues (in eV) at R_{XY} . The third output file is called `fort.7` and contains the "stick" photoelectron spectrum to be read by the `sim1` convolution program. The third line of `ad3d.par` contains the masses of the three atoms (in amu) in the order X, H, Y. Row 4 specifies the type of potential to be used: LEPS (4) or a fitted potential (3). If a fitted potential is to be used, the file `pot.par`, which defines the fitted potential, is required. The format of the `pot.par` file is given in sec. 1.2.3, and the extension of the (collinear) fitted potential to three dimensions is described in Ch. 4, sec. 3.6. Lines 5-7 are the LEPS parameters. The LEPS parameters define the entire potential if a LEPS potential is to be used and define the "reference potential" if a fitted potential is to be used. Line 5 refers to the diatom HX, line 6 is for HY and line 7 is for XY. The first entry is r_e in nm, the second is D_e in kJ/mol, the third is β in nm^{-1} , and the fourth is the Sato parameter.

2.2.2 Keyboard Inputs

Various parameters determining the type of calculation and anion potential are read from the keyboard. Usually, one creates a file containing these parameters (ie, inbrhi) and then executes the program using this file for keyboard input

```
ad3d5 < inbrhi
```

As the program generates quite a lot of output to the screen, it is useful to also send the output to a file:

```
ad3d5 < inbrhi > out
```

A typical keyboard input file is shown below. Again, line numbers are not part of the file, but are used for reference only. The comments can be part of the file, as the program only reads the first entry in each line.

```
1      1          ; 1= do ad. curves, 2= sim. spectrum, 3 = 1 and 2
2      100        ; nshow
3      3.1        ; Rac min (A)
4      4.7        ; Rac max (A)
5      40         ; nrhopts
6      0          ; iverbose
7      800.0      ; r frequency for basis set
8      190.0     ; z frequency for basis set
9      -0.4       ; re (z) for basis (A)
10     200.0     ; rho frequency
11     90         ; number of basis fcns in rho
12     8          ; number of segments between points for rho integral
13     7          ; number of different basis sets to use for rho integral
14     1276.3     ; anion a.s. freq
15     178.1     ; anion a. s. anharmonicity
16     100.0     ; anion s.s. freq
17     700.0     ; anion bend freq
18     -0.39     ; Re (z) in anion (A)
```

```

19  3.88      ; R X-Y in anion
20   2        ; idvr (1 = gaussian quad; 2 = DVR)
21  18        ; max qn in r
22  78        ; max qn in z
23   0        ; 0 = no symmetry in z, 1 = yes
24   1        ; 1 = even fcns only; 2 = odds only

```

Line 1 selects the type of calculation to be done. A value of 1 means that *only* adiabatic curves should be calculated. The adiabatic result file `brhi.g` and the adiabatic curves `brhiN.gv` (see section 2.2.1) will be generated. If a value of 2 is used, the adiabatic result file is read and eigenvalues are calculated on each adiabatic curve and their overlap with the anion is computed to generate a "stick" photoelectron spectrum. A value of 3 will generate a full simulation - it is equivalent to running the code with option 1, then again with option 2. Running the two parts of the code separately is useful as it allows one to obtain a coarse simulation, then refine it (see line 13, below), or to change the anion potential or equilibrium geometry along the R_{XY} (symmetric stretch) coordinate and see how it affects the spectrum. If the anion asymmetric stretch or bend potential is changed, then part 1 of the calculation must be rerun. The same basis *must* be used when the full adiabatic output file is read as when it was generated.

Line 2 is the number of adiabatic curves to be calculated, up to a maximum of 100. Lines 3 and 4 define the range of interhalogen distances R_{XY} (in Å) over which the adiabatic curves will be calculated. As a rule of thumb, the interhalogen distance in the anion should be at about the midpoint. The adiabatic curves flatten out at large R_{XY} . Do not extend the

calculation to excessively large values of R_{XY} , as good convergence in this region requires a large basis. Line 5 is nrhopts, the number of values of R_{XY} at which the Schrodinger Equation for hydrogen motion will be solved. A value of 30 or 40 is fine for simulations; you may wish to use 60 if the adiabatic curves are to be plotted. The maximum value is 100. Line 6 determines the amount of output generated by the program. Use a value of 0, unless you are debugging the program.

Lines 7-9, along with lines 21 and 22, define the basis set used to solve the Schrodinger Equation for hydrogen motion. It is extremely important that optimum values of these parameters be used - use of poor values will result in the calculation taking much longer than it should or in poor convergence of the results. Unlike in the collinear program, none of the basis set frequencies is optimized by the ad3d5 program. Line 7 is the basis set frequency along the bend (r) coordinate (in cm^{-1}), line 8 is the basis set frequency along the asymmetric stretch (z) coordinate (also in cm^{-1}). Note that the basis set frequency used along z is much smaller than ν_3 in the anion. A low frequency is used to allow hindered rotor states such as H-Y...X) to be described correctly. Line 9 is the value of z (in Å) where the basis set is centered. Use a value of 0 for symmetric systems; for asymmetric systems, using the equilibrium value of z in the negative ion (line 18) will ensure optimal convergence of states that have good Franck-

Condon overlap with the anion. A value of 0 can also be used for asymmetric systems, although it may require a slightly larger basis along z.

Lines 10-13 control the calculation of the heavy atom Schrodinger Equation on each adiabatic curve and the subsequent calculation of overlap integrals. Line 10 is the basis set frequency (in cm^{-1}) in R_{XY} , a value of 200 cm^{-1} is fine; line 11 is ndvrrho, the number of basis functions used along R_{XY} , up to a maximum of 100. Do not use values less than 60. Part of the Franck-Condon integral is calculated using quadrature (see Ch. 4, sec. 2); line 12 is the number of quadrature points that are taken between each pair of points in R_{XY} . A value of 8 is fine; the maximum value is 10. Line 13 is nrhoits, the number of basis sets that are used along R_{XY} . This is discussed in detail in Ch. 3, sec. 5.4. Use a value of 1 for preliminary calculations, and try 5 or so for later calculations. If nrhoits > 1, the Schrodinger Equation for halogen atom motion will be solved using nrhoits different basis sets, ranging in size from (ndvrrho-40) to ndvrrho in steps of $40/(\text{nrhoits}-1)$. There is no maximum value for this parameter, but a value of 5 is usually sufficient. Calculating halogen atom wavefunctions on each adiabatic curve and their overlap with the anion takes 2 min CPU time on a Cray XMP with nrhoits = 5.

Lines 14-19 describe the XHY^- anion. Lines 14 and 15 are the frequency and anharmonicity, respectively, for the antisymmetric stretch (or high frequency hydrogen stretch, in asymmetric systems), in cm^{-1} . Morse

potentials (unlike harmonic oscillators) do not have a axis of symmetry, so it is *extremely* important to correctly define the direction of the hydrogen stretching coordinate for asymmetric systems. The hydrogen stretching coordinate is mostly an HY stretch. Thus, the input file ad3d.par places the atoms in the order I - H - Br and the high frequency stretch is the H-Br stretch. Line 16 is the symmetric (or interhalogen) stretch frequency in cm^{-1} , and line 17 is the bend frequency in the anion, also in cm^{-1} . The symmetric stretch and bend are assumed to be harmonic. Lines 18 and 19 describe the geometry of XHY^- . Line 18 is $\frac{1}{2} (R_{\text{HY}} - R_{\text{HX}})$. Note that this is zero for symmetric systems and is negative for BrHI^- . Line 19 is the X - Y distance. Both values are in Å.

Line 20 is 1 if the Gaussian quadrature scheme (the two-dimensional analogue of the HEG quadrature) is to be used and 2 for the DVR quadrature scheme. Use 2, as the DVR is *much* faster than the Gaussian quadrature. If symmetry is to be used (see line 23), you *must* use the DVR.

Lines 21 and 22 define the size of the basis used to solve the Schrodinger equation for hydrogen motion at each value of R_{XY} . Line 21 is the maximum quantum number (and size of the basis N_r) in r. Line 22 is the maximum quantum number in z. For asymmetric systems, this is also the size of the basis; for symmetric systems, the basis N_z is half this size. The overall basis set size is $N_r N_z$. This should be about 1200 for symmetric systems and about 1400 for asymmetric systems. Use of basis sets larger

than 1410 requires modifying the file ad3d.inc and recompiling the program (see sec. 2.3). Do not use overly large basis sets, as the time required to calculate adiabatic curves is proportional to $(N_x N_y)^3$. As was noted in Ch. 4, sec. 2, the basis set used is poorly suited to calculating asymptotic states at large R_{xy} . The best way to test for convergence of the basis is to calculate a few points at large R_{xy} (see lines 3-5) and vary the size and frequency of the basis set until the eigenvalues no longer change. If the calculation is converged at the largest values of R_{xy} , it will be converged at all (reasonable) smaller values.

Line 23 determines if symmetry along the hydrogen stretching coordinate is to be used in calculating the hydrogen motion wavefunctions (1 for symmetry, 0 for no symmetry). If the anion is symmetric, use symmetry - it will greatly speed the calculation. If symmetry is used, basis functions should be centered at 0.0 (line 9). Line 24 is used if symmetry is employed. A value of 1 indicates that functions of even (*gerade*) parity are to be calculated, use a value of 2 to calculate odd (*ungerade*) functions. Only even functions have Franck-Condon overlap with the $v_3 = 0$ wavefunction of XHX^- . If odd functions are calculated, their overlap with the $v_3 = 1$ state of XHX^- is calculated.

2.2.3 Using a Fitted Potential

A completely empirical, fitted potential can be used for the three-dimensional simulations. The construction and application of this potential to the Br + HBr system is discussed in Ch. 4, sec. 3.6. A potential type of 3 is chosen and collinear portion of the fitted potential is read from the file pot.par. This file is discussed in sec. 1.2.3. The potential was extended for the three-dimensional calculations; the additional parameters are given in Table 4-2. The three-dimensional calculation also requires a "reference" LEPS potential. This is defined in lines 5-7 of the ad3d.par input file.

2.3 How the ad3d5 Program Works

This section is intended for the user who wishes to modify the ad3d5 program; casual users who wish to use the program as a "black box" should refer to section 2.2. The mathematical framework behind the DVR method and our implementation of the adiabatic method are discussed in detail in Ch. 3. The principles behind the three-dimensional adiabatic method are discussed in Ch. 4. The three-dimensional adiabatic program is very similar to the collinear program, so only differences between the two programs are explained in detail. The method is implemented as follows (subroutine names are given in italics):

1) The initial parameters for the calculation are read. Physical constants are defined (*setconst*). The anion potential and the basis set are read from the keyboard (subroutine *readin*); keyboard inputs are described in section 2.2.2. The LEPS subroutine is initialized (*initleps*, *dojsign*), and names for output files and the form of the neutral potential energy surface are read from the file *ad3d.par* (*initpars*). Section 2.2.1 describes the format of the *ad3d.par* input file. If a fitted potential is used, the fitted potential subroutines are initialized and the parameters for the fitted potential are read from the file *pot.par* (*initpot* and *splint*). This file is described in sections 1.2.3 and 2.2.3.

2) Adiabatic curves are calculated by solving the Schrodinger Equation for hydrogen atom motion at many values of R_{XY} , which is assumed to be proportional to the scattering coordinate. The overlap between the anion and neutral wavefunctions along the hydrogen motion coordinates is also determined at each value of R_{XY} , so the wavefunctions of hydrogen motion for the anion are calculated, though they are only calculated once, as the anion potential is assumed to be separable (ie, the potential for hydrogen motion is assumed to be independent of R_{XY}). First, the product basis is set up, using symmetry along z , if desired (*setbasis*). Then, the DVR is set up - for each of the coordinates (r and z), the position matrix \mathbf{X} is constructed in the appropriate basis and diagonalized to obtain

T_r and T_z (*setdur*). If the DVR is to be used, the matrices K_r and K_z are calculated.

At each value of R_{xy} the potential is found along z and r (*setv*) and the Hamiltonian matrix is constructed in the HEG quadrature (*finda* and *rvint*) or DVR. The Hamiltonian matrix is then diagonalized (*findeig*). Tailoring the diagonalization subroutine the problem to be solved saves a great deal of computer time and memory. The ~10 lowest eigenvalues and corresponding eigenvectors of a real, symmetric matrix of order ~1200 are required, so the eispack library path appropriate for finding < 10% of the eigenvalues and eigenvectors of a real, symmetric matrix is used (see Lecture Notes in Computer Science: 6. Matrix Eigensystem Routines - EISPACK Guide by B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler (Springer-Verlag, Berlin), 1976, sec. 2.1.13 and 2.2.8). As the Hamiltonian matrix in the DVR is sparse (approximately 93% of the elements are zero), an even more efficient path would be to use a subroutine specifically geared to solving sparse matrices. One such subroutine is *sspotrs*, available as part of the Cray math library on systems with Unicos 6.0 and above. Use of a sparse matrix solver may be necessary to obtain converged results on deuterated, asymmetric systems such as BrDI^- . The overlap along the hydrogen motion coordinates between the ground state anion wavefunction and each adiabatic wavefunction on

the neutral surface is also found. The adiabatic results file is then saved (*saveall*), as are the adiabatic curves (*savewf*).

The program can use several functional forms for the potential energy surface for the neutral, based in the *ipot* flag. The subroutine *V* calculates the Morse potential for the anions and calls the subroutines that generates the neutral potentials. If *ipot* = 4, a LEPS potential is used (*doQJsum*). A fitted potential, using a LEPS "reference potential" is used if *ipot* = 3. This potential is described in Ch. 4, sec. 3.6. The fitted potential is generated by subroutines *pot6* and *spline*. The *spline* and *splint* subroutines are from Chapter 3.3 of Numerical Recipes (Cambridge Univ. Press, Cambridge, 1986). The zero-point bend term used in this potential is calculated by subroutines *bn dh* and *depesdr*, which were provided by Prof. Joel Bowman (Emory Univ.). Modifying the program to use other functional forms for the potential energy surface thus involves modifying *poten* and including a subroutine to calculate the desired potential.

3) A "stick" photoelectron spectrum is generated by calculating halogen atom wavefunctions on each adiabatic curve and determining their overlap with the anion ground state. The user has the option of only generating the adiabatic curves (parts 1 and 2, above) and later running the program to simulate the spectrum (skipping part 2, above). In this case, the file containing the hydrogen motion wavefunctions and adiabatic curves is read by *readall*.

Subroutines used to solve the Schrodinger Equation along R_{XY} are very similar to those used in the collinear program *sd2d7*. The HEG quadrature is always used to solve the Schrodinger Equation along R_{XY} , even if the DVR was used to solve the Schrodinger Equation along z and r . Thus, *setdvr* sets up the matrix T used in the HEG quadrature, *setvOrho* sets up the reference potential corresponding to the basis set in R_{XY} , *setva* sets up the anion potential along R_{XY} (for the neutral, *setvrho* sets up the potential by linearly interpolating along each adiabatic curve using *FindV*), *setArho* constructs the Hamiltonian matrix along R_{XY} using *Findarho*, and eigenvalues and corresponding eigenvectors are found using *findeigr*. The Schrodinger Equation is solved using a 200 cm^{-1} basis set frequency. Subroutines (*optimrho*, *findnupr*, *findtrar*) to optimize the basis set frequency are included, but not used, as optimizing the basis set frequency does not work well on repulsive potentials.

Halogen atom wavefunctions are calculated on each adiabatic curve. First, the adiabatic curve is analyzed as to whether it is purely repulsive or contains a minimum, and how deep this minimum is (*findmin*). This is carried out so that states can be classified as bound (within the adiabatic approximation) or scattering, but does not affect the results of the calculation. Subroutine *boundrho* then calculates halogen atom wavefunctions and eigenvalues on the adiabatic curve, using the same method as for the anion wavefunction in R_{XY} .

Franck-Condon factors are calculated for each halogen atom wavefunction on each adiabatic curve. This is fairly easy to do, within the adiabatic approximation. The Franck-Condon integral is Ch. 4, eq. 29. The integral is done by quadrature using either the trapezoid rule or Simpson's rule (the program is currently set up to use Simpson's rule, as this is more accurate). The integrals over r and z have already been done (see part 2, above), so *findwf* explicitly calculates the basis functions for the harmonic oscillator basis (the product of a Hermite polynomial and a Gaussian), so that *rover* can calculate the overlap integral along R_{XY} using a Simpson's Rule quadrature with weights provided by *dorw*. Subroutine *dofcf* calculates the Franck-Condon integral using an alternative extended Simpson's Rule quadrature (Numerical Recipes, eq. 4.1.14) (provided by *doweight*) and the overlaps along R_{XY} and along z and r . The Schrodinger Equation for hydrogen atom motion is solved at *ndvrrho* (typically 30-40) points in R_{XY} . Computing the integral Franck Condon integral using only *ndvrrho* points in the quadrature results in very inaccurate results. Thus, *nirho* (typically 5-10) quadrature points are used between each pair of points in ρ . Thus, the integral is usually carried out using a ~ 400 point quadrature, ensuring sufficient accuracy. Note that this requires the hydrogen motion wavefunctions to be interpolated between the points in R_{XY} at which the Schrodinger Equation for hydrogen motion is solved.

The cumulative Franck-Condon factors are kept and are printed out. The Franck-Condon factors should sum to 1. A smaller total usually indicates that there is some intensity due to transitions to adiabatic curves lying above the calculated curves. Transitions having Franck-Condon factors greater than 0.00005 are saved to the file fort.7 for plotting by the program sim1. The energy saved is $1.0 - (E_i + D_{e_{\max}})$, where E_i is the energy of each transition relative to three-atom dissociation and $D_{e_{\max}}$ is the dissociation energy of the most stable diatomic fragment.

Limitations on the maximum basis set size, number of adiabatic curves and number of points along R_{XY} can all be relaxed by changing the ad3d5.inc include file. If this file is modified, the ad3d5 program should be recompiled. This can be done using the UNIX make utility via

```
make -f makead3d5
```

2.4 Source Code Listing

2.4.1 makead3d5

```
ad3d5: ad3d5.o leps3d2.o pot6.o bndhbr.o findmin3d.o boundr3d.o dof3d.o matrix.o
      segldr -o ad3d5 ad3d5.o leps3d2.o pot6.o bndhbr.o findmin3d.o boundr3d.o dof3d.o
matrix.o
```

```
ad3d5.o: ad3d.inc
leps3d2.o: ad3d.inc
findmin3d.o: ad3d.inc
boundr3d.o: ad3d.inc
dof3d.o: ad3d.inc
```

2.4.2 ad3d.inc

```
c
c include file for ad3d5.f
c
c implicit real (a-h,o-z)
c parameter(nsmall = 81, nlarge = 1410, nad = 101, maxrpts=41)
c parameter(nrdvrmax =100,maxirho=10)
c
c nsmall = max quantum # along each coord (r, z)
c nlarge = max basis set along r,z (2D product basis)
c nad = max # ad. curves
c maxrpts = max # pts. in Rac
c nrdvrmax = max q # along Rac
c maxirho = max # pts between pairs of Rac points
c for overlap integral in Rac
c
c dimension indexr(nlarge),indexz(nlarge)
c
c common/units/pi,harev,evwn,hbar,a0,amu,emu,clight,harwn,amass
c common/rdvr/rgs(nsmall),tr(nsmall,nsmall),ndvrr
c common/zdvr/zgs(nsmall),tz(nsmall,nsmall),ndvrz
c common/debug/iverbose
c common/number/indexr,indexz,indexzp(nlarge),ndvr,nshow
c common/doV/Varray(2,nsmall,nsmall)
c common/def/eharmr,basiskr,betar,rer,eharmz,basiskz,betaz,rez,
$ eharrho,betarho,basiskrho,rerho
c common/sym/isym,ieven,ndvrz0
c common/pottype/ipot
```



```

common/step/istart,istep
common/bigR/Rac,Racmin,Racmax,nrhopts,delrac,rtoRxx
common/anion/isp,ankr,ankz,rveca(nrdvrmax),
$      veca(maxrpts,nlarge)
common/lepstuff/jsign(3,3),Rel(3),De(3),Bl(3),S(3),Te(3)
common/masses/aMa(3)
common/results/cs(nad,maxrpts),val(nad,maxrpts)
common/fcfs/fcf(nad,nrdvrmax),fcfrho(nad,nrdvrmax)
common/arrays/ar(nrdvrmax,nrdvrmax),ar2(nrdvrmax,nrdvrmax),
$      s1(nrdvrmax),s2(nrdvrmax)
common/savepot/v0(nrdvrmax), rvarray(nrdvrmax)
common/zcoord/zr(nrdvrmax)
common/rcoord/rvec(nrdvrmax,nrdvrmax)
common/gauss/xgs(nrdvrmax),t(nrdvrmax,nrdvrmax)
common/rhostuff/rhomin,rhomin,max,delrho,ndvrho,nirho
common/anionpot/anwe,anwexe,anweA,anwexeA,xalphaA,
$      xdeA,reza,iantype
character*6 molecule, ext
character*1 numbers(15)
common/filestr/numbers,molecule,ext
common/fileno/imollen,iextlen

```

2.4.3 ad3d5.f

```

PROGRAM AD3D5
c
c   DVR code for constructing adiabatic curves by
c   solving H-L-H vibrations at
c   fixed R(H-H). Includes (degenerate) bend and
c   asymmetric stretch. Assumes that angular motion
c   about the z axis has been removed, and that m
c   (the quantum number for this motion) is 0.
c
c   Solves the problem in cylindrical coordinates, where
c   z is the distance along the molecular axis (in the
c   collinear configuration) and r is the distance
c   perpendicular to this axis.
c
c   Two dimensional DVR is used to solve the problem.
c   The product basis consists of a 1D harmonic oscillator
c   along z and a 2D harmonic oscillator in r.
c
c   Written by RBM 12/9/90 (dvr2d1.f)
c   dvr2d2.f RBM 12/13/90 :
c     Can use symmetry in z coordinate.
c     Can do calculation in DVR or using Gaussian Quadrature.

```

```

c     Can solve FHF- as well as quartic oscillator.
c     dvr2d4.f RBM 12/90 :
c     Can also solve I + HI (or any other atom + diatom)
c     on LEPS surface at fixed R(I-I).
c     Does FCFs from v=0 of IHI-.
c     dvr2d41.f RBM 12/25/90 :
c     Change in calculation of FCFs
c     Finds trace of Hamiltonian matrix
c     dvr2d42.f RBM 1/2/91 :
c     Saves wavefunctions
c     scat3d1.f RBM 2/16/91 :
c     Solves problem at various values of rho to
c     construct adiabatic curves
c     scat3d2.f RBM 2/18/91 :
c     Solves for wavefunctions along scattering coord. and
c     does overlaps w/ anion wavefunction to construct
c     simulated photoelectron spectrum.
c     scat3d3.f RBM 2/23/91 :
c     Eliminated some unnecessary arrays, cleaned up output.
c     scat3d4.f RBM 3/3/91 :
c     Find a. s. and bend overlaps with anion at each value
c     of rho while finding adiabatic curves. Decreases
c     memory requirements by 2/3 and makes output file
c     about 500x smaller.
c     scat3d5.f RBM 4/11/91 :
c     Include information from fitted potential in doing
c     3D Br + HBr.
c
c     Includes :
c     main program segment
c     setconst
c     readin
c     setbasis
c     setdvr
c     Finda
c     rvint
c     setV
c     V
c     saveall
c     readall
c     savewf
c     ShowArr
c
c     include 'ad3d.inc'

c     dimension a(nlarge,nlarge)
c     dimension sl1(nlarge), sl2(nlarge), sl3(nlarge), sl4(nlarge)
c     dimension sl5(nlarge),sl6(nlarge),sl7(nlarge),sl8(nlarge)
c     dimension iv1(nlarge)
c     real kr(nsmall,nsmall), kz(nsmall,nsmall)
c     dimension ss1(nsmall), ss2(nsmall)

```

```

dimension epsr(nsmall,nsmall), epsz(nsmall,nsmall)
dimension ttrans(nsmall,nsmall),et(nsmall,nsmall)
dimension Veff(maxrpts),cstar(maxrpts)
dimension wfx(nrdvrmax,maxrpts,maxirho),wfint(nrdvrmax)
dimension tempval(nrdvrmax),tempvec(nrdvrmax,nrdvrmax)
dimension temprval(nrdvrmax), rfct(7000),rval(7000)
dimension totfcfad(nad)
dimension eval(nlarge),evec(nlarge,nad)
common/integ/weight(maxrpts)

write(*,*) '1 : Neutral and anion adiabatic curves only'
write(*,*)
$ '2 : Simulation of spectrum using existing ad. curves'
write(*,*) '3 : Both 1 and 2'
write(*,*) 'Pick one : '
read(*,*) isim
if ((isim .eq. 1) .or. (isim .eq. 3)) isp = 1
c
c   isp = 1 -> calculate anion wavefunction.
c   isp = 2 -> calculate neutral adiabatic curves.
c
ns = nsmall
nl = nlarge

call setconst
call readin

write(*,*) 'nshow = ',nshow

ifcf = 1

if (ipot .eq. 4) write(*,*) 'Using LEPS potential'
if (ipot .eq. 3) write(*,*) 'Using fitted Br + HBr potential !'
c
c   ipot = 4 for normal LEPS
c   ipot = 3 for LEPS + collinear Br + HBr fitted surface.
c

call initleps
if (ipot .eq. 3) call initpot
rmu = amu*aMa(2)*(aMa(1)+aMa(3))/(aMa(1) + aMa(2) + aMa(3))
rtoRxx = sqrt(aMa(1)*aMa(3)/(aMa(1) + aMa(3)))/aMa(2) + 0.5

write(*,*) 'rmu = ',rmu

write(*,*) 'Input iverbose : '
read(*,*) iverbose

write(*,*) 'Input basis function frequency in r (cm-1)'
read(*,*) basefrqr
write(*,*) 'Input basis function frequency in z (cm-1)'
read(*,*) basefrqz

```

```

write(*,*) 'Input re (basis) in z (A) (0 for XHX systems !!)'
read(*,*) rez
rez = rez/a0

write(*,*) 'Input basis function frequency in rho (cm-1)'
read(*,*) basefrho

write(*,*) 'Input number of basis functions in rho'
read(*,*) ndvrrho

write(*,*) 'Number of segments between point in rho integral'
read(*,*) nrho

write(*,*) 'Number of times rho overlaps are done',
$ ' with different basis'
read(*,*) nrhoits

if (isim .eq. 2) call readall(basefrqr,basefrqz)

betar = sqrt(2.0*pi*clight*basefrqr*rmu*amass
# /hbar/amu)*a0*1.0e-10

rer = 0.0

betaz = sqrt(2.0*pi*clight*basefrqz*rmu*amass
# /hbar/amu)*a0*1.0e-10

eharmr = (hbar*betar*betar/1.0e-20)/
# (2.0*pi*a0*a0*rmu*emu*harev*evwn*clight)
basiskr = (betar**4)/rmu

betar = betar*betar

eharmz = (hbar*betaz*betaz/1.0e-20)/
# (2.0*pi*a0*a0*rmu*emu*harev*evwn*clight)
basiskz = (betaz**4)/rmu

eharmrho = basefrho/harwn
betarho = sqrt(rmu*eharmrho)
basiskrho = rmu*eharmrho*eharmrho
rhomin = Racmin*rtoRyx
delrho = delRac * rtoRxx
rhomax = rhomin + (nrhopts-1)*delrho
rerho = rhomin + (nrhopts/2-1)*delrho

write(*,*) 'Input Anion a. s. frequency (cm-1)'
read(*,*) aasfrq
anwe = aasfrq
write(*,*) 'Input Anion a. s. anharmonicity (cm-1)'
read(*,*) anwexe
write(*,*) 'Input Anion s. s. frequency (cm-1)'

```

```

read(,*) assfrq
write(,*) 'Input Anion bend frequency (cm-1)'
read(,*) abendfrq
write(,*) 'Re (z) in anion (A)'
read(,*) reza
write(,*) 'Rac in anion (A)'
read(,*) arii

ankr = rmu* (abendfrq/harwn)**2
if (anwexe .eq. 0) then
c   harmonic oscillator
   ankz = rmu* (aasfrq/harwn)**2
   iantype = 1
else
c   Morse oscillator
   iantype = 2
   anweA = anwe/harwn
   anwexeA = anwexe/harwn
   xalphaA=sqrt(2.0*rmu*anwexeA)
   xdeA=anweA**2/(4.0*anwexeA)
endif

ankrho = rmu* (assfrq/harwn)**2
rerhoa = arii*rtoRxx/a0
reza = reza/a0

write(,*) 'Basis functions :'
write(,800) basefrho,basefrqr,basefrqz
write(,*) 'For anion :'
write(,800) assfrq,abendfrq,aasfrq
800  format('v1 = ',f8.2,' cm-1, v2 = ',f8.2,' cm-1, v3 = ',
$      f8.2,' cm-1')

write(,*) 'idvr :'
read(,*) idvr

call setbasis
call setdvr

8   if (isim .eq. 2) goto 205
   if (isp .eq. 1) then
     nrr = 1
   else
     nrr = nrhopts
   endif

   if (idvr .ge. 2) then
c
c   Do calculation by DVR quadrature.
c

```

```

    epsr = 0.0
    do 100 k = 1, ndvrr
        epsr(k,k) = eharmr*(2.0*(k - 1) + 1.0)

        call transpos(ns,tr,ndvrr,ndvrr,ttrans,ndvrr,ndvrr)
        call matmult(ns,epsr,ndvrr,ndvrr,
#         ttrans,ndvrr,ndvrr,et,ndvrr,ndvrr,ierr)
        call matmult(ns,tr,ndvrr,ndvrr,et,ndvrr,ndvrr,
#         kr,ndvrr,ndvrr,ierr)

        if(iverbose .eq. 1) then
            write(*,*) 'Kr'
            call ShowArr(kr,ns,ndvrr)
        endif

        epsz = 0.0
        ik = 0
        do 110 k = istart,ndvrr0,istep
            ik = ik + 1
            epsz(ik,ik) = eharmz*(k - 1 + 0.5)
110        continue

        call transpos(ns,tz,ndvrrz,ndvrrz,ttrans,ndvrrz,ndvrrz)
        call matmult(ns,epsz,ndvrrz,ndvrrz,
#         ttrans,ndvrrz,ndvrrz,et,ndvrrz,ndvrrz,ierr)
        call matmult(ns,tz,ndvrrz,ndvrrz,et,ndvrrz,ndvrrz,
#         kz,ndvrrz,ndvrrz,ierr)

        if(iverbose .eq. 1) then
            write(*,*) 'Tz'
            call ShowArr(tz,ns,ndvrrz)
            write(*,*) 'epsz'
            call ShowArr(epsz,ns,ndvrrz)
            write(*,*) 'Kz'
            call ShowArr(kz,ns,ndvrrz)
        endif
    endif

c
c   Loop over R (I-I) to get neutral adiabatic
c   curves if isp = 2;
c   Just solve for anion (one value of R (I-I)) if isp = 1.
c
    do 500 i = 1,nrr
        Rac = Racmin + (i-1)*delRac

        call setV

        if (idvr .ge. 2) goto 95
        write(*,*) 'A'
        do 20 k = 1,ndvr
            do 10 j = 1,k

```

```

    a(k,j) = Finda(indexr(k),indexr(j),indexz(k),indexz(j))
    a(j,k) = A(k,j)
10  continue
20  continue

    if (iverbose.ge.1) then
        write(*,*) 'Array A : '
        call ShowArr(a,nl,ndvr)
    endif

    goto 135

c
c  Do calculation by DVR quadrature.
c
95  A = 0.0
    do 130 ii = 1,ndvr
        ir = indexr(ii)
        iz = indexzp(ii)
        do 120 j = 1,ii
            jr = indexr(j)
            jz = indexzp(j)
            if (ir .eq. jr) a(ii,j) = a(ii,j) + kz(iz,jz)
            if (iz .eq. jz) a(ii,j) = a(ii,j) + kr(ir,jr)
            if ((ir .eq. jr) .and. (iz .eq. jz))
                #    a(ii,j) = a(ii,j) + Varray(isp,ir,jz)

                a(j,ii) = a(ii,j)
120  continue
130  continue

    if (iverbose .eq. 1) then
        write(*,*) 'A'
        call ShowArr(a,nl,ndvr)
    endif

    trace = 0.0
    do 133 k = 1,ndvr
133  trace = trace + a(k,k)
        write(*,*) 'Trace = ',trace

c
c  diagonalize Hamiltonian matrix A.
c  Only need < 10% of eigenvalues and corresponding
c  eigenvectors, so use appropriate EISPACK routine.
c
135  call tred1(nl,ndvr,a,sl1,sl2,sl3)
        eps1 = 0.0
        nlow = 1
        call tridib(ndvr,eps1,sl1,sl2,sl3,rlb,rub,nlow,nshow,eval,
$         iv1,ierr,sl4,sl5)
        call tinvt(nl,ndvr,sl1,sl2,sl3,nshow,eval,iv1,evec,ierr,
$         sl4,sl5,sl6,sl7,sl8)

```

```

call trbak1(nl,ndvr,a,sl2,nshow,evec)

c call rs(nl,ndvr,a,eval,1,evec,sl1,sl2,ierr)
c rs finds ALL eigenvalues and eigenvectors.
if (isp .eq. 1) then
do 24 k = 1, nrhopts
do 23 j = 1, ndvr
veca(k,j) = evec(j,1)
23 continue
24 continue
c
c assumes anion potential is separable in
c (rho) and (r,z). ie, wavefunctions for
c H atom motion are same, regardless of rho.
c
endif
if (ierr .ne. 0) goto 9999

if (isp .eq. 2) then
c
c save neutral eigenvalues (points on ad. curves)
c
do 5 j = 1, nshow
val(j,i) = eval(j)
5 continue
c
c Calculate a.s. and bend overlaps at this value
c of R (I-I), for each adiabatic curve. This avoids
c having to save eigenvectors (wavefunctions).
c
do 7 j = 1, nshow
sum = 0.0
do 6 k = 1, ndvr
sum = sum + veca(i,k)*evec(k,j)
6 continue
cs(j,i) = abs(sum)
7 continue

if (nshow .gt. 10) then
n = 10
else
n = nshow
endif

write(*,1010) i,rac*a0,(eval(j)*harev, j = 1,n)
1010 format(i3,' (' ,f7.3,' A)',10f7.3,2x)
endif

500 continue

if (isp .eq. 2) then

```



```

c
c Save overlaps and adiabatic curves,
c so that spectrum can be simulated later.
c
call saveall(basefrqr,basefrqz)
c
c save adiabatic curves for plotting on violet
c
call savewf

write(*,*) 'Results of DVR :'

do 201 i = 1,nshow
  if (i .gt. 1) then
    write(*,901) i,eval(i),eval(i)*hareV,
#     hareV*(eval(i)-eval(1))
    else
    write(*,900) i,eval(i),eval(i)*hareV
  endif

201 continue
endif

if (isp .eq. 1) then
  write(*,*) ' Anion a.s., bend eigenvalues'
  do 220 i = 1,5
220   write(*,1020) i, eval(i)*harwn
1020   format(i4,' ',f10.3,' cm-1')
  endif

if (isp .eq. 1) then
  isp = 2
  goto 8
endif

if (isim .eq. 1) goto 9998

c
c Simulate spectrum
c
c Solve for eigenvalues along rho for anion
c and for each neutral adiabatic curve, then
c simulate photoelectron spectrum.
c
205 call setdvrr
call setv0rho
call setva(ankrho,rerhoa,2)
call setArho
i = 0
j = 0
x = 500.0/evwn

```

```

call findeigr(tempval,tempvec,x,i,j)
iverbose = 0
do 28 i = 1, ndvrrho
  rveca(i) = tempvec(i,1)
28  continue
write(*,*) 'Anion rho eigenvalues : '
do 29 i = 1,3
  write(*,*) i,tempval(i)*harwn,' cm-1'
29  continue

call doweight(2)
call dorw
call findwf(wfx)
call testover(wfint,wfx)
if (iverbose .eq. 1) then
  write(*,*) 'Wavefunction overlaps : should all = 1'
  do 31 j = 1, ndvrrho
31    write(*,1090) j, wfint(j)
1090  format(i3,' ',f9.6)
  endif

if (De(1) .gt. De(2)) then
  diss = De(1)/96.485
else
  diss = De(2)/96.485
endif

totfcf = 0.0
nline = 0
do 25 j = 1, nshow
  call findmin(j,lmflag,imin,imax,vmin,vmax,vasy)
  if (lmflag.eq.0) then
    write(*,1100) j
1100  format('Curve # ',i2,' is purely repulsive')
    vmin = val(j,ndvrrho)
  else
    write(*,1110) j,vmin*harev,a0*(rhomin+(imin-1)*delrho),
    $  vmax*harev,a0*(rhomin+(imax-1)*delrho),val(j,nrhopts)*harev
1110  format('Curve # ',i2,' has a min of ',f9.4,' at rho = ',
    $  f7.4,' and a max of ',f9.4,' at rho = ',f7.4,
    $  ' and an asymptote of ',f9.4,' eV')
  endif
endif
c  re = rhomin + (nrhopts/2 - 1) * delrho
do 30 i = 1, nrhopts
  Veff(i) = val(j,i)
30  continue
totfcfad(j) = 0.0
nsubdel = 40/nrhoits
do 35 nit = 1, nrhoits
  nsub = (nit - 1)*nsubdel
  call boundrho(Veff,0,temprval,rvec,vmax,nbound,nsub)

```

```

write(*,1120) nbound
1120 format('There are ',i4,' bound levels with energies :')
c write(*,1130) (i,tempval(i)*harev, i=1,nbound)
1130 format(i3,' ',f12.6)

call dofcf(j,ndvrrho,wfx)
do 40 i = 1, ndvrrho
  if (fcrho(j,i) .ge. 0.005) then
    write(*,1140) j,i,tempval(i)*harev,fcrho(j,i),fcf(j,i)
1140 format(i3,' ',i3,' ',f8.5,' ',f5.3,' ',f8.6)
    endif
    if ((fcf(j,i) .ge. 0.00005) .and.
      $ (diss+tempval(i)*harev .lt. 2.0))then
      nline = nline + 1
      rval(nline) = tempval(i)*harev
      rfcf(nline) = fcf(j,i)
    endif
    if (nit .eq. 1) totfcfad(j) = totfcfad(j) + fcf(j,i)
40 continue
35 continue
totfcf = totfcf + totfcfad(j)
write(*,1150) totfcfad(j)
1150 format('FCF (ad curve) = ',f8.6)
25 continue
write(*,*) 'ad. curve          FCF'
write(*,1155) (i, totfcfad(i),i=1,nshow)
1155 format(i4,' ',f8.6)
write(*,1160) totfcf
1160 format('Total FCF = ',f8.6)
c
c Save results in fort.7 file for plotting using readfcf15
c
i = 1
write(7,1170) i
write(7,*) nline
1170 format(i3)
do 200 i = 1,nline
  write(7,1180) rfcf(i)
  write(7,1180) 1.0 - (diss + rval(i))
200 continue
1180 format(f6.4)
write(*,*) 'Results in fort.7 - ',nline,' lines saved'

goto 9998

9999 write(*,*) 'ERROR !!! (number ',ierr,')
9998 stop

900 format('eigenvalue # ',i2,' is ',f9.6,' h, or ',f12.4,' eV')
901 format('eigenvalue # ',i2,' is ',f9.6,' h, or ',f12.4,
# ' eV. (' ,f10.4,')')

```

```

910 format(6f12.6)
920 format('FCF = ',f7.4)
925 format(i3)
930 format(f6.4)

```

```

end

```

```

c *****

```

```

subroutine setconst

```

```

include 'ad3d.inc'

```

```

pi = 3.141592654
hbar = 1.0545887e-34
harev = 27.211608
evwn = 8065.479
a0 = 0.52917706
amu = 1822.882
emu = 9.109534e-31
clight = 2.997925e10
harwn = harev*evwn
amass = 1.66056e-27

```

```

return
end

```

```

c *****

```

```

subroutine readin

```

```

c *****

```

```

c
c read 'normal' parameters from keyboard :
c
c ndvr - number of basis functions to be used.
c nshow - number of wavefunctions to be saved.
c rhomin - smallest value of rho.
c delrho - spacing between rho points.
c nrhopts - number of points in rho.
c nirho - for overlaps in rho, the number of points BETWEEN
c         each pair of points in rho. Use an even number.
c

```

```

include 'ad3d.inc'

```

```

write(*,*) 'readin'
write(*,1020)

```

```

1020 format('Number of adiabatic curves to calculate : ', $)

```

```

read(*,*) nshow
write(*,1030)

```

```

1030 format('Minimum value of Rac (in A) : ', $)

```

```

read(*,*) Racmin
Racmin = Racmin/a0
write(*,1050)

```

```

1050 format('Maximum value : ', $)

```

```

read(*,*) Racmax

```

```

Racmax = Racmax/a0
write(*,1060)
1060 format('Number of points in rho : ', $)
read(*,*) nrhopts
delRac = (Racmax - Racmin)/(nrhopts - 1)
return
end
c *****
c subroutine setbasis

include 'ad3d.inc'

write(*,*) 'Input maximum quantum number in r'
read(*,*) ndvrr
write(*,*) 'Input maximum quantum number in z'
read(*,*) ndvrz0

c
c query for use of symmetric basis. This is more efficient
c for symmetric systems (such as IHI-). Calculate all
c even levels first, then do all odd levels (if desired).
c v=0 of IHI- only has overlap with even levels.
c For nonsymmetrical systems (ie, BrHI-) don't use a
c symmetric basis.
c
c ndvrz0 = quantum number of highest order 1D HO
c ndvrz = number of basis functions in z.
c
c Note that basis in r is already symmetric. Only
c sigma states will be found. This means that you can't
c calculate hot bands due to bend-excited XHY-.
c
c write(*,*) 'Use symmetry in z ? (1 = yes, 0 = no)'
read(*,*) isym

if (isym .eq. 1) then
  write(*,*) 'Even functions only (1) or odds only (2) :'
  read(*,*) ieven
  ndvrz = 1 + (ndvrz0 - ieven)/2
  istep = 2
  istart = ieven
else
  istep = 1
  istart = 1
  ndvrz = ndvrz0
endif

c
c set up product basis - it will be of dimension ndvrr=ndvrr*ndvrz
c
c if (iverbose .eq. 1) write(*,*) 'num r z'
ndvrr = 0
do 20 j = 1,ndvrr

```

```

do 10 i = istart,ndvrz0,istep
  ndvr = ndvr + 1
  indexr(ndvr) = j
  indexz(ndvr) = i
  if (isym .eq. 1) then
    indexzp(ndvr) = 1 + (i -ieven)/2
  else
    indexzp(ndvr) = i
  endif
  if (iverbose.eq.1) write(*,100) ndvr,,i
10  continue
20  continue
100 format(i4,' ',i3,' ',i3)
write(*,*) ndvr,' basis functions'

return
end
c *****
c subroutine setdvr

include 'ad3d.inc'

dimension wgs(nsmall,nsmall),ss1(nsmall),ss2(nsmall)
dimension r(nsmall,nsmall)
dimension tzt(nsmall,nsmall)

c
c Find T matrix for dvr - do this by finding
c R matrix, where  $R_{ij} = \langle Q_i | r | Q_j \rangle$  and finding eigenvectors,
c which give the T matrix. Do this for both r and z.
c
c In r : do change of variables  $r = \rho^2/2$ , then
c 1) volume element is  $dr dz d\phi$ , rather than  $r dr dz d\phi$ .
c 2) can find  $\langle Q_i | r | Q_j \rangle = \langle Q_i | \rho^2 | Q_j \rangle / 2$  analytically,
c see Bell, Warshop.....
c
c In z :  $\langle P_i | z | P_j \rangle$  is given in Atkins, QM p. 102.
c
write(*,*) 'ndvrz0 = ',ndvrz0
ns = nsmall
sqrt2 = sqrt(2.0)

do 10 i = 1,ndvrz0
  r(i,2) = 0.0
  r(i,1) = sqrt(1.0*(i-1))
c could use  $\sqrt{(i-1)/2}$  and skip dividing eigenvalues later
10  continue
write(*,*) 'diagonalizing Rz'

c
c diagonalize Rz matrix.
c R is real, symmetric, tridiagonal, so use EISPACK routine
c for banded matrices.

```

```

c
call rsb(ns,ndvrz0,2,r,zgs,1,wgs,ss1,ss2,ierr)
c
c   normalize eigenvectors
c
do 40 i = 1,ndvrz0
  anorm = 0.0
  do 20 j = 1,ndvrz0
    anorm = anorm + wgs(j,i)*wgs(j,i)
20  continue
  anorm = sqrt(anorm)
  do 30 j = 1,ndvrz0
    tz(j,i) = wgs(j,i)/anorm
30  continue
40  continue

do 50 i = 1,ndvrz0
  zgs(i) = zgs(i)/sqrt2
50  continue
c
c   When using symmetry, only use half the number of points
c   in z, and chop Tz matrix accordingly.
c
write(*,*) 'Chopping Tz w/symmetry'
if (isym .eq. 1) then
  do 58 i = 1,ndvrz
    jk = 0
    do 55 j = ieven,ndvrz0,2
      jk = jk + 1
      tzt(jk,i) = tz(j,i)*sqrt2
55  continue
58  continue
    do 65 i = 1,ndvrz
      do 60 j = 1,ndvrz
        tz(j,i) = tzt(j,i)
60  continue
65  continue
    endif

if (iverbose .eq. 1) then
  write(*,*) 'In z : '
  write(*,*) 'Eigenvalues:'
  do 90 i = 1,ndvrz
    write(*,*) zgs(i)
    write(*,*) 'Tz array:'
    call ShowArr(tz,ns,ndvrz)
90  continue
  endif

c
c   Now do r direction
c
write(*,*) 'doing r'

```

```

do 110 i = 1,ndvrr
  r(i,2) = (2.0*(i-1) + 1.0)/2.0
  r(i,1) = ((i-2) + 1.0)/2.0
110  continue
c
c  diagonalize R.
c  R is real, symmetric, tridiagonal, so use EISPACK routine
c  for banded matrices.
c
call rsb(ns,ndvrr,2,r,rgs,1,wgs,ss1,ss2,ierr)
c
c  normalize eigenvectors
c
do 140 i = 1,ndvrr
  anorm = 0.0
  do 120 j = 1,ndvrr
    anorm = anorm + wgs(j,i)*wgs(j,i)
120  continue
  anorm = sqrt(anorm)
  do 130 j = 1,ndvrr
    tr(j,i) = wgs(j,i)/anorm
130  continue
140  continue

if (iverbose .eq. 1) then
  write(*,*) 'In r : '
  write(*,*) 'Eigenvalues:'
  do 160 i = 1,ndvrr
160  write(*,*) rgs(i)
  write(*,*) 'Tr array:'
  call ShowArr(tr,ns,ndvrr)
endif

return
end
c *****
function Finda(i,j,im,in)

include 'ad3d.inc'

c
c  Finds  $\langle Q_i | H | Q_j \rangle$ .
c  The  $h_0$  basis set satisfies
c   $h_0 Q_i = e_i Q_i$ , and  $\langle Q_i | Q_j \rangle = \delta(i,j)$ .
c  Let  $H = h_0 + v$ , then
c   $\langle Q_i | H | Q_j \rangle = \langle Q_i | h_0 + v | Q_j \rangle$ 
c   $= \langle Q_i | h_0 | Q_j \rangle + \langle Q_i | v | Q_j \rangle$ 
c   $= e_i * \delta(i,j) + \langle Q_i | v | Q_j \rangle$ , and
c   $e_i = e_{\text{harm}}*(i + 1/2)$ 
c

if (isym .eq. 1) then
  imp = 1 + (im - ieven)/2

```



```

    inp = 1 + (in - ieven)/2
  else
    imp = im
    inp = in
  endif

  sum = 0.0
  do 5 l = 1,ndvz
    sum = sum + tz(imp,l)*rvint(i,j,l)*tz(inp,l)
5  continue
    if ((i.eq.j).and.(im.eq.in)) then
      Finda = sum + eharmr*(2.0*(i - 1) + 1.0)
1     + eharmz*((im-1) + 0.5)
    else
      Finda = sum
    endif

  return
end
C *****
function rvint(i,j,l)
C
C finds Integral(Axelement(x) dx)
C
  include 'ad3d.inc'

  sum = 0.0
  do 5 k = 1,ndvrr
    sum = sum + tr(i,k)*Varray(isp,k,l)*tr(j,k)
5  continue
  rvint = sum

  return
end
C *****
subroutine setV

  include 'ad3d.inc'

  write(*,*) 'betar = ',betar
  write(*,*) 'betaz = ',betaz
  do 20 j = 1,ndvrr
    do 10 i = 1,ndvz
      Varray(isp,j,i) = V(rgs(j)/betar + rer,zgs(i)/betaz + rez)
10  continue
20  continue
  return
end
C *****
function V(r,z)

```

```

include 'ad3d.inc'

dimension dr(3),dz(3),Ra(3)

dmorse(xx,x0,dee,xalpha)=dee*(1.0-exp(-xalpha*(xx-x0)))**2

if ((isp.eq.1) .and. (iantype .eq. 1)) then
c
c   Anion potential : harmonic osc.
c
  V = ankr*r + ankz*(z-reza)*(z-reza)/2.0
  goto 900
endif
if ((isp.eq.1) .and. (iantype .eq. 2)) then
c
c   Anion potential : Morse osc.
c
  V = ankr*r + dmorse(z,reza,xdeA,xalphaA)
c   write(*,850) z, reza, dmorse(z,reza,xdeA,xalphaA),V
850   format(4(f9.5,' '))
  if (V .gt. 5.0) V = 5.0
c   shelf potential at VERY high values - avoids errors
c   in diagonalization associated with having some values too large
  goto 900
endif

if ((ipot .eq. 3) .or. (ipot .eq. 4)) then
c
c   LEPS potential
c
  c = a0/10.0
  Ra(1) = sqrt((Rac/2.0 + z)**2 + 2.0*r) * c
  Ra(2) = sqrt((Rac/2.0 - z)**2 + 2.0*r) * c
  Ra(3) = Rac * c
  call doQJsum(Ra,Qsum,aJsum)
  V = (Qsum - sqrt(aJsum))/harev/96.485

  if (ipot .eq. 3) then
c
c   use information from collinear fit pot'l for Br + HBr
c   V(Rab,Rbc,Rac) = VLEPS(Rab,Rbc,Rac) +
c   [Vfit(Rab,Rbc,Rab+Rbc) - ZPE(Rab,Rbc,Rab+Rbc) -
c   VLEPS(Rab,Rbc,Rab+Rbc)],
c   where Vfit is fitted (collinear) potential for Br+HBr and
c   ZPE is zero point energy (on a LEPS surface) at this
c   collinear geometry.
c
  Ra(3) = Ra(1) + Ra(2)
  call doQJsum(Ra,Qsum,aJsum)
  VlinLEPS = (Qsum - sqrt(aJsum))/harev/96.485
  Vlinfit = pot6(10.0*Ra(1),10.0*Ra(2))

```

```

        Vzeropt = bndh(Ra(1)/c,Ra(2)/c)
        V = V + Vlinfit - Vzeropt - VlinLEPS
    endif
endif

900  V = V - basiskr*(r - rer) - 0.5*basiskz*(z-rez)**2

    return
    end
c *****
    subroutine saveall(basefrqr,basefrqz)
c *****
    include 'ad3d.inc'
    character*15 fname

    write(*,*) 'saveall'

    fname = molecule(1:imollen) // ext(1:iextlen)

    open(2,file = fname)
    write(2,900) nshow,ndvrr,ndvrz0,ndvrz,ndvr,isym,ieven
900  format(7i5)
    write(2,905) nrhopts,Racmin,Racmax
905  format(i5,2f12.5)
    write(2,907) basefrqr,basefrqz
907  format(2f12.5)
c
c  save adiabatic curves (eigenvalues)
c
    do 10 i = 1,nshow
10   write(2,910) (val(i,j),j=1,nrhopts)
910  format(f9.6)
c
c  save a.s., bend overlap for each
c  adiabatic curve, at each value of R (I-I).
c
    do 20 j = 1,nrhopts
        write(2,920) (cs(i,j), i=1,nshow)
20   continue
920  format(f9.6)
    close(2)
    return
    end
c *****
    subroutine readall(basefrqr,basefrqz)
c *****
    include 'ad3d.inc'
    character*15 fname

    write(*,*) 'readall'

```

```

nshowold = nshow
fname = molecule(1:imollen) // ext(1:iextlen)

open(2,file = fname)
read(2,900) nshow,ndvrr,ndvrz0,ndvrz,ndvr,isym,ieven
900 format(7i5)
read(2,905) nrhopts,Racmin,Racmax
905 format(i5,2f12.5)
read(2,907) basefrqr,basefrqz
907 format(2f12.5)

nshow = nshowold
do 10 i = 1,nshow
10 read(2,910) (val(i,j),j=1,nrhopts)
910 format(f9.6)

do 20 j = 1,nrhopts
read(2,920) (cs(i,j), i=1,nshow)
20 continue
920 format(f9.6)
close(2)
return
end
c *****
c subroutine savewf
c *****
include 'ad3d.inc'

character*15 fname

write(*,*) 'savewf'

ntens = nshow/10
nleft = nshow - 10*ntens

do 20 j = 1, ntens
fname = molecule(1:imollen) // numbers(j+1)
$ // ext(1:iextlen)
c
c save adiabatic curve in plotfile for violet
c each plot file will have (at most) 10 adiabatic curves
c
open(2,file = fname(1:imollen+iextlen+1) // 'v')

do 10 i = 1, nrhopts
write(2,950) (Racmin+(i-1)*delrac)*a0,
$ (val(10*(j-1) + k,i)*harev, k = 1, 10)
950 format(f8.4,10(2x,f8.4))
10 continue
close(2)
20 continue

```

```

if (nleft .gt. 0) then

  fname = molecule(1:imollen) // numbers(ntens+1)
  $ // ext(1:iextlen)
c
c   save adiabatic curve in plotfile for violet
c
  open(2,file = fname(1:imollen+iextlen+1) // 'v')

  do 30 i = 1, nrhopts
    write(2,950) ((Racmin+(i-1)*deirac)*a0,
  $   val(10*(j-1) + k,i)*harev, k = 1, nleft)
30  continue
    close(2)
  endif
  return
end

c *****
c   subroutine ShowArr(a,nm,n)
c *****
  implicit real (a-h,o-z)
  dimension a(nm,nm)

  write(6,*) ' '
  do 10 i = 1,n
10   write(6,100) (a(j,i), j=1,n)
100  format(10f10.5,/)
  return
end

```

2.4.4 leps3d2.f

```

c leps3D2.f -- Create a LEPS potential for a triatomic system.
c
c Created from leps6.f by RBM 12/13/90
c LEPS potential for triatomic system
c Optimized for use on a Cray !!
c
c See Smith p. 44
c
c Indices :
c   1 = ab
c   2 = bc
c   3 = ca
c

```

```

c Variables:
c   R[1] = ab distance (nm)
c   S[1] = Sato Parameter
c   De[1] = dissociation limit of ab (kJ/mol)
c   Rel[1] = equilibrium ab bond length (nm)
c   B[1] = width of potential for ab (nm-1)
c   M[1] = mass of atom a (amu)
c
c Functions:
c   V(R[1], R[2]) = LEPS potential (kJ/mol)
c   Q(1, R[1]) = Q for ab and Rab
c   aJ(1, R[1]) = J for ab and Rab
c   VM(1, R[1]) = Morse potential for ab and Rab
c   VaM(1, R[1]) = anti-Morse potential for ab and Rab
c
c Includes:
c   initleps
c   initpars
c   doQJsum
c   doJsign
c *****
c   subroutine initleps
c *****
c   call initpars
c   call doJsign
c   return
c   end
c *****
c   subroutine initpars
c *****
c
c   Read masses, potential parameters, etc from
c   the file 'ad3d.par' - this avoids having to
c   recompile the code when calculating the effect
c   of isotopic substitution, or when modifying the
c   potential.
c
c   include 'ad3d.inc'
c
c   write(*,*) 'initpars'
c   numbers(1) = '0'
c   numbers(2) = '1'
c   numbers(3) = '2'
c   numbers(4) = '3'
c   numbers(5) = '4'
c   numbers(6) = '5'
c   numbers(7) = '6'
c   numbers(8) = '7'
c   numbers(9) = '8'
c   numbers(10) = '9'
c   numbers(11) = 'A'

```

```

numbers(12) = 'B'
numbers(13) = 'C'
numbers(14) = 'D'
numbers(15) = 'E'

open(1,file = 'ad3d.par')
read(1,200) molecule
200 format(A6)
imollen = index(molecule,' ') - 1
read(1,200) ext
iextlen = index(ext,' ') - 1
read(1,*) (aMa(i), i = 1,3)
100 format(3(f9.5,2x))
read(1,*) ipot
do 30 i = 1, 3
  read(1,*) Re(i), De(i), Bl(i), S(i)
30 continue
do 40 i = 1,3
  Te(i) = 1.0/(1.0 + S(i))
40 continue
130 format(4f9.5)
close(1)

write(*,*) 'Input file read'
write(*,*) 'Masses : '
write(*,100) (aMa(i), i = 1,3)
write(*,*) 'LEPS parameters : Re (nm), De (kJ/mol), B (1/nm), S'
do 60 i = 1, 3
  write(*,150) Re(i), De(i), Bl(i), S(i)
60 continue
150 format(4(f9.5,2x))

return
end

c *****
c subroutine doQJsum(R,Qsum,aJsum)
c *****
include 'ad3d.inc'
dimension R(3), taJ(3)

Qsum = 0.0
aJsum = 0.0
do 10 i = 1,3
  x = -Bl(i)*(R(i)-Re(i))
  aVM = De(i)*(exp(2.0*x)-2.0*exp(x))
  aVaM = De(i)*(exp(2.0*x) + 2.0*exp(x))/2.0
  taJ(i) = ((1.0 + S(i)) * aVM - (1.0 - S(i))*aVaM)/2.0
  Qsum = Qsum + Te(i)*((1.0+S(i))*aVM +
$ (1.0-S(i))*aVaM)/2.0
10 continue

```

```

do 30 i = 3,1,-1
  do 20 k = 1,i
    aJsum = aJsum + taJ(i)*jsign(i,k)*Te(i)*taJ(k)*Te(k)
20  continue
30  continue
  return
end
c *****
c  subroutine dojsign
c *****
c  include 'ad3d.inc'

do 10 i = 1,3
  do 5 k = 1,3
    if (i.eq.k) then
      jsign(i,k) = 1
    else
      jsign(i,k) = -1
    endif
5  continue
10 continue
  return
end

```

2.4.5 findmin.f

```

c *****
c  subroutine findmin(jcurve,lmflag,imin,imax,vmin,vmax,vasy)
c *****
c
c  Analyze adiabatic curve to determine if it
c  is purely repulsive or contains local minima.
c  If it has local minima, find the deepest, and
c  also find the highest local maximum.
c
c  include 'ad3d.inc'
c
c  jcurve : adiabatic curve number
c  lmflag : 0 if purely repulsive curve, 1 if it has a local minimum
c  imin : pt # for deepest local minimum
c  imax : pt # for local max
c  vmin : potential at deepest local minimum
c  vmax : potential at local maximum
c  vasy : potential at last point (nrhopts)
c

```



```

    lmflag = 1
c
c  find first local minimum
c
    i = 0
5  i = i + 1
    if (val(jcurve,i+1).gt.val(jcurve,i)) then
        ismall = i
        k = nrhopts + 1
7  k = k - 1
        if (val(jcurve,k-1).gt.val(jcurve,k)) goto 7
        ilarge = k
        vmin = val(jcurve,ismall)
        vmax = val(jcurve,ilarge)
        imin = ismall
        imax = ilarge
        do 8 k = ismall, ilarge
            if (val(jcurve,k) .lt. vmin) then
                vmin = val(jcurve,k)
                imin = k
            endif
            if (val(jcurve,k) .gt. vmax) then
                vmax = val(jcurve,k)
                imax = k
            endif
6  continue
    else
        if (i .lt. nrhopts-1) goto 5
        lmflag = 0
    endif

999 vasy = val(jcurve,nrhopts)
    return
    end

```

2.4.6 boundr3d.f

```

c -----
c  file : boundr3d.f
c -----
c
c Includes :
c  boundrho
c  FindV
c  setv0rho

```

```

c  setvrho
c  setva
c  setArho
c  Findarho
c  findnupr
c  findtrar
c  setdvrr
c  findeigr
c
c *****
c   subroutine boundrho(Veff,iaflag,tempval,tempvec,
c   $  vtop, neigen,nsub)
c *****
c
c   Find shape and Feshbach resonances on the adiabatic
c   curve Veff. (if iaflag = 0) or anion wavefunctions
c   (if iaflag = 1). vtop is the maximum energy at which
c   wavefunction are considered bound. On return, neigen
c   is the number of 'bound' states; tempval() is the
c   eigenvalues, and tempvec() is the wavefunctions.
c
c   include 'ad3d.inc'
c
c   dimension Veff(*),tempval(nrdvrmax)
c   dimension tempvec(nrdvrmax,nrdvrmax)
c
c   could optimize basis set frequency - but don't
c
c   call optimrho(Veff,iaflag)
c
c   set up A matrix w/ optimized basis set
c
c   ndvrsave = ndvrrho
c   ndvrrho = ndvrrho - nsub
c
c   write(*,*) 'ndvrrho = ',ndvrrho
c   call setdvrr
c   iverbose = 0
c   call setv0rho
c   call setvrho(Veff, iaflag)
c   call setArho
c
c   get the eigenvalues
c
c   call findeigr(tempval, tempvec, vtop, neigen, iaflag)
c   ndvrrho = ndvrsave
c
c   write(*,1000) neigen
c   1000 format('In boundrho, neigen = ',i4)
c   return
c   end

```

```

c *****
c function FindV(Veff, rho2)
c *****
c
c include 'ad3d.inc'
c
c dimension Veff(*)
c
c if (rho2.le.rhomin) then
c   FindV = Veff(1) + (Veff(1)-Veff(2))*(rhomin-rho2)/delrho
c endif
c if (rho2.ge.rhomas) then
c   FindV = Veff(nrhopts)
c endif
c if ((rho2.gt.rhomin).and.(rho2.lt.rhomas)) then
c   pt = (rho2 - rhomin)/delrho + 1
c   ipt = int(pt)
c   p = pt - ipt
c   FindV = (1.0 - p) * Veff(ipt) + p * Veff(ipt + 1)
c endif
c10 if (iverbose.eq.1) write(*,1000) rho2, ipt, p, Veff(ipt)*harev,
c   $ Veff(ipt+1)*harev,FindV*harev
c 1000 format('rho : ',f8.3,' pt ',i3,' p ',f6.5,' Ve ',f8.3,
c   $ ' ',f8.3,' V ',f8.3)
c   return
c   end
c *****
c subroutine setv0rho
c *****
c
c set up reference potential -
c this is the potential implied by the basis set used...
c this routine is used once, if the basis set is
c not modified.
c
c include 'ad3d.inc'
c
c if (iverbose.eq.1) then
c   write(*,*) 'setv0rho'
c endif
c do 10 i = 1, ndvrrho
c   zr(i) = xgs(i)/betarho + rerho
c   dx = zr(i) - rerho
c   v0(i) = basiskrho*dx*dx/2.0
c 10 continue
c if (iverbose .eq. 1) then
c   write(*,*) 'V0'
c   do 20 i = 1, ndvrrho
c 20   write(*,100) i, v0(i)
c 100  format(i4,' ',f12.5)
c   endif

```

```

return
end
c *****
c subroutine setvrho(Veff,iaflag)
c *****
c
c set up the difference potential array (rvarray).
c rvarray is the difference between the actual potential
c and the reference potential (v0).
c
c include 'ad3d.inc'
c dimension Veff(*)

do 10 i = 1, ndvrrho
  if (iaflag.eq.0) rvarray(i) = FindV(Veff,zr(i))
  if (iverbose.eq.1) write(*,1010) zr(i),rvarray(i)*harev
1010 format('z : ',f9.4,' : ',f12.4,' eV')
10 continue
c
c find difference potential by subtracting off
c reference potential.
c
c if (iverbose .eq. 1) write(*,1000) (rvarray(i),i=1,ndvrrho)
1000 format(10f9.6,/)
  do 100 i = 1, ndvrrho
    rvarray(i) = rvarray(i) - v0(i)
100 continue
  if (iverbose .eq. 1) then
    write(*,*) 'VRHO'
    write(*,1000) (rvarray(i),i=1,ndvrrho)
  endif

return
end
c *****
c subroutine setva(ank,rea,ittype)
c *****
c
c For ANION, along rho:
c set up the difference potential array (rvarray).
c rvarray is the difference between the actual potential
c and the reference potential (v0).
c
c include 'ad3d.inc'

if (ittype .eq. 1) then
  n = ndvr
  do 10 i = 1, ndvr
    zt = zr(i) - rea
    rvarray(i) = ank*zt*zt/2.0
10 continue

```

```

else
  n = ndvrrho
  do 20 i = 1, ndvrrho
    rt = zr(i) - rea
    rvarray(i) = ank*rt*rt/2.0
20  continue
endif

if (iverbose .eq. 1) then
  write(*,*) 'AVRHO'
  write(*,1000) (rvarray(i),i=1,ndvrrho)
endif
1000 format(f9.6)
c
c   find difference potential by subtracting off
c   reference potential.
c
  do 100 i = 1, n
    rvarray(i) = rvarray(i) - v0(i)
100  continue
if (iverbose .eq. 1) then
  write(*,*) 'AVRHOafter'
  write(*,1000) (rvarray(i),i=1,ndvrrho)
endif

return
end
c *****
  subroutine setArho
c *****
c
c   Construct the Hamiltonian matrix
c
  include 'ad3d.inc'

if (iverbose.eq.1) then
  write(*,*) 'setArho'
endif
do 20 i = 1,ndvrrho
  do 10 j = 1, i
    ar(i,j) = Findarho(i,j)
    ar(j,i) = ar(i,j)
10  continue
20  continue
return
end
c *****
  function Findarho(i,j)
c *****
c
c   see FindA for comments

```

```

c
include 'ad3d.inc'

sum = 0.0
do 5 k = 1,ndvrrho
  sum = sum + t(i,k)*rvarray(k)*t(j,k)
5 continue
if (i.eq.j) then
  Findarho = sum + eharmrho*((i-1) + 0.5)
else
  Findarho = sum
endif
return
end
c *****
function findnupr(Veff,iaflag,anu)
c *****
c
c find derivative of trace w.r.t. the frequency anu.
c
include 'ad3d.inc'

dimension Veff(*)

hanu = 0.0001
call setbeta(anu-hanu)
c write(*,160) beta,anu,eharm,eharm*harev*evwn
160 format('beta : ',f9.6,' nu : ',f9.3,' e : ',f9.7
$ , ' or ',f13.4' cm-1')
call setv0rho
call setvrho(Veff, iaflag)
call findtrar(trace1)

call setbeta(anu+hanu)
call setv0rho
call setvrho(Veff, iaflag)
call findtrar(trace2)
findnupr = (trace2 - trace1)/hanu/2.0
c write(*,180) anu,findnupr
180 format(' nu : ',f13.7,' deriv : ',f16.7)
return
end
c *****
subroutine findtrar(trace)
c *****
include 'ad3d.inc'

trace = 0.0
do 10 j = 1,ndvrrho
  trace = trace + FindArho(j,j)
10 continue

```

```

return
end
c *****
c      subroutine setdvrr
c *****
c
c      set up for dvr-type quadrature.
c      see setdvr for comments
c
c      include 'ad3d.inc'

      dimension wgs(nrdvrrho,nrdvrrho),r(nrdvrrho,nrdvrrho)
c
c      <Qn | x | Qn+1> = sqrt(n+1)/(beta*sqrt(2))
c      all other matrix elements are 0
c      (x can be written in terms of raising/lowering operators..)
c      see Atkins, Molecular QM p. 102
c
c      write(*,*) 'setdvrr'
c      do 47 i = 1,nrdvrrho
c          r(i,2) = 0.0
c          r(i,1) = sqrt(1.0*(i-1))
47      continue
c
c      Diagonalize R. R is real, tridiagonal, so
c      use eispack routine for banded matrices
c
c      call rsb(nrdvrrho,nrdvrrho,2,r,xgs,1,wgs,s1,s2,ierr)
c
c      eigenvectors are orthogonal, now normalize
c
c      do 70 i=1,nrdvrrho
c          xnorm = 0.0
c          do 50 j = 1,nrdvrrho
c              xnorm = xnorm + wgs(j,i)*wgs(j,i)
50      continue
c          xnorm = sqrt(xnorm)
c          do 60 j = 1,nrdvrrho
c              t(j,i) = wgs(j,i)/xnorm
60      continue
70      continue
c          sqrt2 = sqrt(2.0)
c          do 80 i = 1,nrdvrrho
c              xgs(i) = xgs(i)/sqrt2
80      continue
c          if(iverbose.eq.1) then
c              write(*,*) 'eigenvalues :'
c              do 90 i = 1, nrdvrrho
c                  write(*,100) xgs(i)
90      continue
c          call ShowArr(t,nrdvrrho,nrdvrrho)

```

```

endif
100 format(f12.6)
return
end
c *****
c subroutine findeigr(tempval,tempvec,vtop,neigen,iaflag)
c *****
c
c uses eispack library routines to diagonalize A,
c finding the eigenvalues and eigenvectors.
c The nshow lowest eigenvalues/eigenvectors are
c saved in tempval(1..nshow), tempvec(1..nshow,1..ndvr),
c respectively.
c
c include 'ad3d.inc'
c dimension tempval(nrdvrmax), tempvec(nrdvrmax,nrdvrmax)
c
c if (iverbose.eq.1) then
c   write(*,*) 'findeigr'
c endif
c if (iverbose.eq.1) then
c   write(*,*) 'A matrix : '
c   call ShowArr(ar,nrdvrmax,ndvrrho)
c endif
c call rs(nrdvrmax,ndvrrho,ar,tempval,1,tempvec,s1,s1,ierr)
c if (ierr.ne.0) then
c   write(*,1000) ierr
c stop
1000 format('ERROR # ',i3,' in diagonalization in findeigr')
c endif
c
c wavefunctions are ok, so normalize them
c
c 1030 format('iaflag: ',i7)
c if (iaflag.eq.1) then
c
c anion - only return wavefunction w/ quantum # neigen
c
c   i = neigen
c   xnorm = 0.0
c   do 25 j = 1, ndvrrho
c     xnorm = xnorm + tempvec(j,i)*tempvec(j,i)
25   continue
c   xnorm = sqrt(xnorm)
c   do 28 j = 1, ndvrrho
c     tempvec(j,i) = tempvec(j,i)/xnorm
28   continue
c   endif
c   if (iaflag.eq.0) then
c
c neutral -- find number of 'bound' states (states

```



```

c      with energies below vtop) and return this number
c      (neigen) and eigenvalues/vectors.
c
      write(*,1020) (tempval(i)*harev, i = 1,4)
1020  format('eigenvalues: ',4f15.4)
      i = 0
29    i = i + 1
      if ((tempval(i) .le. vtop).and.(i .lt. ndvrrho)) goto 29
      neigen = i - 1
      write(*,1010) neigen
1010  format(i3,' bound states')
      do 50 i = 1, neigen
          xnorm = 0.0
          do 30 j = 1, ndvrrho
              xnorm = xnorm + tempvec(j,i)*tempvec(j,i)
30    continue
          xnorm = sqrt(xnorm)
          do 40 j = 1, ndvrrho
              tempvec(j,i) = tempvec(j,i)/xnorm
40    continue
50    continue
      endif
      write(*,1100) neigen
1100 format('at end, neigen = ',i4)
      return
      end

```

2.4.7 dofcf3d.f

```

c -----
c      File : dofcf3d.f
c      Includes :
c      dofcf
c      doweight
c      rover
c      findwf
c      dorw
c      testover
c
c *****
c      subroutine dofcf(iad,issmax,wfx)
c *****
c
c      Finds FCFs from v=0 of anion to all 1..issmax rho levels
c      supported by adiabatic curve iad.
c

```

```

include 'ad3d.inc'

common/integ/weight
dimension cstar(maxrpts), wfx(nrdvrmax,maxrpts,maxirho)
dimension dstar(maxrpts),weight(maxrpts)

sumrtot = 0.0
do 90 iss = 1, issmax
  sumr = 0.0
  call rover(iss,wfx,dstar)
  sum = 0.0
  do 80 k = 1, nrhopts
c     if (iss .eq. 1) write(*,1100) k, iad, cstar(k)
1100  format('pt # ',i3,' C*(',i3,') = ',f9.6)
      sum = sum + weight(k)*cs(iad,k)*dstar(k)
      sumr = sumr + weight(k)*dstar(k)
c     write(*,1000) iss,k,weight(k),cstar(k),dstar(k)
1000  format(i3,' ',i3,' ',f5.3,' ',f10.7,' ',f10.7)
80    continue
      fcf(iad,iss) = sum*sum
      fcfrho(iad,iss) = sumr*sumr
      sumrtot = sumrtot + sumr*sumr
c     write(*,1200) iad, iss, sumr*sumr
1200  format('Curve # ',i3,' ss # ',i3,' fcf(r) = ',f9.6)
c     if (sum .gt. 0.0) write(*,*) iad, iss, sum
90    continue
c     write(*,*) 'Tot fcf(r) = ',sumrtot
      return
      end
c *****
      subroutine doweight(itype)
c *****
c
c     Find weighing factors for
c     Trapezoid rule integration (itype = 1)
c     Simpson's rule integration (itype = 2)
c     (Alternative Extended Simpson's Rule,
c     Numerical Recipes 4.1.14)
c
c     include 'ad3d.inc'

common /integ/weight(maxrpts)

rstep = (rhomax - rhomin)/(nrhopts - 1)
if (itype .eq. 1) then
  weight(1) = rstep/2.0
  weight(nrhopts) = rstep/2.0
  do 10 i = 2, nrhopts-1
10    weight(i) = rstep
  endif
if (itype .eq. 2) then

```

```

nr = nrhopts
weight(1) = rstep*17.0/48.0
weight(2) = rstep*59.0/48.0
weight(3) = rstep*43.0/48.0
weight(4) = rstep*49.0/48.0
weight(nr) = weight(1)
weight(nr-1) = weight(2)
weight(nr-2) = weight(3)
weight(nr-3) = weight(4)
do 20 i = 5, nr-4
20   weight(i) = rstep
endif
c   write(*,*) 'WEIGHTS'
c   do 40 i = 1, nrhopts
c40  write(*,1000) i, weight(i)
1000 format(i3,' : ', f5.3)
return
end
c *****
c   subroutine rover(iss,wfx,dstar)
c *****
c
c   Calculate product of sym. stretch neutral and anion
c   wavefunctions at every value of rho.
c
c   include 'ad3d.inc'
c   common /rinteg/rweight(maxirho)
c   dimension dstar(maxrpts),wfx(nrdvrmax,maxrpts,maxirho)

dstar = 0.0
do 30 k = 1, nrhopts
do 25 i = 1, nirho

c
c   find anion wavefunction da at current value of r.
c
c   da = 0.0
do 10 j = 1, ndvrrho
da = da + rveca(j)*wfx(j,k,i)
10  continue

c
c   find neutral wavefunction dn at current value of r.
c
c   dn = 0.0
do 20 j = 1, ndvrrho
dn = dn + rvec(j,iss)*wfx(j,k,i)
20  continue
dstar(k) = dstar(k) + da * dn * rweight(i)
25  continue
30  continue
c   write(*,*) 'DSTAR'
c   do 40 i = 1,nrhopts

```

```

c40  write(*,1000) i, dstar(i)
1000 format(i3,' ',f10.6)
      return
      end
c *****
      subroutine findwf(wfx)
c *****
      include 'ad3d.inc'

      dimension wfx(nrdvrmax,maxrpts,maxirho),xnorm(nrdvrmax)
      dimension xval(maxrpts,maxirho)

      dx = (rhomax - rhomin)/(nrhopts - 1)
      dxi = dx/(nirho-1)
c      divide by nirho, not (nirho-1)
      do 7 j = 1, nirho
        do 5 i = 1, nrhopts
          5  xval(i,j) = rhomin + (i-1)*dx + (j-1)*dxi - dx/2.0 - rerho
          7  continue
c
c      first set h0 and h1
c
      do 15 j = 1, nirho
        do 10 i = 1, nrhopts
          wfx(1,i,j) = 1.0
          wfx(2,i,j) = 2.0*betarho*xval(i,j)
        10  continue
      15  continue

c
c      Now get the rest of the hermite polynomials
c      recursively, using
c       $h[n+1] = 2*y*h[n] - 2*n*h[n-1]$ ,
c      where  $y = betarho*(xmin + i*xh - re)$ 
c
      do 30 k = 3,ndvrrho
        do 25 j = 1, nirho
          do 20 i = 1, nrhopts
            wfx(k,i,j) = 2.0*betarho*xval(i,j)*
              1  wfx(k-1,i,j) - 2.0*(k-2)*wfx(k-2,i,j)
          20  continue
        25  continue
      30  continue

c
c      Include normalization and  $\exp(-y*y/2)$ 
c
      xnorm(1) = sqrt(betarho/sqrt(pi))
      do 40 k = 2,ndvrrho
        xnorm(k) = xnorm(k-1)/sqrt(2.0*(k-1))
      40  continue

```

```

do 65 j = 1, nirho
  do 60 i = 1, nrhopts
    y = betarho*xval(i,j)
    yexp = exp(-y*y/2.0)
    do 50 k = 1, ndvrrho
      wfx(k,i,j) = wfx(k,i,j) * xnorm(k) * yexp
50    continue
60    continue
65    continue

  return
end
c *****
  subroutine dorw
c *****
  include 'ad3d.inc'
  common /rinteg/rweight(maxirho)

  rweight(1) = 0.5/(nirho-1)
  rweight(nirho) = rweight(1)
  do 10 i = 2, nirho-1
10    rweight(i) = 1.0/(nirho-1)

  return
end
c *****
  subroutine testover(wfint,wfx)
c *****
  include 'ad3d.inc'

  dimension wfint(nrdvrmax),wfx(nrdvrmax,maxrpts,maxirho)
  common /integ/weight(maxrpts)
  common /rinteg/rweight(maxirho)

  if (iverbose .eq. 1) then
    write(*,*) 'WEIGHT'
    do 3 i = 1, nrhopts
3      write(*,1000) i,weight(i)
      write(*,*) 'RWEIGHT'
      do 5 i = 1, nirho
5        write(*,1000) i,rweight(i)
1000    format(i3,' ',f10.6)
      endif

    do 20 k = 1, ndvrrho
      sum = 0.0
      do 15 j = 1, nirho
        do 10 i = 1, nrhopts
          sum = sum + wfx(k,i,j)*wfx(k,i,j)*weight(i)*rweight(j)
10        continue
15      continue

```

```

      wfint(k) = sum
20  continue

      return
      end

```

2.4.8 pot6.f

```

c  file pot6.f
c
c  FORTRAN version of pot6.p (PASCAL).
c  does  $x^2/x^4/x^6$  potential
c  Application : fitted surface for Br + HBr reaction.
c
c  Converted by RBM 4/9/91
c
c  Includes :
c  testpot
c  pot
c  pot6
c  Initpot
c  spline
c  splint
c *****
      subroutine testpot
c *****
      implicit real (a-h,o-z)
      real Ra(3)

      call setconst
      call initleps
      call initpot

      harev = 27.211608
      a0 = 0.52917706
      c = a0/10.0

10  write(*,*) 'Input r1, r2, r3 (A)'
      read(*,*) ra(1),ra(2),ra(3)
      do 20 i = 1,3
20  Ra(i) = Ra(i)/10.0
      call doQJsum(Ra,Qsum,aJsum)
      V = (Qsum - sqrt(aJsum))/harev/96.485

c
c  use information from collinear fit pot'l for Br + HBr
c  V(Rab,Rbc,Rac) = VLEPS(Rab,Rbc,Rac) +

```

```

c      [Vfit(Rab,Rbc,Rab+Rbc) - ZPE(Rab,Rbc,Rab+Rbc) -
c      VLEPS(Rab,Rbc,Rab+Rbc)],
c      where Vfit is fitted (collinear) potential for Br+HBr and
c      ZPE is zero point energy (on a LEPS surface) at this
c      collinear geometry.
c
c      write(*,*) 'V LEPS, 3D = ',V*harev,' eV'
c      Ra(3) = Ra(1) + Ra(2)
c      call doQJsum(Ra,Qsum,aJsum)
c      VlinLEPS = (Qsum - sqrt(aJsum))/harev/96.485
c      write(*,*) 'V lin LEPS = ',VlinLEPS*harev,' eV'
c      Vlinfit = pot6(10.0*Ra(1),10.0*Ra(2))
c      write(*,*) 'Vlinfit = ',Vlinfit*harev,' eV'
c      Vzeropt = bndh(Ra(1)/c,Ra(2)/c)
c      write(*,*) 'Vzeropt = ',Vzeropt*harev,' eV'
c      V = V + Vlinfit - Vzeropt - VlinLEPS
c      write(*,*) 'V 3D, fit = ',v*harev,' eV'
c
c      write(*,*) ' '
c      goto 10
c      return
c      end
c *****
c      function pot6(rq1,rq2)
c *****
c
c      given r1 (r Br - H) and r2 (r H - Br) in angstroms, returns
c      the potential for that (collinear) arrangement in hartree
c      using x^2/x^4/x^6 potential
c
c      implicit real (a-h,o-z)
c      parameter(npn = 21)
c      npn is number of nodes for cubic spline interpolation
c      common/splpot/pX1(npn),pal(npn),pA(npn),pB(npn),pR(npn),px4(npn),
c      $ pX12(npn),pal2(npn),pA2(npn),pB2(npn),px42(npn),
c      $ pR2(npn),pAA(npn),pXX1(npn),pAA2(npn),pXX12(npn)
c
c      real curX1, cural, curA, curB, Rxx, curX, curx4
c      real x2, x12, pola, polb, polc
c
c      Rxx = rq1 + rq2
c      find R Br-Br
c
c      if ((Rxx .le. pR(npn)) .and. (Rxx .ge. pR(1))) then
c      if (Rxx .le. pR(4)) then
c      curX1 = 0.0
c      curA = 0.0
c      else
c      call splint(pR2,pXX1,pXX12,npn-3,Rxx,curX1)
c      call splint(pR2,pAA,pAA2,npn-3,Rxx,curA)
c      endif

```

```

call splint(pR,pal,pal2,npn,Rxx,cural)
call splint(pR,pB,pB2,npn,Rxx,curB)
call splint(pR,px4,px42,npn,Rxx,curx4)

curX = abs(rq1 - rq2)/sqrt(2.0)
if (curX .lt. curX1) then
  x12 = curX1 * curX1
  pola = 3*curA/(x12*x12) - curB/(4*x12)
  polb = curB/8 + pola*x12
  polc = (curB - 8*pola*x12)/(24*x12*x12)
  x2 = curX * curX
  pot6 = cural + curA - polb*x2 + pola*x2*x2 + polc*x2**3
else
  pot6 = cural + 0.5*curB*(curX - curX1)**2 +
$   curx4*(curX-curX1)**4
  endif
  pot6 = pot6/96.485/27.211608
else
  pot6 = 0.0
endif
return
end
c *****
  subroutine Initpot
c *****
  implicit real (a-h,o-z)
  parameter(npn = 21)
c   npn is number of nodes for cubic spline interpolation
  common/splpot/pX1(npn),pal(npn),pA(npn),pB(npn),pR(npn),px4(npn),
$ pX12(npn),pal2(npn),pA2(npn),pB2(npn),px42(npn),
$ pR2(npn),pAA(npn),pXX1(npn),pAA2(npn),pXX12(npn)

  real py1, pyn

  write(*,*) 'Reading potential parameters'

  open(2,file = 'pot.par')
  read(2,*) n
  do 10 i = 1, n
10   read(2,*) pR(i),pX1(i),pal(i),pA(i),pB(i),px4(i)

  py1 = 1.0e40
  pyn = py1
c   use 'natural' splines - 2nd deriv = 0 at endpoints

  call spline(pR,pX1,npn,py1,pyn,pX12)
  call spline(pR,pal,npn,py1,pyn,pal2)
  call spline(pR,pA,npn,py1,pyn,pA2)
  call spline(pR,pB,npn,py1,pyn,pB2)
  call spline(pR,px4,npn,py1,pyn,px42)

```



```

do 20 i = 1, npn-3
  pAA(i) = pA(i+3)
  pXX1(i) = pX1(i+3)
  pR2(i) = pR(i+3)
20 continue

call spline(pR2,pAA,npn-3,py1,pyn,pAA2)
call spline(pR2,pXX1,npn-3,py1,pyn,pXX12)
return
end
c *****
c  subroutine spline(x,y, n, yp1,ynpn,y2)
c *****
c
c  procedure translated to FORTRAN from
c  spline.p (Pascal)
c  From Numerical Recipes
c
c  parameter(npn = 21)
c  implicit real (a-h,o-z)
c  real x(npn),y(npn),y2(npn),yp1,ynpn
c  integer n

c
c  real p,qn,sig,un
c  real u(npn)

c
c  if (yp1 .gt. 0.99e30) then
c    y2(1) = 0.0
c    u(1) = 0.0
c  else
c    y2(1) = -0.5
c    u(1) = (3.0/(x(2)-x(1)))*((y(2)-y(1))/(x(2)-x(1))-yp1)
c  endif

c
c  do 10 i = 2, n-1
c    sig = (x(i)-x(i-1))/(x(i+1)-x(i-1))
c    p = sig*y2(i-1)+2.0
c    y2(i) = (sig-1.0)/p
c    u(i) = (y(i+1)-y(i))/(x(i+1)-x(i))
c    $   -(y(i)-y(i-1))/(x(i)-x(i-1))
c    u(i) = (6.0*u(i)/(x(i+1)-x(i-1))-sig*u(i-1))/p
10 continue

c
c  if (ynpn .gt. 0.99e30) then
c    qn = 0.0
c    un = 0.0
c  else
c    qn = 0.5
c    un = (3.0/(x(n)-x(n-1)))*(ynpn-(y(n)-y(n-1))/(x(n)-x(n-1)))
c  endif

```

```

y2(n) = (un-qn*u(n-1))/(qn*y2(n-1)+1.0)

do 20 k = n-1, 1, -1
20  y2(k) = y2(k)*y2(k+1)+u(k)

return
end

c *****
c  subroutine splint(xa,ya,y2a,n,x,y)
c *****
c
c  FORTRAN version of subroutine splint
c  Pascal version is splint.p
c  From Numerical Recipes
c
c  implicit real (a-h,o-z)
c  parameter(npn = 21)
c  real xa(npn),ya(npn),y2a(npn),x,y

real h,b,a

klo = 1
khi = n
10  if ((khi-klo) .gt. 1) then
    k = (khi+klo)/2
    if (xa(k) .gt. x) then
        khi = k
    else
        klo = k
    endif
    goto 10
endif

h = xa(khi)-xa(klo)
if (h .eq. 0.0) then
    write(*,*) 'pause in routine splint'
    write(*,*) ' ... bad xa input'
    stop
endif

a = (xa(khi)-x)/h
b = (x-xa(klo))/h
y = a*ya(klo) + b*ya(khi) +
$  ((a**3-a)*y2a(klo) + (b**3-b)*y2a(khi))*(h*h)/6.0
return
end

```

2.4.9 bndhbr.f

```
c _____
c file bndhbr.f
c _____
c...program to calculate the bending eigenvalue for an abc triatomic
c...in the harmonic approximation.
c Subroutines provided by and written by Prof. J. M. Bowman
c
c Includes :
c   bndh
c   dpesdr
c
c Uses leps3d2.f for potential
c Use with ad3d4.f or higher !
c
c...the bend energy bndh is calculated in the harmonic approximation
c...given by:  $bndh = \hbar \omega (1.0 + nb)$  (in atomic units). this
c...is specific for a triatom with a linear minimum, i.e., bend is doubly
c...degenerate. for a discussion of this and a test of the method see
c...j.m. bowman, chem. phys. lett. 124 (1986) 263.
c
c...rac=rab+rbc for linear geom.
c...gam1 is abc bond angle. Bond lengths Rab, Rbc are input in au
c...assume that potential is calculated in au
c
c some results you should get in tests:
c
c for chlcl bcmr leps surface—rhcl=2.77 au at saddle point and
c ground bend energy should be 509 cm-1 (plus or minus 2 cm-1)
c
c for h+h2 on lsth surface at saddle point rhh = 1.757 au
c ground bend should be 906 cm-1.
c _____
c   function bndh(Rab, Rbc)
c _____
c   implicit real (a-h,o-z)
c   common/masses/aMa(5)
c
c   real ma,mb,mc,nb, h, pi, amu
c   real Rac, g, omegb, dpg1, fgam, deriv
c   real r3save, gam1
c
c   h = 0.01
c   pi = 3.14159265
c   amu = 1822.887
c   ma = aMa(1) * amu
c   mb = aMa(2) * amu
c !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```

      mb = 1.5 * amu
c #####
c      Use this JUST for 3D BrHBr- and BrDBr- !!!
c
      mc = aMa(3)* amu
      nb = 0.0

      Rac = Rab + Rbc
c
c...calculate the g-matrix element
c
      g = 1.0/(ma*Rab*Rab) + 1.0/(mc*Rbc*Rbc) +
$      ((1.0/Rab + 1.0/Rbc)**2)/mb
c
c...calculate the second derivative of the potential wrt gamma using
c...first derivative of pot wrt Rac holding Rab and Rbc fixed.
c...calculated in dpesdr and finite difference approx and chain rule.
c
      r3save = Rac
      gam1 = pi+h
      Rac = sqrt(Rab*Rab + Rbc*Rbc - 2.0*Rab*Rbc*cos(gam1))

c
c now get the first derivative of the potential wrt Rac
c
      deriv = dpesdr(Rab, Rbc, Rac)

      Rac = r3save
      dpg1 = deriv*sin(gam1)*Rab*Rbc/Rac
      fgam = dpg1/h

      if (fgam .ge. 0.0) then
        omegb = sqrt(g*fgam)
        bndh = omegb*(1.0+nb)
      else
        bndh = 0.0
      endif

      return
      end
c -----
      function dpesdr(Rab, Rbc, Rac)
c -----
c...obtain numerical first derivative
c...of potential surface wrt r(3)
      implicit real (a-h,o-z)
      real h,e1,e2, Ra(3)

      h = 0.001
      harev = 27.211608
c

```

```

c this is a sample call to the potential
c which returns the energy in au. user supplies this
c in whatever form is handy.
c
  c = 0.52917706/10.0
c
c used to convert from au (ie, a0) to nm
c
Ra(1) = Rab * c
Ra(2) = Rbc * c
Ra(3) = (Rac - h) * c
call doQJsum(Ra,Qsum,aJsum)
e1 = (Qsum - sqrt(aJsum))/harev/96.485
Ra(3) = (Rac + h) * c
call doQJsum(Ra,Qsum,aJsum)
e2 = (Qsum - sqrt(aJsum))/harev/96.485

dpedr = (e2 - e1)/h/2.0

return
end

```

2.4.10 matrix.f

This is the same subroutine as is used in the collinear program and is listed in Appendix A, sec. 1.4.10.

2.4.11 pot.par

```

21
2.6 0.0 -50.0 0.0 5000.0 1300.0
2.8 0.0 -140.0 0.0 4000.0 1600.0
3.0 0.0 -215.0 0.0 2900.0 2100.0
3.29 0.0 -330.5 0.0 800.0 2000.0
3.4 0.31 -343.5 15.0 1600.0 18000.0
3.5 0.465 -360.5 33.5 1100.0 62000.0
3.6 0.541 -370.0 51.0 800.0 22000.0
3.7 0.613 -374.9 84.0 900.0 6000.0
3.85 0.719 -376.8 135.0 1000.0 2200.0
4.0 0.826 -377.0 170.0 1050.0 1200.0

```

4.2 0.968 -377.1 217.0 1080.0 900.0
4.7 1.322 -377.3 300.0 1230.0 300.0
5.2 1.676 -377.5 360.0 1220.0 100.0
5.7 2.030 -377.65 370.0 1180.0 50.0
6.2 2.3835 -377.8 376.0 1150.0 20.0
7.0 2.949 -378.0 377.0 1130.0 5.0
8.0 3.656 -378.15 377.8 1130.0 3.0
9.0 4.363 -378.25 378.0 1130.0 2.0
10.0 5.070 -378.28 378.1 1130.0 1.0
11.0 5.778 -378.3 378.2 1130.0 0.5
12.0 6.485 -378.31 378.3 1130.0 0.2

3. The Simulation Program sim1

3.1 Running the sim1 Program

The sim1 simulation program convolutes a stick photoelectron spectrum with a resolution function and plots it. The stick spectrum is in the file fort.7, and sim1 is designed to use the fort.7 files produced by ad2d7 and ad3d5. The user is prompted for the type of energy scale to be used (electron kinetic energy or scattering energy), the x axis range, and the resolution (fixed resolution, or the instrumental resolution function (see Ch. 2), if the electron kinetic energy scale is used. The program produces a 1000 point simulation, saves it (x, y pairs) in the file fort.11, and then plots it by remotely running the program plot2d on the computer violet. A tek4014 (or equivalent) terminal should be used if the spectrum is to be plotted on the screen.

3.2 The File sim1.f

```
PROGRAM SIM1
c
c Convolute line intensities in 3D simulation (from
c scat3d4) with Gaussians to produce a simulated spectrum.
c
c 1. Reads fort.7 output file from scat3d4.
c 2. Prompts for x axis (eKE or scattering energy).
c 3. X axis range.
c 4. Gaussian width.
c
implicit real (a-h,o-z)
```

```

parameter(nmax=7000,nsim=1000)
real nrg(nmax)
common/data/fcf(nmax),nrg,n
common/sim/simnrg(nsim),size(nsim)

call readin
write(*,*) 'X axis in 1) eKE or 2) scattering energy'
read(*,*) ixscale
if (ixscale .eq. 2) then
  do 10 i = 1, n
10   nrg(i) = 1.0 - nrg(i)
     xs = nrg(1)
     xl = nrg(n)
  else
    write(*,*) 'eKE = A - Scattering energy, where'
    write(*,*) 'A = hv + Opt (HY) - Do (XHY-) - EA(X)'
    write(*,*) 'A = 1.717 (BrHBr)'
    write(*,*) ' 1.678 (BrDBr)'
    write(*,*) ' 2.231 (BrHI)'
    write(*,*) ' 1.006 (IHI)'
    write(*,*) ' 0.967 (IDI)'
    write(*,*) 'Enter A'
    read(*,*) x
15   nrg(i) = x - 1.0 + nrg(i)
     xs = nrg(n)
     xl = nrg(1)
  endif
  write(*,500) xs,xl
500 format('Data is from ',f5.3,' to ',f5.3,' eV')
  write(*,*) 'Xmin (eV)'
  read(*,*) xmin
  write(*,*) 'Xmax (eV)'
  read(*,*) xmax
  write(*,*) 'FWHM of Gaussians (in meV) (0 for expt. resolution)'
  read(*,*) res
  width = res/2.35482e3
  call dosim(xmin,xmax,width)
  call savesim
  write(*,*) 'Plot (1 = yes) ?'
  read(*,*) iplot
  if (iplot .eq. 1)
    $ i = ishell('cat fort.11 | rsh violet plot2d -y 0.0 1.05')
  stop
end

```

c

c


```

c
subroutine readin
implicit real (a-h,o-z)
parameter(nmax=7000,nsim=1000)
real nrg(nmax)
common/data/fcf(nmax),nrg,n

read(7,*) i
read(7,*) n
write(*,*) 'Reading ', n, ' lines from fort.7'
if (n .gt. nmax) then
  write(*,*) 'ERROR : cannot read all ',n,' lines in file ',
$         ' can only read ',nmax
  stop
endif
500 format(i3)
do 10 i = 1, n
c   write(*,*) i
   read(7,510) fcf(i)
   read(7,*) nrg(i)
10 continue
close(7)
510 format(f6.4)
write(*,*) n, ' lines read from fort.7'
return
end

```

c

c-----

c

```

subroutine dosim(xmin,xmax,width0)
parameter(nmax=7000,nsim=1000)
real nrg(nmax)
common/data/fcf(nmax),nrg,n
common/sim/simnrg(nsim),simint(nsim)

delx = (xmax-xmin)/(nsim - 1)
do 10 i = 1, nsim
  simnrg(i) = xmin + delx*(i-1)
  simint(i) = 0.0
10 continue

do 30 i = 1, nsim
  sum = 0.0
  if (width0 .eq. 0.0) then
    width = sqrt(25.0 + 180.0*simnrg(i)) / 2.35482e3
  else
    width = width0

```

```

endif
do 20 j = 1, n
  x = (simnrg(i) - nrg(j))/width
  z = x*x
  if (z .lt. 13.8) sum = sum + fcf(j)*exp(-z/2.0)/width
c    only keep contributions > 0.001
20  continue
  simint(i) = simint(i) + sum
30  continue
return
end

c
c -----
c
subroutine savesim
parameter(nmax=7000,nsim=1000)
common /sim/simnrg(nsim),simint(nsim)

smax = simint(1)
do 5 i = 2, nsim
  if (simint(i) .gt. smax) smax = simint(i)
5  continue
write(11,500) (simnrg(i),simint(i)/smax,i = 1,nsim)
close(11)
500 format(f6.4,' ',f6.4)
return
end

```

Appendix B: Tenure, The Data Acquisition Program for the Fixed-Frequency Photoelectron Spectrometer

1. Introduction

The tenure program is used for data acquisition, display, and preliminary manipulation on the fixed-frequency photoelectron spectrometer. This experiment was used to acquire the data discussed in Chapters 3 and 5 and most of the data discussed in Chapter 4. The program acquires data using a LeCroy TR8828C Transient Digitizer with LeCroy MM8104 Memory Module and LeCroy 6010 Magic Controller. The tenure program can display data on a video terminal or plot data on a Hewlett-Packard HP7470A plotter. The program can also convert raw time-of-flight data to electron kinetic energy for a photoelectron spectrum, or to masses for a mass spectrum and can manipulate data by smoothing it and subtracting background noise from the raw photoelectron spectrum. The tenure program is written in Borland Turbo Pascal 3.01A and runs on an IBM AT (or equivalent) computer. Offspring of this program are used by Prof. Irene Waller (U. of British Columbia) and by Prof. Mitchio Okumura (Calif. Institute of Technology), where Alan Frankel has extensively modified the program (called TOFView) and converted it to Turbo Pascal 5.0.

The tenure program is intended to be fairly user-friendly. The

various data-acquisition parameters are displayed on the text screen and can easily be set by the user by using the \uparrow and \downarrow keys to select a parameter and then modifying the parameter by using the \rightarrow and \leftarrow keys to increment and decrement the parameter, or by typing in $\langle\text{CR}\rangle$ followed by the new value of the parameter and $\langle\text{CR}\rangle$. This portion of the program was adapted from a program written for data acquisition on an Ion Cyclotron Resonance Spectrometer by Dr. Thomas Buckley of the National Institute of Standards and Technology. Various commands can also be entered using the $\langle\text{Alt}\rangle$ key plus another key. For example, $\langle\text{Alt}\rangle$ E starts a new experiment. These commands are listed on two help screens ($\langle\text{Alt}\rangle$ H displays these screens, the files HELP1.PAR and HELP2.PAR) and are also described below. When data is being collected, it is displayed on the graphics screen as it comes in. The user selects whether to plot the raw (time-of-flight) data or to convert the data to electron kinetic energy or mass (for a mass spectrum) before plotting.

The primary function of the tenure program is data acquisition. The data acquisition hardware consists of a LeCroy TR8828C 200 MHz Transient Digitizer with MM8104 memory. This is controlled by a LeCroy 6010 Magic Controller, which is also used as a CAMAC crate controller and fast averager for the 8828. Communications with the crate controller are through a National Instruments GPIB-PCIIA interface card. The hardware involved in the experiment is described in Chapter 3 and, in much greater

detail, in Alexandra Weaver, Ph. D. Thesis, University of California, Berkeley (1991). The timing and triggering of the electronics used for data acquisition are accomplished in hardware (using a Stanford Research Systems DG535 Pulse Generator) so, while the experiment is running, the software passively acquires data, rather than control the experiment. Before acquiring data, the program sets up the initial operating parameters for the transient digitizer. These include the active memory size (16 or 32K bytes - 32K is used), number of channels (settable by the user, up to 2500), fraction of pretrigger samples (0/8 to 8/8 is allowed, in steps of 1/8. 0/8 is used), sampling period (5, 10, 20, 40, 80, 160, or 320 ns, settable by the user), and voltage offset (settable by the user). Initial values of these parameters (and other default operating parameters) are read from the file `expt.par` when the program is started. The program also sets up the parameters that determine how often the program reads the averaged data from the Magic Controller. In each scan, the Transient Digitizer receives a trigger pulse (confusingly called the STOP pulse in LeCroy documentation) that initiates data acquisition. The data is acquired and is transferred to the Magic Controller, where it is added to existing data. After a certain number of scans (settable by the user), the summed data is read from the Magic Controller and it is cleared. If the data is averaged for too many scans, overflow can cause the data to be corrupted. Thus, the data is usually averaged for about 250 scans before it is averaged. At the typical

20 Hz rep rate of the experiment, this means data is read about every 12 s.

This slow reading rate allows the computer to perform some data manipulation and to plot the results during the experiment. The data is displayed on the screen using subroutines in the file `rmgrap15.pas`. These subroutines use graphics primitives (ie, subroutines that plot lines, characters, etc.) developed by W. Polik (available through the University of California, Berkeley in the file `graphlib.tgt`). The file `graphli2.tgt` contains some of these subroutines, and is used in lieu of the complete (and significantly longer) version due to limitations on the amount of code space in a single program in the Turbo Pascal 3.0 environment.

The user specifies whether the raw time-of-flight data or the converted data should be plotted. Scans shorter than 10 μ s are assumed to be photoelectron spectra and the flight time is converted to electron kinetic energy via

$$eKE_{\text{raw}} = \frac{2,843,174 \text{ l}^2}{t - t_0} \quad (1)$$

and

$$eKE = eKE_{\text{raw}} + 5.4863 \times 10^{-4} \frac{U}{M} \quad (2)$$

where eKE_{raw} is the electron kinetic energy (in eV) without the correction for the center of mass velocity of the beam, eKE includes this correction, l is the length of the flight tube (in m), t is the electron flight time (in ns) and t_0

is the offset between the ion-laser interaction and the transient digitizer trigger pulse. The ion beam energy (in eV) is U , and the mass of the ion (in amu) is M . Note that l and t_0 are treated as adjustable parameters in calibrating the photoelectron spectrum (see below). Scans longer than 10 μ s are assumed to be mass spectra and the flight time is converted to mass via

$$M = U \left(\frac{t - t_{0,m}}{72,024.3 d} \right)^2 \quad (3)$$

where $t_{0,m}$ is the mass spectrum trigger offset (in ns) and d is the length of the mass spectrometer in m. Both $t_{0,m}$ and d are adjustable parameters used to calibrate the mass spectrum (see below). While the experiment is running, only a few keyboard commands are active. $\langle \text{Alt} \rangle L$ is a toggle that determines if the data points should be plotted with dots or lines, and F9 and F10 are seldom-used commands that select which graphics screen to display. The IBM AT has two graphics screens, so data is plotted on the screen that is not currently being displayed, then the two screen are switched. Thus the user does not see the data being plotted. In addition, the user can stop the experiment by $\langle \text{Alt} \rangle S$. Once the experiment has stopped, the \leftarrow and \rightarrow keys can be used to move the cursor to a desired data point a point at a time, while $\langle \text{Shift} \rangle \leftarrow$ and $\langle \text{Shift} \rangle \rightarrow$ move the cursor in 10 point increments, and $\langle \text{Ctrl} \rangle \leftarrow$ and $\langle \text{Ctrl} \rangle \rightarrow$ move the cursor in 100 point increments. Once the cursor is on the desired point, its data value (along with those of neighboring points) can be determined with $\langle \text{CR} \rangle$. The data

can be replotted by reading data (<Alt> R), then typing <CR> at the filename prompt. <Alt> T switches the display to the text screen, where the experimental parameters can be changed, and <Alt> G switches to the graphics screen. The F1 key exits the program.

The mass and electron kinetic energy scales can be calibrated within the program. The mass scale is calibrated by converting to mass (where the spectrum is plotted using the most recent values of $t_{0,m}$ and d), then choosing two mass peaks to use to calibrate the mass scale. The calibration is more accurate if the masses of these peaks are well separated (ie, don't calibrate on $^{35}\text{Cl}^-$ and $^{37}\text{Cl}^-$). The user moves the cursor to one of the peaks (the order in which they are chosen doesn't matter), presses <Alt> 1, and types in the mass at the prompt. He then moves the cursor to the second peak, presses <Alt> 2, and enters the mass at the prompt. The calibration parameters $t_{0,m}$ and d are then calculated and displayed. The length d should be about 1.7 m. The electron kinetic energy subroutine was written by Steven Bradforth. In order to calibrate the energy scale, one first takes the photoelectron spectrum of two (or more) of the halogens F^- , Cl^- , Br^- , I^- and measures the time-of-flight of transitions to the $^2\text{P}_{3/2}$ ground and $^2\text{P}_{1/2}$ excited states. The calibration is accomplished by pressing <Alt> C and entering the appropriate times of flight in the table. The calibration parameters are determined by linear least squares using the subroutine fit from Ch. 14.2 of Numerical Recipes: The Art of Scientific Computing by W.

H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (Cambridge U. Press, Cambridge), 1989.

The tenure program contains several file-manipulation features. <Alt> D is used to change the default directory, <Alt> W saves the experimental spectrum to a file. <Alt> R reads an experimental file (overwriting existing data) and <Alt> A adds a data file to existing data. Files should only be added in the time-of-flight display. If the electron kinetic energy is being displayed, <Alt> A will add a data file to existing data, determine the baseline of the data (the average of the 10 points at highest electron kinetic energy) and will subtract this from the entire data set. This is done because the intensity of a point with electron kinetic energy E is given by

$$I(E) \propto \frac{i(t)}{\left(\frac{dx}{dt}\right)} \quad (4)$$

$$I(E) \propto i(t) E^{-3/2}$$

where $i(t)$ is the intensity in the time-of-flight spectrum, so a small offset in the baseline is magnified at low electron kinetic energy. The photoelectron spectrum (in electron kinetic energy) can be saved with <Alt> U. This file cannot be reread by the program, so this is not a substitute for saving the file with <Alt> W, but is intended to save files that can then be read into sophisticated plotting programs such as Freelance (on the IBM) or TempleGraph (on the Sun 360). <Alt> P writes the current operating

parameters (number of scans, ion mass, energy calibration value, etc.) to the file `expt.par`. These will be the initial default values used the next time the program is run.

The data can also be plotted on a HP7470A plotter. The file `plotlib9.pas` contains both the plotting primitives (which translate simple commands like "draw a line from x_1, y_1 to x_2, y_2 " to HPGL and send it to the plotter) and the subroutines that use these primitives to plot the spectrum. Data can be plotted using lines (F7) or points (F8). Plotting data using points is not recommended, as it tends to destroy the tips of plotter pens.

The tenure program also has subroutines to modify the experimental data. These fall into two classes: smoothing and background subtraction. Pressing F3 initiates the smoothing subroutine. The user is prompted for the desired resolution (in meV). The spectrum will be convoluted with a Gaussian of the specified FWHM. Note that, due to the time-of-flight \rightarrow energy conversion, more points are averaged at low eKE than at high eKE. The smoothed intensity $I(E)$ of a point at energy E is given by

$$I(E) = \frac{\sum_i I^*(E_i) e^{-\alpha(E - E_i)^2}}{\sum_i e^{-\alpha(E - E_i)^2}} \quad (5)$$

where $I^*(E_i)$ is the unsmoothed intensity and

$$\alpha = \frac{2.773 \times 10^6}{res^2}$$

where res is the desired resolution in meV.

Background signal due to scattered light can be subtracted from the spectrum in two ways. Conceptually, the simplest way is to take the photoelectron spectrum with the laser interacting with the ion of interest, then take it again with the laser missing (in time) the ion, and subtract the two results. This is accomplished by setting the Background Subtraction flag to Yes before running the experiment. The program takes the desired number of points per scan (usually 250) with the laser hitting the ion of interest, then sets the laser timing 20 μs earlier (by changing the timing of the appropriate channel on the pulse generator), takes another 250 shots of data, subtracts the two and stores (and averages) the result. This is mainly used if the signal/background ratio is small (< 1) and one wishes to make sure that there really is signal. It takes twice as long to obtain a spectrum in this fashion, and the resulting signal-to-noise ratio is poorer than it would be without background subtraction (although the "noise" is now due to shot noise, rather than to background from scattered light). A more efficient way to remove the background due to scattered light (devised by Steven Bradforth, who also wrote the subroutines) takes advantage of the observation that this background varies smoothly in time and the shape of the background spectrum does not change from day to day (unless the laser is re-aligned). Thus, one takes a well-averaged background spectrum

(averaging 250,000 shots or more) and saves it. This file could be scaled and subtracted from subsequent experiments, but a better approach is to use Wiener filtering and a Fourier Transform to smooth the background spectrum, save this smoothed background, and subtract it from subsequent spectra. The Fourier Transform smoothing is invoked with <Alt> F and uses subroutines from Ch. 12 of Numerical Recipes. Wiener filtering is described in Ch. 12.6 of Numerical Recipes. <Alt> N is then used to scale a background spectrum and subtract it from the experimental spectrum.

The files that comprise the tenure program are briefly described in section 2 and are listed in section 3.

2. Description of Files

mgplt20.pas contains the main program segment and the subroutines used for data acquisition.

plotlib9.pas subroutines to plot data on a HP7470A plotter. It also contains subroutines to convert data from time of flight to electron kinetic energy or mass and to scale data for plotting.

rmgrap15.pas subroutines used to plot data on the screen (as the name implies), but also contains the key-handling subroutine (which determines what the program should do in response to the user hitting a key), initialization, parameter display,

help, Gaussian smoothing, and file-handling subroutines. It also contains the subroutine used to clear a graphics screen. This subroutine is hardware dependent, so if the graphics card is changed, this subroutine may have to be modified.

calib1.pas subroutines used to calibrate the time of flight → electron kinetic energy conversion.

scale.pas subroutine used to subtract background due to scattered light from a spectrum.

ftshort.pas subroutines for Wiener filtering of a background spectrum.

four1.pas used by subroutines in ftshort.pas.

modfile.pas defines types used by Numerical Recipes subroutines.

tp3decl3.pas shortened version of tpdecl3.pas, National Instruments GPIB subroutines.

graphli2.tgt shortened version of graphlib.tgt, graphics subroutines written by William Polik.

expt.par contains default experimental parameters. Each parameter has six lines - the parameter name, units, current default value, maximum value allowed, minimum value allowed, and amount to increment value.

help1.par and help2.par contain text of help screens.

3. Source Code Listing

3.1 mgplt20.pas

(* mgplt20.pas

INCLUDED NEW TPDECL code as TP3DECL3.PAS when changed motherboard on Lab computer. Must run computer at 8 MHz clock speed with current GPIB-PC board hardware. Also installed new tp3ib.com, gpib.com and new set of GPIB commands into c:\GPIB-PC. That version of GPIB software (C11, C7 version of TURBO) actually can handle clock speeds and boards up to 486/33MHz. 10/30/91 SEB, DWA

LeCroy 6010 Magic Controller and 8828 Transient Recorder control program. The 6010 is used as a fast averager for the 8828. All communications to the 8828 are through the 6010. ** New slot location for 8828. *)

(* Version 1.00

RBM 10/17/88

SEB 5/2/90

Electron Energies plotted on screen and on plotter

Scans <= 25000 points are allowed

Mass correction for energy scale

Default directory

Fourier Transform and Data Store as function of Energy Added

Chaining Analysis program removed *)

program main (input,output);

(* New include file here 10/30/91 *)

(\$I \turbo\gpib\tp3decl3.pas)

Const

ERR = \$8000; (* Error Detected *)

MAXSIZE = 2010; (* maximum size of data array *)

Text = FALSE;

Graphics = TRUE; (* used for Mode (ie, if Mode = TRUE, then Graphics are being displayed) *)

LabelColor = LightRed;

AxesColor = LightRed;

DataColor = LightGreen;

CursorColor = White;

DoneColor = LightMagenta;

NumPars = 12; (* Number of experimental parameters *)

TimeScan = 0; (* Constants for ScanType *)

MassScan = 1;

EnergyScan = 2;

DivFactor = 1; (* Number from transient digitizer that corresponds to one electron hitting detector *)

Type

```
gldarray = array [1..2048] of real;
glndata = ARRAY [1..20] OF real;
BufString = string[50];
rmbuf = array[1..MAXSIZE] of real;
rminbuf = array[1..MAXSIZE] of integer;
YNStringType = array[0..1] of string[5];
DVar = record
    (* display variable *)
    What : string[50];    (* name of variable *)
    Units : string[5];   (* units of variable *)
    Value : real;        (* current value *)
    UpLimit : real;      (* upper limit *)
    LowLimit : integer;  (* lower limit *)
    Incr : integer;      (* increment *)
end;
```

Var

```
Param : array[1..NumPars] of DVar; (* experimental parameters *)
DlyVal : array[0..6] of integer; (* time per channel data for 8828 *)
devname : ibstring; (* device name *)
pulse, magic : integer; (* device descriptor *)
rdbuf, rtest, rtest2 : BufString; (* read data buffer *)
j, terr, result, k, numblocks : integer;
hbuf : io1buf;
data : rmbuf;
Mode : boolean; (* whether text or graphics is shown *)
ExpDone : boolean; (* true if experiment done *)
ScanNumLo, ScanNumHi : integer; (* ScanNum = 10000*Hi + Low *)
CurrPage : integer; (* graphics page currently displayed - 0 or 1 *)
outbuf : BufString;
maxval, minval : real; (* min and max values in data array *)
lastpt, currpt : integer; (* previous and current cursor location *)
ValuesShown : boolean; (* TRUE if data values are being displayed *)
ConnectPoints : boolean; (* TRUE = connect points in graph *)
Background : boolean; (* TRUE = background scan *)
BackData : rmbuf; (* background data array *)
MassLength, MassSlope, MassInt : real; (* Time = Slope*sqrt(Mass) + Int *)
YNString : YNStringType;
Time1, Time2, Mass1, Mass2 : real; (* time, mass pairs for mass calibration *)
PulseDelay : real; (* trigger delay (nsec) for energy scans *)
PulseSlope : real;
LasereV : array [0..8] of real; (* laser energies *)
LaserGas : array [0..8] of String[10]; (* laser types *)
MaxEnergy, MinEnergy : real;
ScanType : integer; (* time/mass/energy scan *)
Dots : boolean; (* true if plotter does dots *)
LastFile : BufString; (* name of last file loaded or saved *)
oldtotsum, totsum : real;
Directory : BufString; { default directory }
FileVar : file; (* for chain file *)
ExpSaved : Boolean; (*Flag for prevention of data loss*)
```

```

InitSub: Boolean;
function itohex (i:integer):str4;
(* Convert integer to hex string. *)
(* Useful for printing hex value of IBSTA *)
var k : integer; s : string[4]; nib : integer;
begin
k := 12; s := "";
while (k >= 0) do
begin
nib := (i shr k) and $f;
if ($A <= nib) and (nib <= $F)
then nib := nib + $37
else nib := nib + $30;
s := s + chr(nib);
k := k - 4;
end;
itohex := s;
end;
procedure ExitGraphics; forward;

($I \turbo\graphics\graphli2.tgt) (* include Polik's graphics routines *)

procedure prvars (ibsta:integer;iberr:integer;ibcnt:integer);
(* print gpib status *)
var stas : string[4];
begin
(* if mode = graphics then ExitGraphics;
stas := itohex(ibsta);
writeln (Lst,'ibsta=0x',stas,'iberr=0x',iberr,'ibcnt=0x',ibcnt);
writeln(Lst,outbuf); *)
writeln(#7);
end;

procedure finderr;
begin
(* This routine would notify you that
the ibfind call failed, and refer you
to the handler software configuration
procedures. *)
writeln (' Find error Name: ', devname);
end;

procedure error;
begin
(* This routine would, among other
things, check iberr to determine
the exact cause of the error
condition and then take action
appropriate to the application.
For errors during data transfers,
ibcnt may be examined to determine

```



```

        the actual number of bytes
        transferred. *)
    (* writeln (' Error');
    *) prvars (ibsta,iberr,ibcnt);
end;

procedure magerr;
var sprchk : str4;
begin
    (* This routine would analyze the fault
    returned in the magic controller's status
    byte and take appropriate action. *)

    writeln (' MAGIC error');
    prvars (ibsta,iberr,ibcnt);
end;

procedure mwrite(wrbuf: BufString);
    (* transfer string to character array, send to magic controller,
    and check for gpib errors *)

var rlen, i : integer;
begin
    rlen:= length(wrbuf);
    outbuf := wrbuf;
    for i:= 1 to rlen do ibbuf[i]:= wrbuf[i];
    ibwrt(magic,ibbuf,rlen);
    if ((ibsta AND ERR) <> 0) then error;
end;

procedure mread(numchars : integer);
    (* read numchars characters from magic controller,
    transfer to string, and check for gpib errors *)

var i : integer;
begin
    ibrd(magic,ibbuf,numchars);
    for i:= 1 to ibcnt do rdbuf[i]:= ibbuf[i];
    if ((ibsta AND ERR) <> 0) then error;
end;

procedure pread(numchars : integer);
    (* read numchars characters from pulse generator,
    transfer to string, and check for gpib errors *)

var i : integer;
begin
    ibrd(pulse,ibbuf,numchars);
    for i:= 1 to ibcnt do rdbuf[i]:= ibbuf[i];
    if ((ibsta AND ERR) <> 0) then error;
end;

```

```

end;

procedure pwrite(wrbuf: BufString);
  (* transfer string to character array, send to pulse generator,
  and check for gpib errors *)

  var rlen, i : integer;
  begin
    rlen:= length(wrbuf);
    outbuf := wrbuf;
    for i:= 1 to rlen do ibbuf[i]:= wrbuf[i];
    ibwrt(pulse,ibbuf,rlen);
    if ((ibsta AND ERR) <> 0) then error;
  end;

procedure char_to_int(var res : integer; var errcode : integer);
  (* Convert a character string to an integer.
  The integer is from the first numeric character
  to the last. Note that 0-9 are the only numeric
  characters recognized. ('-', '.', and 'E' are not
  recognized as numeric characters *)

  var i,j : integer;
  resbuf : BufString;

function is_num(c : char) : boolean;

  begin
    if c in ['0'..'9'] then
      is_num := TRUE
    else is_num := FALSE;
  end;

  begin
    i := 1;
    j := 1;
    resbuf := ' ';
    while NOT (is_num(ibbuf[i])) i := i + 1;
    while (is_num(ibbuf[i]))
      begin
        resbuf[j] := ibbuf[i];
        j := j + 1;
        i := i + 1;
      end;
    val(resbuf,res,errcode);
  end;

procedure char_to_real(var res : real; var errcode : integer);
  (* Convert a character string to an integer.
  The integer is from the first numeric character after a comma
  to the last. Note that 0-9 are not the only numeric

```

characters recognized. ('.', and 'E' are also
recognized as numeric characters *)

```
var i,j : integer;  
    resbuf : BufString;
```

```
function IsComma(c : char) : boolean;
```

```
begin  
    if c = ',' then  
        IsComma := TRUE  
    else IsComma := FALSE;  
end;
```

```
function IsRealNum(c : char) : boolean;
```

```
begin  
    if (c in ['0'..'9']) OR (c = '.') OR (c = 'E') OR (c = 'e') then  
        IsRealNum := TRUE  
    else IsRealNum := FALSE;  
end;
```

```
begin  
    i := 1;  
    j := 1;  
    resbuf := ' ';  
    while NOT (IsComma(ibbuf[i])) i := i + 1;  
    while NOT (IsRealNum(ibbuf[i])) i := i + 1;  
    while (IsRealNum(ibbuf[i]))  
    begin  
        resbuf[j] := ibbuf[i];  
        j := j + 1;  
        i := i + 1;  
    end;  
    val(resbuf,res,errcode);  
end;
```

```
($I c:\turbo\gpib\plotlib9.pas) (* include plotting routines *)  
($I c:\turbo\gpib\modfile.pas)
```

```
Function Exist(FileName : BufString) : boolean; forward;  
procedure CheckDir(var FileName : BufString); forward;  
Procedure InitData; forward;  
($I c:\turbo\gpib\calib1.pas)(* Calibration routine*)  
($I c:\turbo\gpib\four1.pas) (* Required for ftshort.pas *)  
($I c:\turbo\gpib\ftshort.pas) (* Short Fourier Transformation for Background Subtraction *)  
($I c:\turbo\gpib\rmgrap15.pas) (* include graphing and keyboard routines *)  
($I c:\turbo\gpib\scale.pas) (* noise subtraction routine *)
```

```
procedure InitPulse;  
    (* initialize Stanford Pulse Generator *)
```

```

begin

  (* Assign a unique identifier to the
  device "PULSE" *)
  devname := 'PULSE';
  pulse := ibfind (devname);

  (* Check for ibfind error. *)
  if (pulse < 0) then
  begin
    writeln('Pulse Find error ');
    finderr;
  end;
end;

procedure InitMag1;
(* initialize 6010 Magic Controller *)
begin

  (* Assign a unique identifier to the
  device "MAGIC" *)
  devname := 'MAGIC';
  magic := ibfind (devname);

  (* Check for ibfind error. *)
  if (magic < 0) then
  begin
    writeln('Magic Find error ');
    finderr;
  end;

  (* Clear the device. *)
  ibclr (magic);

  (* Check for an error on each GPIB call to
  be safe. *)
  if ((ibsta AND ERR) <> 0) then error;

  rdbuf:= ' ';

end;

(* -----*)
(* Set up transient digitizer *)
(* -----*)

procedure InitMag2;
begin
  ibclr(magic);
  delay(200);
  ibclr(magic);
  delay(200);

```

```

mwrite('ECHO OFF ');
mwrite('A 0 '); (* set secondary address of 8828 *)
mwrite('N 12 '); (* set primary address of 8828 *)
(* AGAIN (8/16/91) AND AGAIN (9/4/91) NEW SLOT NUMBER *)
mwrite('Z '); (* clear 8828 *)

```

```

(* 8828 configuration:
R 9      R 5-R 7      R 1-R 4
Memory Size  Sampling Period  Pretrigger
0 16 K      0 5 nsec      0 0/8 of active memory
1 32 K      1 10          1 1/8
                2 20          2 2/8
                3 40          3 3/8
                4 80          4 4/8
                5 160         5 5/8
                6 320         6 6/8
                7 ext. clock  7 7/8
                        8 8/8

```

Bytes 10-15 should be 0. Byte 8 is ignored.
Normally, pretrigger is 0/8 and memory is 32 K.
Then, the control word is 256 + 16*sampling. *)

```

mwrite('F 26 '); (* enable LAM *)
mwrite('E ');

```

```

(* ----- *)
(*      Set up magic controller      *)
(* ----- *)

```

```

mwrite('COMM_FORMAT OFF,WORD,BINARY ');
mwrite('COMM_BLOCKSIZE 200 ');
mwrite('COMM_ORDER LOFIRST ');
(* mwrite('COMM_DMA OFF '); *)
delay(100);
mwrite('USING 12 '); (* new N number of 8828 *)
mwrite('AVGENABLE ON');
delay(100);
end;

```

```

(* ----- *)
(*      Acquire data      *)
(* ----- *)

```

```

procedure GetScanType;
(* determine type of scan. If it is not a time scan, and it is
longer than 10 usec, then it is a mass scan, otherwise it is an
electron energy scan *)
begin
if Param[6].Value = 1 then
ScanType := TimeScan
else

```

```

begin
  if DlyVal[round(Param[4].Value)] * Param[3].Value > 10000 then
    ScanType := MassScan
  else
    ScanType := EnergyScan;
  end;
end;

procedure FindDelay;
(* find trigger delay for electron energy. This only
  makes sense if the Stanford box is being used to
  generate the stop pulse *)
begin
  pwrite('DT 6 ');
  pread(30);
  char_to_real(PulseDelay,terr);
  PulseDelay := PulseDelay * 1E9;
end;

procedure ChangeMode; forward;

procedure StartScan;
(* tell the 6010 to start a scan *)
begin
  ChangeMode;
  mwrite('AVG START ');
end;

procedure InitScan;
(* initialize 6010 and 8828 in preparation for a scan *)
begin
  InitMag2;
  i := round(Param[5].Value);
  str(i,rtest2);
  mwrite('W ' + rtest2 + ' '); (* set offset *)
  mwrite('F 19 ');
  mwrite('E ');
  i := 256 + 16*round(Param[4].Value);
  str(i,rtest2);
  mwrite('W ' + rtest2 + ' '); (* configure 8828 : pretrigger 0/8,
    requested sampling period, 32K memory *)
  mwrite('F 16 '); (* set control word *)
  mwrite('E ');
  i := round(Param[3].Value) + 10;
  str(i,rtest2);
  mwrite('AVGLEN ' + rtest2 + ' '); (* set # of channels to accumulate *)
  i := round(Param[2].Value);
  str(i,rtest2);
  mwrite('AVGN ' + rtest2 + ' '); (* set # of times to average *)
end;

```

```

procedure InitData;
(* initialize data and Background array *)
begin
  InitSub:=FALSE; (*Subtraction set now redundant, and flag reinitilaized*)
  for i := 1 to MAXSIZE do
    begin
      data[i] := 0.0; (* clear data array *)
      BackData[i] := 0.0;
    end;
  ScanNumLo := 0;
  ScanNumHi := 0;
  LastFile := '';
end;

procedure StartExp;
(* begin an experiment *)
BEGIN
If ExpSaved=FALSE THEN
BEGIN
  GoToText;
  write(#7+'Overwrite Experimental Data from previous Run ????? ');
  readln(rtest2);
  IF (rtest2='Y') OR (rtest2='y') THEN ExpSaved:=TRUE;
END;
IF ExpSaved=TRUE THEN
  begin
    InitData;
    ExpDone := FALSE;
    Mode := GRAPHICS;
    GetScanType;
    if ScanType = EnergyScan then FindDelay;
    CurrPage := 1;
    CurrPt := round(Param[3].Value/2); (* put cursor at center of screen *)
    PulseDelay := Param[10].Value + 40.0;
    PulseSlope := 2843174.0 * Sqr(Param[11].Value);
    oldtotsum := 0;
    totsum := 0;
    InitMag1;
    SetClearOn;
    EnterGraphics;
    SetClearOff;
    DoGraph;
    InitScan;
    Background := FALSE;
    StartScan;      (* do a ChangeMode in this routine *)
  end
ELSE
  BEGIN
    writeln;
    writeln('OLD EXPERIMENTAL DATA RESTORED, PLEASE SAVE. ');
    Delay(2000);
  END;

```

```

    ClrScr;
    ShowLabels;
    END;
END;(*procedure*)

procedure DoScan;
(* do a scan - read data, save in array, call
graphing routine (if necessary) *)
begin
    if 10000.0*ScanNumHi + ScanNumLo < Param[1].Value then
    begin
        result := 0;
        mwrite('AVGDONE '); (* check if done *)
        mread(20);
        char_to_int(result,terr);
        if result > 0 then
        begin
            (* Read Data *)
            numblocks := round(Param[3].Value/100);
            str(round(200*numblocks - 1),rtest);
            mwrite('READ -A,0,' + rtest + ' '); (* read averaged data *)
            for j:= 0 to numblocks-1 do
            begin
                ibrdi(magic,hbuf,200);
                if Background then (* read background data *)
                for k:= 1 to 100 do
                    BackData[k + 100*j] := int(hbuf[k])/DivFactor
                else
                for k:= 1 to 100 do (* read data *)
                    Data[k + 100*j] := -Data[k + 100*j] + int(hbuf[k])/DivFactor - BackData[k +
100*j];
                end;
                if NOT Background then (* update current scan number *)
                    ScanNumLo := ScanNumLo + round(Param[2].Value);
                if ScanNumLo >= 10000 then
                begin
                    ScanNumLo := ScanNumLo -10000;
                    ScanNumHi := ScanNumHi + 1;
                end;
                data[1] := data[2]; (* fudge 1st data pt *)
                { for j:= 3 to 80 do }
                { data[j] := data[1]; }
                if 10000.0 * ScanNumHi + ScanNumLo >= Param[1].Value then
                begin (* finished experiment *)
                    ExpDone := TRUE;
                    writeln(#7);
                    If ScanNumHi >= 1 THEN ExpSaved := FALSE; (* Set flag only for completed
scan over 10K*)
                    Background := TRUE;
                end
                else (* start new scan *)
                begin

```



```

        InitScan;
        StartScan;
    end;
    if (Mode = Graphics) AND (Background OR (Param[7].Value = 0)) then
        begin (* graph data *)
            CurrPage := 1 - CurrPage;
            DoGraph;
        end;
    end;
end;
end;
end;

```

procedure ChangeMode;

(* change 'C' output on Stanford box by 1 usec.
ie, put laser on and off resonance for background
subtraction *)

```

begin
    if Param[7].Value = 1 then
        begin
            pwrite('CS 0;DL 1,0,3;SC 12');
            if Background then
                begin pwrite('IC 1'); pwrite('IC 1'); end (* Return laser to resonance *)
            else
                begin pwrite('IC 0'); pwrite('IC 0'); end; (* Set signal off resonance *)
            pwrite('SC 15');
            Background := NOT Background;
        end;
    end;
end;

```

procedure StopExp; (* halt experiment currently in progress *)

```

begin
    ExpDone := TRUE;
end;

```

procedure DoPlot; (* plot data *)

```

var i,j,k : integer;
begin
    getpen(2);
    i := round(Param[3].Value/5);
    j := round(i/2);
    PlotAxes(0,round(Param[3].Value),j,i,minval,maxval,DlyVal[round(Param[4].Value)]);
    plot_data(1,round(Param[3].Value),minval,maxval,data);
    endplt;
end;

```

procedure DoGraph; (* normalize and graph data *)

```

begin
    SetDrawPage(CurrPage);
    ClearPage(CurrPage);
    ScreenAxes;
    ( maxval := find_max(1,numblocks*100,data); )

```

```

minval := find_min(1,numblocks*100,data);
Justify(1,numblocks*100,minval);
( maxval := maxval - minval; )
maxval := find_max(1,numblocks*100,data);
minval := 0.0;
Invert(1,numblocks*100,maxval);
maxval := find_max(1,numblocks*100,data);
oldtotsum := totsum;
totsum := find_sum(1,numblocks*100,data);
GraphData(1,numblocks*100,0.0,maxval,data);
LabelAxes;
end;

begin ( program )
  InitParam;      (* initialize experimental parameters *)
  InitPlt;        (* initialize plotter *)
  InitPulse;      (* initialize pulse generator *)
  SetClearOff;
  ClrScr;
  ShowLabels;
  ShowParam;      (* display experimental parameters *)

  (* Main Loop *)
  while NOT Done do
    begin
      if KeyPressed then GetKey;
      if NOT ExpDone then DoScan;
    end;
    (* done, so clean up and exit program *)
    if mode = graphics then
      begin
        SetClearOn;
        ExitGraphics;
      end;
    normvideo;
  end.

```

3.2 plotlib9.pas

```

(* PLOTLIB9.PAS - Plotter library routines *)
(* Written by RBM 1/87 *)
(* Last Modified by RBM 12/9/87 *)
(* Version 0.75
Plot Mass Scale
Electron Energy Calcs
Plot Electron Energies with COM correction *)

```

```

Var
plotter : integer; (* device descriptor *)
outstring : BufString;

function COMEnergy(RawE, IonMass : real) : real;
(* Convert raw energy to COM corrected energy *)
begin
  COMEnergy := RawE + 5.4863E-4*Param[12].Value/IonMass;
end;

function Convert(i : integer) : real;
(* Convert time to mass or electron energy *)
var RawEnergy : real;
begin
  case ScanType of
    TimeScan :
      Convert := 0.1*i*DlyVal[round(Param[4].Value)];
    MassScan :
      Convert := sqr((int(DlyVal[round(Param[4].Value)])*(i+8) + MassInt)/MassSlope);
    EnergyScan :
      begin
        RawEnergy := PulseSlope/sqr(int(DlyVal[round(Param[4].Value)])*(i+8) - PulseDelay);
        Convert := COMEnergy(RawEnergy,Param[9].Value);
      end;
  end;
end;

function Intensity(XValue, YValue : real) : real;
(* Convert intensities from time to mass or energy *)
begin
  case ScanType of
    TimeScan :
      Intensity := YValue;
    MassScan :
      Intensity := YValue;
      { Intensity := YValue/sqrt(XValue); } (* M ^ -.5 *)
    EnergyScan :
      If XValue=0 THEN Intensity := YValue ELSE
        Intensity := YValue/(XValue * sqrt(XValue)); (* E ^ -1.5 *)
  end;
end;

type
  DateStr = string[10];

function Date: DateStr; { see DOSFCALL.DOC for documentation }
type
  regpack = record
    ax,bx,cx,dx,bp,si,di,ds,es,flags: integer;
  end;

```

```

var
  reckpt:   reckpt;           (record for MsDos call)
  month,day: string[2];
  year:     string[4];
  dx,cx:    integer;

```

```

begin
  with reckpt do
  begin
    ax := $2a shl 8;
  end;
  MsDos(reckpt);           ( call function )
  with reckpt do
  begin
    str(cx,year);          (convert to string)
    str(dx mod 256,day);    ( " )
    str(dx shr 8,month);    ( " )
  end;
  date := month+'/' + day+'/' + year;
end;

```

```

procedure plt(rmbuf: BufString);
  (* transfer string to character array *)
  var rlen, i : integer;
  begin
    rlen:= length(rmbuf);
    for i:= 1 to rlen do ibbuf[i]:= rmbuf[i];
    ibwrt(plotter,ibbuf,rlen);
    if ((ibsta AND ERR) <> 0) then error;
  end;

```

(* Plotting primitives :
 most of these routines are just mnemonics for
 standard HP Graphics Language commands.
 Refer to HP 7470 manual for more detail on
 individual commands *)

```

procedure getpen(pen : integer);
  (* select current pen *)
  begin
    plt('PU ');
    str(pen,outstring);
    outstring := 'SP' + outstring;
    plt(outstring);
  end;

```

```

procedure setwindow(xs, xl, ys, yl : integer);
  var s1,s2,s3,s4 : BufString;
  (* set plotter window *)
  begin
    str(xs,s1);

```

```

    str(xl,s2);
    str(ys,s3);
    str(yl,s4);
    s1 := s1 + ',' + s3 + ',' + s2 + ',' + s4;
    outstring := 'IP' + s1;
    plt(outstring);
end;

procedure usercoords(xs, xl, ys, yl : real);
var s1,s2,s3,s4 : BufString;
(* set user coords *)
begin
    str(xs:3:3,s1);
    str(xl:4:3,s2);
    str(ys:1:0,s3);
    str(yl:1:0,s4);
    outstring := 'SC' + s1 + ',' + s2 + ',' + s3 + ',' + s4;
    plt(outstring);
end;

procedure moveto(xcoord, ycoord : real);
var rndstring : BufString;
(* raise pen and move it to the selected point *)
begin
    str(xcoord:4:2,outstring);
    str(ycoord:1:5,rndstring);
    outstring := 'PU;PA' + outstring + ',' + rndstring;
    plt(outstring);
end;

procedure quickmoveto(xcoord, ycoord : real);
var rndstring : BufString;
(* don't raise pen, just move to selected point *)
begin
    str(xcoord:4:3,outstring);
    str(ycoord:1:3,rndstring);
    outstring := 'PA' + outstring + ',' + rndstring;
    plt(outstring);
end;

procedure move(xcoord, ycoord : real);
var rndstring : BufString;
(* raise pen, move the selected offset (relative to current point) *)
begin
    str(xcoord:4:3,outstring);
    str(ycoord:6:0,rndstring);
    outstring := 'PU;PR' + outstring + ',' + rndstring;
    plt(outstring);
end;

procedure lineto(xcoord, ycoord : real);

```

```

var rndstring : BufString;
(* lower pen and draw a line from the current pen position
to the desired point *)
begin
  str(xcoord:4:3,outstring);
  str(ycoord:2:5,rndstring);
  outstring := 'PD;PA' + outstring + ',' + rndstring;
  plt(outstring);
end;

procedure quicklineto(xcoord, ycoord : real);
var rndstring : BufString;
(* don't lower pen (pen is assumed down);
draw a line from the current pen position to the desired point *)
begin
  str(xcoord:1:3,outstring);
  str(ycoord:1:5,rndstring);
  outstring := 'PA' + outstring + ',' + rndstring;
  plt(outstring);
end;

procedure line(xcoord, ycoord : real);
var rndstring : BufString;
(* draw a line at the desired offset *)
begin
  str(xcoord:4:1,outstring);
  str(ycoord:6:0,rndstring);
  outstring := 'PD;PR' + outstring + ',' + rndstring;
  plt(outstring);
end;

procedure PlotAxes(xstart, xend, minxtick, maxxtick : integer; ystart, yend : real; timeperpoint :
integer);
(* draw and label axes *)
type
  shortstring = String[5];
var
  i : integer;
  endstr : shortstring;
  numstring : BufString;
  MaxMass,r, MaxEnergy, MinEnergy : real;
begin
  moveto(0,1000); (* Plot axes *)
  lineto(0,0);
  lineto(1000,0);
  moveto(850,1100);
  endstr := #13 + #10 + #3 + ' ';
  plt('SI.12,.18 ');
  plt('LB' + LastFile + endstr);
  plt('LB' + Date + endstr);
  str(Param[1].Value:0:0,numstring);

```

```

plt('LB' + numstring + ' Scans' + endstr);
if Param[7].Value = 1 then
  plt('LBBack. subtraction' + endstr);
str(Param[12].Value:4:0,numstring);
plt('LBFloat : ' + numstring + ' V' + endstr);

MaxEnergy := int(LasereV[round(Param[8].Value)] + 0.99);
if ScanType = EnergyScan then
begin
  str(Param[11].Value:3:2,numstring);
  plt('LLength : ' + numstring + ' cm' + endstr);
  str(Param[10].Value:3:2,numstring);
  plt('LBTo : ' + numstring + ' ns' + endstr);
  usercoords(0.0,MaxEnergy,ystart,1.0);
end
else
begin
  plt('LBLaser : ' + LaserGas[round(Param[8].Value)] + endstr);
  usercoords(Convert(xstart),Convert(xend),ystart,1.0);
end;
  (* Plot Y Axis *)
plt('TL 2 '); (* set initial tick length *)
for i:= 1 to 5 do
begin
  moveto(0.0,0.2*i);
  plt('YT '); (* plot minor tick marks y-axis *)
end;
str(yend:6:0,numstring);
plt('SI.2,.35 '); (* set character size *)
plt('CP-7,-.25 ');
plt('LB' + numstring + #3 + ' '); (* label Y Axis *)
quickmoveto(0,0.3);
plt('DI0,1,SI.3,.5,CP3,1 '); (* write vertically *)
plt('LBCounts' + #3 + ' ');
plt('DI1,0,SI.2,.35 ');
plt('TL1 '); (* set initial tick length *)

  (* Draw X Axis : 3 cases :
  1. Time scan
  2. Mass Scan
  3. Energy Scan *)
case ScanType of
TimeScan : begin
  i := xstart;
  while i <= xend do
  begin
    moveto(Convert(i),0.0);
    plt('XT '); (* plot minor tick marks x-axis *)
    i := i + minxtick;
  end;
  plt('TL2 ');

```

```

i := xstart;
plt('SI.2,,35 ');
if int(xend-xstart+1)*timeperpoint < 5000 then
begin
  while i <= xend do
  begin
    quickmoveto(Convert(i),0);
    str(i*timeperpoint,numstring);
    plt('XT;CP-1.33,-1 ');
    plt('LB' + numstring + '#3 + ' ');
    i := i + majxtick;
  end;
  moveto(Convert(round(xend/2)),0);
  plt('SI.3,,5 ');
  plt('CP-9,-2.5 ');
  plt('LBElapsed Time (nsec)' + '#3+' ');
  plt('SI.2,,35 ');
end
else
begin
  while i <= xend do
  begin
    quickmoveto(Convert(i),0);
    plt('XT;CP-2.33,-1 '); (* plot major tick marks x-axis *)
    str((int(i)/1000)*timeperpoint:2:1,numstring);
    plt('LB' + numstring + '#3 + ');
    i := i + majxtick;
  end;
  moveto(Convert(Round(xend/2)),0);
  plt('SI.3,,5 ');
  plt('CP-9,-2.5 ');
  plt('LBElapsed '+' + '#3+' ');
  plt('LBTime (usec)' + '#3+' ');
  plt('SI.2,,35 ');
end;
end; (* Time Plot *)
MassScan : begin
  MaxMass := Convert(xend);
  case Round(0.5 + MaxMass) of
    0..50 : begin minxtick := 2; majxtick := 5; end;
    51..100 : begin minxtick := 5; majxtick := 10; end;
    101..200 : begin minxtick := 10; majxtick := 20; end;
    201..400 : begin minxtick := 20; majxtick := 50; end;
    401..1000 : begin minxtick := 50; majxtick := 100; end;
    else begin minxtick := 200; majxtick := 500; end;
  end; { case }
  (* Draw Minor Mass Ticks X Axis *)
  plt('TL1 ');
  i := 0;
  while(i <= MaxMass) do
  begin

```



```

        quickmoveto(int(i),0.0);
        plt('XT ');
        i := i + minxtick;
    end;

    (* Draw Major Mass Ticks X Axis *)
    plt('TL2 ');
    i := 0;
    while(i <= MaxMass) do
    begin
        quickmoveto(int(i),0.0);
        plt('XT;CP-1.33,-1 ');
        str(i,numstring);
        plt('LB' + numstring + #3 + ' ');
        i := i + majxtick;
    end;
    moveto(MaxMass/2,0);
    plt('SI.3,.5;CP-4,-2.5 ');
    plt('LBMass'+#3+' ');
    plt('SI.2,.35 ');
end; (* Mass Scan *)

EnergyScan : begin
    MinEnergy := 0.0;
    (* Draw Minor Mass Ticks X Axis *)
    plt('TL1 ');
    r := (int(MinEnergy * 4) + 1)/4;
    while(r <= MaxEnergy) do
    begin
        quickmoveto(r-MinEnergy,0.0);
        plt('XT ');
        r := r + 0.25;
    end;

    (* Draw Major Mass Ticks X Axis *)
    plt('TL2 ');
    r := (int(MinEnergy * 2) + 1)/2;
    while(r <= MaxEnergy) do
    begin
        quickmoveto(r-MinEnergy,0.0);
        plt('XT;CP-1.33,-1 ');
        str(r:1:1,numstring);
        plt('LB' + numstring + #3 + ' ');
        r := r + 0.5;
    end;
    moveto(MaxEnergy/2,0);
    plt('SI.3,.5;CP-14,-2.5 ');
    plt('LBElectron Energy (eV)'+#3+' ');
    plt('SI.2,.35 ');
end; (* Mass Scan *)
end; (* case *)

```

```
end; ( procedure )
```

```
procedure plot_data(xstart, xend : integer; ystart, yend : real; buffer : rmbuf);
(* plot data, connecting points (plotting dots is
too hard on the pen) *)
var i : integer;
    r : real;
begin
  if ScanType = EnergyScan then
    xstart := Round((sqrt(PulseSlope/LasereV[round(Param[8].Value)]) + PulseDelay)
/DlyVal[round(Param[4].Value)]) - 7;
    r := Convert(xstart);
    moveto(r,Intensity(r,buffer[xstart]/yend));
    plt('PD ');
    for i := xstart to xend do
      begin
        r := Convert(i);
        if Dots = TRUE then
          begin
            moveto(r,Intensity(r,buffer[i]/yend));
            plt('PD ');
          end
        else
          quicklineto(r,Intensity(r,buffer[i]/yend));
        end;
      plt('PU ');
      usercoords(0,1000,0,1000);
    end;
end;
```

```
function find_max(xstart, xend : integer; buffer : rmbuf) : real;
(* find maximum value in data set *)
var i : integer;
    max : real;
begin
  if ScanType = EnergyScan then
    xstart := Round((sqrt(PulseSlope/LasereV[round(Param[8].Value)]) + PulseDelay)
/DlyVal[round(Param[4].Value)]) - 7;
    max := Intensity(Convert(xstart),buffer[xstart]);
    for i := xstart+1 to xend do
      if (Intensity(Convert(i),buffer[i]) > max) then max := Intensity(Convert(i),buffer[i]);
    if max < 1.0 then max := 1.0;
    find_max := max;
  end;
```

```
function find_sum(xstart, xend : integer; buffer : rmbuf) : real;
(* find total sum of data points (for background tweaking) *)
var i : integer;
    tot : real;
begin
  tot := 0.0;
```

```

    for i := xstart+1 to xend do
        tot := tot + buffer[i];
        find_sum := tot;
    end;

function find_min(xstart, xend : integer; buffer : rmbuf) : real;
(* find minimum value in data set *)
var i : integer;
    min : real;
begin
    if ScanType = EnergyScan then
        begin
            xstart := Round((sqrt(PulseSlope/LasereV[round(Param[8].Value)]) + PulseDelay)
                /DlyVal[round(Param[4].Value)]) - 7;
            min := 0.0;
            for i:= xstart - 9 to xstart do
                min := min + buffer[i];
            end;
            find_min := min/10;
        end
    else
        begin
            min := Intensity(Convert(xstart),buffer[xstart]);
            for i := xstart+1 to xend do
                if (Intensity(Convert(i),buffer[i]) < min) then min := Intensity(Convert(i),buffer[i]);
                (* if min < 0.0 then min := 0.0; *)
            end;
            find_min := min;
        end;
    end;

procedure justify(xstart, xend : integer; smallest : real);
(* subtract off smallest intensity from all data points *)
var i : integer;
begin
    for i := xstart to xend do
        begin
            data[i] := data[i] - smallest;
            if data[i] < 0 then data[i] := 0.0; (* this should never occur *)
        end;
    end;

procedure Invert(xstart, xend : integer; largest : real);
(* invert data set so negative voltages give positive peaks *)
var i : integer;
begin
    for i := xstart to xend do
        data[i] := largest - data[i];
    end;

procedure InitPlt;
(* initialize plotter *)
begin

```

```

plotter := ibfind ('PLOTTER');

(* Check for ibfind error.          *)
if (plotter < 0) then
begin
  writeln('Plotter Find Error');
  finderr;
end;

(* Clear the device.                *)
ibclr (plotter);
(* Check for an error on each GPIB call to
   be safe.                          *)
if ((ibsta AND ERR) <> 0) then error;
setwindow(1000,9000,1500,7000);
usercoords(0,1000,0,1000);
end;

procedure endplt;
(* clean up and replace pen *)
begin
  moveto(0,0);
  plt('SP 0 ');
  plt('SP 0 ');
  ibclr (plotter);
  (* ibonl (plotter,0); *)
end;

```

3.3 rmgrap15.pas

```

(* rmgrap15.pas
  routines for data display on screen
  keyboard input and interpretation routines *)

(* Version 1.00
  4/26/87 : Mass Scans
  5/8/87  : Text File for Data Storage
  5/18/87 : Electron Energy Scale
  3-5/88  : Data Smoothing
  5/24/88 : weighed (gaussian) smoothing, improved
            data value (numbers) display
  6/29/88 : help screen
  7/7/88  : default directory
  10/17/88: noise subtraction substituted for primitive smoothing
  11/13/88: Some errors trapped for, check on starting new expt.
            without save

```

5/2/90 : Add all features of an.pas program, viz Fourier
 Transform and Store in Energy and remove Chaining
 two programs together option *)

```

var
  c : char;
  i, xpos, ypos, CPos : integer;
  Done, funckey : boolean;

procedure WriteParams; forward;
procedure ReadParams; forward;

procedure InitParam;
  (* initialize display of experimental parameters *)
  (* Called once, at beginning of program *)

  var i : integer;
      T0, L : real;
begin
  InitGraphics; (* initialize Polik's graphics routines *)
  CurrPage := 0; (* current graphics display and drawing page *)
  CPos := 1; (* cursor position for parameter display *)
  oldtotsum := 0.0;
  Done := FALSE; (* TRUE only if you wish to exit program *)
  Mode := Text; (* displaying text on screen *)
  ExpDone := TRUE; (* FALSE only if experiment in progress *)
  ConnectPoints := FALSE; (* for data graphing on screen *)
  Dots := FALSE; (* plot lines *)
  numblocks := 5;
  currpt := 1; (* current cursor position *)
  T0 := -12.26;
  L := 0.9330;
  PulseDelay := T0 + 40.0; (* stop pulse delay from laser firing (nsec) *)
  PulseSlope := 2843174.0*L*L;
    (* Electron Energy = PulseSlope/((Time + PulseDelay)^2)
  (* Before 8/11, PulseDelay = 1764, PulseSlope = 2244000)
  (* 8/11-8/18 PulseDelay = 1727, PulseSlope = 2461500)
  (* 8/19 and later, PulseDelay = 1427, PulseSlope = 2388700)
  (* 9/29 and later, T0 = 18.02, L = 90.94, PulseSlope = 2843174*L*L)
  (* 10/9 and later, T0 = -7.12, L = 93.05, PulseSlope = " ")
  (* 10/15 and later, T0 = -12.26, L = 93.30)
  (* 10/21 and later, T0 = 84.64, L = 85.12)
  (* 10/23 and later, T0 = -12.26, L = 93.30)
  (* EE in eV, time in ns)
  (* Nsec/channel for transient digitizer *)
  DlyVal[0] := 5;
  DlyVal[1] := 10;
  DlyVal[2] := 20;
  DlyVal[3] := 40;
  DlyVal[4] := 80;
  DlyVal[5] := 160;

```

```
DlyVal[6] := 320;
```

```
(* Various laser gases and corresponding energies -  
for Electron Energy display *)
```

```
LaserGas[0] := '213 nm';  
LasereV[0] := 5.825; (* Laser energy (in eV) *)  
LaserGas[1] := '266 nm';  
LasereV[1] := 4.66;  
LaserGas[2] := '299 nm';  
LasereV[2] := 4.144;  
LaserGas[3] := '355 nm';  
LasereV[3] := 3.495;  
LaserGas[4] := '416 nm';  
LasereV[4] := 2.979;  
LaserGas[5] := '532 nm';  
LasereV[5] := 2.33;  
LaserGas[6] := '2.0 V';  
LasereV[6] := 2.0;  
LaserGas[7] := '1.5 V';  
LasereV[7] := 1.5;  
LaserGas[8] := '1.0 V';  
LasereV[8] := 1.0;
```

```
YNString[0] := 'No';  
YNString[1] := 'Yes';  
LastFile := '';
```

```
InitData;
```

```
(* initialize parameters ie, # of scans, laser type, etc *)
```

```
ReadParams;
```

```
MassLength := 1.72; (* effective length of mass spec. in meters *)  
MassSlope := 72024.3 * MassLength/sqrt(Param[12].Value); (* Values are in nsec *)  
MassInt := 250.0; (* Values for Mass Display *)  
ExpSaved := TRUE; (* Initialise flag for preventing overwriting data*)  
Directory := '\turbo\data\'; { default directory }  
end;
```

```
procedure ReadParams;
```

```
(* read parameter file. *)
```

```
var
```

```
FileVar : text;
```

```
i : integer;
```

```
begin
```

```
Assign(FileVar,'EXPT.PAR');
```

```
Reset(FileVar);
```

```
for i := 1 to NumPars do { read parameters }
```

```
with Param[i] do
```

```
begin
```

```

        Readln(FileVar,what);
        Readln(FileVar,units);
        Readln(FileVar,Value);
        Readln(FileVar,UpLimit);
        Readln(FileVar,LowLimit);
        Readln(FileVar,Incr);
    end;
    Close(FileVar);
end;

procedure WriteParams;
    (* save parameter file. *)
var
    FileVar : text;
    i : integer;
begin
    Assign(FileVar,'EXPT.PAR');
    Rewrite(FileVar);
    for i := 1 to NumPars do { read parameters }
        with Param[i] do
            begin
                Writeln(FileVar,what);
                Writeln(FileVar,units);
                Writeln(FileVar,Value);
                Writeln(FileVar,UpLimit);
                Writeln(FileVar,LowLimit);
                Writeln(FileVar,Incr);
            end;
        end;
    Close(FileVar);
end;

```

```

procedure ShowLabels;
    (* Display parameter labels *)
var i : integer;
begin
    lowvideo;
    TextColor(15);
    for i := 1 to NumPars do
        begin
            gotoxy(10,6+i);
            write(Param[i].What);
            gotoxy(50,6+i);
            write(Param[i].Units);
        end;
    gotoxy(21,24);
    TextColor(11);
    write('Type Alt-H for help');
end;

```

```

procedure ShowParam;

```

```

(* display parameter values *)
(* note that for some parameters, the value itself
   is not displayed - rather it is used as an index
   to a value that is displayed. ie, if Param[4].Value
   is 2, then the time per point is 20 nsec
   (as DelayValue[2] is 20) *)

var i : integer;
begin
  for i := 1 to NumPars do
    begin
      if i = CPos then TextColor(12)
      else TextColor(15);
      gotoxy(40,6+i);
      case i of
        4 : writeln(DlyVal[round(Param[i].Value)],' ');
        6,7 : writeln(YNString[round(Param[i].Value)],' ');
        8 : writeln(LaserGas[round(Param[i].Value)],' ');
        10,11 : writeln(Param[i].Value:0:2,' ');
        else writeln(Param[i].Value:0:0,' ');
      end;
    end;
  end;

type mybuf = string[50];

procedure ShowFile(fname : mybuf);
  (* display the contents of a file *)
var
  FileVar : text;
  scratch : String[128];
begin
  ClrScr;
  normvideo;
  gotoxy(1,1);
  Assign(FileVar,fname);
  ReSet(FileVar);
  while NOT eof(FileVar) do
    begin
      Readln(FileVar,scratch);
      Writeln(scratch);
    end;
  end;

procedure Help;
  (* display help screens *)
begin
  if mode = GRAPHICS then
    begin
      mode := TEXT;
      ExitGraphics;
    end;
end;

```



```

end;    { if }
ShowFile('HELP1.PAR');
while NOT KeyPressed do
  ;
ShowFile('HELP2.PAR');
while NOT KeyPressed do
  ;
ClrScr;
ShowLabels;
end;

```

```

function Exist;
(* returns TRUE if the file already exists *)
var Fil : file;
    Extant : boolean;
begin
  Assign(Fil,FileName);
  ($I-)
  Reset(Fil);
  ($I+)
  Extant := (IOResult = 0);
  if Extant then Close(Fil);
  Exist := Extant;
end;

```

```

function cube(r : real) : real;
{ returns r^3 }
begin
  cube := r*r*r;
end;

```

```

function weigh(r,alpha : real) : real;
{ weighing function for smoothing - use a Gaussian }
begin
  weigh := exp(-alpha*r*r);
end;

```

```

procedure smooth(res : real);
{ Smooth data using weighing function weigh.
  ie, Data'(Eo) = sum(i) (weigh(Eo - Ei) * Data(Ei)) / sum(i) (weigh(Eo - Ei))
  Res is fwhm resolution in eV. }

```

```

type dptr = ^rmbuf;
var
  a, w, realdE, maxdE, totd, totw : real;
  i,j, maxi, mini, range, npts : integer;
  tpc : integer; { time per channel (ie, 5 (ns)) }
  temp : dptr; { (dynamic) temporary storage of modified data }
begin
  tpc := DlyVal[round(Param[4].Value)];

```

```

npts := round(Param[3].Value);
a := 2.773/sqr(res);
maxdE := 2.146/sqrt(a);
{ smooth iff abs(Eo - Ei) <= maxdE
  ie, iff weigh(Eo - Ei) >= 0.01 }
New(temp);
for j := 1 to npts do { loop over data points }
begin
  { find 1st and last points to smooth over for a given data point }
  realdE := 2*tpc*PulseSlope/cube(tpc*(j+8)-PulseDelay);
  range := trunc(maxdE/realdE);
  if range > 1 then { smooth this point ... }
  begin
    maxi := j + range;
    if maxi > npts then maxi := npts; { off the end of the data set }
    mini := j - range;
    if mini < 1 then mini := 1;
    totd := 0.0; { hold sum weigh*Data }
    totw := 0.0; { hold sum weigh }
    for i := mini to maxi do { sum over i }
    begin
      w := weigh((i-j)*realdE,a);
      totd := totd + data[i] * w;
      totw := totw + w;
    end;
    temp^[j] := totd/totw; { save (smoothed) data point }
  end { if }
  else
    temp^[j] := data[j];
  end; { loop over data set }
for j := 1 to npts do { transfer data set to real storage }
  Data[j] := temp^[j];
Dispose(temp);
end;

```

(* Some useful Graphing routines *)

```

procedure ClearPage(WhichPage : integer); (* Clear graphics screen *)
begin
  if WhichPage = 0 then FillChar(Mem[$A000:$0000], $8000, $00);
  if WhichPage = 1 then FillChar(Mem[$A800:$0000], $8000, $00);
  (* Note: use $B000 and $B800 for monochrome display *)
end;

```

```

procedure ScreenAxes;
(* Draw axes - don't label them, just draw them *)
begin
  SetColor(AxesColor);
  MoveScreen(100,900);
  DrawLineScreen(100,100);
  DrawLineScreen(950,100);

```

```

DefineWindow(100,100,950,900);
SetColor(White);
end;

procedure LabelAxes;
(* label axes - note that the are 4 distinct cases for X Axis:
  1. short time scan ( < 1000 nsec)
  2. long time scan
  3. mass scan
  4. energy scan *)

var r, timewidth, MaxMass, XScale : real;
    MinTick : integer;
begin
  SetColor(AxesColor);
  DefineWindow(2,10,1000,1000);
  MoveScreen(15,875);
  str(maxval:6:0,rtest); (* display total # of counts on y axis *)
  SetTextDirection(0);
  DrawText(rtest);
  MoveScreen(750,950);
  r := 100.0*(10000.0 * ScanNumHi + ScanNumLo)/Param[1].Value; (* % done *)
  str(r:3:1,rtest);
  SetColor(DoneColor);
  DrawText(rtest + ' % Done'); (* display % done *)
  MoveScreen(500,950);
  str(totsum:0:0,rtest);
  DrawText('Sum = ' + rtest);
  MoveScreen(500,900);
  str(totsum-oldtotsum:0:0,rtest);
  DrawText('Change = ' + rtest);
  MoveScreen(70,450);
  SetTextDirection(1);
  SetColor(LabelColor);
  DrawText('Counts'); (* Y Axis label *)
  SetColor(AxesColor);
  for i := 1 to 5 do
  begin
    MoveScreen(90,100+160*i);
    DrawLineRelScreen(20,0); (* Y Axis tick marks *)
  end;
  (* Label X Axis *)
  case ScanType of
  TimeScan : begin
    timewidth := int(DlyVal[round(Param[4].Value)]) * Param[3].Value/5.0;
    (* timewidth is scan length in nsec *)
    if timewidth < 1000 then
    begin (* short time scan *)
      SetTextDirection(0);
      MoveScreen(400,10);
    end;
  end;
end;

```

```

SetColor(LabelColor);
DrawText("Time (nanoseconds)");
SetColor(AxesColor);
for i:= 0 to 5 do
begin
  MoveScreen(100 + 170*i,115);
  DrawLineRelScreen(0,-30); (* tick marks *)
  MoveScreen(80 + 170*i,40);
  str(timewidth*i:4:0,rtest);
  DrawText(rtest); (* X Axis labels *)
end;
end { if timewidth < 1000 }
else (* long time scan *)
begin
  timewidth := timewidth/1000; (* convert to usec *)
  SetTextDirection(0);
  MoveScreen(400,10);
  SetColor(LabelColor);
  DrawText("Time (microseconds)");
  SetColor(AxesColor);
  for i:= 0 to 5 do
  begin
    MoveScreen(100 + 170*i,115);
    DrawLineRelScreen(0,-30); (* X Axis tick marks *)
    MoveScreen(80 + 170*i,40);
    str(timewidth*i:3:1,rtest);
    DrawText(rtest); (* X Axis labels *)
  end;
end; { else - if timewidth >= 1000 }
end; { Time Scan }
MassScan : begin
  SetTextDirection(0);
  MoveScreen(500,10);
  SetColor(LabelColor);
  DrawText("Mass (amu)");
  SetColor(AxesColor);
  MaxMass := Convert(round(Param[3].Value));
  (* maximum mass to be displayed *)
  (* choose minor tick intervals *)
  case Round(0.5 + MaxMass) of
    0..50 : MinTick := 5;
    51..100 : MinTick := 5;
    101..200 : MinTick := 10;
    201..400 : MinTick := 20;
    401..1000 : MinTick := 50;
    else MinTick := 200;
  end; { case }
  i := 0;
  while(i <= MaxMass) do
  begin
    MoveScreen(100 + 850*(int(i)/MaxMass),110);

```

```

        DrawLineRelScreen(0,-20);
        i := i + MinTick; (* draw minor ticks *)
    end;
    i := 0;
    while(i <= MaxMass) do
    begin
        MoveScreen(100 + 850*(int(i)/MaxMass),115);
        DrawLineRelScreen(0,-30);
        MoveScreen(80 + 850*(int(i)/MaxMass),40);
        str(i,rtest);
        DrawText(rtest);
        i := i + 2*MinTick; (* draw major ticks *)
    end;
end; { Mass Scan }
EnergyScan : begin
    SetTextDirection(0);
    MoveScreen(400,10);
    SetColor(LabelColor);
    DrawText('Electron Energy (eV)');
    SetColor(AxesColor);
    MaxEnergy := LasereV[round(Param[8].Value)];
    (* Max electron energy is laser energy *)
    MinEnergy := 0; (* minimum EE to display *)
    XScale := 850.0/(MaxEnergy - MinEnergy);
    r := (int(MinEnergy * 4) + 1)/4;
    while(r <= MaxEnergy) do
    begin
        MoveScreen(100 + (r - MinEnergy)*XScale,110);
        DrawLineRelScreen(0,-20);
        r := r + 0.25;
    end; (* draw minor ticks (every .25 eV) *)
    r := (int(MinEnergy * 2) + 1)/2;
    while(r <= MaxEnergy) do
    begin
        MoveScreen(100 + (r - MinEnergy)*XScale,115);
        DrawLineRelScreen(0,-30);
        MoveScreen(82 + (r - MinEnergy)*XScale,40);
        str(r:1:1,rtest);
        DrawText(rtest);
        r := r + 0.5; (* draw major ticks *)
    end;
end; { Energy Scan }
end; { case }
DefineWindow(100,100,950,900);
SetDisplayPage(CurrPage);
SetColor(White);
end;

procedure GraphData(xstart, xend : integer; ystart, yend : real; buffer : rmbuf);
    (* graph data, connecting points, if necessary *)
    var i : integer;

```

```

r : real;
begin
  if ScanType = EnergyScan then
    begin
      xstart := Round((sqrt(PulseSlope/LasereV[round(Param[8].Value)]) + PulseDelay)
        /DlyVal[round(Param[4].Value)]) - 7;
      DefineUserCoordinates(0.0,ystart,LasereV[round(Param[8].Value)],yend);
    end
  else
    DefineUserCoordinates(Convert(xstart),ystart,Convert(xend),yend);
    SetColor(DataColor);
    If NOT ConnectPoints then
      begin
        for i := xstart to xend do
          begin
            r := Convert(i);
            MoveUser(r,Intensity(r,buffer[i]));
            DrawPoint;
          end;
        end
      end
    else
      begin
        r := Convert(xstart);
        MoveUser(r,Intensity(r,buffer[xstart]));
        for i := xstart+1 to xend do
          begin
            r := Convert(i);
            DrawLineUser(r,Intensity(r,buffer[i]));
          end;
        end;
      end;
    SetColor(White);
  end;
end;

```

```

procedure DoCursor;
  (* take care of cursor movement *)
var i : integer;
    r : real;
begin
  MoveUser(Convert(lastpt),Intensity(Convert(lastpt),data[lastpt]));
  SetColor(Black);
  DrawLineRelScreen(0,50); (* erase old cursor *)
  SetColor(DataColor);
  (* patch up any data points/lines that were deleted *)
  if NOT ConnectPoints then
    begin
      for i := lastpt-1 to lastpt+1 do
        if (i > 0) AND (i <= Param[3].Value) then
          begin
            r := Convert(i);
            MoveUser(r,Intensity(r,data[i]));
            DrawPoint;
          end;
        end;
      end;
    end;
end;

```

```

        end;
    end
    else
    begin
        if lastpt = 1 then MoveUser(Convert(1),Intensity(Convert(1),data[1]))
        else MoveUser(Convert(lastpt-1),Intensity(Convert(lastpt-1),Data[lastpt-1]));
        for i := lastpt to lastpt+1 do
            if i <= Param[3].Value then
                DrawLineUser(Convert(i),Intensity(Convert(i),data[i]));
            end;
        MoveUser(Convert(currpt),Intensity(Convert(currpt),data[currpt]));
        SetColor(CursorColor);
        DrawLineRelScreen(0,50); (* draw new cursor *)
        SetColor(White);
    end;
end;

```

```

procedure MoveCursor(Dist : integer);
begin
    lastpt := currpt;
    if ScanType = EnergyScan then currpt := currpt - Dist
    else currpt := currpt + Dist;
    if currpt < 1 then currpt := 1;
    if currpt > Param[3].Value then currpt := round(Param[3].Value);
    DoCursor;
end;

```

```

procedure CheckDir; (* Arguments defined in forward in parent program file *)
var
    DirSet, ExtSet : boolean;
    j : integer;
begin
    DirSet := FALSE;
    ExtSet := FALSE;
    for j := 1 to Length(FileName) do
        begin
            if FileName[j] = '\' then DirSet := TRUE;
            if FileName[j] = '.' then ExtSet := TRUE;
        end;
    if DirSet = FALSE then FileName := Directory + FileName;
    if ExtSet = FALSE then FileName := FileName + '.dat'
end;

```

```

procedure WriteFile(FileName : BufString);
(* Save data on disk. Have already checked that it is ok
to kill FileName (if it exists).
This routine traps for ioerror if
the assigned directroy does not exist (8/27/89)
Note that some parameters
are not saved (whether background subtraction was active
or not, for instance), also, the time -> energy conversion
factors are not saved. This should be corrected later.

```

(NB : t -> E factors saved 10/23/87)
 (Also note that the raw intensity vs time data is saved,
 not int vs mass or energy
 Flag ExpSaved reset so another spectrum may be started *)

```

var
  FileVar : text;
  DirErr: boolean;
  i : integer;
begin
  ($I-)
  Assign(FileVar,FileName);
  Rewrite(FileVar);
  ($I+)
  DirErr := (IOResult=$F1);
  if DirErr=FALSE then
    begin
      Writeln(FileVar,'A'); (* current format *)
      Writeln(FileVar,Param[4].Value);
      Writeln(FileVar, Param[3].Value);
      Writeln(FileVar,Param[2].Value);
      Writeln(FileVar,Param[1].Value);
      Writeln(FileVar,Param[5].Value);
      Writeln(FileVar,Param[10].Value);
      Writeln(FileVar,Param[11].Value);
      for i:= 1 to round(Param[3].Value) do
        Writeln(FileVar,Data[i]:0:0);
      Close(FileVar);
      Writeln(' ');
      Writeln('Data Saved');
      LastFile := FileName;
      ExpSaved:=TRUE;
      Delay(500);
    end
  else
    begin
      writeln(#7+'IOerror - Directory full or non-existant');
      Delay(1500); (* wait about .5 sec so that user can read message *)
    end;
end; { procedure }

```

```

procedure ReadFile(FileName : BufString);
  (* read data file. Have already checked that FileName exists. *)

```

```

var
  FileVar : text;
  i : integer;
  min : real;
  teststr : BufString;
  NewFormat : boolean;

```



```

begin
  InitSub:=FALSE; (*New File coming, old BackData copy redundant*)
  Assign(FileVar,FileName);
  Reset(FileVar);
  Readln(FileVar,teststr);
  if (teststr[1] = 'A') then NewFormat := TRUE
  else NewFormat := FALSE;
  if NOT NewFormat then Reset(FileVar);
  Readln(FileVar,Param[4].Value);
  Readln(FileVar, Param[3].Value);
  Readln(FileVar,Param[2].Value);
  Readln(FileVar,Param[1].Value);
  Readln(FileVar,Param[5].Value);
  if NewFormat then
  begin
    Readln(FileVar,Param[10].Value);
    PulseDelay := Param[10].Value + 40.0;
    Readln(FileVar,Param[11].Value);
    PulseSlope := 284.3174 * Sqr(Param[11].Value);
  end;
  for i:= 1 to round(Param[3].Value) do
    Readln(FileVar,Data[i]);
  Close(FileVar);
  Writeln(' ');
  Writeln('Data Read');
  min := Data[1];
  for i:= 2 to round(Param[3].Value) do
    if Data[i] < min then min := Data[i];
  for i:= 1 to round(Param[3].Value) do
    Data[i] := Data[i] - min;
  LastFile := FileName;
end; { procedure }

procedure AddFile(FileName : BufString);
  (* Read data file, adding data to data currently in data[] array.
  Note that the time per point for the two MUST be the same.
  The total time for the scan is the smaller of the two times.
  (ie, if a 600 ns scan and a 1000 ns scan are added, the
  resulting scan is 600 ns long.) Also, intensities vs time are
  added, not converted intensities after mass or energy calculations *)

var
  FileVar : text;
  i : integer;
  NumOfScans, PointsPerScan, TimePerPoint, r : real;
  teststr : BufString;
  NewFormat : boolean;

begin
  InitSub := FALSE; (*new file coming, BackData copy now redundant*)
  Assign(FileVar,FileName);

```

```

Reset(FileVar);
Readln(FileVar,teststr);
if (teststr[1] = 'A') then NewFormat := TRUE
else NewFormat := FALSE;
if NOT NewFormat then Reset(FileVar);
Readln(FileVar,TimePerPoint);
if (TimePerPoint = Param[4].Value) then
begin
  Readln(FileVar, PointsPerScan);
  if PointsPerScan < Param[3].Value then Param[3].Value := PointsPerScan;
  Readln(FileVar,Param[2].Value);
  Readln(FileVar,NumOfScans);
  Param[1].Value := Param[1].Value + NumOfScans;
  Readln(FileVar,Param[5].Value);
  if NewFormat then
  begin
    Readln(FileVar,Param[10].Value);
    PulseDelay := Param[10].Value + 40.0;
    Readln(FileVar,Param[11].Value);
    PulseSlope := 284.3174 * Sqr(Param[11].Value);
  end;
  for i:= 1 to round(Param[3].Value) do
  begin
    Readln(FileVar,r);
    Data[i] := Data[i] + r;
  end;
  Close(FileVar);
  LastFile := 'SUM ' + FileName;
  Writeln(' ');
  Writeln('Data Read');
end
else
  Writeln(#7 + ' Time per Point is Wrong : Cannot Add Data Set');
  Delay(500);
end; { procedure }
procedure GetScanType; forward;
procedure StoreEner(FileName : BufString);      (* Energy vs. intensity file*)
var
  FileVar : text;
  xstart,i : integer;
begin
  With Param[6] do
  Begin
    Value:=0; (* Make sure Convert routine uses energy *)
  end;
  GetScanType;
  maxval := find_max(1,round(Param[3].Value),data);
  xstart:= Round((sqrt(PulseSlope/LasereV[round(Param[8].Value)]) + PulseDelay)
    /DlyVal[round(Param[4].Value)]-7;
  Assign(FileVar,FileName);
  Rewrite(FileVar);

```

```

    for i:= round(Param[3].Value) downto xstart do
      writeln(FileVar,Convert(i):1:3,'',(Intensity(Convert(i),Data[i])/maxval):1:6);
    Close(FileVar);
    Writeln(' ');
    Writeln('Data Saved');
    LastFile := FileName;
    Delay(500); (* wait about .5 sec so that user can read message *)
end; { procedure }
procedure GoToText;
  { Prepare to display text on screen }
begin
  if mode = GRAPHICS then
    begin
      mode := TEXT;
      ExitGraphics;
    end; { if }
  ClrScr;
  normvideo;
end;

procedure GoToGraphics;
begin
  CurrPage := 1;
  currpt := round(Param[3].Value);
  SetDrawPage(CurrPage);
  EnterGraphics;
  ClearPage(CurrPage);
  maxval := find_max(1,round(Param[3].Value),data);
  minval := 0.0;
  ScreenAxes;
  LabelAxes;
  GraphData(1,round(Param[3].Value),0.0,maxval,data);
  mode := GRAPHICS;
end;
procedure scale; forward;
procedure DoGraph; forward;
procedure DoPlot; forward;
procedure StopExp; forward;
procedure StartExp; forward;
procedure GetKey; forward;
procedure DoKey(Fnction : boolean; KeyCode : integer);
  (* Interpret keys pressed by user. Called by GetKey. See Turbo
  Reference Manual p. 341 for keycodes. Fnction is TRUE if a
  function key is pressed (ie, F1-F10, the arrow keys, Alt-anything) *)
var
  keep,keep2,keep3,keep4,wait,Crittime,seper : real;
  CritCh :ARRAY[1..5] OF integer;
  npoint :integer;
  NewDir : bufstring;
  Resol :real; {Resolution at best, in energy space, (eV) }
begin

```

```

if Fnction = TRUE then
begin
  case KeyCode of
    68 : if Mode = Graphics then SetDisplayPage(1); (* F10 *)
    67 : if Mode = Graphics then SetDisplayPage(0); (* F9 *)
    38 : ConnectPoints := NOT ConnectPoints; (* Alt L - toggle connect points *)
    31 : StopExp; (* Alt S - stop experiment *)
  end; { case }
end; { if Fnction }

```

```

if ExpDone = TRUE then
begin
  if Fnction = TRUE then
  begin
    case KeyCode of
      20 : begin (* Alt T - display text screen *)
          Mode := Text;
          SetClearOn;
          ExitGraphics;
          SetClearOff;
          ShowLabels;
          end;
      34 : begin (* Alt G - display graphics screen *)
          Mode := Graphics;
          EnterGraphics;
          end;
      18 : begin (* Alt E - start a new experiment *)
          StartExp;
          end;
      19 : begin (* Alt R - read a data file *)
          GoToText;
          IF ExpSaved=FALSE THEN
            BEGIN
              write (#7+'Overwrite experimental data ? ');
              readln(rtest2);
              IF (rtest2 = 'Y') OR (rtest2='y') THEN ExpSaved := TRUE;
            END;
          IF ExpSaved = TRUE THEN
            BEGIN
              gotoxy(1,12);
              write('Name of File to Read ('+Directory+'NAME.dat) : ');
              readln(rtest);
              if Length(rtest) > 0 then
                begin
                  CheckDir(rtest);
                  if Exist(rtest) = TRUE then ReadFile(rtest) (* Read data file *)
                  else
                    begin
                      Writeln(#7 + 'File Not Found');
                      Delay(500);
                    end;
                end;
            END;
          end;
    end;
  end;
end;

```

```

    end;
    ScanNumHi := trunc(Param[1].Value/10000);
    ScanNumLo := round(Param[1].Value - 10000.0 * ScanNumHi);
    PulseDelay := Param[10].Value + 40.0;
    PulseSlope := 284.3174 * Sqr(Param[11].Value);
    GetScanType;
    totsum := 0;
    GoToGraphics;
  END
ELSE
  BEGIN
    Writeln;
    Writeln('EXPERIMENTAL DATA RESTORED. ');
    Delay(1000);
    ClrScr;
    ShowLabels;
  END;
end;
30 : begin (* Alt A - add a data file *)
  GoToText;
  gotoxy(1,12);
  write('Name of File to Add ('+Directory+'NAME.dat) : ');
  readln(rtest);
  if Length(rtest) > 0 then
    begin
      CheckDir(rtest);
      if Exist(rtest) = TRUE then AddFile(rtest)      (* Read data file *)
      else
        begin
          Writeln(#7 + 'File Not Found');
          Delay(500);
        end;
      end;
    end;
  PulseDelay := Param[10].Value + 40.0;
  PulseSlope := 284.3174 * Sqr(Param[11].Value);
  ScanNumHi := trunc(Param[1].Value/10000);
  ScanNumLo := round(Param[1].Value - 10000.0 * ScanNumHi);
  GetScanType;
  totsum := 0;
  minval := find_min(1,round(Param[3].Value),data);
  Justify(1,round(Param[3].Value),minval);
  GoToGraphics;
end;
17 : begin (* Alt W - write (save) a data file *)
  GoToText;
  gotoxy(1,12);
  write('Name of Data File ('+Directory+'NAME.dat) : ');
  readln(rtest);
  if Length(rtest) > 0 then
    begin
      CheckDir(rtest);

```

```

if Exist(rtest) = FALSE then WriteFile(rtest) (* Write data file *)
else
begin
  Write(#7+'File Already Exists. Overwrite Existing File (Y/N) ? ');
  read(rtest2);
  if (rtest2 = 'Y') OR (rtest2 = 'y') then WriteFile(rtest);
end;
end;
ClrScr;
ShowLabels;
end;
25 : WriteParams; (* Alt P - save parameters *)
35 : Help; (* Alt H - display help screens *)
32 : begin (* Alt D - change default directory *)
  GoToText;
  gotoxy(10,12);
  writeln('Default directory is '+Directory);
  gotoxy(10,13);
  write('Input new default directory : ');
  readln(NewDir);
  if length(NewDir) > 0 then
  begin
    if NewDir[1] = '.' then
      for i := 2 to length(Newdir) do
        Directory := Directory + NewDir[i]
      else
      begin
        Directory := ' ';
        for i := 2 to length(NewDir) do
          Directory := Directory + ' ';
        for i := 1 to length(NewDir) do
          Directory[i] := NewDir[i];
        end;
      end;
    end;
  end;
  ClrScr;
  ShowLabels;
end;
59 : BEGIN
  Done := TRUE; (* F1 - exit program *)
END;
61 : begin (* F3 - do weighed (gaussian) smoothing *)
  GoToText;
  gotoxy(10,12);
  writeln('Gaussian Smoothing of Data : ');
  gotoxy(10,13);
  write('Enter fwhm resolution (meV) (or 0 to exit) : ');
  readln(resol);
  if ((resol > 0) AND (resol < 100)) then
    smooth(resol/1000);
  minval := find_min(1,round(Param[3].Value),data);
  Justify(1,round(Param[3].Value),minval);

```

```

GoToGraphics;
end;
49 : begin (*ALT-N*)
GoToText;
scale;
PulseDelay := Param[10].Value + 40.0;
PulseSlope := 284.3174 * Sqr(Param[11].Value);
ScanNumHi := trunc(Param[1].Value/10000);
ScanNumLo := round(Param[1].Value - 10000.0 * ScanNumHi);
GetScanType;
IF (ScanType=EnergyScan) AND (InitSub=TRUE) THEN
BEGIN
writeln('Going into energy : already performing ALT - A');
write('Do you also want to Gaussian Smooth the Data : ');
readln(rtest2);
if (rtest2 = 'Y') OR (rtest2 = 'y') then
begin
write('Enter fwhm resolution (meV) (or 0 to cancel) : ');
readln(resol);
if ((resol > 0) AND (resol < 100)) then
smooth(resol/1000);
end;
END;
minval := find_min(1,round(Param[3].Value),data);
Justify(1,round(Param[3].Value),minval);
GoToGraphics;
end;
46 : begin (* ALT-C *) (*Calibrate spectrum*)
GoToText;
calib;
ClrScr;
ShowLabels;
end;
33 : BEGIN (* ALT-F *) (*Calculate Fast Fourier Transform, and then smooth*)
GoToText;
ftfit(data); (*Fourier routine*)
minval := find_min(1,round(Param[3].Value),data);
Justify(1,round(Param[3].Value),minval);
GoToGraphics;
end;
22 : BEGIN (* ALT-U *)
GoToText;
gotoxy(1,12);
write('Name to store Energy Data File ('+Directory+'NAME.dat) : ');
readln(rtest);
if Length(rtest) > 0 then
begin
CheckDir(rtest);
if Exist(rtest) = FALSE then StoreEner(rtest) (* Write data file *)
else
begin

```

```

Write('File Already Exists. Overwrite Existing File (Y/N) ? ');
read(rtest2);
if (rtest2 = 'Y') OR (rtest2 = 'y') then StoreEner(rtest);
end;
end;
ClrScr;
ShowLabels;
END;
65 : DoPlot; (* F7 - plot data on plotter *)
66 : begin (* F8 - plot dots *)
      Dots := TRUE;
      DoPlot;
      Dots := FALSE;
end;
72 : if CPos > 1 then CPos := CPos - 1 (* Up Arrow *)
      else CPos := NumPars;
80 : if CPos < NumPars then CPos := CPos + 1 (* Down Arrow *)
      else CPos := 1;
75 : begin (* Left Arrow - decrease parameter/move cursor *)
      if mode = TEXT then (* decrease parameter *)
        with Param[CPos] do
          begin
            Value := Value - Incr;
            if Value < LowLimit then Value := LowLimit;
          end
        else
          MoveCursor(-1);
        end;
77 : begin (* Right Arrow - increase parameter/move cursor *)
      if mode = TEXT then (* increase parameter *)
        with Param[CPos] do
          begin
            Value := Value + Incr;
            if Value > UpLimit then Value := UpLimit;
          end
        else
          MoveCursor(1);
        end;
115 : if mode = GRAPHICS then MoveCursor(-100); (* ctrl left arrow *)
116 : if mode = GRAPHICS then MoveCursor(100); (* ctrl right arrow *)
120 : if (mode = GRAPHICS) AND (Param[6].Value = 0) then (* Alt 1 - Mass
Calibration *)
      begin
        ExitGraphics;
        ClrScr;
        Write('Enter Mass : ');
        Readln(Mass1);
        Time1 := int(DiyVal[round(Param[4].Value)]) * (currpt+8);
        EnterGraphics;
      end;
121 : if (mode = GRAPHICS) AND (Param[6].Value = 0) then (* Alt 1 - Mass

```


Calibration *)

```
begin
  ExitGraphics;
  ClrScr;
  Write('Enter Mass : ');
  Readln(Mass2);
  Time2 := int(DlyVal[round(Param[4].Value)]) * (currpt+8);
  if (Time2 <> Time1) and (Mass2 <> Mass1) then
    begin
      MassSlope := (Time1 - Time2)/(sqrt(Mass1) - sqrt(Mass2));
      MassInt := MassSlope*sqrt(Mass1) - Time1;
      MassLength := MassSlope * sqrt(param[12].value)/2024.3;
      writeln('Time  Mass ');
      writeln(Time1,' ',Mass1);
      writeln(Time2,' ',Mass2);
      writeln('Slope : ',MassSlope:1:4);
      writeln('Intercept : ',MassInt:1:4);
      writeln('Length : ',MassLength:0:4,' m');
      delay(500);
      minval := find_min(1,round(Param[3].Value),data);
      Justify(1,round(Param[3].Value),minval);
      GoToGraphics;
    end; { if Time 2 <> Time 1 }
  end; { if Mode = Graphics }
end; { case }
end { if Fnction = TRUE }
else { Fnction = FALSE }
if KeyCode = 52 then MoveCursor(-10);(* shift left arrow = 4 *)
if KeyCode = 54 then MoveCursor(10); (* shift right arrow = 6 *)
if KeyCode = 13 then
begin (* Carriage Return (Enter) - change parameter/display data values *)
  if mode = TEXT then (* change value *)
    with Param[Cpos] do
      begin
        gotoxy(40,6+CPos);
        normvideo;
        read(Value);
        if Value > UpLimit then Value := UpLimit;
        if Value < LowLimit then Value := LowLimit;
      end
    end
  else (* display data values *)
    begin
      if ValuesShown then
        begin
          ValuesShown := FALSE;
          Mode := Graphics;
          EnterGraphics;
          DoCursor;
        end
      else
        begin
```

```

ValuesShown := TRUE;
Mode := Graphics;
ExitGraphics;
ClrScr;
for i:= 1 to 5 do
  writeln;
case ScanType of
  TimeScan : writeln(' Time (ns) Intensity');
  MassScan : writeln(' Mass (amu) Time (ns) Intensity');
  EnergyScan : writeln(' E. E. (eV) Time (ns) Raw Int. Sc. Int. ');
end; { case }
for i := currpt - 5 to currpt + 5 do
  if i>0 then
  begin
    if i= currpt then TextColor(13);
    if i= currpt+1 then TextColor(14);
    case ScanType of
      TimeScan : writeln(' ',int(DlyVal[round(Param[4].Value)]*i/1000:3:3,
        ' ',Data[i]:6:0);
      MassScan : writeln(Convert(i):4:1, ' ',
        int(DlyVal[round(Param[4].Value)]*(i+8)/1000:3:3,
        ' ',Data[i]:6:0);
      EnergyScan : writeln(' ',Convert(i):1:3, ' ',
        int(DlyVal[round(Param[4].Value)]*(i+8)/1000:3:3,
        ' ',Data[i]:6:0, ' ',Intensity(Convert(i),data[i]):6:0);
    end; { case }
  end; { if i > 0 }
end; { else }
end;
end;
end;
if Mode = Text then ShowParam;
end; { procedure }

procedure GetKey;
(* called if a key has been pressed - determines which key, and
  whether it was a function key or not, then calls DoKey to
  interpret the key *)
var
  c : char;
  funckey : boolean;
  i : integer;

begin
  c := '';
  funckey := FALSE;
  read(kbd,c); (* read the key *)
  if (c = #27) AND keypressed then
  begin
    funckey := TRUE; (* if the first character is Esc (ascii 27),
      then it was a function key. Read next

```

```

                                character to determine which function key *)
    read(kbd,c);
  end;
  i := ord(c); (* convert character to ascii equivalent integer *)
  DoKey(funckey,i);
end;

```

3.4 calib1.pas

(*Define some statistical routines from Numerical Recipes*)

```

FUNCTION gamm1n(xx: real): real;
CONST
  stp = 2.50662827465;
  half = 0.5;
  one = 1.0;
  fpf = 5.5;
VAR
  x,tmp,ser: double;
  j: integer;
  cof: ARRAY [1..6] OF double;
BEGIN
  cof[1] := 76.18009173;
  cof[2] := -86.50532033;
  cof[3] := 24.01409822;
  cof[4] := -1.231739516;
  cof[5] := 0.120858003e-2;
  cof[6] := -0.536382e-5;
  x := xx-one;
  tmp := x+fpf;
  tmp := (x+half)*ln(tmp)-tmp;
  ser := one;
  FOR j := 1 to 6 DO BEGIN
    x := x+one;
    ser := ser+cof[j]/x
  END;
  gamm1n := sngl((tmp+ln(stp*ser))
END;

PROCEDURE gcf(a,x: real; VAR gammcf,gln: real);
LABEL 1;
CONST
  itmax=100;
  eps=3.0e-7;
VAR
  n: integer;

```

```

gold,g,fac,b1,b0,anf,ana,an,a1,a0: real;
BEGIN
gln := gammln(a);
gold := 0.0;
a0 := 1.0;
a1 := x;
b0 := 0.0;
b1 := 1.0;
fac := 1.0;
FOR n := 1 to itmax DO BEGIN
  an := 1.0*n;
  ana := an-a;
  a0 := (a1+a0*ana)*fac;
  b0 := (b1+b0*ana)*fac;
  anf := an*fac;
  a1 := x*a0+anf*a1;
  b1 := x*b0+anf*b1;
  IF (a1 <> 0.0) THEN BEGIN
    fac := 1.0/a1;
    g := b1*fac;
    IF (abs((g-gold)/g) < eps) THEN GOTO 1;
    gold := g
  END
END;
writeln('pause in GCF - a too large, itmax too small'); readln;
1: gammcf := exp(-x+a*ln(x)-gln)*g
END;

```

```

PROCEDURE gser(a,x: real; VAR gamser,gln: real);
LABEL 1;
CONST
  itmax=100;
  eps=3.0e-7;
VAR
  n: integer;
  sum,del,ap: real;
BEGIN
  gln := gammln(a);
  IF (x <= 0.0) THEN BEGIN
    IF (x < 0.0) THEN BEGIN
      writeln('pause in GSER - x less than 0'); readln
    END;
    gamser := 0.0
  END ELSE BEGIN
    ap := a;
    sum := 1.0/a;
    del := sum;
    FOR n := 1 to itmax DO BEGIN
      ap := ap+1.0;
      del := del*x/ap;
      sum := sum+del;

```

```

        IF (abs(del) < abs(sum)*eps) THEN GOTO 1
    END;
    writeln('pause in GSER - a too large, itmax too small'); readln;
1:   gamser := sum*exp(-x+a*ln(x)-gln)
    END
END;

FUNCTION gammq(a,x: real): real;
VAR
    gamser,gln: real;
BEGIN
    IF ((x < 0.0) OR (a <= 0.0)) THEN BEGIN
        writeln('pause in GAMMQ - invalid arguments'); readln
    END;
    IF (x < a+1.0) THEN BEGIN
        gser(a,x,gamser,gln);
        gammq := 1.0-gamser
    END ELSE BEGIN
        gcf(a,x,gamser,gln);
        gammq := gamser
    END
END;

PROCEDURE fit(x,y: glndata; ndata: integer; sig: glndata; mwt: integer;
    VAR a,b,siga,sigb,chi2,q: real);
(* Programs using routine FIT must define the type
TYPE
    glndata = ARRAY [1..ndata] OF real;
in the main routine. *)
VAR
    i: integer;
    wt,t,sy,sxoss,sx,st2,ss,sigdat: real;
BEGIN
    sx := 0.0;
    sy := 0.0;
    st2 := 0.0;
    b := 0.0;
    IF (mwt <> 0) THEN BEGIN
        ss := 0.0;
        FOR i := 1 to ndata DO BEGIN
            wt := 1.0/sqr(sig[i]);
            ss := ss+wt;
            sx := sx+x[i]*wt;
            sy := sy+y[i]*wt
        END
    END ELSE BEGIN
        FOR i := 1 to ndata DO BEGIN
            sx := sx+x[i];
            sy := sy+y[i]
        END;
        ss := ndata
    END

```



```

BEGIN
  writeln('Least Squares Fitting of Energy vs. time of flight dataset');
  writeln('For calibration of Spectra. ');
  writeln;
  Float:=Param[12].Value;
  write('Float Voltage = ',Float:4:0,' V Change ? n');
  gotoxy(35,4);
  read(ques);
  IF (ques='y') OR (ques='Y') THEN
  BEGIN
    write(' New Value ? ');
    read(Float);
    Param[12].Value:=Float;
  END;
  writeln;
  cm:=Float*5.485802e-04;
  gotoxy(1,6);
  Wavelength:=213; (*Default*)
  write('Laser wavelength you wish to calibrate (213,266,299,355 nm available) ?
,Wavelength);
  gotoxy(73,6);
  readln(Wavelength);
  gotoxy(1,8);
  writeln('time (ns)', ' :5,'energy (eV)', ' :3,'error (ns)', ' :2,'Mass');
  wavestr:= ' ';
  str(wavelength,wavestr);
  If Exist('calib.'+ wavestr) = FALSE THEN
  BEGIN
    writeln('Not available');
    Delay(1000);
  END ELSE
  BEGIN
    assign(j,'calib.'+ wavestr); reset(j);
    read(j,ndata);
    writeln;
    FOR i:= 1 To ndata DO
    BEGIN
      read(j,T[i]);
      gotoxy(1,9+i);
      write(T[i]:6:1);
      read(j,E[i]);
      gotoxy(17,9+i);
      write(E[i]:7:5);
      read(j,sig[i]);
      gotoxy(30,9+i);
      write(sig[i]:6:1);
      read(j,M[i]);
      gotoxy(41,9+i);
      write(M[i]:5:1);
      writeln;
    END;
  END;

```

```

Close(j);
writeln('Note all times are from Time-of-flight display screen');
gotoxy(1,9+ndata+3);
write('EDIT - Return for no change, Typover for new value, -1 for ignore.');
```

REPEAT

```

  FOR i:= 1 to ndata DO
  BEGIN
    temptime:=0;
    gotoxy(1,9+i);
    readln(temptime);
    IF (temptime<>0) AND (temptime<>-1) THEN BEGIN
      T[i]:=temptime;
      END;
    IF temptime=-1 THEN
      BEGIN
        IF sig[i]=1000 THEN sig[i]:=2.5 ELSE sig[i]:=1000;
        gotoxy(30,9+i);
        write(sig[i]:6:1);
        END;
        gotoxy(1,9+i);
        writeIn(T[i]:6:1);
        END;
    gotoxy(1,9+ndata+4);
    write(' All correct ? n');
    gotoxy(15,WhereY);
    readln(ques);
  UNTIL (ques='y') OR (ques='Y');
```

```

count:=0;
FOR i:= 1 To ndata DO
  BEGIN
    IF sig[i]<>1000 THEN
      BEGIN
        count:=count+1;
        X[count]:=1.0/(sqrt(E[i]-cm/M[i]));
        Y[count]:=T[i];
        sg[count]:=sig[i];
        END;
      END;
    ndataold:=ndata;
    ndata:=count;
  IF ndata<3 THEN
    BEGIN
      writeln('Too few datapoints, need at least three to do a linear regression.');
```

```

      Delay(2000);
    END ELSE
    BEGIN
      writeln;
      FOR MWT:=0 to 1 DO BEGIN
        fit(x,y,ndata,sg,mwt,a,b,siga,sigb,chi2,q);
        writeln;
```



```

IF MWT=0 THEN BEGIN
  writeln('Fit assuming uniform uncertainties in time measurements');
  END ELSE BEGIN
  writeln('Fit using uncertainties as supplied');
  END;
writeln(' :5,'a= ',a:13:6,' :6,'uncert.= ',siga:9:6);
writeln(' :5,'b= ',b:13:6,' :6,'uncert.= ',sigb:9:6);
writeln(' ',chisq= ',chi2:13:6,' :6,'for ',ndata,' datapoints');
IF MWT=1 THEN BEGIN
  writeln('Goodness of fit (Q) = ',q:8:4);
  END;
writeln;
writeln('T0 = ',a:6:2,' ns +/- ',siga:6:2);
writeln('l = ',(b*5.93094e-02):6:2,' cm +/- ',(sigb*5.93094e-02):6:2);
(Fit is to line of the form t = tzero + K sqrt(1/E-DELTA E))

write('Use these calibration parameters ? ');
readln(rtest2);
IF (rtest2 = 'y') OR (rtest2 = 'Y') THEN
  BEGIN
  Param[10].Value:= a; (New T0)
  Param[11].Value:=(b*5.93094e-04)*100.0; (New length l)
  PulseDelay := Param[10].Value + 40.0;
  PulseSlope := 284.3174 * Sqr(Param[11].Value);
  (Write back to calibrant file)
  assign(j,'calib.'+ wavestr);
  rewrite(j);
  writeln(j,ndataold);
  FOR i:= 1 To ndataold DO
  BEGIN
    writeln(j,T[i]:6:1);
    writeln(j,E[i]:7:5);
    writeln(j,2.5:4:1);
    writeln(j,M[i]:5:1);
  END;
  Close(j);
(Now Check on fit)
  writeln('With these parameters, the fit is as follows:');
  writeln('');
  writeln(' Mass',':4,'Exact ',' ',Calib.',' Deviation', ' Ignored');
  writeln(' :9,'Energy Energy');
  rmserr:=0.0;
  FOR i:= 1 To ndataold DO
  BEGIN
    Ecalc[i]:=COMEnergy(PulseSlope/sqr(T[i]-Param[10].Value),M[i]);
    { As calibration numbers are from tof display, they have not been corrected by 40.0
ns
    Thus do not use PulseDelay but Param[10].Value directly for T0 here)
    IF sig[i]<> 1000 THEN
      BEGIN
        writeln(M[i]:6:1,' :3,E[i]:5:3,' :5,Ecalc[i]:5:3,' :4,E[i]-Ecalc[i]:6:3);

```

```

        rmserr:=rmserr + sqr(E[i]-Ecalc[i]);
    END
ELSE BEGIN
    writeln(M[i]:6:1,' :3,E[i]:5:3,' :5,Ecalc[i]:5:3,' :4,E[i]-Ecalc[i]:6:3,' :7,'***');
    END;
END; (* FOR *)
( writeln('NDATA=',ndata, 'rmserr=',rmserr);)
rmserr:=sqrt(rmserr/ndata);
writeln('');
writeln('Root Mean Square Deviation = ',rmserr:6:4,' eV');
writeln('');
write('<Press Any Key to continue>');
WHILE NOT KeyPressed do
;
END; (* If rtest2 = y *)
END; (* FOR MWT loop *)
end; {else part of if ndata<3 }

END; (* ELSE file exists *)

END;

```

3.5 scale.pas

```

procedure scale;

type
    BufString = string[50];
var
    FileVar : text;
    FileName : bufstring;
    i : integer;
    NumOfScans, PointsPerScan, TimePerPoint, r : real;
    teststr : BufString;
    ErrFlag, NewFormat : boolean;
    con : real;
    olddir : BufString;

BEGIN
    ErrFlag:=FALSE; (*Initialize flag*)
    IF round(Param[3].Value) > ((MAXSIZE-10)/2) THEN
        BEGIN
            ErrFlag:=TRUE;
            writeln(#7+'Dataset too many points for subtraction, must be <= 1200 pts');
            Delay(1500);
        END;
    END;

```

```

IF (InitSub=FALSE) AND (ErrFlag = FALSE) THEN
BEGIN
  For i:= 1 to round(Param[3].Value) do
    BackData[i]:=Data[i];
  olddir:=Directory;
  Directory:='\turbo\data\noise\';
  write('Filename (Default is data\noise\): ');
  readln(FileName);
  CheckDir(FileName);
  IF Exist(FileName)=FALSE THEN
  BEGIN
    Writeln('File does not exist');
    Delay(500);
    ErrFlag:=TRUE;
  END;
  If Exist(FileName)=TRUE THEN
  BEGIN
    Assign(FileVar,FileName);
    Reset(FileVar);
    Readln(FileVar,teststr);
    IF (teststr[1] = 'A') then NewFormat := TRUE
    else NewFormat := FALSE;
    IF NOT NewFormat then Reset(FileVar);
    Readln(FileVar,TimePerPoint);
    IF (TimePerPoint = Param[4].Value) then
    BEGIN
      Readln(FileVar, PointsPerScan);
      if PointsPerScan < Param[3].Value then Param[3].Value := PointsPerScan;
      Readln(FileVar,r);
      Readln(FileVar,NumOfScans);
      Param[1].Value := Param[1].Value + NumOfScans;
      Readln(FileVar,r);
      IF NewFormat then
      BEGIN
        Readln(FileVar,r);
        Readln(FileVar,r);
      END;
      for i:= 1 to round(Param[3].Value) do
      BEGIN
        Readln(FileVar,r);
        BackData[round(Param[3].Value)+i]:= r;
      END;
      Close(FileVar);
      Writeln('Data Read');
    END
  else
  BEGIN
    Writeln('#7 + ' Time per Point is Wrong : Cannot Add Data Set');
    Close(FileVar);
    Delay(500);
    ErrFlag:=TRUE;
  END;
END;

```

```

    END; (This ends the Time per point IF !)
    END; (*if Exist FileName*)
    Directory:=Olddir; (reset directory to what it was on entry)
    END
ELSE writeln('Using Original datasets for the raw spectrum and background Fit');

(now if InitSub was set or not)
IF ErrFlag=FALSE THEN
    BEGIN
        Write('Scale by #Scans of data/#Scans of Background = ? ');
        Readln(con);
        for i:=1 to 10 do
            begin
                Data[i]:=0;
            end;
        for i:= 11 to round(Param[3].Value) do
            Data[i] := BackData[i] - con * BackData[round(Param[3].Value)+i];
        IF InitSub=FALSE THEN
            LastFile := 'SUB ' + LastFile;
            InitSub:= TRUE; (* Now true because full file read, and plot filename set*)
            Writeln(' ');
            Writeln('Subtraction Done');
            Delay(500);
        END;
    END; (procedure)

```

3.6 ftshort.pas

```

(*Fourier transformation on time of flight spectrum, expects <1024 point datafile*)
procedure ffit(var totdata : rmbuf);

const
    nn=1024;
    nn2=2048;
    nn3=512;
    nn4=513;

var
    i,ii : integer;
    data : gldarray;
    avg : real;
    olddir : Bufstring;
TYPE
    BufString = String[50];
    parray = array[1..nn4] of real;

```

```

Procedure WriteFilea(writedata : rmbuf);
  (* write data file. *)
var
  FileVar : text;
  i : integer;
  FileName,OldDir : bufstring;
  NewFormat : boolean;
begin
  write(' Name of output file (data\noise\NAME.dat) : ');
  readln(FileName);
  CheckDir(FileName);
  Assign(FileVar,FileName);
  Rewrite(FileVar);
  writeln(FileVar,'A');
  writeln(filevar,round(Param[4].Value));
  writeln(filevar,round(Param[3].Value));
  writeln(filevar,round(Param[2].Value));
  writeln(filevar,'1'); (*Marker to indicate smoothed file*)
  writeln(filevar,round(Param[5].Value));
  writeln(filevar,round(Param[10].Value));
  writeln(filevar,round(Param[11].Value));
  for i:= 1 to round(Param[3].Value) do
  begin
    writeln(FileVar,writedata[i]:0:1);
  end;
  Close(FileVar);
  Writeln(' ');
  Writeln(i,' Data Points written');
end; { procedure }

```

```

Procedure WriteFour;
VAR
  FileVar :text[$800];
  i : integer;
BEGIN
  assign (FileVar,'FOURTR.DAT');
  rewrite(FileVar);
  For i:=1 to nn2 do
    Writeln(FileVar,data[i]);
  Close(FileVar);
END;

```

```

PROCEDURE ReadFour;
VAR
  FileVar : text[$800];
  i: integer;
BEGIN
  assign (FileVar,'FOURTR.DAT');
  reset(FileVar);
  For i:=1 to nn2 do
    Readln(FileVar,data[i]);

```

```
Close(FileVar);
END;
```

```
Procedure Fourier1;
var
```

```
  i,Thecase,ii,io,cutl,cuto : integer;
  Noise,freq,a: real;
  Filter: parray;
```

```
BEGIN
```

```
(* Wiener Filtering - see Numerical Recipes by Press et al..*)
```

```
  Writeln(' Wiener Filter : Smooth Lorentzian Filter');
```

```
  a:=15;
```

```
  Write('FWHM/2 of Lorentzian (default=15) ? ');
```

```
  Readln(a);
```

```
  FOR ii:= 1 to nn3+1 DO
```

```
    Filter[ii]:=1.0/(1+sqr(int(ii-1)))/sqr(a);
```

```
  For ii:=1 to nn3+1 do
```

```
    BEGIN
```

```
      i:=2*ii;
```

```
      data[i-1]:=data[i-1]*filter[ii];
```

```
      data[i]:=data[i]*filter[ii];
```

```
    END;
```

```
  For ii:=nn3+2 to nn do
```

```
    BEGIN
```

```
      i:=2*ii;
```

```
      data[i-1]:=data[i-1]*filter[nn-ii+2];
```

```
      data[i]:=data[i]*filter[nn-ii+2];
```

```
    END;
```

```
end;      (procedure)
```

```
(*Main procedure*)
```

```
BEGIN
```

```
  Olddir:=Directory;
```

```
  Directory:='\turbo\data\noise\';
```

```
  Writeln('Fourier Transformation of a ',round(Param[3].Value),' point time of flight spectrum');
```

```
  Writeln;
```

```
  Write('Is the transformed spectrum already stored as FOURTR.DAT ? (1=yes) ');
```

```
  Readln(i);
```

```
  IF i <> 1 THEN
```

```
    BEGIN
```

```
      For ii:=1 to round(Param[3].Value)-10 do
```

```
        BEGIN
```

```
          data[2*ii-1]:=totdata[ii];
```

```
          data[2*ii]:=0;
```

```
        END;
```

```
      avg:=0.0;
```

```

For ii:= round(Param[3].Value)-9 to round(Param[3].Value) do (*extrapolate the last ten
points out to nn*)
BEGIN
data[2*ii-1]:=totdata[ii];
data[2*ii]:=0;
avg:=avg+totdata[ii];
END;
avg:=avg/10.0;
For ii:=2*round(Param[3].Value)+1 to nn2 do
data[ii]:=avg; (pad end with average of last ten points)
FOUR1(data,nn,1); (transform into frequency space)

WriteFour; ( store the frequency spectrum)

END (if)
ELSE BEGIN
ReadFour; (*Read fourier transform file*)
END;

FOURIER1;

FOUR1(data,nn,-1); (Inverse Transform)

writeln;
For ii:=1 to round(Param[3].Value) do
BEGIN
totdata[ii]:=data[ii*2-1]/nn;
END;
(* Set first two points to min and max of input data for comparison*)
totdata[1]:=maxval;
totdata[2]:=0;

LastFile:='FTFIT'+ LastFile;
Write('Save the smoothed spectrum ? (1=yes, default = no) ');
i:=0; (default)
Readln(i);
IF i=1 THEN
WriteFilea('otdata);
(*in any case the smoothed spectrum is passed back to main routine for display*)
Directory:=OldDir; (reset to old directory)
Param[1].Value:=1; (* Set number of Scans to 1 to mark the fact that this is sub file*)
END;

```

3.7 four1.pas

```

PROCEDURE four1(VAR data: gldarray; nn, isign: integer);

```

(* Programs using routine FOUR1 must define type
TYPE

gldarray = ARRAY [1..nn2] OF real;
in the calling routine, where nn2=nn+nn. *)

VAR

ii,jj,n,mmax,m,j,istep,i: integer;
wtemp,wr,wpr,wpi,wi,theta: real;
tempr,tempi: real;

BEGIN

n := 2*nn;

j := 1;

FOR ii := 1 to nn DO BEGIN

i := 2*ii-1;

IF (j > i) THEN BEGIN

tempr := data[j];

tempi := data[j+1];

data[j] := data[i];

data[j+1] := data[i+1];

data[i] := tempr;

data[i+1] := tempi

END;

m := n DIV 2;

WHILE ((m >= 2) AND (j > m)) DO BEGIN

j := j-m;

m := m DIV 2

END;

j := j+m

END;

mmax := 2;

WHILE (n > mmax) DO BEGIN

istep := 2*mmax;

theta := 6.28318530717959/(isign*mmax);

wpr := -2.0*sqr(sin(0.5*theta));

wpi := sin(theta);

wr := 1.0;

wi := 0.0;

FOR ii := 1 to (mmax DIV 2) DO BEGIN

m := 2*ii-1;

FOR jj := 0 to ((n-m) DIV istep) DO BEGIN

i := m + jj*istep;

j := i+mmax;

tempr := wr*data[j]-wi*data[j+1];

tempi := wr*data[j+1]+wi*data[j];

data[j] := data[i]-tempr;

data[j+1] := data[i+1]-tempi;

data[i] := data[i]+tempr;

data[i+1] := data[i+1]+tempi

END;

wtemp := wr;

wr := wr*wpr-wi*wpi+wr;

wi := wi*wpr+wtemp*wpi+wi


```

    END;
    mmax := istep
  END
END;

```

3.8 modfile.pas

```

TYPE double = real; char12 = string[12];
FUNCTION sngl(x:real):real; BEGIN sngl := x END;
PROCEDURE glopen(VAR infile:text; filename:char12);
  BEGIN assign(infile,filename); reset(infile) END;

```

3.9 tp3decl3.pas

(***** TURBO Pascal Declarations *****)

(* 10/30/91 These are the new routines for TURBO interface with GPIB board running version C11 of the GPIB software. The installation tp3decl.pas has been pruned down to tp3decl3.pas in the same way as original RB Metz code *)

(*V-*) (* relax string length restrictions *)

```

Const MAXIBBUF = $1000; (* maximum buffer size for I/O functions *)
      MAXINTBUF = $1000; (* maximum buffer size for integer I/O functions *)

```

```

Type iobuf = array[1..MAXIBBUF] of char;
     io1buf = array[1..MAXINTBUF] of integer;
     ibstring = string[50];
     str4 = string[4];

```

```

Var ibsta : integer; (* status word *)
    iberr : integer; (* GPIB error code *)
    ibcnt : integer; (* # bytes sent or DOS error code *)
    ibbuf : iobuf; (* I/O buffer for commands/data *)
    intbuf : io1buf; (* integer I/O buffer for data *)
    bdname : ibstring; (* board or device name *)
    bd : integer; (* board descriptor *)

```

vcnt : integer; (* v or byte count *)

```
function ibfn (name:ibstring;
              var iberr,ibcnt:integer;
              var buf:iobuf;
              var buf1:io1buf;
              var vcnt:integer;
              bd,fncode:integer):integer; external 'c:\turbo\gpib\tp3ib.com';
```

(* You MUST include the appropriate declaration, as given below, for each *)
(* procedure or function you call. You may omit declarations for functions *)
(* you do not call. *)

```
procedure ibclr (bd:integer);
begin
  ibsta := ibfn(bdname,iberr,ibcnt,ibbuf,intbuf,vcnt,bd,22);
end;
```

```
procedure ibdma (bd:integer;v:integer);
begin
  ibsta := ibfn(bdname,iberr,ibcnt,ibbuf,intbuf,v,bd,11);
end;
```

```
function ibfind (bdname:ibstring):integer;
var name : ibstring;
begin
  name := bdname + chr(0);
  ibfind := ibfn(name,iberr,ibcnt,ibbuf,intbuf,vcnt,bd,27);
end;
```

```
procedure ibrd (bd:integer;var ibbuf:iobuf;cnt:integer);
begin
  ibsta := ibfn(bdname,iberr,ibcnt,ibbuf,intbuf,cnt,bd,28);
end;
```

```
procedure ibrdi (bd:integer;var intbuf:io1buf;cnt:integer);
begin
  ibsta := ibfn(bdname,iberr,ibcnt,ibbuf,intbuf,cnt,bd,37);
end;
```

```
procedure ibsic (bd:integer);
begin
  ibsta := ibfn(bdname,iberr,ibcnt,ibbuf,intbuf,vcnt,bd,3);
end;
```

```
procedure ibwrt (bd:integer;ibbuf:iobuf;cnt:integer);
begin
  ibsta := ibfn(bdname,iberr,ibcnt,ibbuf,intbuf,cnt,bd,30);
end;
```

```
procedure ibwrti (bd:integer;intbuf:io1buf;cnt:integer);
```

```
begin
  ibsta := ibfn(bdname,iberr,ibcnt,ibbuf,intbuf,cnt,bd,38);
end;
```

3.10 graphli2.tgt

This file is a condensed version of the graphics subroutines in graphlib.tgt which were written by William F. Polik and are available as Turbo Graphic Tools, ver. 1.10, distributed by The Campus Software Office, 295 Evans Hall, University of California, Berkeley, California 94720. As this file is functionally equivalent to the unedited (and commercially available) version, it is not listed here.

3.11 expt.par

```
Accumulate
Scans
  2.000000000000000E+004
  5.000000000000000E+005
1
5000
Update Every
Scans
  4.000000000000000E+002
  5.000000000000000E+003
1
50
Number of Points per Scan

  8.000000000000000E+002
  2.500000000000000E+003
100
100
```

Time per Point
nsec
0.00000000000000E+000
6.00000000000000E+000
0
1
Offset
0.00000000000000E+000
2.55000000000000E+002
0
10
Time Scan
1.00000000000000E+000
1.00000000000000E+000
0
1
Background Subtraction
0.00000000000000E+000
1.00000000000000E+000
0
1
Laser Wavelength
1.00000000000000E+000
8.00000000000000E+000
0
1
Ion Mass
amu
2.10000000000000E+001
2.00000000000000E+004
1
1
T0
nsec
8.06700000000000E+001
1.00000000000000E+004
-10000
1
Length
cm
8.99100000000000E+001
2.00000000000000E+002
50
1
Float
V
1.26000000000000E+003

2.500000000000000E+003
500
100

3.12 help1.par

Current version of program is mgplt19 "TENURE"

(all commands use Alt-key combinations)

while experiment is running :

(S)top experiment
toggle (L)ines <-> dots

when experiment is stopped :

exit program (F1)
(H)elp (display this screen)
change default (D)irectory
start a new (E)xperiment
(R)ead a data file
(A)dd a data file
Subtract a (N)oise Fit file
(W)rite (save) results in a file
(U) Write an energy spectrum file for export
(C)alibrate spectrum

smoothing data :

gaussian smoothing (F3)
(F)ourier Transform Fit
< press any key for next help screen >

3.13 help2.par

while experiment is stopped :

plot data (connect points) (F7)
plot data (don't connect points) (F8)
display (T)ext screen
display (G)raphics screen

in text mode :

select parameter to modify (up/down arrows)

increase/decrease parameter value (right/left arrows)
manually (ENTER) new parameter

in graphics mode :

move cursor :

right/left one data point (right/left arrow)
right/left 10 data points (shift-right/left arrow)
right/left 100 data points (ctrl-right/left arrow)

display data values (Enter)

in a mass scan :

pick first point for a mass calibration (alt - 1)
pick second point and calibrate mass scan (alt - 2)

< press any key to exit help >

Appendix C: Publications from Graduate Work

1. R. B. Metz, T. Kitsopoulos, A. Weaver and D. M. Neumark, "Study of the Transition State Region in the Cl + HCl Reaction by Photoelectron Spectroscopy of ClHCl⁻," J. Chem. Phys. **88**, 1463 (1988).
2. A. Weaver, R. B. Metz, S. E. Bradforth and D. M. Neumark, "Spectroscopy of the I + HI Transition State Region by Photodetachment of IHI⁻," J. Phys. Chem. **92**, 5558 (1988).
3. A. Weaver, R. B. Metz, S. E. Bradforth and D. M. Neumark, "Observation of the $\tilde{A}(^2B_2)$ and $\tilde{C}(^2A_2)$ states of NO₂ by Negative Ion Photoelectron Spectroscopy of NO₂⁻," J. Chem. Phys. **90**, 2070 (1989).
4. R. B. Metz, A. Weaver, S. E. Bradforth, T. N. Kitsopoulos and D. M. Neumark, "Probing the Transition State with Negative Ion Photodetachment: The Cl + HCl and Br + HBr Reactions," J. Phys. Chem. **94**, 1377 (1990).
5. S. E. Bradforth, A. Weaver, D. W. Arnold, R. B. Metz and D. M. Neumark, "Examination of the Br + HI, Cl + HI, and F + HI Hydrogen Abstraction Reactions by Photoelectron Spectroscopy of BrHI⁻, ClHI⁻, and FHI⁻," J. Chem. Phys. **92**, 7205 (1990).
6. A. Weaver, R. B. Metz, S. E. Bradforth and D. M. Neumark, "Investigation of the F + H₂ Transition State Region via Photoelectron

- Spectroscopy of the FH_2^- Anion," *J. Chem. Phys.* **93**, 5352 (1990).
7. R. B. Metz, D. R. Cyr and D. M. Neumark, "Study of the $^2\text{B}_1$ and $^2\text{A}_2$ States of CH_2NO_2 via Ultraviolet Photoelectron Spectroscopy of the CH_2NO_2^- Anion," *J. Phys. Chem.* **95**, 2900 (1991).
 8. R. E. Continetti, D. R. Cyr, R. B. Metz and D. M. Neumark, "Fast Beam Studies of N_3 Photodissociation," *Chem. Phys. Lett.* **182**, 406 (1991).
 9. R. B. Metz, S. E. Bradforth, and D. M. Neumark, "Transition state spectroscopy of bimolecular reactions using negative ion photodetachment," *Adv. Chem. Phys.* (accepted, 1991).
 10. S. E. Bradforth, D. W. Arnold, R. B. Metz, A. Weaver and D. M. Neumark, "Spectroscopy of the Transition State: Hydrogen Abstraction Reactions of Fluorine," *J. Phys. Chem.* (accepted, 1991).
 11. R. B. Metz and D. M. Neumark, "Adiabatic Three-Dimensional Simulations of the IHI^- , BrHI^- , and BrHBr^- Photoelectron Spectra," in preparation.
 12. R. B. Metz, R. E. Continetti, D. R. Cyr and D. M. Neumark, "Photodissociation of CH_2NO_2 at 208, 235, and 270 nm," in preparation.

END

**DATE
FILMED**

8 / 19 / 92

