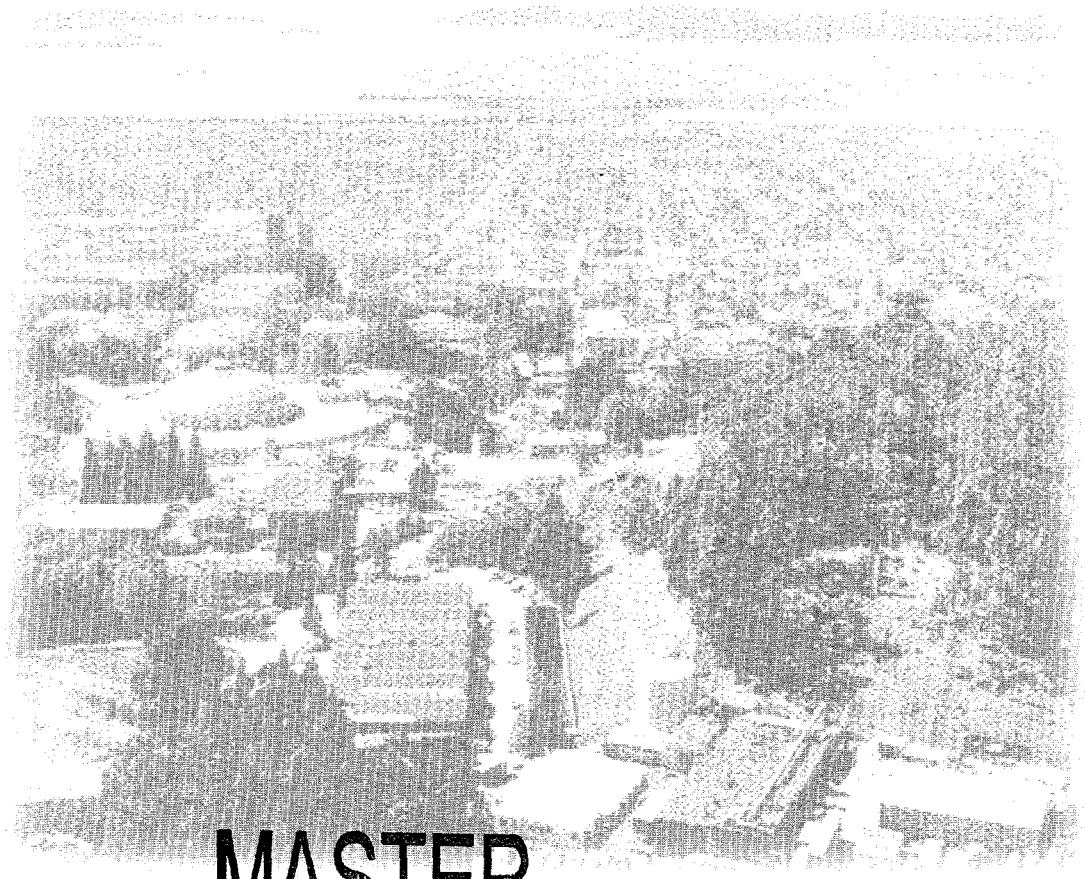
LBNL-40237
UC-411

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

Photodissociation Dynamics and Spectroscopy of Free Radical Combustion Intermediates

David L. Osborn
Chemical Sciences Division

December 1996
Ph.D. Thesis



LBNL-40237
UC-1600

Photodissociation Dynamics and Spectroscopy of Free Radical Combustion Intermediates

David Lewis Osborn
Ph.D. Thesis

Department of Chemistry
University of California, Berkeley

and

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

December 1996

This work 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.

DISCLAIMER

Photodissociation Dynamics and Spectroscopy of Free Radical Combustion Intermediates

by

David Lewis Osborn

Bachelor of Science (University of Chicago) 1991

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Doctor of Philosophy

in
Chemistry

in the
GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Daniel M. Neumark, Chair
Professor Richard J. Saykally
Professor Jeffrey A. Reimer

Fall 1996

This work 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.

Photodissociation Dynamics and Spectroscopy of Free Radical Combustion Intermediates

by

David Lewis Osborn

ABSTRACT

Photodissociation Dynamics and Spectroscopy of Free Radical Combustion Intermediates

by

David Lewis Osborn

Doctor of Philosophy in Chemistry

University of California, Berkeley

Professor Daniel M. Neumark, Chair

The photodissociation spectroscopy and dynamics of free radicals is studied by the technique of fast beam photofragment translational spectroscopy. Photodetachment of internally cold, mass-selected negative ions produces a clean source of radicals, which are subsequently dissociated and detected. The photofragment yield as a function of photon energy is obtained, mapping out the dissociative and predissociative electronic states of the radical. In addition, the photodissociation dynamics, product branching ratios, and bond energies are probed at fixed photon energies by measuring the translational energy, $P(E_T)$, and angular distribution of the recoiling fragments using a time- and position-sensitive detector. *Ab initio* calculations are combined with dynamical and statistical models to interpret the observed data.

The photodissociation of three prototypical hydrocarbon combustion intermediates forms the core of this work. The methoxy radical (CH_3O), the vinoxy radical (CH_2CHO)

and the ketenyl radical (HCCO), are representative examples of alkanoxy, alkenoxy, and alkynoxy radicals, respectively. The dominant channel in methoxy dissociation following ultraviolet excitation is $CH_3O \rightarrow CH_3 + O$. Vibrational structure is resolved in the $P(E_T)$ distribution, and a new heat of formation is obtained for CH_3O. Two primary channels are active in vinoxy photodissociation: $CH_2CHO \rightarrow CH_3 + CO$, and $CH_2CHO \rightarrow CH_2CO + H$. The dissociation dynamics in vinoxy occur by internal conversion to the ground state followed by dissociation over a barrier to products. The ketenyl radical is observed for the first time in the ultraviolet, dissociating via two distinct channels: $HCCO \rightarrow CH\ X(^2\Pi) + CO$, and $HCCO \rightarrow CH\ a(^4\Sigma^-) + CO$. The dynamics in this case occur by internal conversion and intersystem crossing, followed by dissociation on potential surfaces with negligible barriers to products. The branching ratio is a function of photon energy, and a heat of formation is determined for ketenyl.

In addition, the photodissociation of a negative ion radical, $N_2O_2^-$, is also explored. This ion photodissociates via two channels: $N_2O_2^- \rightarrow O^- + N_2O$, and $N_2O_2^- \rightarrow NO^- + NO$. Product vibrational structure is observed in the N_2O bending mode, and a heat of formation is obtained for $N_2O_2^-$.

DEDICATION PAGE

To my father

Donald Lewis Osborn

and my mother

Wanda Kay Hodges Osborn

TABLE OF CONTENTS

Acknowledgments

I would first and foremost like to thank Dan Neumark for inviting me to join his research group and the fast radical beam machine project in the fall of 1991. As my understanding of academic research in general and physical chemistry in particular has increased over the years, I am ever more impressed by Dan's ability to conceive novel experiments and bring them rapidly to fruition. His advice on matters both experimental and theoretical has been indispensable, and I am especially grateful that he allowed me to pursue the projects which I felt had merit rather than dictate the course of our research.

I could not have made it this far without the help and guidance provided by my colleagues on the FRBM team. Working with Bob Continetti, the first post-doc on the FRBM project, was like shifting into the fast lane of the autobahn—the only option is to accelerate quickly. His amazing intensity and passion for science set a high standard in the laboratory. The successful development of the time- and position-sensitive detector is a testament to his experimental expertise, and he laid the framework for the data analysis code presented in Appendix A. Although he characterized me as an "outlaw biker," upon leaving Berkeley for his new position at UC San Diego Bob entrusted to me a big, leafy schefflera, which he had acquired from Gil Nathanson, and which is doubtless endowed with science karma, judging from the success of its two former owners.

Doug Cyr deserves credit for bringing the FRBM project from ideas and drawings to the huge mass of aluminum, lasers, and electronics which is ensconced in D-21 Latimer Hall. Under his patient guidance I learned how to operate and understand the beast of an apparatus he had constructed. Doug made me appreciate attention to detail in design and taught me most of my drafting skills. Always planning for the future, he amassed an

immense stockpile of nuts, bolts, vacuum hoses, saw blades, big red warning lights, and most of all the ubiquitous wall of paper towels at the back of the lab, which I think we have finally depleted. Doug pioneered the pulsed discharge ion source in the group, which made the study of all three alkoxy radicals in this dissertation possible. He also taught me valuable lessons on balancing science with the rest of life, being very dedicated to his (continually growing) family while simultaneously accomplishing first-rate research.

David Leahy succeeded Bob Continetti as the second talented post-doc with whom I have had the opportunity to work. An exceptional scientist, David brought fresh ideas and exactness to the experiment, along with an electron talisman which still hangs over the electron detector. His persistence in developing the factor determination scheme described in Chapter 2 led to a crucial improvement in energy resolution, without which the experiments on CH_3O would have been much less informative. David taught me discipline in programming and in scientific writing, and demonstrated how to ski straight down a steep hill with only momentary lapses of control. His amiable character, agile hacky-sack moves, and goofy exclamations made the long hours in the lab much more enjoyable.

In 1994, Hyeon Choi joined the FRBM project with a sound theoretical background from his education in Korea. Ever polite, he listened patiently to my tirades about safety and procedure, and soon became an asset in the lab. It was Hyeon who bravely mixed acetylene with oxygen to develop a reliable (and safe) method for the production of $HCCO^-$ ions, making the study of the ketenyl radical possible. Hyeon deserves special mention for taking care of my "growling" cats during my trips away from Berkeley.

Another energetic post-doc, David Mordaunt, joined the project in 1995. Already a honed translational spectroscopist, with impressive theoretical credentials to boot, he contributed immediately to the HCCO experiments and developed the statistical adiabatic impulsive model. Always competitive, he outdid previous colleagues by skiing with nearly complete disregard for his own well-being, and has since become a shredding snowboarder.

The most recent addition to the team is Ryan Bise, whose skill at synthesis and dedication in the lab have already brought him success. I am confident he will lead the FRBM through new and exciting experiments in the coming years.

Many other members of the Neumark group, both past and present, have provided an important storehouse of knowledge and ideas for which I am grateful. Although I only overlapped with them for a month, Dan's first students, Theo Kitsopoulos, Ricky Meets, and Alex Weaver have in the past years generously provided me with advice on life after Berkeley and accounts of the early Stonehenge and Acropolis eras of the group. Steve Bradforth introduced me to the wonders of UNIX, cheap burritos, and propagation in imaginary time, and has recently provided much appreciated perspective on the academic track and femtosecond chemistry experiments.

Don Arnold was a great source of experimental knowledge, and introduced me to the fields of electronic structure calculations and photoelectron spectroscopy. His leadership on the group softball team and perpetual optimism contributed greatly to the spirit of the group. ZEKE-queen and fellow doughnut-lover Caroline Chick deserves recognition for instructing me on the Jahn-Teller effect and the subtleties of the thighmaster. Gordon Burton, a truly punny fellow, shared his adventurous culinary

exploits with me, and together with Esther deBeer and Eun Ha Kim, acquired photoelectron spectra on Stonehenge to characterize the discharge source. I am indebted to Cangshan Xu, a true master of personal computers, for helping me with UNIX administration. Together with Travis Taylor, we acquired further photoelectron spectra which were crucial for characterizing the FRBM radical production scheme on each new species. I commend Yuexing Zhao and Ivan Yourshaw of ZEKE fame for loaning me laser dye and laser parts every time I needed them to continue an experiment. Mike Furlanetto, the latest addition to the ZEKE crew, takes the hat of UNIX administrator from me, which I'm sure he will wear well. Finally, Jeff Greenblatt and Marty Zanni, proprietors of the impressive femtosecond photoelectron spectrometer, have been gracious enough to share their laser expertise with me before I, too, jump into femtoland.

Several other members of the chemistry department deserve recognition for the support they have given me. I thank Cheryn Gliebe, the most capable secretary I have ever met, for her advice on matters as mundane as purchasing, and as magnificent as Yosemite. Eric Granlund, Hans Graetsch, and Dave Murai of the machine shop made numerous parts for me, sometimes on very short notice, and often with improvements to my design. Tim Robinson of the graphics lab frequently regaled me with jargon I had no hope of understanding, and together with Dan Gezelter of the Miller group helped me through several ridiculous UNIX fiascoes.

I would like to thank my examination committee chair, Y. T. Lee, for his inspiration and perspective, and the rest of my committee, Rich Saykally, Evan Williams, and John Clarke. Rich Saykally in particular has provided many insights on spectroscopic issues which have arisen during my research. I thank Jeff Reimer of chemical engineering

for his enthusiasm in reading this dissertation. I also acknowledge the National Defense Science and Engineering Graduate Fellowship program for support during my first three years at Berkeley.

Over the last year, Celeste Rohlfing of Sandia National Labs has taught me a great deal about electronic structure, explaining the procedures and terminology of *ab initio* computation while helping me avoid some of the vagaries of CASSCF calculations. I have truly enjoyed collaborating with her on the CH_2CHO and HCCO projects.

I am very grateful to Don Levy at the University of Chicago for allowing me to join his research group, introducing me to the world of spectroscopy during my junior and senior years. Joe Alfano, Jeff Elam, and the rest of the Levy group deserve credit for fielding the barrage of questions I asked them. Graham Fleming, Jeremy Burdett, and Laurie Butler were helpful mentors who encouraged me to pursue graduate work in chemistry. Finally, I would like to thank Bruce Holder, my high school chemistry teacher, who first introduced me to chemistry and Monty Python.

Even with a compelling project and the supportive environment of the Neumark group, I could never have finished graduate school without the avocations that kept me sane throughout these years. Ronald Barnes, my carillon teacher and good friend, always provided a place for me in Sather Tower, where I could escape to the world of music and the strange art of campanology. I feel privileged to have studied with him during my time at Berkeley and am honored by the compositions and arrangements he created for me. The rest of the carillon staff, John Agraz, Jeff Davis, David Hunsberger, and Richard Strauss also deserve sincere thanks for accepting me into their ranks and providing much needed breaks from scientific endeavors.

My climbing, skiing, and kayaking partners have helped me explore the great outdoors of California, and I am thankful that I have had the opportunity to enjoy their company. The time I have spent on the walls of Yosemite and in the snow and rivers of the Sierra Nevada has given me perspective and motivation I never could have acquired from the vantage point of the city or the laboratory.

I am grateful for the support of my father and sister, who have encouraged without pressuring and given me the benefit of the doubt even though I am still in school at age twenty-eight. Finally, I thank Julie for a constant supply of granola, for diligent proofreading of this work, and for enriching my life with her presence.

1 Introduction

The science of chemistry is first and foremost an attempt to understand the process of formation and cleavage of chemical bonds, the interactions which bind nuclei together in particular arrangements through electron-nuclear attraction. The core of physical chemistry, in turn, is an effort to uncover the fundamental mechanisms of chemical transformations. Toward this end, perhaps the most compelling system to study is a bimolecular reaction

$$AB + CD \rightarrow [ABCD]^* \rightarrow products$$

in the gas phase at very low pressure, where $[ABCD]^*$ represents a transient excited species. In this case, it is hoped that the essence of the reaction will be revealed due to the absence of external perturbations, such as collisions with solvent molecules or the walls of the reaction vessel. In the last forty years, great strides toward this goal have been made in the field of chemical reaction dynamics.[1,2]

However, even in the relatively well-defined limit of a crossed molecular beam experiment, in which the reactants approach and the products are detected along defined trajectories, the results are inherently averaged over a range of impact parameters describing the difference between head-on and glancing collisions of the reactants. One way to avoid averaging over the impact parameter in the bimolecular "full collision" described above is to simplify the system by studying the unimolecular "half collision" of the dissociation reaction

$$[ABCD]^* \rightarrow AB + CD, \tag{1}$$

in which the positions of the *AB* and *CD* moieties have a definite relative orientation in the excited $[ABCD]^*$ species. The "reactant" $[ABCD]^*$ can be prepared in a highly specific manner by excitation of the molecule *ABCD* with a photon of known energy, in which case the process is termed photodissociation.

Photodissociation dynamics experiments provide a direct means of studying the forces which act on a molecule as a bond is broken, and hence are an important method in studying the fundamental nature of chemical reactions, as indicated by the large body of experimental and theoretical research in this field.[3,4] In photodissociation experiments, some of the most important questions are:

What are the nascent products?

Is there more than one product channel participating?

What is the bond strength, or ΔH_{rxn} for each channel?

How fast does dissociation take place?

How is the excess energy distributed among the products?

Can we determine the dissociation mechanism from this information?

For the sake of experimental simplicity, the vast majority of photodissociation experiments to date have been performed on stable, closed-shell molecules. Of course, all chemical reactions must involve open-shell, free radical species as intermediates between reactants and products, and it is clearly of interest to subject this class of molecules to the same rigorous photodissociation investigations as employed on closed-shell molecules.

Despite the importance of free radicals in chemistry, little spectroscopic information is available on their dissociative electronic states, and even less is known about their dissociation dynamics.

The principal impediment to photodissociation studies of free radicals has been the generation of a pure sample of these intermediates. To circumvent this problem, a general technique has been developed in our laboratory which prepares well-defined, mass-selected samples of free radicals by photodetachment of negative ion precursors. Recognizing that free radicals have positive electron affinities, the corresponding negative ion can be produced and separated from contaminant species by mass spectrometry. The free radical of interest can be generated from the anion by laser photodetachment with one photon, after which the radical itself can be dissociated with a second photon. The most important aspect of this approach is that the anion precursors, by virtue of their charge, can be separated from the host of other species present, thereby eliminating the ambiguity that plagues most neutral radical experiments, which must in some way identify which species is producing the experimental signal. Using this technique, known as Fast Radical Beam Photofragment Translational Spectroscopy, I address in this dissertation the questions presented above with experiments and calculations on the photodissociation spectroscopy and dynamics of neutral and negative ion free radicals.

$$ABCD^- \xrightarrow{h\nu_1} ABCD + e^- \xrightarrow{h\nu_2} AB + CD$$

photodetachment photodissociation

Principles of photodissociation dynamics

The goal of photodissociation dynamics experiments is to describe as rigorously as possible the potential energy surface(s) (PES) on which the dissociation reaction takes place. In the most general picture, the PES is a function of the coordinates of all the electrons and nuclei in the molecule, describing the energy associated with every possible configuration. The description of the potential energy of a molecule can be simplified by invoking the Born-Oppenheimer approximation, in which it is assumed that electronic motion can be separated from nuclear motion because the light electrons adjust essentially instantaneously to a displacement of the ponderous nuclei. The Born-Oppenheimer approximation in the adiabatic representation is exact in the limit that the nuclei move infinitely slowly, i.e., adiabatically. Adiabatic Born-Oppenheimer surfaces describe the potential energy of

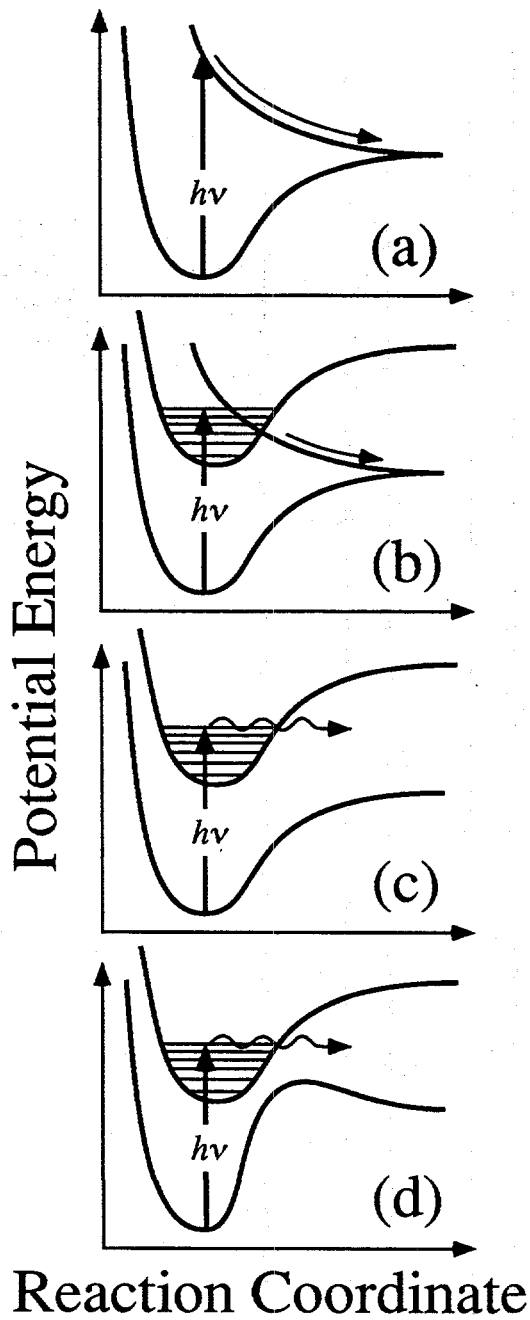


Figure 1: Four canonical dissociation schemes: (a) direct excitation to a repulsive surface, (b) predissociation via coupling to a repulsive curve, (c) ISC / IC to a barrierless potential, (d) ISC / IC to a potential with an exit barrier.

the molecule as a function of nuclear configuration for each distinct electronic state. A schematic diagram of four surface configurations common to photodissociation reactions is given in Fig. 1. The slope of the PES along the nuclear coordinates, $\partial V/\partial Q$, describes the forces which act on the molecule as a bond is broken, and it is these forces which determine the dynamics of the dissociation and the disposal of energy among translational, vibrational, rotational, and electronic degrees of freedom of the products. As long as the electronic potential V (which depends parametrically on the nuclear coordinates Q) changes slowly as a function of Q, the adiabatic representation is accurate.

The present experiments are designed to probe the potential energy surfaces of free radicals by measuring the final state distributions of the two molecular fragments created in a photodissociation event. This inquiry is comprised of two separate experiments with complementary objectives. First, the relative positions of PESs which are dissociative, or lead to dissociation, must be mapped out as a function of the photon energy which initiates the reaction. Second, for those photon energies which cause photodissociation, the products are interrogated to determine the disposal of energy among product degrees of freedom, which, as stated above, is a direct consequence of the PES and the dynamics which take place on it.

I have chosen to present the four potential curves in Fig. 1 because each represents an important class of photodissociation reactions, and each system presented in this work can be classified in one of these categories. In all cases the molecule makes an electronic transition from its ground electronic and vibrational state to an excited electronic state by absorption of a single visible or ultraviolet photon of energy $h\nu$. According to the Franck-Condon approximation, the transition to the excited electronic state occurs

instantaneously, such that the geometry of the molecule does not change during the transition. The nuclear positions then evolve under the influence of the forces $\partial V/\partial Q$, which may lead to dissociation if $h\nu > D_0$, one of the bond strengths. Note that the potentials in Fig. 1 describe motion along one nuclear coordinate Q; for triatomic and larger molecules the PES is multidimensional, and this diagram represents only the reaction coordinate along which the bond is broken.

In Fig. 1a, excitation to a directly repulsive surface results in an absorption spectrum which is lifetime broadened by the prompt dissociation. The shape of the repulsive potential generally deposits a large amount of energy into product translation, giving a non-statistical distribution of energy among the product states. For the remaining three cases in Fig. 1, the Born-Oppenheimer approximation breaks down when the molecule makes a non-radiative transition from the initially excited surface (which is bound with respect to dissociation) to another PES which is unbound along the reaction coordinate Q, an effect known as predissociation. In Fig. 1b, the initially excited state is bound, but couples to a purely repulsive surface on which dissociation proceeds rapidly. The product state distribution is often similar to that observed in direct dissociation (Fig. 1a), because the final surfaces have the same shape in both cases. Figure 1b describes the dissociation of the CH_3O radical presented in Chapter 3.

Figures 1c and d also depict predissociation, but in this case the initially excited molecule couples back to the ground state (or possibly a lower-lying excited state) surface and dissociates either with (Fig. 1d), or without (Fig. 1c), a barrier to products. The former situation describes the dissociation dynamics of the CH_2CHO radical presented in Chapter 4, while the latter case exemplifies the PES of the HCCO radical in Chapter 5. If

the coupling occurs to a state with identical spin multiplicity as the initially excited state, the coupling process is called internal conversion (IC), while if the spin multiplicity changes, the process is known as intersystem crossing (ISC). The important distinction in either case is that the final surface on which the dissociation takes place has a potential well along the reaction coordinate, in contrast to the situation in Figs. 1a and b. The presence of the well often traps the molecule for some time before dissociation occurs, leading to a more or less statistical redistribution of the available energy among some or all degrees of freedom. Consequently, little energy is deposited in translational motion of the products because tetra-atomic and larger molecules have more vibrational than translational degrees of freedom. Note that the portion of the ground state PES to the right of the barrier in Fig. 1c has the same repulsive nature as seen in Fig. 1b, and for this reason there are some similarities between the two situations. In all cases, the salient feature is that *the PES determines the final state distribution of the products, so that measurement of these distributions provides information about the potential energy surfaces of the system.*

Photodissociation of open-shell vs. closed-shell molecules

Before discussing the application of photodissociation experiments to free radicals, recall that a free radical is physically stable, but, with rare exceptions, is chemically unstable. Physical stability means that the ground state PES is a potential well, and as such, *isolated* free radicals are indefinitely stable as long as the well depth is significantly larger than the thermal energy k_bT. Chemical instability, on the other hand, denotes the fact that free radicals are extremely reactive in the presence of other molecules, often

reacting after a single collision, and must be studied on a timescale shorter than the collision lifetime. The meaning of the word *stable* is therefore context-sensitive.

While the dynamics occurring on a PES generally do not depend on whether the molecule is closed or open-shell, the potential surfaces themselves are often qualitatively different. This distinction arises for two principal reasons. First, due to their open-shell electronic structure, radicals tend to have low-lying (i.e., 0 - 5 eV) electronic states. Second, closed-shell molecules often dissociate to give two free radicals, for example $CH_3I \rightarrow CH_3 + I$, or $H_2O_2 \rightarrow OH + OH$. In this case, the PES for the reverse reaction, radical-radical recombination, typically has no activation barrier. Free radicals, on the other hand, often dissociate to an open-shell and a closed-shell pair of fragments, e.g., $CH_2CHO \rightarrow CH_3 + CO$, or $CH_2NO_2 \rightarrow NO + CH_2O$, which implies that the reverse reaction is usually impeded by a barrier representing the energy required to convert a quite content closed-shell molecule into a reactive species. For these reasons, the dissociation energies of free radicals are typically less than for similar closed-shell molecules, and the potential energy surfaces describing closed-shell vs. open-shell dissociation are qualitatively different, leading to different reaction dynamics.

As an example, consider the photodissociation of nitromethane (CH_3NO_2) vs. the nitromethyl radical (CH_2NO_2). Upon absorption of a single ultraviolet photon, the only product channel observed in nitromethane dissociation is simple C–N bond fission, $CH_3NO_2 \rightarrow CH_3 + NO_2$. In contrast, the nitromethyl radical shows two different dissociation channels: simple N–O bond fission, $CH_2NO_2 \rightarrow O + CH_2NO$, and a rearrangement channel, $CH_2NO_2 \rightarrow NO + CH_2O$. This example succinctly demonstrates

how the unpaired electron in a radical may dramatically change the dissociation dynamics compared to a similar closed-shell molecule.

Photofragment translational spectroscopy

The technique employed in these studies is photofragment translational spectroscopy (PTS), originally developed by Wilson and coworkers,[5] with significant improvements developed over the last 25 years.[6-8] In this technique, internally cold molecules are dissociated with a photon of known energy $h\nu$, and the photofragments are detected, almost always by ionization. Although many PTS experiments used fixed frequency lasers, allowing only one or a few choices for the photon energy $h\nu$, in general it is wise to explore the dependence of the photofragment yield as a function of photon energy. The photofragment yield (PFY) spectrum obtained such an experiment, depicted schematically in Fig. 2, maps out the position of the electronic states which either predissociate or directly dissociate. In predissociation, the initially excited state is bound, but coupled to another electronic state which is unbound giving rise to narrow

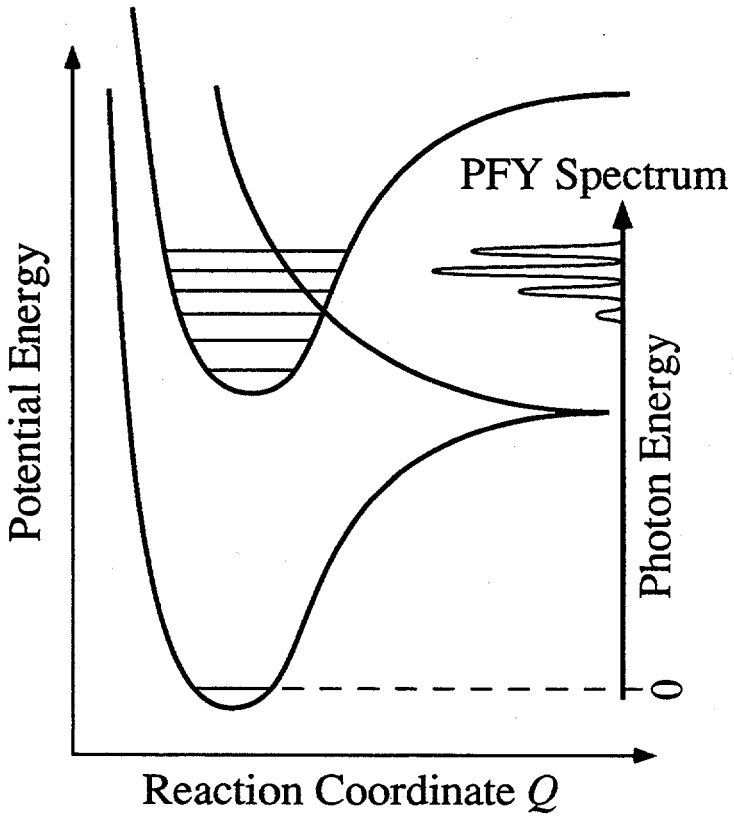


Figure 2: The PFY spectrum is shown on the right as a function of photon energy.

peaks in the PFY spectrum. In contrast, direct dissociation results in a PFY spectrum which has little, if any, structure.

The main thrust of PTS, however, is to measure the translational energy, E_T, of the two separating fragments. The energy balance of this process is given by:

$$E_T = h\nu - D_0 - E_{int}(AB) - E_{int}(CD), \tag{2}$$

where D_0 is the bond dissociation energy, and E_{int} is the sum of the internal energies (vibrational, rotational, electronic) of each fragment. This process is depicted graphically in Fig. 3, with the resulting translational energy distribution, $P(E_T)$, shown in the upper right. The structure in the distribution is due to the quantized vibrational levels of the products, depicted by the harmonic well orthogonal to the reaction coordinate at the asymptote. If, for some energy E_T, it can be determined that the internal energies of the products are zero, Eq. (2) gives D_0 directly. In the present work, the bond strength has been measured for three of the four radicals presented in this dissertation, namely CH_3O, HCCO, and $N_2O_2^-$.

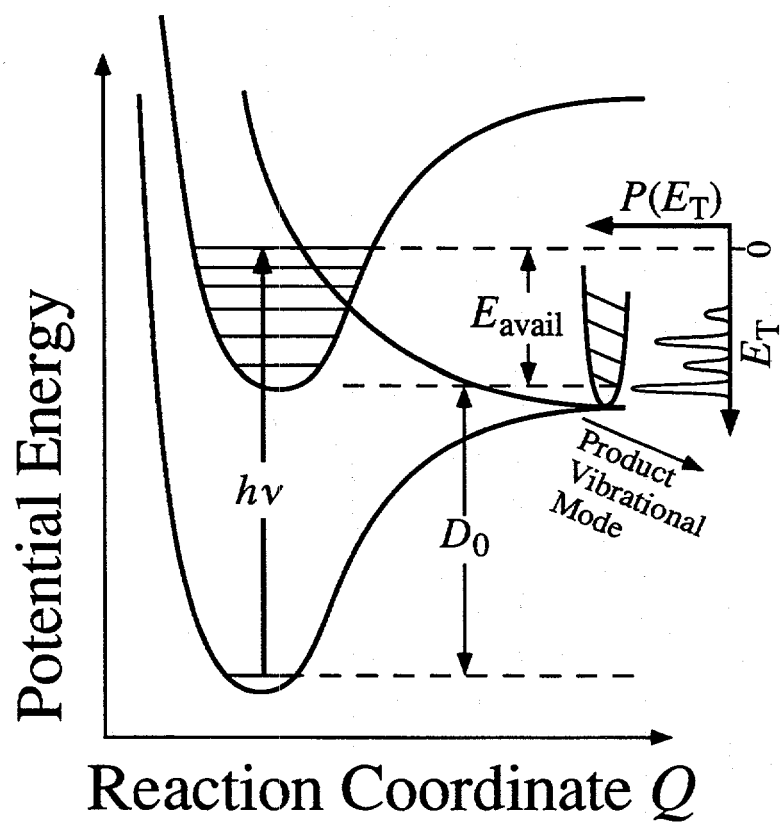


Figure 3: The $P(E_T)$ distribution, shown on the right, with structure due to product vibrational excitation.

In fact, PTS can also measure the angular distribution of fragments with respect to a polarized excitation. To be more precise, the coupled two-dimensional energy and angular distribution of the photofragments, $P(E_T,\theta)$, can be reduced to two one-dimensional distributions by the relation

$$P(E_T,\theta) \propto P(E_T) \bullet \left[1+\beta(E_T)P_2(\cos\theta)\right] \tag{3}$$

where $\beta(E_T)$[9] is the energy dependent anisotropy parameter describing the angular distribution of the fragments. The values of the anisotropy parameter range from $-1 \leq \beta \leq 2$, where the limiting values correspond to a $\sin^2\theta$ or $\cos^2\theta$ distribution of fragments, respectively, and $\beta = 0$ is an isotropic distribution.

The anisotropy parameter provides two types of information. First, a positive value of β indicates a parallel transition in which the recoil axis is parallel to the transition dipole moment, while a negative value indicates a perpendicular transition, with the direction of recoil perpendicular to the transition dipole. Second, the anisotropy parameter is reduced if rotational motion of the excited molecule takes place prior to dissociation, and β is therefore related to the dissociative lifetime. For photodissociation of a diatomic molecule, instantaneous fragmentation leads classically to the limiting values of β (i.e., +2 or -1, depending on whether the transition is parallel or perpendicular), while fragmentation after many rotations of the excited molecule gives the values +0.5 or -0.25, respectively. Intermediate values of β imply a dissociative lifetime on the order of a rotational period. For polyatomic molecules, the dissociative lifetime also affects β, but there may not be a straightforward relationship between the measured value of β and the lifetime, as discussed in Chapter 5 for the HCCO radical.

Previous photodissociation experiments on radicals

As mentioned above, most photodissociation experiments (and most physical chemistry experiments in general) are performed on closed-shell molecules simply because it is convenient to use a sample which is stable in a gas cylinder or a beaker. Nevertheless, free radicals are ubiquitous in chemistry, especially in high-energy or low-density environments such as combustion and atmospheric chemistry, and it is the purpose of this section to explain how photodissociation of such transient species can be investigated.

There are a few free radicals, such as O_2, ClO_2, NO, and NO_2, which are sufficiently chemically stable that traditional photodissociation techniques may be employed to study these systems. However, most free radicals are not amenable to traditional sample handling techniques, and instead must be generated *in situ* and studied before a collision between the radical and another reactive substance can occur. The latter requirement is generally achieved by conducting the entire study in a vacuum chamber of sufficiently low pressure that the collision probability on the timescale of the experiment is much less than unity. In fact, most modern photodissociation experiments on closed-shell species are conducted in such vacuum chambers, so that the only barrier to extending these studies to radicals is that of generating a sufficiently pure, high density sample of these transient molecules.

Generation of radicals by dissociation of a (closed-shell) precursor molecule requires a rather violent input of energy (e.g., laser photolysis, electric discharge, electron impact), often generating a wide range of species including negative and positive ions and a variety of neutral radicals and chemical reaction products. This single fact is responsible

for the paucity of radical dissociation experiments, because the many species created usually obscure the detection of the one radical of interest. Alternatively, the radical may be formed with so much internal energy that it dissociates spontaneously. Nevertheless, several clever experimentalists have succeeded in studying photodissociation of CCl_3,[10] CH_3,[11] HCO,[12,13] CH_3S,[14] NCO,[15] and DNF.[16] In related experiments, fragmentation of other radicals such as CH_3CO and C_2H have been studied as secondary processes following the primary photodissociation of a closed-shell molecule. These latter examples are not well-defined because the radicals are born with a broad distribution of internal energies rather than a single energy as achieved by laser excitation. In any case, the methods of radical preparation in these studies were developed on a case-specific basis, and are not generally transferable to other radicals.

The fast radical beam translational spectrometer

As mentioned earlier, the radical production scheme employed in this research is novel because the radicals are produced from negative ions, and may therefore be mass-selected. The Fast Radical Beam Translational Spectrometer (or FRBM), constructed for these studies by Cyr, Continetti, Metz, and Neumark,[17-18] is shown in Fig. 4. A complete description of this apparatus is given in the Ph.D. dissertation of Douglas Cyr.[19] A

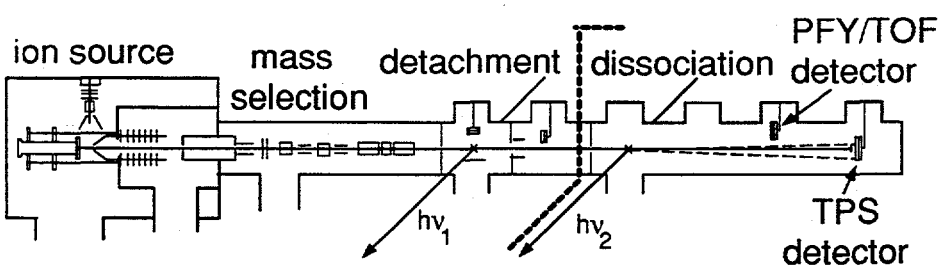


Figure 4: The Fast Radical Beam Translational Spectrometer.

conceptual description of the experimental design is presented here, with information specific to particular experiments given in subsequent chapters. Modifications to the apparatus are discussed in Chapter 2.

Ion production begins with a precursor molecule seeded in a carrier gas (e.g., CH_3OH in 5 atm Ne), and expanded into the source chamber through a pulsed valve operating at 60 - 100 Hz. Ions are generated in the throat of the ensuing supersonic expansion by one of two methods. In the first method, the free jet is crossed with a 1 keV electron beam, creating low energy secondary electrons which produce negative ions. A newer method, using a pulsed electric discharge to create anions, is described in detail in Chapter 4 and has been used for all the alkoxy radicals in this work. In either method, the ions are cooled by the supersonic expansion to their ground vibrational state and to rotational temperatures of 35 - 50 K.

The ion beam passes through a skimmer and is accelerated to high laboratory energy (typically 5 - 9 keV). The ions are then re-referenced to ground by a potential switch[20] discussed in Chapter 2. Ions of the desired mass are selected by a Bakker type time-of-flight mass spectrometer and photodetached with a pulsed dye laser to create the corresponding neutral. As shown graphically in Fig. 5, the energy of the detachment laser is just above threshold, so that only radicals in their zero-point vibrational level are produced. Any residual ions are deflected from the beam, leaving a packet of internally cold, contaminant free neutral radicals moving at a high laboratory velocity (typically 10^7 cm/s). Note that if there is a significant geometry change from anion to neutral, the Franck-Condon factors governing the photodetachment step may give a low cross section

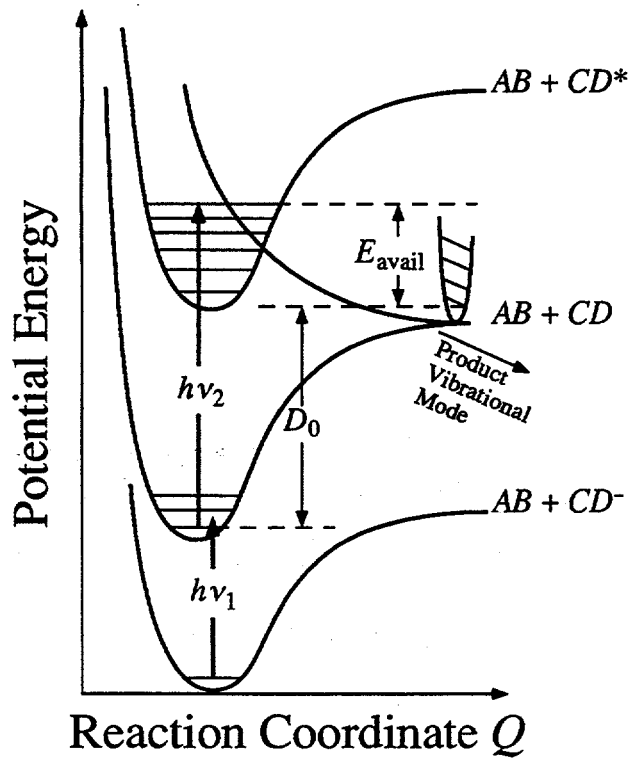


Figure 5: Potential surfaces of the complete system for radical production and dissociation.

for production of $\upsilon = 0$ radicals, making the experiments more difficult on a signal-to-noise basis.[19]

A second linearly polarized pulsed dye laser dissociates the radicals, which are detected using one of two microchannel plate (MCP) detectors. The first MCP detector is of traditional design, measuring the PFY spectrum (or time-of-flight distributions).[21] The second detector, similar to one developed by de Bruijn and Los,[22] is a time- and position-sensitive (TPS) coincidence wedge- and strip-anode detector[18,23] used for dissociation dynamics experiments. A schematic side-view of the TPS detector is shown in Fig. 6. Undissociated radicals are prevented from impinging on the detectors by a beam block across the detector center. Neutral fragments are observed directly (i.e., without an ionization step) due to the high kinetic energy of the particles in the laboratory frame. For fragment kinetic energy > 1 keV, and fragment mass > 12 a.m.u., the detection efficiency is approximately 50% and is independent of mass or chemical composition.[24]

This universal detector is an important aspect of the experimental design allowing many product channels to be simultaneously detected with equal probability. The high detection efficiency is crucial since the density of radicals in the laser interaction region is

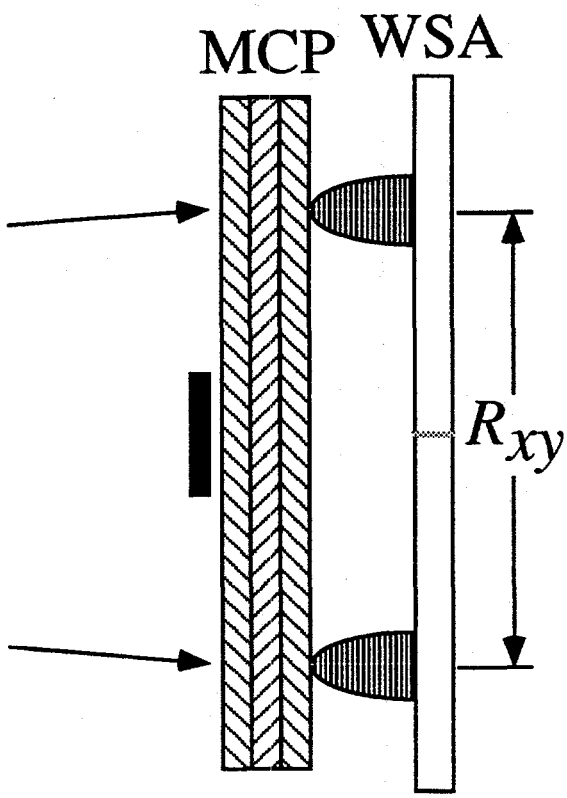


Figure 6: Side view of the TPS detector. Arrows indicate the impact position of each photofragment on the first of three microchannel plates (MCP) arranged in a z-stack configuration. The beam block is shown in solid black. Electrons exit the final MCP and spread before impinging on the anode (WSA). The distance between the centroids of the two electron clouds gives the recoil R_{xy}.

on the order of $10^3/cm^3$. As discussed in Chapter 4, detection of H or D atoms is more difficult than detection of heavier species. An additional advantage of conducting photodissociation experiments in a fast beam is that the laboratory scattering angles of the fragments are very small ($\Theta < \pm 1°$), giving a high collection efficiency. These favorable kinematics are a direct consequence of the fact that the center-of-mass energy of the beam (typically 8,000 eV) is much greater than the relative recoil energy of the two fragments ($\approx D_0 \approx 0$ - 5 eV). As a result, collection efficiencies of 100% are routinely achieved, with a detector of radius 20 mm located 1 - 2 m from the dissociation laser.

Two types of experiments are performed to characterize a given radical dissociation. First, the dissociation laser wavelength is scanned, and the total flux of photofragments arriving at the first MCP detector is monitored. This experiment gives the photofragment yield (PFY) spectrum, mapping out the spectroscopy of the dissociative electronic states. Once the wavelength dependence of process is determined, a second experiment is performed to probe dissociation dynamics using the time- and position-

sensitive MCP detector. In this experiment the position of both photofragments from a single parent radical, and the time difference between their arrival at the detector are measured. The coincidence detection scheme gives significantly better energy resolution (at best $\Delta E_T/E_T = 0.7\%$)[25] than experiments which detect only one of the fragments.[26]

The spatial and temporal signals from the wedge- and strip-anode are amplified and converted to digital values as described in the dissertation of Douglas Cyr.[19] The raw data (charge fraction and time-to-amplitude converter output) is converted to the spatial separation of the two fragments (x, y), and time difference between their arrival (Δt, or alternatively, the z coordinate), by means of proportionality factors whose determination is discussed in Chapter 2. The (x, y, Δt) values can then be analytically converted to the desired experimental quantities, namely the fragment masses, their relative translational energy, and their scattering angle with respect to the $\vec{E}$ vector of the dissociation laser. The kinematics of the fast beam PTS process shown in Fig. 7 are described by the equations:

$$m_{AB} = M \bullet R_{CD} \frac{\left(1 - \frac{v_0 \Delta t}{L}\right)}{\left\{R_{xy} - \left(\frac{v_0 \Delta t}{L}\right) R_{CD}\right\}} \tag{4}$$

$$E_T = \left(\frac{E_0}{L^2}\right)\left(\frac{m_{AB} m_{CD}}{M^2}\right)\left[(v_0 \Delta t)^2 + R_{xy}{}^2\right]\left(1 + 2\frac{m_{CD} - m_{AB}}{M}\frac{v_0 \Delta t}{L}\right) \tag{5}$$

$$\theta = \arctan\left(\frac{R_{xy}}{v_0 \Delta t}\right) \tag{6}$$

in which m_{AB}, m_{CD}, and M are the masses of the two fragments and the parent radical, respectively, $R_{xy} = \sqrt{x^2 + y^2}$ is the distance between the two fragment impacts on the

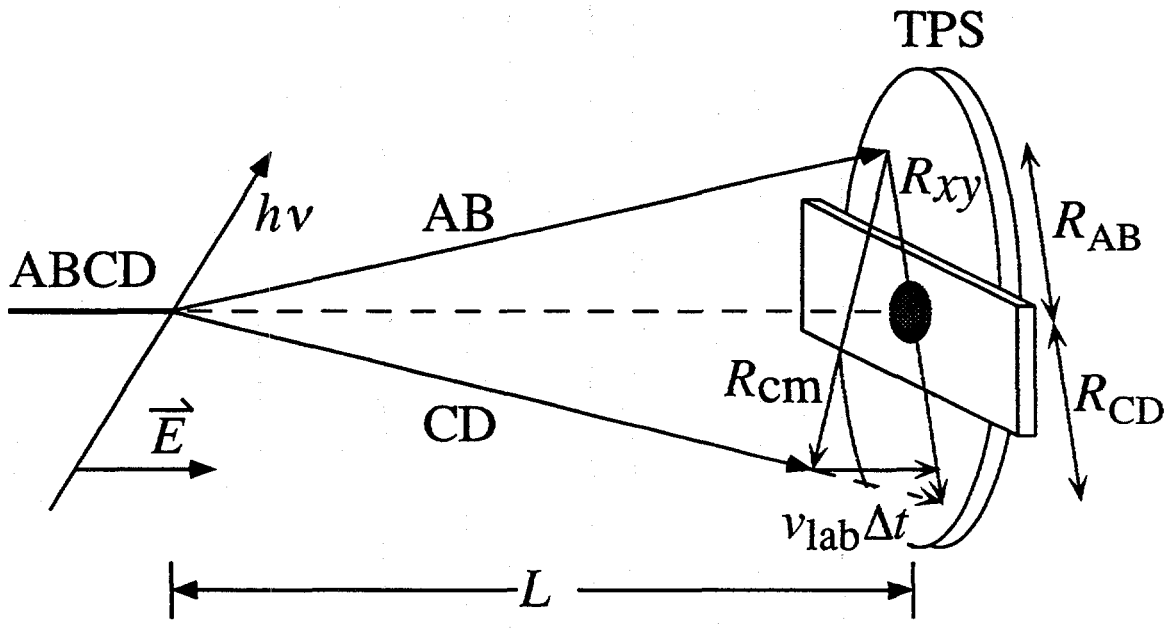


Figure 7: Fast beam kinematics of photodissociation using the time- and position-sensitive detector (TPS). The three dimensional recoil vector is given by $R_{CM} = \sqrt{R_{xy}^2 + (v_{lab}\Delta t)^2}$.

detector face, R_{CD} is the distance from the radical beam to fragment CD on the detector face, v_0 and E_0 are the velocity and kinetic energy, respectively, of the parent radical beam, L is the flight length between laser and detector, θ is the scattering angle, and E_T is the center-of-mass relative translational energy of the two fragments. The FORTRAN codes that perform this conversion are described in Appendix A.

Comparison of fast beam translational spectroscopy with other techniques

For the sake of perspective, it is worth comparing the fast beam implementation of PTS in this work with other common techniques for studying photodissociation. State-resolved fragment detection experiments, in which single rovibrational states of one product are detected, can give the rotational distribution, and with more effort the vibrational distribution, of one fragment.[4] No information on the translational energy distribution or the rovibronic distribution of the other product is obtained, and therefore

these experiments are often conducted close to the photodissociation threshold, where the other product is constrained to have little energy. A closely related technique, photofragment excitation spectroscopy (PHOFEX),[2] monitors the population dependence of a single rovibrational product state on the energy of the dissociation laser. High-resolution Doppler spectroscopy[27] goes one step further, using a narrow-bandwidth probe laser to resolve the Doppler profile of one photofragment, thereby obtaining some information on both the translational energy and angular distribution, and consequently a measure of the amount of energy deposited in the undetected fragment. In addition, information on the vector correlation of the recoil velocity with the rotation of the detected fragment can be obtained, lending considerable insight into the dissociation mechanism. Similarly, photofragment imaging[28] experiments give the spatial image of one state-specifically ionized fragment, yielding a low-resolution measurement of the translational energy and angular distribution for a specific rovibronic state of this product.

All these techniques measure a single rovibronic state of one fragment (via laser induced fluorescence or resonance enhanced multi-photon ionization), enabling one to construct quite detailed information on one or a few of the many possible final product states. One of the drawbacks to these methods is that the detection schemes are specific to each fragment (e.g., NO, CO, I, etc.) and must be revamped each time one wishes to detect a different product. In addition, only a very small fraction (i.e., a single rovibronic level) of the total dissociation products are detected at any one time. In contrast, PTS, as implemented here, is inherently multiplexed, probing all final states, at all scattering angles, of all product channels simultaneously, although resolution of individual product rotational states is not possible. In general, most other techniques take a high-resolution view of a

small part of the dissociation dynamics, while PTS takes a lower-resolution view of the entire picture. Both approaches clearly have their advantages, and together are quite complementary. However, the fast beam PTS approach is certainly a more general technique for radical photodissociation because of its high sensitivity (essential for experiments at densities of $10^3/cm^{-3}$) and universal detection capabilities.

Interpretation of Results

To summarize, all the information from the Fast Radical Beam Translational Spectrometer is given in three types of spectra.

- The spectroscopy of the dissociative excited states is revealed in the photofragment yield (PFY) spectrum.
- The disposal of energy in the products (for each product channel) is contained in the translational energy distribution, $P(E_T)$.
- The angular distribution for each product channel is given by the anisotropy parameter distribution, $\beta(E_T)$.

The remaining task is to rationalize these results and develop a dissociation mechanism in a consistent manner by comparison with semi-empirical models, which have the advantage of being easy to implement, or better yet, with classical or quantum mechanical reaction dynamics calculations based on *ab initio* potential energy surfaces.

Today, the field of photodissociation dynamics has progressed to a point where accurate, full-dimensional *ab initio* surfaces are available for several triatomic molecules, and the dynamics calculations based on these surfaces can be compared with detailed experimental results. As advances in accurate PES calculations for four, five, and six

atom molecules continue, a similar interplay between experiment and theory promises great insight into these larger systems. At the same time, the experimental approach utilized in the work presented here offers a general technique for the study of free radical dissociation. The availability of these experimental results has already spawned further calculations of the many electronic surfaces responsible for CH_3O photodissociation, and one hopes that these results will be interesting and useful to other scientists trying to unravel the chemistry of free radicals.

References

[1] R. D. Levine and R. B. Bernstein, *Molecular Reaction Dynamics and Chemical Reactivity* (Oxford, New York, 1987).

[2] C. B. Moore and I. W. M. Smith, J. Phys. Chem. **100**, 12848 (1996).

[3] R. Schinke, *Photodissociation Dynamics* (Cambridge University Press, Cambridge, 1993).

[4] L. J. Butler and D. M. Neumark, J. Phys. Chem. **100**, 12801 (1996).

[5] G. E. Busch, J. F. Cornelius, R. T. Mahoney, R. I. Morse, D. W. Schlosser, and K. R. Wilson, Rev. Sci. Instrum. **41**, 1066 (1970); G. E. Busch and K. R. Wilson, J. Chem. Phys. **56**, 3626 (1972).

[6] A. M. Wodtke and Y. T. Lee, *Molecular Photodissociation Dynamics*, M. N. R. Ashfold and J. E. Baggot, Eds. (Royal Society of Chemistry, London, 1987), p. 31.

[7] M. N. R. Ashfold, I. R. Lambert, D. H. Mordaunt, G. P. Morley, and C. M. Western, J. Phys. Chem. **96**, 2938 (1992).

[8] M. N. R. Ashfold, D. H. Mordaunt, and S. H. S. Wilson, Adv. Photochem. **21**, 217 (1995).

[9] R. N. Zare, Mol. Photochem. **4**, 1 (1972).

[10] E. J. Hintsa, X. Zhao, W. M. Jackson, W. B. Miller, A. M. Wodtke, and Y. T. Lee, J. Phys. Chem. **95**, 2799 (1991).

[11] S. W. North, D. A. Blank, P. M. Chu, and Y. T. Lee, J. Chem. Phys. **102**, 792 (1995).

[12] D. W. Neyer, X. Luo, P. H. Houston, and I. Burak, J. Chem. Phys. **98**, 5095 (1993), and references therein.

[13] J. D. Tobiason, J. R. Dunlop, and E. A. Rohlfing, J. Chem. Phys. **103**, 1448 (1995).

[14] C. W. Hsu, D. L. Liao, Z. X. Ma, P. J. H. Tjossem, and C. Y. Ng, J. Chem. Phys. **97**, 6283 (1992).

[15] X. Liu and R. D. Coombe, J. Chem. Phys. **91**, 7534 (1989).

[16] J. Chen and P. J. Dagdigian, J. Chem. Phys. **98**, 3554 (1993).

[17] R. E. Continetti, D. R. Cyr, R. B. Metz, and D. M. Neumark, Chem. Phys. Lett. **182**, 406 (1991).

[18] R. E. Continetti, D. R. Cyr, D. L. Osborn, D. J. Leahy, and D. M. Neumark, J. Chem. Phys. **99**, 2616 (1993).

[19] D. R. Cyr, Ph.D. Dissertation, University of California, Berkeley, 1993.

[20] R. E. Continetti, D. R. Cyr, and D. M. Neumark, Rev. Sci. Instrum. **63**, 1840 (1992).

[21] D. R. Cyr, R. E. Continetti, R. B. Metz, D. L. Osborn, and D. M. Neumark, J. Chem. Phys. **97**, 4937 (1992).

[22] D. P. de Bruijn and J. Los, Rev. Sci. Instrum. **53**, 1020 (1982).

[23] C. Martin, P. Jelinsky, M. Lampton, R. F. Malina, and H. O. Anger, Rev. Sci. Instrum. **52**, 1067 (1981).

[24] H. Kersten, *Microchannel Plate Report,* (FOM-Institute for Atomic and Molecular Physics, Amsterdam, 1987)

[25] D. J. Leahy, D. L. Osborn, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **103**, 2495 (1995).

[26] The exception to this statement is the case of H-atom photofragment translational spectroscopy, in which only the H atom is detected, but the energy resolution is exceptional (on the order of 2 - 3 cm^{-1}) due to favorable kinematics.

[27] J. P. Simons, J. Phys. Chem. **91**, 5378 (1987); P. L. Houston, J. Phys. Chem. **91**, 5388 (1987).

[28] P. L. Houston, J. Phys. Chem. **100**, 12757 (1996).

2 Improvements to the experiment and analysis[a]

This chapter discusses improvements to the Fast Radical Beam Photofragment Translational Spectrometer in both experimental design and data analysis. Three major improvements have been accomplished beyond the description of the experiment given in the Ph.D. dissertation of Douglas Cyr.[1] First, the pulsed discharge source for ion generation has been substantially improved, and is described in detail in Chapter 6. Second, the high-voltage switch used for re-referencing the ions to ground potential has been completely rebuilt and is briefly discussed in this chapter. Finally, the procedure for obtaining the proportionality factors which convert charge-fraction and timing information (from the time- and position-sensitive detector) into x, y, and z spatial information is also presented here.

I. The high voltage switch (a. k. a. the death pulser)

In order to avoid floating either the source region or the detection region of the apparatus at the ion beam voltage (typically V_{beam} = 5 - 9 keV), a fast MOSFET potential switch is used to re-reference the ion beam to ground potential. The principle of the switch is straightforward. After the ions are accelerated to a potential V_{beam}, they pass through a 3 mm aperture into a stainless steel cylinder, 25 cm long by 8 cm in diameter, held initially at $V_{cyl} = V_{beam}$. While the ions are inside the cylinder, the potential V_{cyl} is

[a] Portions of this chapter are published in the Journal of Chemical Physics **103**, 2495 (1995).

dropped rapidly (≈ 250 ns) to ground. Because there can be no electric field inside a conductor, no force is exerted on the ions inside the cylinder during the switching time. These ions exit the cylinder through a 3 mm hole with a velocity equal to $\sqrt{\frac{2V_{\text{beam}}}{m}}$, but referenced to the final potential of V_{cyl}, which is nominally within a few volts of ground.

The design and operation of this home-built switch was presented by Continetti, Cyr, and Neumark.[2] The original switch consisted of ten 1000 V varistor-bypassed MTP1N100 MOSFET stages in series driven by two Siliconix D469CJ quad MOSFET drivers and was capable of switching up to 8 keV. Although generally stable, certain components failed periodically, necessitating replacement. The layout of the circuit, in which all ten identical stages were individually screwed into the chassis, linked with soldered electrical connections, made troubleshooting and repair a tedious job at best. In addition, the upper limit on V_{beam} imposed by this switch was around 8 keV. To circumvent both of these limitations, a new switch was constructed based on the same electrical design, but using a modular approach (to facilitate quick identification and repair of faulty components) with sixteen 1000 V MOSFET stages (to allow operation beyond well beyond $V_{\text{beam}} = 10$ keV).

Two improvements were made in the electrical components. First, the MOSFETs were replaced with Harris RFP4N100 N-Channel 1000 V MOSFETs which can switch more current (4.3 A) than the previous transistors. Second, four Teledyne TC4422CPA MOSFET drivers replace the two D469CJ drivers. The TC4422CPAs drive four instead of five stages each, and operate with an "on" voltage of 15 V compared to 12 V, allowing

a more rapid accumulation of the requisite 120 nC of gate charge required to turn on each MOSFET.

Realizing that failure of certain components on the high-voltage stages is inevitable, a modular design was developed in which each stage is constructed on a printed circuit board, as shown in Fig. 1, which is connected via a PC card slot to the rest of the switch. Not only does this design ease the task of troubleshooting, it also allows rapid replacement of a defective state with a spare stage so that data collection may be continued with minimal interruption.

It should be noted that in testing the performance of the switch, the calibration of the high voltage oscilloscope probe used (normally a Tektronix P6015) can affect the appearance of the oscilloscope trace. Under certain conditions the switch may not drop V_{cs} completely to ground. While this may indicate a defective stage, it may also be an artifact of the probe, and care must be exercised to obtain a reliable diagnostic.

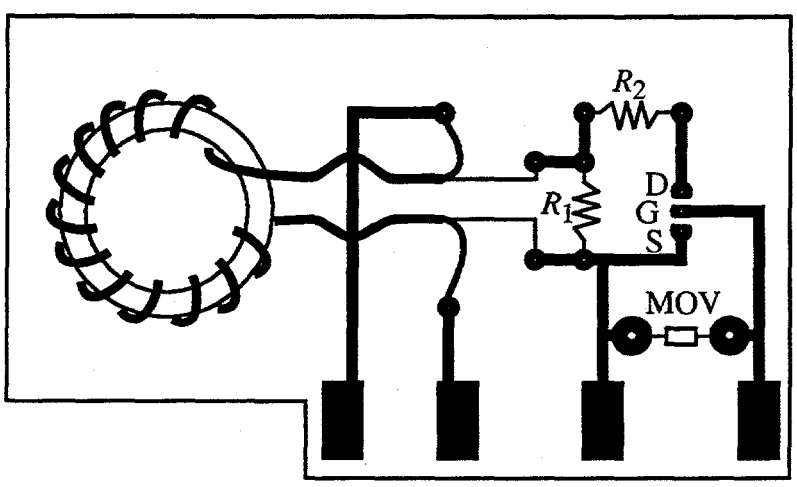


Figure 1: MOSFET card for the high voltage switch. The MOSFET is denoted by (D, G, S), and bypass varistor by MOV. $R_1 = 50\ \Omega$, $R_2 = 250\ \Omega$.

II. Determination of detector factors

As mentioned in the introduction, several proportionality factors are required to convert the experimentally measured charge-fraction and Δt values from the time- and position-sensitive detector into the three physically useful observables we wish to obtain: the mass ratio of the two photofragments $\frac{m_1}{m_2}$, the relative translational energy E_T, and the angle θ between the center-of-mass recoil vector and $\vec{E}$ of the linearly polarized dissociation laser. The optimization of these proportionality constants, or "factors," by least-squares fitting to data from the dissociation $O_2 \rightarrow O + O$ was a major advance in our time- and position-sensitive coincidence detection scheme. Prior to this development, we derived factors by covering the microchannel plates with a mask of regularly spaced pinholes, illuminating the mask with ultra-violet radiation to stimulate electron emission, and adjusting the factors by hand until the best correspondence between the observed and true pattern was obtained. Upon joining the experiment, David Leahy recognized that the energy resolution of the experiment at that time (60 meV) was far worse than what one calculates from the 100 μm spatial and 500 ps time resolution of the detector. It is due to his aptitude and perseverance through many months of attacking this problem that we have attained a resolution of 6.8 meV, or $\frac{\Delta E_T}{E_T} = 0.6$ %. I cannot overstate the importance of this increased resolution to the experiments on the CH_3O radical presented in Chapter 3, and I give David Leahy the credit for this significant advance in our detection scheme.

For each observed coincidence event, the time- and position-sensitive detector records the impact positions of both fragments with an accuracy of about 100 μm, while a time-to-amplitude converter (TAC) simultaneously records the time interval between the arrival of the two fragments to an accuracy of 500 ps. Together, these measurements yield the center-of-mass translational energy release and the recoil angle of each photodissociation event. For fragments of equal mass (as in the case of O_2), these relationships are given by

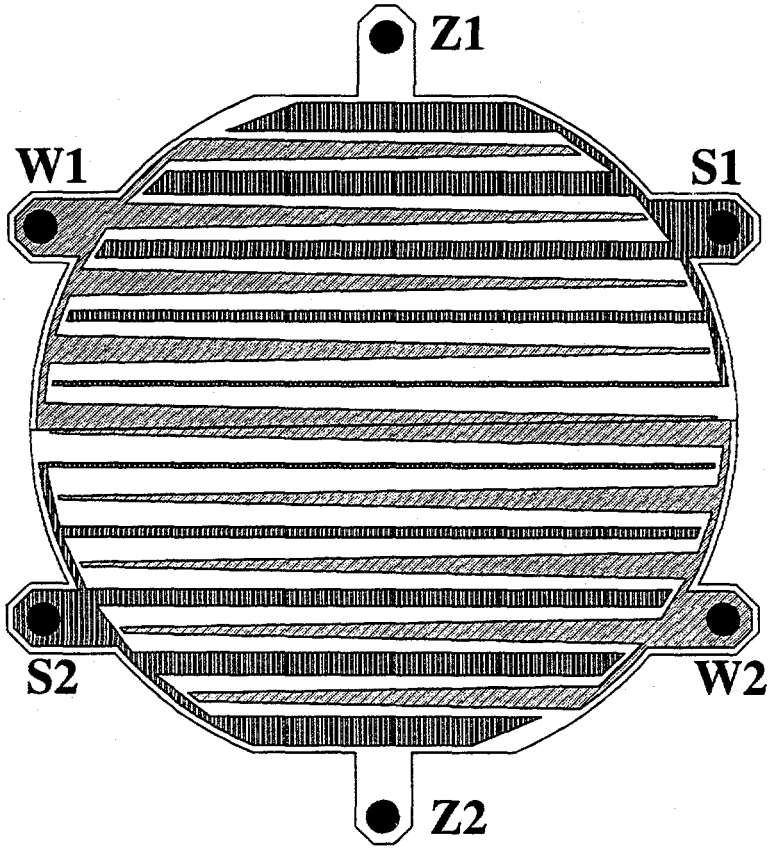


Figure 2: The wedge-and-strip anode of the time- and position-sensitive detector. The wedge, strip, and zig-zag conductors are labeled for both the upper half (1) and lower half (2) of the anode

$$E_T = \frac{1}{2} E_{\text{beam}} \frac{R_{xy}^2 + \left(v_{\text{beam}} \Delta t\right)^2}{L^2} \tag{1}$$

and

$$\theta = \arctan\left(\frac{R_{xy}}{v_{\text{beam}} \Delta t}\right), \tag{2}$$

where E_{beam} and v_{beam} are the laboratory energy and velocity of the O_2 beam, respectively; R_{xy} is the observed recoil distance perpendicular to the beam axis (as measured by the position-sensitive detector); Δt is the time interval between the arrivals of the two photofragments at the detector; and L is length of the drift region.

The fragment impact positions are observed using of a pair of wedge-and-strip anodes that collect the charge from the detector's microchannel plates.[3,4] A schematic of the wedge- and strip anode pattern is shown in Fig. 1; the actual pattern is considerably denser. This pattern consists of two separate wedge-and-strip anodes (one upper and one lower, which we will label 1 and 2, respectively), comprised of three conductors each ("wedge", "strip" and "zigzag"). These anodes divide the ~10^7 electrons from the microchannel plates between the three conductors in a spatially specific manner. The wedge conductor tapers in the horizontal direction while the strip conductor changes in width along the vertical direction, so that the two electrodes determine the horizontal and vertical position, respectively, of the centroid of the electron cloud. The following equations relate these charge fractions to the impact positions:

$$x_1 = a_{1x}F_{\text{wedge}} + b_{1x} \tag{3a}$$

$$y_1 = a_{1y}F_{\text{strip}} + b_{1y} + c_{1y}F_{\text{strip}}^2 \tag{3b}$$

$$x_2 = a_{2x}F_{\text{wedge}} + b_{2x} \tag{3c}$$

$$y_2 = a_{2y}F_{\text{strip}} + b_{2y} + c_{2y}F_{\text{strip}}^2 \tag{3d}$$

The Cartesian coordinates x and y are the horizontal and vertical positions, respectively, of the fragment impacts relative to the origin, which we define to lie at the center of the radical beam. The multiplicative constants a_k and additive constants b_k define

the nearly linear relationship between the anode charge fractions F and the Cartesian coordinates. In addition, we have included a small quadratic term in the y coordinate equations to model an inherent nonlinearity in the detector's charge division. Equations (3a - d) are different and somewhat simpler than those that appear in Ref. 3; for our charge amplification scheme, the cross-talk corrections have proven to be of negligible importance. The crux of the data analysis is the determination of the values of the parameters a_k, b_k and c_k. However, in addition to these ten parameters, four additional parameters are used to obtain maximum energy resolution.

Ideally, the normal to the detector face lies coincident with the main axis of the FRBM, such that the distances from the photodissociation region to all points in a given annulus of the detector face are equal. In practice the detector is aligned by reflecting a He-Ne laser beam off the beam-block of the detector and aligning the back reflection with the main laser beam. To correct for the inevitable slight inaccuracy in this procedure, two additional factors, *tilt-x* and *tilt-y*, are used to account for the rotated position of the detector about the two axes orthogonal to the radical beam axis. The final two factors are *TAC-m* and *TAC-b*, the slope and intercept respectively of the linear relationship between the TAC output and the true Δt value for the dissociation.

While rough guesses for a_k and b_k can be made based on the anode geometry, in a quantitative analysis the fourteen constants must be determined from the O_2 dissociation data directly. We accomplish this by treating the constants as parameters in a non-linear least squares fit to the oxygen data, with the merit function

$$\chi^2 = \sum_i \frac{(R_{\text{calc}} - R_{\text{obs},i})^2}{\sigma^2_{\text{recoil}}} + \frac{R^2_{\text{centroid},i}}{\sigma^2_{\text{centroid}}}, \tag{4}$$

where

$$R_{\text{obs},i} = [(x_1^i - x_2^i)^2 + (y_1^i - y_2^i)^2 + (v_{\text{beam}}\Delta t_i)^2]^{1/2} \tag{5}$$

and

$$R_{\text{centroid},i} = \{[(x_1^i + x_2^i)/2]^2 + [(y_1^i + y_2^i)/2]^2\}^{1/2}. \tag{6}$$

The index i stands for the independent coincident events. R_{calc} is the calculated recoil distance given by inverting Eq. 1, using the known oxygen bond strength (5.117 eV, Ref. 5), the photon energy, the parent oxygen beam velocity v_{beam}, and the length of the fragment drift region. R_{calc} is first determined using the kinetic energy release corresponding to the ($j_1 = 2$, $j_2 = 1$) final state; which was the most probable final state from predissociation of the $\upsilon' = 7$ level.[6] In subsequent iterations, the mean kinetic energy release for each data set is used. R_{obs} is the length of the observed three dimensional recoil vector between the two fragments, which depends on the adjustable parameters a_k, b_k and c_k. Similarly, R_{centroid} is the parameter-dependent distance from the center of the radical beam to the parent radical impact (as inferred from the fragment impacts, using conservation of momentum). The constants σ_{recoil} and σ_{centroid} characterize the standard deviations in the recoil measurement (150 μm) and centroid position (1.6 mm, i.e., the radical beam width).

The non-linear least squares fit utilizes a Levenberg-Marquardt algorithm.[7] The fit is provided with data sets of a few tens of thousands coincident events in the form of the raw charge fractions of Eqs. (3a-d). Following an initial estimate, the adjustable

parameters a_k, b_k, c_k, and the *tilt* and *TAC* parameters are optimized by the algorithm to minimize χ^2. This results in a set of parameters that yield the high translational energy resolution necessary to resolve the correlated spin-orbit distributions in O_2 dissociation[8] and the vibrational product state of the CH_3 radical in CH_3O dissociation presented in Chapter 3 and Ref. 9. In general, the oxygen data fits provide an excellent absolute calibration for the position sensitive detector, which is an essential first step for all our photodissociation studies of polyatomic free radicals.

The FORTRAN code XZYLSQ, which accomplishes the fitting procedure described above, is given in Appendix D, along with practical information needed to implement the program.

References

[1] D. R. Cyr, Ph.D. Dissertation, University of California, Berkeley, 1993.

[2] R. E. Continetti, D. R. Cyr, and D. M. Neumark, Rev. Sci. Instrum. **63**, 1840 (1992).

[3] R.E. Continetti, D.R. Cyr, D.L. Osborn, D.J. Leahy, and D.M. Neumark, J. Chem. Phys. **99**, 2616 (1993).

[4] O.H.W. Siegmund, M. Lampton, J. Bixler, S. Bowyer, and R.F. Malina, IEEE Trans. Nucl. Sci. **NS-33**, 724 (1986); J.V. Vallerga, G.C. Kaplan, O.H.W. Siegmund, M. Lampton, and R.F. Malina, *ibid.* **NS-36**, 881 (1989).

[5] P.C. Cosby and D.L. Huestis, J. Chem. Phys. **97**, 6109 (1992).

[6] D.J. Leahy, D.R. Cyr, D.L. Osborn, and D.M. Neumark, Chem. Phys. Lett. **216**, 503 (1993).

[7] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vettering, Numerical Recipes (Cambridge University, Cambridge, England, 1986), pp. 523-528.

[8] D. J. Leahy, D. L. Osborn, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **103**, 2495 (1995).

[9] D. L. Osborn, D. J. Leahy, E. M. Ross, and D. M. Neumark, Chem. Phys. Lett. **235**, 484 (1995).

3 Photodissociation spectroscopy and dynamics of the $\tilde{A}$ ($^2A_1$) electronic state of CH_3O and CD_3O.

I. Introduction

The methoxy radical (CH_3O) is a remarkable molecule, and has become one of the most widely studied polyatomic free radicals since it was first observed in emission of ethyl nitrate photolysis in 1953.[1] Methoxy has drawn such intense scrutiny because it serves as a paradigm for a surprisingly large number of fundamental physical and chemical interactions. It is important in the atmosphere due to its participation in smog chemistry[2] and is an intermediate in the combustion of cool flames of methane,[3] oxygenated fuels containing CH_3OH, and fuel-lean ignition processes.[4] Although not yet detected in interstellar gas, it is quite likely that CH_3O exists in this medium because its close relatives CH_3OH and CH_2O are found in abundance.[5]

In the field of photodissociation dynamics, the rich photochemistry of methoxy provides archetypal cases of two distinct dissociation mechanisms. On the ground state surface, dissociation to H + CH_2O is impeded by an activation barrier. Stimulated emission pumping (SEP) experiments show narrow resonances both below and above the barrier, with important implications to the long-standing question of mode-specific vs. statistical unimolecular decomposition.[6] In a preliminary report,[7] we presented the first unambiguous evidence for predissociation of CH_3O following excitation of the ultraviolet $\tilde{A}\,(^2A_1) \leftarrow \tilde{X}\,(^2E)$ transition. In contrast with barrier impeded ground state dissociation,

ultraviolet dissociation of methoxy, yielding CH_3 + O, provides a textbook example of predissociation via coupling to purely repulsive excited state potential energy surfaces. In this paper we give a full exposition covering our experiments on the photodissociation dynamics of CH_3O and CD_3O.

Although the literature on CH_3O is quite extensive, the most germane works, from the viewpoint of photodissociation dynamics, are the spectroscopic[6-31] and theoretical investigations[32-40] of the rotational, vibrational, and electronic structure of methoxy. Within the C_{3v} point group, methoxy's ground state is of E symmetry due to a two-fold electronic degeneracy. Consequently, CH_3O is subject to vibronic coupling via the Jahn-Teller effect[41,42] in which nuclear motion along one (or more) of the e symmetry vibrations splits the electronic degeneracy. In addition, the unpaired electron spin can couple to the magnetic field generated by the electronic orbital motion around the symmetry axis. As a result, methoxy is a prototypical example of the complications that arise when both Jahn-Teller and spin-orbit effects must be treated together.[43]

The spectroscopy of CH_3O is far from straightforward, with a substantial amount of controversy surrounding the proper values for the spectroscopic parameters. As noted by Foster *et al.*,[23] there has been a great deal of confusion in determining the true location of the 0_0^0 band in the $\tilde{A}\,(^2A_1) \leftarrow \tilde{X}\,(^2E)$ transition. The confusion stems in part from the fact that all three non-totally symmetric (e) vibrations (ν_4, ν_5, ν_6) are active in the $\tilde{A}$ state, in addition to the three totally symmetric (a_1) vibrations (ν_1, ν_2, ν_3). The resulting spectra are rather complex, even at the very low rotational and vibrational temperature achieved in a supersonic jet expansion. Extensive vibrational[30] and rotational[25,30] analyses of the $\tilde{A} \leftarrow \tilde{X}$ LIF spectrum give the accepted value of the rovibronic band origin

ν_0 = 31,614.51 cm^{-1}. The vibrational frequencies for several modes have proven even more difficult to evaluate, although two recent investigations appear to have settled all the major discrepancies.[30,31] In addition, several incongruous determinations of the fluorescence lifetime of the $\tilde{A}$ state have been reported. Three investigations found τ_{FL} = 2.1 μs for 3_0^n (n = 0, 1, 2),[17-19] a fourth reported τ_{FL} = 1.5 μs for n = 0-6,[12] while another investigation found a variation in lifetime as a function of n, with values ranging from $\tau_{FL}(3_0^0)$ = 2.23 μs to $\tau_{FL}(3_0^7)$ = 0.89 μs.[22] Recent measurements in a supersonic jet give $\tau_{FL}(3_0^6)$ = 0.35 - 0.38 and $\tau_{FL}(3_0^7)$ < 0.02 μs.[44]

However, all experimental and theoretical investigations agree that the main progression in the $\tilde{A} \leftarrow \tilde{X}$ transition is ν_3, the C–O stretch. The extended Franck-Condon progression in this mode arises from the large change in C–O bond length between ground and excited states (r_{CO}' = 1.58 Å $\leftarrow$ r_{CO}'' = 1.37 Å).[25] The increase in r_{CO} results from excitation of a CO σ-bonding electron to a non-bonding p_π orbital localized on the oxygen atom, leaving essentially half a CO bond in the $\tilde{A}$ ($^2A_1$) state.

Among the theoretical studies of methoxy's electronic structure, the pioneering work of Jackels was the first systematic investigation of the many excited electronic states of CH_3O.[33] In addition to the bound $\tilde{X}$ (^2E) and $\tilde{A}$ ($^2A_1$) states, he reported three electronic states of $^2A_2$, $^4A_2$, and ^4E symmetry which are purely repulsive along the C–O coordinate, shown schematically in Fig. 1. The curve crossings between the lowest of these states and the $\tilde{A}$ state was predicted to lie at 36,500 cm^{-1}, slightly above the energy of the 3_0^7 transition. More detailed *ab initio* investigations, including the coupling

strengths between different excited electronic states, have recently been undertaken by Pederson and Yarkony,[39] and by Cui and Morokuma.[40]

There are at least four photodissociation product channels open to methoxy for the excitation energies used in this study:

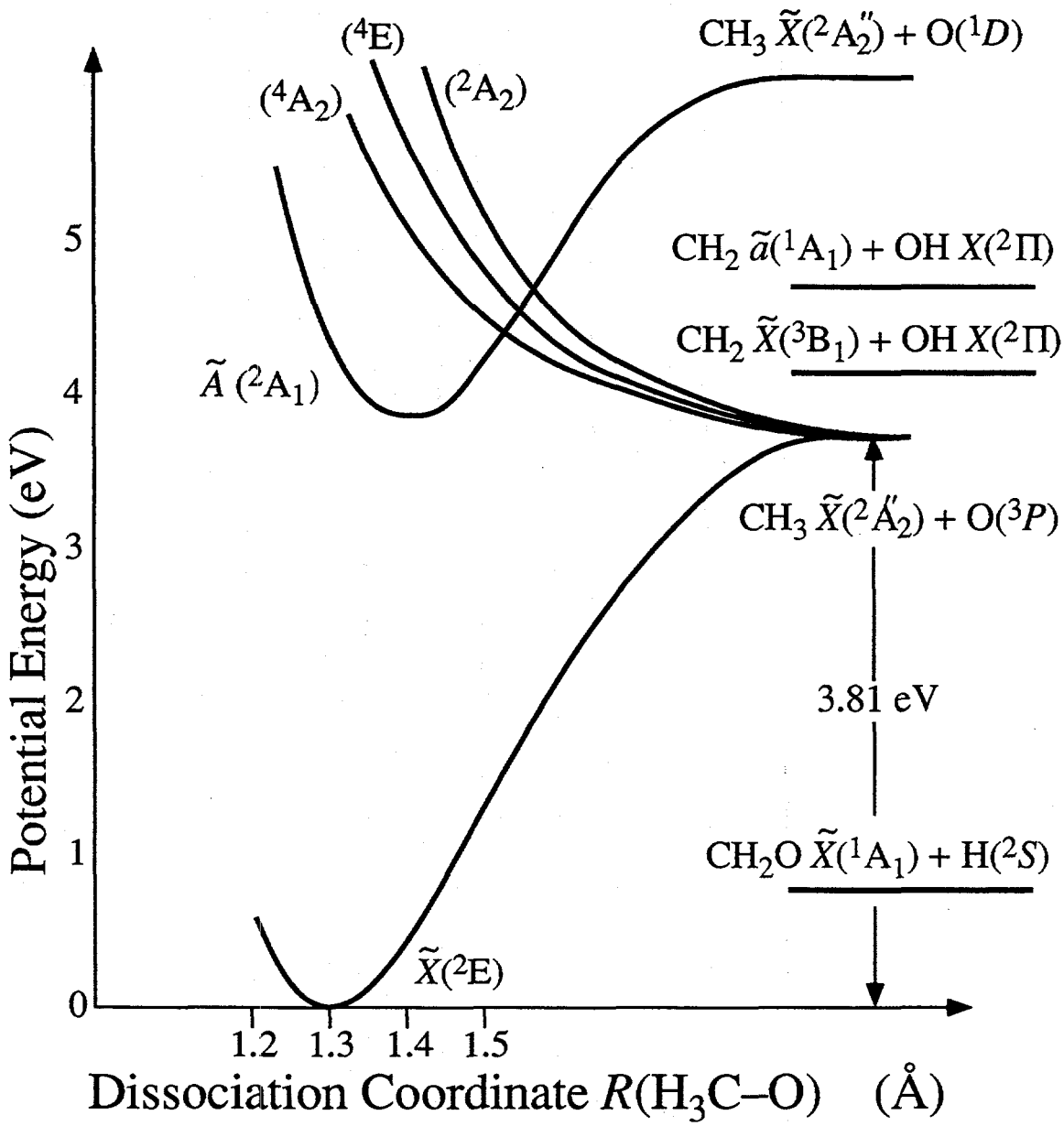


Figure 1: Schematic C_{3v} potential energy surfaces for the methoxy radical as a function of CO bond length.

$$CH_3O\,\tilde{X}\,(^2E) \xrightarrow{h\nu} CH_3O\,\tilde{A}\,(^2A_1) \rightarrow$$

$$CH_3\,\tilde{X}\,(^2A_2'') + O\,(^3P) \qquad \Delta H_{rxn,0} = 3.807 \pm 0.013\ \text{eV} \qquad (I)$$

$$CH_2\,\tilde{X}\,(^3B_1) + OH\,X\,(^2\Pi) \qquad \Delta H_{rxn,0} = 4.10 \pm 0.05\ \text{eV} \qquad (II)$$

$$CH_2\,\tilde{a}\,(^1A_1) + OH\,X\,(^2\Pi) \qquad \Delta H_{rxn,0} = 4.49 \pm 0.05\ \text{eV} \qquad (III)$$

$$CH_2O\,\tilde{X}\,(^1A_1) + H\,(^2S) \qquad \Delta H_{rxn,0} = 0.78 \pm 0.07\ \text{eV} \qquad (IV)$$

The energetics are based on the heat of formation of methoxy $\Delta H^{\circ}_{f,0}$ (CH_3O) = 0.29 ± 0.02 eV (6.8 ± 0.4 kcal/mol) found in our preliminary investigation,[7] and the literature values for other species.[45] Channel (*I*) is the major product channel for CH_3O at all photon energies. The formation of CH_2 + OH via channels (*II*) and/or (*III*) is observed as a minor product channel in CH_3O photodissociation. The corresponding channels are also observed with CD_3O. As discussed elsewhere,[46] detection of H atom loss is difficult though not impossible with our apparatus. For methoxy we have evidence that channel (*IV*) plays a very minor role in methoxy dissociation at high photon energies.

Our experimental results for both isotopes are presented in section III, and analyzed in section IV. In section V we discuss the dissociation dynamics of methoxy in light of the experimental distributions and comment on the possibilities for further investigations.

II. Experimental

The experimental apparatus used in these studies, the fast beam photofragment translational spectrometer (Fig. 2), has been described in detail elsewhere,[47,48] and only a brief description is given here. In order to produce a well-characterized sample of free radicals, we rely on the fact that open-shell species have positive electron affinities and form stable negative ions. The neutral free radical of interest can then be generated by laser photodetachment of the mass-selected negative ion precursor, and subsequently photodissociated by a second laser:

$$CH_3O^- \xrightarrow{h\nu_1} CH_3O + e^- \xrightarrow{h\nu_2} \text{fragments} \tag{1}$$

Consequently, the apparatus in Fig. 2 is divided into two main regions. In the first region

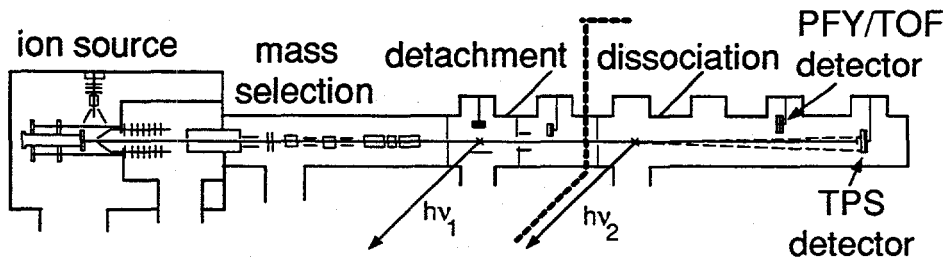


Figure 2: The fast radical beam translational spectrometer. The dotted line separates the radical production section on the left from the photodissociation section on the right. The flight distance L between the dissociation laser and a detector is 0.68 m for the TOF detector, and either 1.00 or 2.00 meters for the TPS detector.

the ions are generated and photodetached, while radical photodissociation occurs in the second region.

Methoxide ions are formed in a 60 Hz pulsed supersonic expansion of CH_3OH seeded in 5 atm Ne at room temperature. A pulsed electric discharge in the throat of the expansion creates negative ions,[49] which cool both rotationally and vibrationally during the expansion. The ions are accelerated to 8,000 eV laboratory energy, and mass-selected by the Bakker time-of-flight method.[50] The output of a pulsed dye laser intersects the ion beam at the appropriate time such that only methoxide ions are photodetached. Furthermore, the detachment energy is just above threshold[9] ($h\nu$ = 1.71 eV for CH_3O, $h\nu$ = 1.66 eV for CD_3O), so that methoxy is created only in its ground vibrational state.

In the second region of the apparatus, a frequency-doubled pulsed dye laser, operating between 35,000 - 40,500 cm^{-1} with a bandwidth of 0.3 cm^{-1}, intersects the packet of methoxy radicals. The dissociation laser is calibrated against the absorption spectrum of I_2,[51] with an absolute accuracy of 1 cm^{-1}. Fragments from a photodissociation event are detected directly, without an ionization step, using one of two microchannel plate (MCP) detectors. An aluminum strip across the center of each detector prevents undissociated radicals from striking the MCPs, so that all signal observed is due to nascent photofragments.

Three types of experiments are performed to characterize the photodissociation of methoxy. First, photofragment yield (PFY) spectra are obtained by monitoring the total flux of fragments at the TOF detector, located 0.68 m from the dissociation laser, as a function of laser wavelength, giving spectroscopic information complementary to absorption and fluorescence measurements.

Second, after the PFY spectrum is acquired, the photodissociation laser is fixed at a specific energy, and both fragments arising from a single parent radical are detected in coincidence using a time- and position-sensitive (TPS) wedge- and strip-anode detector,[52,53] located 2.0 m from the dissociation laser. Due to the favorable kinematics of this fast beam experiment, the data may be directly inverted to produce the 2-dimensional coupled translational energy and angular distribution, $P(E_T,\theta)$, where E_T is the center-of-mass relative translational energy of the recoiling fragments, and θ is their scattering angle with respect to the electric field of the linearly polarized dissociation laser. This distribution can be separated according to

$$P(E_T,\theta) = P(E_T)\cdot\{1 + \beta(E_T)\cdot P_2(\cos\theta)\} \tag{2}$$

into the angle-independent translational energy distribution, $P(E_T)$, and the energy-dependent anisotropy parameter,[54] $\beta(E_T)$, which describes the angular distribution of fragments. The high laboratory kinetic energy of the fragments affords an MCP detection efficiency of ≈ 50%, which is independent of the identity of the fragment (except for H or D atoms, as noted below). For the experiments presented here, the translational energy resolution is given by $\Delta E_T / E_T = 2.2\%$. This coincidence detection scheme is only feasible when the mass ratio of the two fragments $m_1/m_2 \leq 5$.

Because of this restriction, detection of channel (*IV*) requires a different approach. The third mode of operation is a non-coincidence experiment, in which the time-of-flight of the photofragments is recorded by digitizing the signal from the TOF detector in 1 ns time bins.[46] The observable is a projection of the 3-D velocity distribution onto the radical

beam axis, averaging out much of the detail which is present in the full $P(E_T,\theta)$ distribution. In return, this sacrifice of detail allows us to detect hydrogen atom loss channels. Given sufficient data, a Monte Carlo forward convolution routine can be implemented to determine translational energy and angular distributions which are consistent with the TOF data. In practice, the experiment is much more sensitive to D atoms than H atoms, and this experiment has only been performed for CD_3O. Even for D atoms, the detection efficiency of ≈ 8% is significantly reduced from that of "heavy" fragments such as CH_3.

III. Results

A. Photofragment Yield spectra of CH_3O and CD_3O

The PFY spectrum shown in Fig. 3 is the first direct observation of photodissociation in the $\tilde{A}\,(^2A_1) \leftarrow \tilde{X}\,(^2E)$ transition of methoxy,[7] and also represents the first spectroscopic transitions observed for $h\nu > 37{,}300$ cm^{-1}. The spectrum is composed of two prominent vibrational progressions superimposed on a small but finite continuum of photofragment signal. No signal was observed below 35,300 cm^{-1}. Rotational resolution was not a priority in this study, and although it has been resolved, such spectra are very congested, in part due to the estimated rotational temperatures of 35 - 50 K.[52,55] Therefore a relatively large laser step size of ≈ 6 cm^{-1} was adopted to allow increased signal averaging at each photon energy. However, because of the underlying rotational structure, each individual scan was offset slightly in energy from the previous data in a given scan range in order to average over rotational features, which might otherwise

provide spurious structure since the laser step size is larger than the laser bandwidth (0.3 cm^{-1}).

The two progressions in Fig. 3 have a spacing of ≈ 600 cm^{-1}, which can be identified with progressions involving ν_3, the CO stretch. The CO stretching mode also

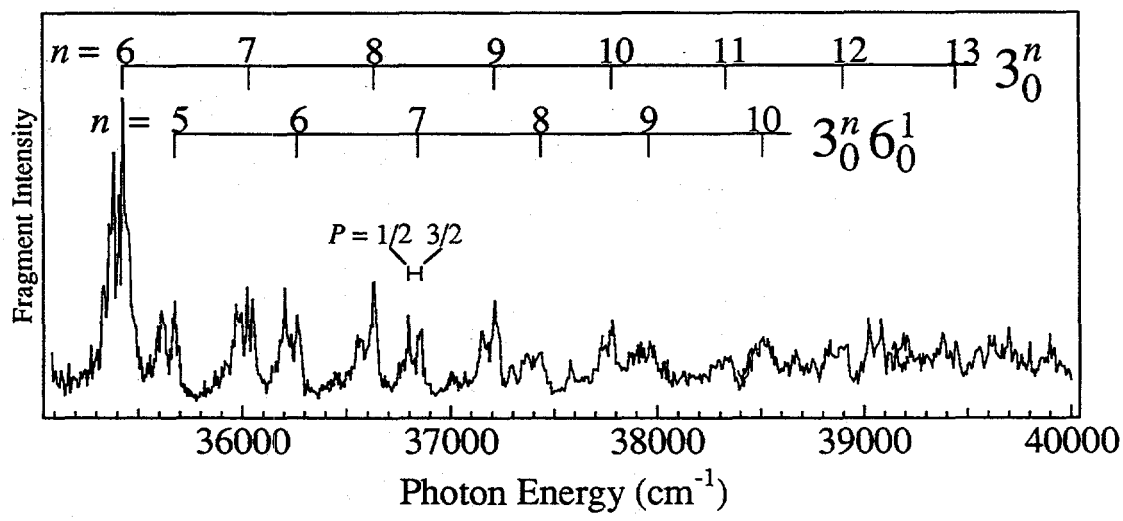


Figure 3: Photofragment yield spectrum of CH_3O. The two quantum numbers for each peak in the $3_0^n e_0^1$ progression refer to the values of n if the e symmetry mode is ν_5 or ν_6. The 60 cm^{-1} spin-orbit splitting between the ^2E_P, P = 3/2, 1/2 levels of the $\tilde{X}$ state is shown.

dominates the absorption and LIF spectra, where it has a fundamental frequency of ν_3 = 662 cm^{-1}.[30] The progressions are assigned as the pure CO stretch, 3_0^n (n = 6 - 13), and a combination band $3_0^n 6_0^1$ (n = 5 - 10), which involves one quantum of the non-totally symmetric methyl rock vibration. The assignment differs from that presented in our preliminary investigation,[7] and is discussed in section IV.A. On a finer scale, each peak shows a splitting of approximately 60 cm^{-1}, corresponding to the spin-orbit splitting between the $^2E_{3/2}$ and $^2E_{1/2}$ components of the ground electronic state. Both spin components are present because each represents an allowed transition in the

photodetachment of CH_3O^- ($^1A_1$). However, the "hot" spin-orbit distribution does not imply a hot vibrational distribution. In fact, the photoelectron spectrum of CH_3O^- (acquired on a different apparatus but with the identical ion source) shows no hot bands, implying that all the ions are in their vibrational ground state.[56] To aid in comparison with LIF spectra[13,16,23-28,30] of jet-cooled methoxy, which probed only transitions from the lower $^2E_{3/2}$ component of the ground state, the vibrational combs in Fig. 3 are centered on these transitions.

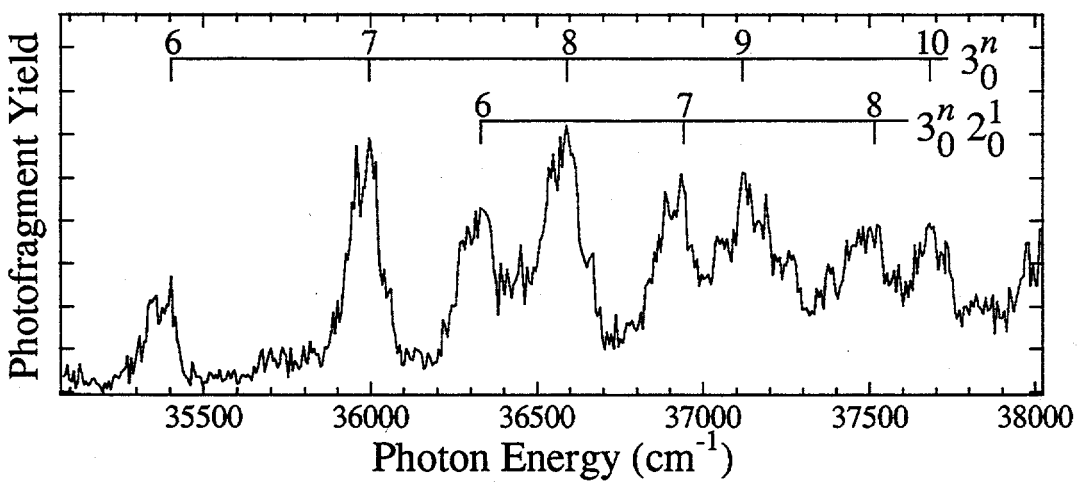


Figure 4: Photofragment yield spectrum of CD_3O.

Figure 4 shows the corresponding PFY spectrum for the CD_3O radical. The general features of the spectrum are analogous to CH_3O, i.e., 3_0^n is the main progression and is accompanied by a combination band. However, the continuum signal underneath the vibrational structure is more intense at lower energies for CD_3O than for CH_3O. In addition, the combination band observed for CD_3O is assigned to a different progression, $3_0^n 2_0^1$, as discussed in Section IV.A.

B. Translational energy and angular distributions

The photofragment translational energy distributions, $P(E_T)$, arising from CH_3O and CD_3O, are given in Figs. 5 and 6 for excitation of several vibrational levels shown in the PFY spectra of Fig. 3 and 4, respectively. The translational energy is binned in intervals of 10 meV. For photon energies $h\nu < 37{,}500$ cm^{-1}, all signal arises from simple C–O bond fission, i.e., channel (*I*). It is obvious that most of the available energy is deposited into translation of the recoiling fragments, because the distributions peak near the maximum translational energy available,[7] as denoted by the dashed vertical line in each distribution of Figs 5 and 6. The most striking feature in the CH_3O $P(E_T)$ distributions is the progression with a spacing of ≈ 74 meV (600 cm^{-1}), which is easily seen in the 3_0^n data sets. As detailed in Section IV.B, these features arise from excitation of the ν_2 umbrella mode of the CH_3 fragment. Note that the vibrational features are much less distinct in the 3_0^6 and $3_0^n 6_0^1$ bands.

In the CD_3O $P(E_T)$ distributions (Fig. 6), most of the available energy again goes into product translation, with a product vibrational progression of ≈ 57 meV (460 cm^{-1}) observed, corresponding to excitation of the ν_2 umbrella mode in CD_3 photofragment. The most surprising results in Fig. 6 are the product state distributions observed when the $3_0^n 2_0^1$ combination band is excited. In section IV.B we will show that these $P(E_T)$ distributions are the result of a bimodal distribution in the ν_2 mode of CD_3, with the two most intense peaks corresponding to of $\upsilon_2 = 0$ and $\upsilon_2 = 2$. Note the stark contrast between the distributions arising from $3_0^n 2_0^1$ excitation of CD_3O and those arising from $3_0^n 6_0^1$ excitation of CH_3O.

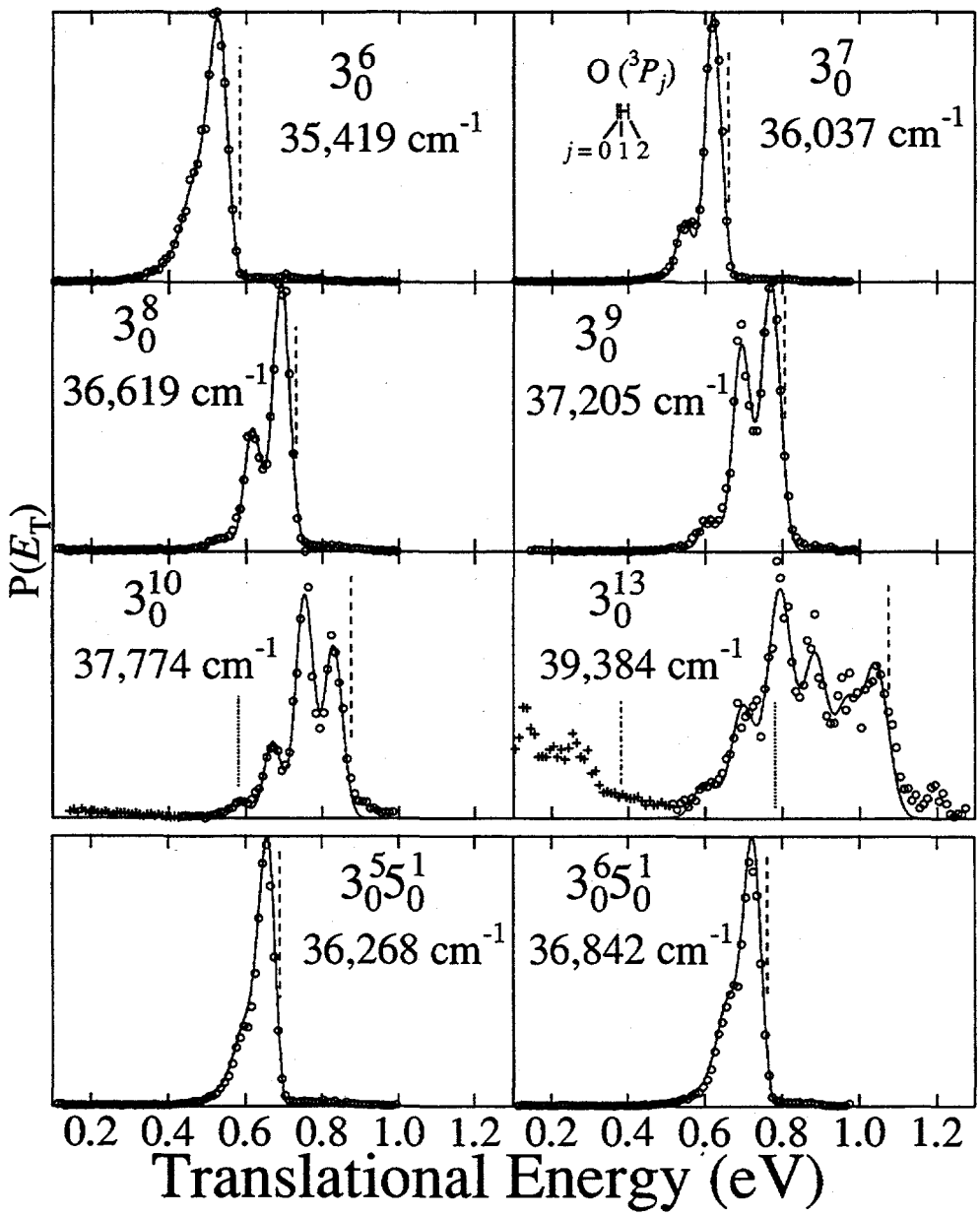


Figure 5: $P(E_T)$ distributions for CH_3O. Circles show CH_3 + O data, crosses show CH_2 + OH data, while the line represents the fit to CH_3 + O discussed in the text. Thermodynamic thresholds are given for CH_3 + O (– – –), CH_2 $\tilde{X}$ ($^3B_1$) + OH (----), and CH_2 $\tilde{a}$ ($^1A_1$) + OH (- - - - -). The spin-orbit levels of O(^3P_j) are shown in the 3_0^7 panel.

All of the $P(E_T)$ distributions presented arise from excitation of the $^2E_{3/2}$ level of the $\tilde{X}$ state. Excitation from the $^2E_{1/2}$ levels exhibits identical behavior, which is to be expected since the level excited in the $\tilde{A}$ ($^2A_1$) state is identical. In addition, $P(E_T)$ distributions acquired at three different photon energies within the rotational envelope of the 3_0^8 band of CH_3O show no significant differences. Therefore, further investigation of

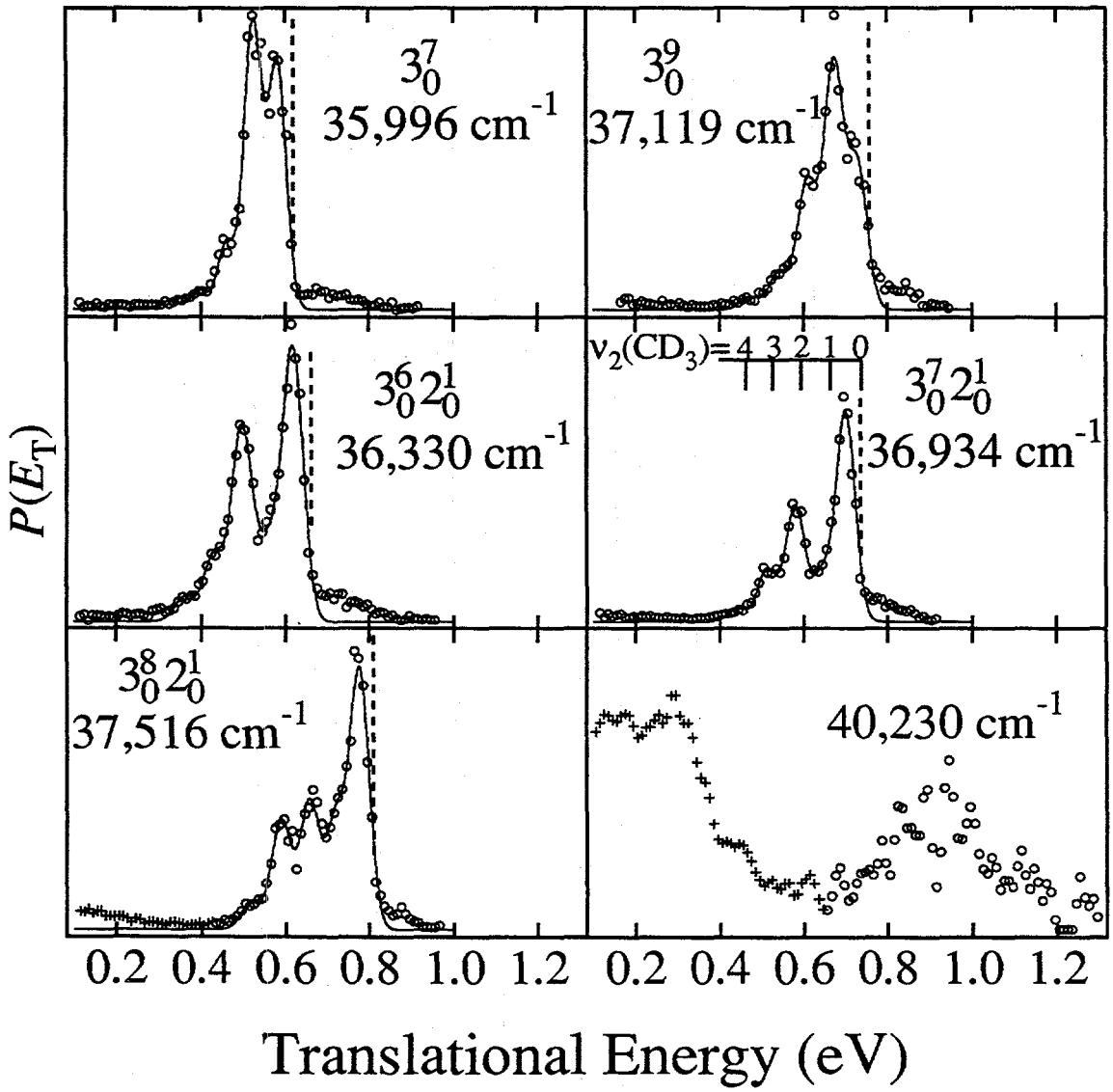


Figure 6: $P(E_T)$ distributions for CD_3O. The legend is identical to Fig. 5.

the dependence of the photodissociation dynamics on the excited state rotational level was not pursued.

For $h\nu > 37{,}500$ cm^{-1}, the structured features in the $P(E_T)$ distributions arising from C–O bond fission remain, but a new feature is observed in both isotopes for $E_T < 0.5$ eV which is due to production of CH_2 + OH (CD_2 + OD). The evidence for this new

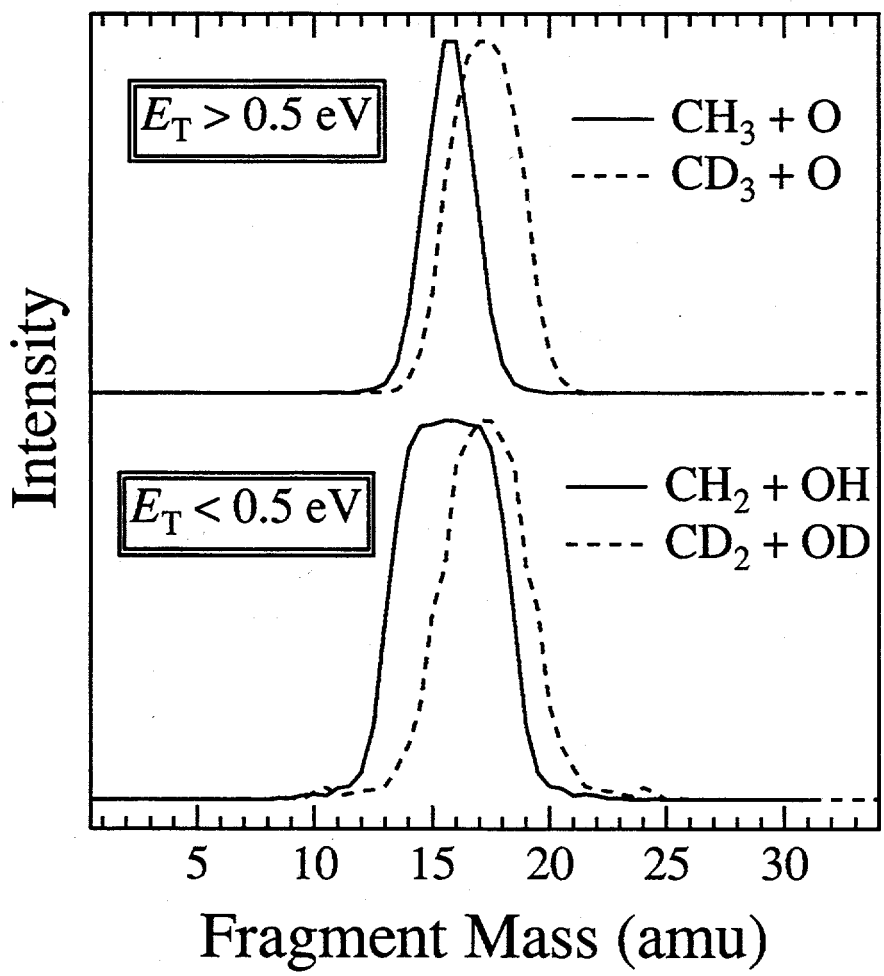


Figure 7: Photofragment mass spectrum exciting 3_0^{13} of CH_3O (—) and 3_0^{10} of CD_3O (---). Note that for $E_T > 0.5$ eV, the deuterium isotope has the broader FWHM, implying fragmentation to channel (*I*), while for $E_T < 0.5$ eV the hydrogen isotope has the broader FWHM, implying fragmentation to channels (*II*) and/or (*III*). The peak heights are normalized to unity and do not represent branching ratios.

channel is shown in Figure 7, which gives the product mass spectra of CH_3O and CD_3O for 3_0^{13} and 3_0^{10} excitation respectively, divided into contributions for $E_T > 0.5$ eV and $E_T < 0.5$ eV. Note that for CD_3O, the two fragments have masses 16 and 18 *regardless* of whether channel (*I*) (CD_3 + O) or channel (*II*)/(*III*) (CD_2 + OD) is produced, resulting in an identical product mass spectrum regardless of the branching ratio of these two channels. In CH_3O, the analogous decay pathways have product masses 15 and 16, or 14 and 17, respectively, such that formation of channel (*I*) should give rise to a narrower peak for CH_3O than for CD_3O, while channel (*II*)/(*III*) products will give a broader peak for CH_3O than for CD_3O. From these observations, the only consistent conclusion is that channel (*I*) products dominate for $E_T > 0.5$ eV, while channel (*II*) / (*III*) products dominate when $E_T < 0.5$ eV, as shown by the legend in Fig. 7.

The branching ratio of CH_3 + O : CH_2 + OH increases with photon energy, and is 3:1 for 3_0^{13}. The same trend occurs in CD_3O, where the extreme case of excitation at $h\nu = 40{,}230$ cm^{-1} results in a branching ration CD_3 + O : CD_2 + OD of ≈ 1:1, as shown in Fig. 6.

In all the data sets acquired for methoxy, the anisotropy parameter $\beta(E_T)$ is independent of E_T. For CH_3O, the 3_0^n transitions show angular distributions which are isotropic to within experimental error, while the $3_0^n 6_0^1$ transitions are slightly anisotropic, described by $\beta = 0.4 \pm 0.1$ for the transition at 36,268 cm^{-1}, and $\beta = 0.3 \pm 0.1$ for the transition at 36,842 cm^{-1} in Fig. 5. In CD_3O, all the transitions give isotropic angular distributions within experimental error.

C. Non-coincidence time-of-flight distributions

As mentioned in the experimental section, a non-coincidence experiment must be performed in order to assess the importance of the hydrogen atom loss pathway, channel (*IV*). In our preliminary work on methoxy, we reported a branching ratio for channel (*IV*) in CD_3O of $2\%^{+5}_{-1}$, and $10\%^{+10}_{-5}$ for the 3^8_0 and 3^{10}_0 bands respectively.[7] The experiments were repeated recently, and the data again shows evidence of D atom loss from CD_3O, but with less intensity than our earlier data. Based on our experience with photodissociation of the vinoxy radical, in which D atom loss is a major channel,[46] it appears that the branching ratio for CD_2O + D is likely smaller than the values reported previously, representing no more than 5% of the branching ratio at 3^{10}_0. In any event, this decay channel seems to be of minor importance in methoxy, at least for the photon energy range used in this experiment.

IV. Analysis

A. Photofragment Yield spectra of CH_3O and CD_3O

The rotational term value of the $\tilde{A}\,(^2A_1) \leftarrow \tilde{X}\,(^2E)$ transition in CH_3O was given by Liu *et al.*[25] as 31,614.51 cm^{-1} from rotational analysis of the combined LIF and microwave[14] data. The effective 0^0_0 transition, measured from the lowest level of the $^2E_{3/2}$ component of the ground state, is 31,644.6 cm^{-1}; it is convenient to use this value when comparing the vibronically resolved $\tilde{A}\,(^2A_1) \leftarrow \tilde{X}\,(^2E_{3/2})$ components in our PFY spectra with jet-cooled LIF experiments (in which all transitions originate in the $^2E_{3/2}$ component).

The significant increase in CO bond length for the $\tilde{A} \leftarrow \tilde{X}$ transition gives rise to an extended Franck-Condon progression in ν_3 for both absorption and LIF measurements, and it is reasonable to assume that the same progression will dominate the PFY spectrum. While fluorescence is observed in CH_3O up to ≈ 37,000 cm^{-1}, spectroscopic assignments from LIF are complicated in this energy range due to spectral congestion and Fermi resonances. In our previous analysis of the PFY spectrum of CH_3O,[7] we assigned the two progressions shown in Fig. 3 as 3_0^n and $3_0^n 5_0^1$, where the identity of the latter progression was the only feasible assignment based on the accepted vibrational frequencies of the $\tilde{A}$ state at that time.[23] Due to a recent reassignment[30] of the ν_6 fundamental frequency in the $\tilde{A}$ state, (previously 595 cm^{-1}, currently 929.5 cm^{-1}), it is unclear from the peak positions alone whether the combination band we observe in the PFY spectrum of CH_3O arises from the ν_5 (1,403 cm^{-1}) or the ν_6 mode. The ambiguity arises because the $3_0^n 5_0^1$ and $3_0^{n+1} 6_0^1$ progressions have nearly identical frequencies. However, our dynamics results discussed in section V.B.1, together with recent LIF results,[44] suggest that the combination band observed in dissociation involves the ν_6 methyl rocking mode rather than ν_5.

We assign the lowest energy dissociative transition as 3_0^6, corresponding to an energy 3,775 cm^{-1} above the zero-point level of the $\tilde{A}$ state. The assignment of the main progression as purely 3_0^n is complicated because of the Fermi resonance between ν_2 and $2\nu_3$ in the $\tilde{A}$ state, discussed further in section V. Based on the most recent rotationally resolved LIF data,[30] the 3_0^6 and $3_0^5 6_0^1$ transitions occur at 35,437 and 35,673 cm^{-1} respectively, compared with the two lowest transitions in our PFY spectrum at 35,419 and

35,666 cm^{-1}. The comparison of LIF with PFY transition frequencies agrees to within experimental uncertainty when one considers that our rotational temperature (35 - 50 K) is greater than in the LIF spectra (5 K), displacing the band maxima (which we measure) from the true rovibronic origin. In addition, our laser step size of ≈ 6 cm^{-1} prevents us from determining band maxima more precisely than this value.

The PFY spectrum of CD_3O (Fig. 4) shows some interesting differences with that of CH_3O. Assignment of the 3_0^n progression is straightforward, due to the small isotope shift in this mode (ν_3 = 663 cm^{-1}).[23] However, the combination band we observe is more difficult to assign. While it is reasonable to assume that the same $3_0^n 6_0^1$ progression found in CH_3O will be active in CD_3O, this assignment gives poor agreement with the data. The ν_5 fundamental in CD_3O is reported as 1,047 cm^{-1},[23] and we estimate the ν_6 fundamental as 693 cm^{-1} (based on the measured isotope shift in ν_5). From our observation of the 3_0^6 transition at 35,404 cm^{-1}, we expect the $3_0^6 5_0^1$ transition at ≈ 36,451 cm^{-1}, which is in fact a minimum between two peaks in Fig. 4, and the $3_0^6 6_0^1$ transition at ≈ 36,097 cm^{-1}, which lies on the blue edge of the 3_0^7 transition. By contrast, the ν_2 fundamental (CD_3O umbrella) is 971 cm^{-1}, from which the $3_0^6 2_0^1$ transition is predicted at ≈ 36,375 cm^{-1}, in better agreement with the first peak at 36,330 cm^{-1} in the strong combination band of CD_3O. Therefore we assign the combination band in the PFY spectrum of CD_3O as $3_0^n 2_0^1$, as corroborated by the dissociation dynamics discussed in section V.B.2.

B. Translational energy distributions

1. CH_3O

Analysis of the $P(E_T)$ distributions will focus primarily on the CH_3 + O product channel, for which the most detailed information is obtained. The main goals of the analysis are to determine: (i) the best value of the bond dissociation energy $D_0(CH_3\text{–}O)$, (ii) the CH_3 fragment vibrational distribution, and (iii) a measure of rotational excitation of the CH_3 fragment. The analysis results are unchanged from those presented previously, and a fuller description of the analysis is given in this section.

The balance of energy for photodissociation of methoxy is given by

$$h\nu + E_{int}(CH_3O) = D_0(CH_3\text{–}O) + E_T + E_V(CH_3) + E_R(CH_3) + E_{SO}(O\,^3P_j) \qquad (3)$$

where E_{int} is the most probable rotational energy of the parent radical, E_V and E_R are the product vibrational and rotational energy, and E_{SO} is the spin-orbit energy of the oxygen atom. Based on a rotational temperature of 50 K, we estimate $E_{int}(CH_3O) = 35\ cm^{-1}$. We can determine D_0 since we measure E_T directly, but only if we can locate an energy in the $P(E_T)$ distribution for which the last three terms of Eq. (3) are zero. The steep falloff in intensity on the high energy side of each distribution in Fig. 5 is evidence for this thermodynamic limit, in which all the available energy goes into product translation, forming CH_3 ($\upsilon = 0$, $J = 0$) + O ($^3P_2$). For each data set we obtain an independent estimate of D_0 by extrapolating to the energy where the $P(E_T)$ distribution reaches zero

intensity. An average of these thresholds, deconvoluted from the experimental resolution, is used for the parameter D_0 in the fitting procedure discussed below.

In order to determine the product vibrational distributions, we fit the data to a set of rotational / spin-orbit distribution functions separated by the accurately known term energies of the CH_3 umbrella mode.[57] The distribution functions, $f_n(E_T)$, are nearly Gaussian in shape, with an asymmetric tail extending to lower translational energy (i.e., towards higher rotational energy). The shape of this function is identical for every vibrational peak in a given spectrum. The model distribution, $F(E_T)$ is given by:

$$F(E_T) = \sum_{n=0}^{n'} \alpha_n f_n \{E_T - (h\nu - D_0 - n\omega_2 - \Delta); \Gamma\} \tag{4}$$

Here n labels the number of quanta in the CH_3 ν_2 umbrella mode, Γ is the FWHM of the combined rotational / spin-orbit envelope and instrumental resolution, and Δ is the energy difference between the maximum possible E_T for each n $\{E_T^{max}(n) = h\nu - D_0 - n\omega_2\}$ and the peak of the nth rotational distribution function.

The coefficients α_n give the CH_3 product vibrational distribution, while the instrumental resolution can be deconvoluted from the width Γ to give an indication of the energy balance between the last two terms of Eq. (3).[58] The solid line in Fig. 5 shows the best non-linear least squares fit to each data set, with the results given in Table I. We note that the fits are sensitive to the strong negative anharmonicity in the ν_2 mode of CH_3, confirming our assignment that the highest energy peak in each spectrum corresponds to $\upsilon_2 = 0$.

The third column of Table I shows the FWHM of the rotational envelope in meV for two limiting cases. The larger values fit the data if it is assumed that all the O atoms are produced in their $^3P_2$ ground state, while the smaller value assumes a statistical 5:3:1 population of the $^3P_{2:1:0}$ spin-orbit states. The magnitude of the spin-orbit splittings are shown in the 3_0^7 panel of Fig. 5. Both limits reproduce the data well, implying that this data is not sufficiently sensitive to distinguish between the two cases.

Fitting all the data sets for CH_3O gives the best value of $D_0(CH_3\text{–}O) = 3.807 \pm 0.013$ eV. From the known heats of formation of CH_3 and O,[45] we obtain the heat of formation $\Delta H^\circ_{f,0}$ (CH_3O) = 0.29 ± 0.02 eV (6.8 ± 0.4 kcal/mol). Additional thermodynamic quantities are derived from this value in Ref. 7.

Transition	$h\nu$ (cm^{-1})	FWHM[a] (meV)	$\langle E_V \rangle$[b] (meV)	υ=0	υ=1	υ=2	υ=3	υ=4	υ=5
3_0^6	35,419	68 / 63	12	84	14	1	0	0	0
3_0^7	36,037	47 / 39	14	82	16	1	0	0	0
3_0^8	36,619	50 / 41	26	67	28	3	1	0	0
3_0^9	37,205	53 / 43	44	52	40	7	1	0	0
3_0^{10}	37,774	53 / 43	73	35	45	16	4	1	0
3_0^{13}	39,384	73 / 70	179	19	15	20	28	14	4
$3_0^5 5_0^1$	36,268	53 / 45	10	87	11	1	0	0	0
$3_0^6 5_0^1$	36,842	58 / 50	14	80	17	1	0	0	0

Table I: Product branching ratio for $CH_3O \rightarrow CH_3 + O$.

[a] FHWM of vibrational peaks after deconvolution of experimental resolution. The left value in this column assumes production of only $O(^3P_2)$, while the right value assumes a statistical $O(^3P_j)$ distribution (see text).
[b] Average vibrational energy

Finally, we consider product state distributions for channels (*II*) and (*III*), producing CH_2 + OH, which are observed in the energy range $E_T < 0.5$ eV upon excitation of 3_0^{10} and 3_0^{13}. While there is no reproducible vibrational structure in this data, we can speculate on the relative importance of CH_2 $\tilde{X}$ ($^3B_1$) vs. $\tilde{a}$ ($^1A_1$). Referring to Fig. 5, any data with $E_T > E_T^{max}$ (*III*) can only be due to fragmentation via channel (*II*). For $E_T < E_T^{max}$ (*III*), both channels can contribute, but the significant rise in the signal concurrent with the opening of the electronically excited methylene channel implies that channel (*III*) is the major source of CH_2 + OH.

Transition	$h\nu$ (cm^{-1})	FWHM[c] (meV)	$\langle E_V \rangle$[d] (meV)	υ=0	υ=1	υ=2	υ=3	υ=4	υ=5
3_0^6	35,404	80	62	37	38	14	7	4	0
3_0^7	35,996	47	48	39	46	10	4	1	0
3_0^8	36,590	51	65	28	47	17	5	2	1
3_0^9	37,119	52	70	25	43	23	7	2	0
3_0^{10}	37,679	63	88	18	39	28	10	4	1
$3_0^6 2_0^1$	36,330	57	74	43	11	31	11	4	0
$3_0^7 2_0^1$	36,934	51	69	47	11	27	12	3	0
$3_0^8 2_0^1$	37,516	52	79	41	17	20	17	4	1

Table II: Product branching ratio for $CD_3O \rightarrow CD_3$ + O.

[c] FWHM of vibrational peaks after deconvolution of experimental resolution.
[d] Average vibrational energy

2. *CD_3O*

A similar data analysis is undertaken for the CD_3 + O product channel in the deuterated species, with the results given in Table II. Deuteration lowers the frequency of the methyl radical umbrella mode from 606 to 458 cm^{-1},[59] which explains the less resolved vibrational structure observed in the top two panels of Fig. 6. Two important differences are observed in comparison to CH_3O dissociation. First, for a given level of excitation in the 3_0^n progression, the CD_3 vibrational distributions are substantially hotter than those observed for CH_3. Second, the vibrational distributions for the $3_0^n 2_0^1$ series are bimodal, with the population of $\upsilon_2 = 0$ and $\upsilon_2 = 2$ always greater than that for $\upsilon_2 = 1$. The origin of these effects will be discussed in the next section. Fitting all the data sets for CD_3O gives the best value of $D_0(CD_3\text{–}O) = 3.85 \pm 0.02$ eV, and therefore $\Delta H^\circ_{f,0}$ $(CD_3O) = 0.16 \pm 0.04$ eV (3.7 ± 0.9kcal/mol).[60] Note the significant difference of 0.13 eV (3.1 kcal/mol) compared to the heat of formation of CH_3O.

V. Discussion

A. Photofragment Yield spectra of CH_3O and CD_3O

One of the important conclusions from the PFY spectrum is that, within our detection limits, CH_3O does not dissociate for photon energies below the 3_0^6 transition, implying that the crossing of the $\tilde{A}$ state by the lowest of the three repulsive curves shown in Fig. 1 occurs in the vicinity of 3,775 cm^{-1} above the $\tilde{A}$ state zero-point level. The threshold for dissociation and the energy of the curve crossing has been the subject of

much debate among both experimentalists and theorists, with our value being the lowest reported. Our threshold is unambiguous because we detect dissociation directly by measuring the presence of primary photofragments. A very recent determination of the dissociation threshold by Powers *et al.*,[44] via LIF lifetime measurements and fluorescence depletion spectroscopy (FDS), gives 3,793 cm^{-1}, in agreement with our value.

One of the main complexities in the $\tilde{A}\,(^2A_1)$ state of CH_3O arises because two quanta of ν_3 establish a Fermi resonance with ν_2 (CH_3O umbrella, 1289 cm^{-1}), giving rise to a series of multiplets at each overtone in the 3_0^n series.[16] By fitting the observed multiplet spectra to an anharmonic oscillator Hamiltonian for ν_2 and ν_3, Powers *et al.*[30] obtained the dominant character of each member of the Fermi multiplet, although the two modes are strongly mixed. An interesting question is whether the competition between predissociation and fluorescence depends on which member of the multiplet is excited.

The splittings between Fermi multiplet components are significant. For example, the 3_0^5, $3_0^1 2_0^2$, and $3_0^3 2_0^1$ bands are separated by 35 and 43 cm^{-1} respectively,[30] giving a total width to this feature of 78 cm^{-1}. However, the peak widths in Fig. 3 are also broadened due to the width of the rotational bands (at least 20 cm^{-1}) and the fact that we observe transitions in the PFY spectrum from both $^2E_{3/2}$ and $^2E_{1/2}$ components ($\Delta E \approx 60$ cm^{-1}) of the ground state. It is therefore difficult to determine whether different multiplet components decay with different rates from the PFY spectrum. In fact, the LIF and FDS experiments of Powers *et al.*[44] show no significant differences in lifetimes for different members of a particular multiplet.

In CD_3O, the Fermi resonance between ν_2 and ν_3 is lifted, and the vibrational frequencies of all modes except ν_3 are substantially lowered. The combination band in CD_3O is assigned to the $3_0^n 2_0^1$ progression, with the first member appearing in the PFY spectrum as $3_0^6 2_0^1$. While there is certainly enough energy for dissociation of $3_0^5 2_0^1$, this transition is not observed in Fig. 4, implying that for CD_3O, energy deposited in ν_2 is not available to facilitate fragmentation. In contrast, CH_3O dissociates at $3_0^5 6_0^1$, i.e., with five quanta in ν_3. The isotopic comparison implies that excitation of an *e* symmetry mode (observed for CH_3O) promotes dissociation, while energy deposited in the a_1 symmetry ν_2 mode (observed for CD_3O) is not strongly coupled to the reaction coordinate. We give further evidence for these conclusions in section V.B, based on the translational energy and angular distribution data.

B. Channel (*I*) dissociation dynamics

Unimolecular photodissociation is a powerful technique because the disposal of energy among translation, vibrational, rotational, and electronic degrees of freedom in the products is a direct consequence of the potential energy surfaces on which this half-collision occurs.[61] Consequently, the main information we obtain on the dissociation dynamics of the methoxy radical is derived from the $P(E_T)$ distributions presented graphically in Figs. 5 and 6, and numerically in Tables I and II. In this section we discuss the dissociation mechanism for channels (*I* - *III*) in light of the experimental product state distributions.

1. $CH_3 + O$ products

It is evident from Fig. 5 that most of the available energy in the $\tilde{A} \leftarrow \tilde{X}$ photodissociation of CH_3O goes into product translation for channel (*I*). Although radicals similar to methoxy, such as CH_2CHO and HCCO, dissociate in the ultraviolet by internal conversion (IC) to the ground state potential energy surface (PES), the $P(E_T)$ distributions for methoxy are inconsistent with such a mechanism. Indeed, SEP studies[6] on CH_3O prove that if IC were facile, H + CH_2O would be the dominant product channel. Instead, the large recoil energies observed in our data are indicative of dynamics on a purely repulsive PES, in which potential energy is converted efficiently into translational kinetic energy. The results of *ab initio* calculations by Jackels,[33] represented schematically in Fig. 1, predict three excited electronic states of $^4A_2$, ^4E, and $^2A_2$ symmetry, which are repulsive along the C–O bond and correlate asymptotically to $CH_3\,\tilde{X}(^2A_2'') + O(^3P)$. All our results for this channel are consistent with a mechanism in which non-adiabatic coupling between these surfaces and the optically prepared levels of the $\tilde{A}\,(^2A_1)$ state leads to predissociation of CH_3O. Given this basic description of the dissociation mechanism, we can explore further mechanistic details in the experimental data.

First, we consider the flow of energy into product vibration. Based on the fitting procedure presented in section IV.B, the $P(E_T)$ distribution is described accurately by excitation in a single CH_3 mode, the ν_2 umbrella vibration. Focusing on the 3_0^n progression, and ignoring for the moment $n = 6$, a smooth increase in the average CH_3 vibrational excitation $\langle E_V \rangle$ is observed with increasing photon energy (Table I). Very little

product excitation arises from the 3_0^7 level, in contrast to the inverted distributions for 3_0^{10} and 3_0^{13}.

The sudden approximation is often invoked to explain dissociation dynamics on purely repulsive surfaces. In this model the fragments are assumed to dissociate sufficiently rapidly that there is negligible coupling between translational and vibrational degrees of freedom. In the sudden limit, the CH_3 vibrational distribution is obtained by a Franck-Condon projection of the CH_3 moiety in the $\tilde{A}$ state onto the asymptotic ν_2 vibrational levels of the free methyl radical. Specifically, the $\angle$HCO in the $\tilde{A}$ state is 106°, while asymptotically the methyl radical is planar, corresponding to $\angle$HCO = 90° within C_{3v} symmetry. Using *ab initio* force constants for CH_3, the sudden approximation predicts a most probable value of $\upsilon_2 = 5$ for the CH_3 ν_2 distribution, somewhat hotter than the distribution observed at 3_0^{13}, and clearly incorrect for all lower photon energies.

The failure of the sudden approximation implies substantial coupling of translational and vibrational degrees of freedom as the dissociation proceeds on the repulsive surface, allowing $\angle$HCO to relax smoothly with increasing CO bond length. In other words, at low photon energies (e.g. 3_0^7 excitation), CH_3O appears to evolve adiabatically to products. This situation is entirely reasonable when one considers that the light H atoms can respond quickly to motion of the heavy C and O atoms as r_{CO} increases, maintaining an equilibrium value of $\angle$HCO at each value of r_{CO}. This picture also explains the behavior of the 3_0^n $P(E_T)$ distributions with increasing photon energy. The recoil velocity between the C and O atoms increases with increasing available energy, such that

the H atoms cannot respond as accurately to the changing potential along the ∠HCO coordinate, resulting in increased asymptotic CH_3 vibrational excitation.

The photodissociation dynamics of methyl iodide show a strong similarity to our results on methoxy. For the $CH_3I \rightarrow CH_3 + I^*$ channel, which occurs on a purely repulsive surface, the methyl fragment is produced with little vibrational or rotational excitation. An adiabatic mechanism very similar to that described above for methoxy governs the dynamics of this channel in CH_3I, as elucidated by classical trajectory calculations on *ab initio* potential energy surfaces for this system.[62]

We now consider partitioning of energy between the CH_3 rotational energy and the spin-orbit energy of O (^3P_j) (Eq. 3). As shown in column 3 of Table I, the vibrational peak widths increase monotonically in the 3_0^n progression for $n \geq 7$, denoting increasing rotational and/or spin-orbit product excitation. The oxygen atom can make a significant contribution to the peak width because the $^3P_0$ and $^3P_1$ levels of the oxygen atom lie 28 and 20 meV above the ground state $^3P_2$ level respectively.[63] Although we present the rotational envelope widths for the two limiting cases in which no energy or a statistical distribution of energy is deposited in the oxygen atom, there is no reason to believe that either limit reflects reality, nor that the spin-orbit state distribution is independent of photon energy, as our two limiting cases assume. Indeed, by analogy with the spin-orbit distributions measured in O_2 dissociation[64] it is likely that the j distributions of the oxygen atoms from CH_3O are not statistical, but fluctuate in a seemingly random fashion depending the value of n. Subject to this assumption, we conclude that the increasing peak widths in Table I reflect an increase of rotational excitation as a function of photon energy.

However, the amount of the available energy deposited in product rotation is still relatively small. For example, in the 3_0^{10} $P(E_T)$ distribution (Fig. 5), the energy difference between $CH_3(\upsilon = 0, J = 0, K = 0)$ and the peak of the $CH_3(\upsilon = 0)$ distribution is 50 meV (6% of the available energy), corresponding to a $CH_3(\upsilon = 0, J = 6, K = 0)$ fragment. Simply put, the fact that we resolve vibrational structure at all in the $P(E_T)$ distributions means that product rotational excitation is not extensive. We propose that the small amount of product rotation indicates a dissociation pathway which deviates little from C_{3v} symmetry for 3_0^n excitation.

In contrast to the $P(E_T)$ distributions just discussed, the distributions for 3_0^6 and $3_0^n 6_0^1$ lack the well-resolved vibrational features present in the other spectra in Fig. 5. For 3_0^6, the much broader vibrational peak is likely due to an experimental effect rather than an increase in rotational excitation with respect to 3_0^7. The most recent literature value for the fluorescence lifetime of 3_0^6 is 0.37 μs,[44] a substantial fraction of the 9 μs flight time required for a CH_3O radical with 8 keV center-of-mass energy to travel from the dissociation laser to the TPS detector. Excited state lifetimes in this range degrade the $P(E_T)$ distributions to lower E_T because long-lived radicals remain intact for a portion of the flight time, resulting in a smaller recoil vector measured at the detector, which translates to an artificially small value of E_T for that event. Using the fluorescence lifetime given above, this effect causes a 1/e broadening of 51 meV.[65] When this width is deconvoluted from the 68 meV width given in Table I, the remaining rotational envelope has a width of 45 meV, slightly narrower than the 47 meV width measured for 3_0^7.

Therefore, rotational excitation of the CH_3 + O products following 3_0^6 excitation follows the same trend as the other members in the 3_0^n progression despite the broad peak observed in Fig. 5.

The $P(E_T)$ distributions following $3_0^6 6_0^1$ and $3_0^7 6_0^1$ excitation show broader peaks than their quasi-isoenergetic counterparts in the 3_0^n progression. Powers *et al.*[44] found that the fluorescence lifetime for $3_0^6 6_0^1$ lies between the limits 20 ns $\geq \tau_{FL} \geq$ 20 ps. As a result of this short lifetime, the broadening effect described above for 3_0^6 is negligible compared to the experimental resolution, and the increased width of the peaks is likely a result of greater product rotational excitation. It is quite plausible that sampling of non-C_{3v} geometries due to *e* symmetry vibration would result in greater torque in the exit channel, and hence more product rotational excitation. This explanation is reasonable when one considers that the ν_6 methyl rocking motion would naturally develop into rotational excitation of the methyl radical about the *b*-axis, further dynamical evidence that the combination band in Fig. 3 is principally due to $3_0^n 6_0^1$ rather than $3_0^n 5_0^1$.

Recall that the product angular distributions for $3_0^6 6_0^1$ and $3_0^7 6_0^1$ excitation are described by anisotropy parameters of $\beta = 0.4$ and $\beta = 0.3$ respectively, while the angular distributions of the 3_0^n progression are isotropic ($\beta = 0$). However, the $\tilde{A}\,(^2A_1) \leftarrow \tilde{X}\,(^2E)$ transition has a perpendicular transition dipole moment, which should result in an anisotropy parameter $\beta \leq 0$. Although the vibronic symmetry of $3_0^n 6_0^1$ is parallel ($e \leftarrow e$), the electronic part of the transition dipole remains ($a \leftarrow e$) in the absence of perturbations by other electronic states,[30] and hence should still give $\beta \leq 0$. Therefore the positive

values of β measured for $3_0^6 6_0^1$ and $3_0^7 6_0^1$ excitation implies that the $\tilde{A}$ state is perturbed by another electronic state of e symmetry (such as the nearby ^4E state shown in Fig. 1), which mixes an electronic $e \leftarrow e$ component into the transition dipole. These results are consistent with the observation of both parallel and perpendicular rotational structure resolved in bands containing fundamentals of e vibrations.[30]

As a final point on the angular distributions, we note that the observation of $\beta \neq 0$ for $3_0^n 6_0^1$ does not necessarily imply that the excited levels have a shorter lifetime than those in the 3_0^n progression, for which $\beta = 0$. Classically, β reaches the limiting value of 0.4 for a parallel transition of a symmetric top with a dissociative lifetime much longer than a rotational period.[66] Quantum mechanically, however, the anisotropy can average to zero due to interference between rotational wavefunctions. As a result, we do not speculate on the lifetime of the excited levels based on their angular distributions.

2. *CD_3 + O products*

The C–O bond fission channel in CD_3O shows three main differences from that of CH_3O. First, as stated in section V.A, the active combination band is $3_0^n 2_0^1$, in which all vibrational motion is totally symmetric. Second, the CD_3 vibrational distributions resulting from 3_0^n excitation are substantially more excited than at similar photon energies in CH_3O, as verified by comparison of the average product vibrational energies $\langle E_V \rangle$ in Tables I and II. Finally, the CD_3 vibrational distributions resulting from $3_0^n 2_0^1$ excitation are bimodal, as shown in Fig. 6.

Increased vibrational excitation of the CD_3 umbrella motion resulting from 3_0^n excitation follows directly from the adiabatic dissociation mechanism proposed for channel (*I*) dynamics in CH_3O. When H atoms are replaced with D atoms in methoxy, the increased mass inhibits the ability of these atoms to track the motion of the C and O atoms during dissociation. As a result, the ∠DCO angle will not relax to its equilibrium value as readily as in the lighter isotope, resulting in more excitation of the ν_2 mode in CD_3.

A similar isotope effect is observed in CD_3I dissociation, and the complete description of these effects found in Ref. 62 is most likely applicable to CD_3O. However, for the present purposes, the simple explanation given above is sufficient.

By far the most surprising result from the CD_3O $P(E_T)$ distributions is the bimodal CD_3 product state distribution observed following excitation of the $3_0^n 2_0^1$ combination band (cf. Table II). To the best of our knowledge, methoxy photodissociation represents the first observation of such an isotope effect on the *product* state distributions. A compelling hypothesis for the origin of this dynamical effect lies in the relationship between the umbrella mode frequency in the $\tilde{A}$ state of CD_3O (971 cm^{-1}), and the corresponding umbrella frequency in the CD_3 fragment (458 cm^{-1}). Due to the negative anharmonicity in the methyl radical, the first overtone of $\nu_2(CD_3)$ occurs at 966 cm^{-1}.[59] Therefore, if the single quantum of $\nu_2(CD_3O)$ excited in the $3_0^n 2_0^1$ combination band does not couple well to the reaction coordinate, the 971 cm^{-1} of energy deposited into this motion will result in a propensity to populate the nearly degenerate first overtone of $\nu_2(CD_3)$, i.e., the amount of energy in umbrella motion will be conserved throughout the dissociation. This hypothesis is also supported by the fact that $3_0^5 2_0^1$ does not dissociate

(cf. Fig. 4), even thought this state has more energy than the 3_0^6 level, which does dissociate. This intriguing hypothesis calls for a comparison with reaction dynamics calculations based on the new potential energy surfaces which are now becoming available for methoxy.[39,40]

3. Curve crossing mechanism

Until now we have discussed the dissociation dynamics without consideration of the mechanism that couples the optically accessed $\tilde{A}\,(^2A_1)$ state with the $^4A_2$, ^4E, and $^2A_2$ repulsive states. The calculations of Jackels demonstrated that the ^4E state is reached via a one-electron transition from the $\tilde{A}$ state, while the two A_2 configurations require the rearrangement of two electrons.[33] He argued that the $^2A_1 \Leftrightarrow {}^4E$ interaction should be the strongest, because these states have a first-order spin-orbit interaction, analogous to the $^2\Sigma \Leftrightarrow {}^4\Pi$ coupling in a diatomic radical. The $^2A_1 \Leftrightarrow {}^4A_2$ interaction can occur by a second-order spin-orbit interaction, while the $^2A_1 \Leftrightarrow {}^2A_2$ interaction is most likely dominated by off-diagonal elements in the nuclear kinetic energy operator.

Recent *ab initio* calculations by Pederson and Yarkony,[39] and by Cui and Morokuma[40] have defined the minimum seam of crossings between the $\tilde{A}$ state and each repulsive curve. The new calculations place the crossing points at lower energy than those obtained by Jackels, in better agreement with our photodissociation threshold. The coupling constants between the states are also calculated, along with the dependence of the coupling at geometries away from C_{3v} symmetry.[39]

Until now, the experimental data has been discussed for simplicity under the tacit assumption that one repulsive curve accounts for the entire dynamics of channel (*I*). One

feature in Fig. 5 may shed some light on whether coupling with multiple curves contributes to the final product state distributions. The CH_3 vibrational excitation resulting from 3_0^{13} excitation is bimodal (cf. Table I), which could plausibly result from the contribution of an additional repulsive curve at this high photon energy leading to vibrationally cold products. Again, detailed assessment of this issue should soon be possible in light of the new potential energy surfaces being developed.

C. Channel (*II*) and (*III*) dissociation mechanism

Our experimental information on the CH_2 + OH product channel is limited. We find that this product channel is absent for transitions with $h\nu$ < 37,500 cm^{-1}, and competes with the dominant CH_3 + O channel at higher photon energies. As pointed out in section IV.B, on energetic grounds it appears that channel (*III*), producing electronically excited methylene fragments, dominates production of ground state methylene, channel (*II*). Concerning the mechanism for this channel, the recent *ab initio* study of the methoxy system by Cui and Morokuma[40] explored the excited state isomerization of $CH_3O \rightarrow CH_2OH$, finding a transition state 1.00 eV above the $\tilde{A}$ state minimum. They point out that the $\tilde{A}$ state of CH_3O correlates through this transition state to channel (*III*) products, while the $\tilde{X}$ state correlates to channel (*II*) products. Our data are consistent with the excited state mechanism, and the observed production of some ground state methylene could easily be explained by a transition to a lower surface along the isomerization or dissociation coordinate. A similar mechanism is probably responsible for CD_2 + OD dissociation in CD_3O.

VI. Conclusions

The ultraviolet $\tilde{A}\,(^2A_1) \leftarrow \tilde{X}\,(^2E)$ transition in both CH_3O and CD_3O is observed to predissociate due to curve crossings with one or more of the excited $^2A_2$, $^4A_2$, and ^4E repulsive states. Several interesting differences in the photodissociation dynamics have been observed between the two isotopes. The photofragment yield spectrum of CH_3O shows a main progression in the C–O stretch, and a combination band with one quantum of the e symmetry ν_6 methyl rock. The main progression is identical in CD_3O, but the combination band is assigned as $3_0^n 2_0^1$, involving the a_1 symmetry ν_2 umbrella motion. The 3_0^6 transition is the lowest energy transition observed to dissociate in either isotope, placing the threshold for CH_3O dissociation 3,775 cm^{-1} above the zero point level of the $\tilde{A}$ state. In CH_3O the combination band $3_0^n 6_0^1$ dissociates for $n = 5$, while in CD_3O combination band dissociation is not observed unless $n \geq 6$. This mode-specific effect demonstrates that the way in which energy is distributed in the $\tilde{A}$ state effects the ultimate outcome.

The major product channel is C–O bond fission giving a methyl radical and an oxygen atom. The translational energy distributions for both isotopes show that the ν_2 umbrella mode of the $CH_3(CD_3)$ fragment is excited. For 3_0^n excitation of both isotopes, the product state distribution is described by an adiabatic picture in which the product excitation slowly increases with increasing photon energy. Excitation of $3_0^n 6_0^1$ in CH_3O may result in more rotational excitation of the products, consistent with sampling of non-C_{3v} geometries in the $\tilde{A}$ state. Excitation of $3_0^n 2_0^1$ in CD_3O shows a surprising dynamical

effect: the product state distribution of the CD_3 fragment is bimodal, with excitation of $\upsilon_2 = 0$ and $\upsilon_2 = 2$ always greater than that for $\upsilon_2 = 1$.

The results given in this article should enable a rigorous comparison between experiment and theory because the data comprise *both* translational and vibrational product state distributions, acquired at many different photon energies for two isotopes. New *ab initio* potential energy surfaces are now becoming available for methoxy, and it appears that the methoxy radical will continue its tradition as a benchmark molecule in the fields of spectroscopy and dynamics.

Acknowledgments

This research is supported by the Director, Office of Energy Research, Office of Basic Energy Science, Chemical Sciences Division of the US Department of Energy on Contract No. DE-AC03-76SF00098.

References

[1] D. W. G. Style and J. C. Ward, Trans. Faraday Soc. **49**, 999 (1953).

[2] P. H. Leighton, *Photochemistry of Air Pollution* (Academic Press, New York, 1961); B. J. Finlayson-Pitts and J. N. Pitts, Jr., *Atmospheric Chemistry* (Wiley, New York, 1986); C. T. Pate, B. J. Finlayson and J. N. Pitts, Jr., J. Am. Chem. Soc. **96**, 6554 (1974); H. Levy II, Planet. Spac. Sci. **21**, 575 (1973); A. C. Baldwin, J. R. Barker, D. M. Golden, and D. G. Hendry, J. Phys. Chem. **81**, 2483 (1977).

[3] C. K. Westbrook and F. L. Dryer, Progr. Energy Combustion Sci. **10**, 1 (1984); W. Tsang and R. F. Hampson, J. Phys. Chem. Ref. Data **15**, 1087 (1986); B. A. Williams and J. W. Fleming, Chem. Phys. Lett. **221**, 27 (1994); S. Zabarnick, Combustion Flame **85**, 27 (1991); D. B. Olsen and W. C. Gardiner Jr., J. Phys. Chem. **81**, 2514 (1977).

[4] J. Warnatz, *Combustion Chemistry*, ed. W. C. Gardiner (Springer, Berlin) 258 (1984).

[5] P. Thaddeus, private communication.

[6] A. Geers, J. Kappert, F. Temps, and J. W. Wiebrecht, J. Chem. Phys. **99**, 2271 (1993); *ibid.* **101**, 3618 (1994); *ibid.* **101**, 3634 (1994).

[7] D. L. Osborn, D. J. Leahy, E. R. Ross, and D. M. Neumark, Chem. Phys. Lett. **235**, 484 (1995).

[8] K. Ohbayashi, H. Akimoto, and I Tanaka, J. Phys. Chem. **81**, 798 (1977).

[9] P. C. Engelking, G. B. Ellison, and W. C. Lineberger, J. Chem. Phys. **69**, 1826 (1978).

[10] H. R. Wendt and H. E. Hunziker, J. Chem. Phys. **71**, 5202 (1979).

[11] D. K. Russell and H. E. Radford, J. Chem. Phys. **72**, 2750 (1980).

[12] G. Inoue, H. Akimoto, and M. Okuda, J. Chem. Phys. **72**, 1769 (1980).

[13] D. E. Powers, J. B. Hopkins, and R. E. Smalley, J. Phys. Chem. **85**, 2711 (1981).

[14] Y. Endo, S. Saito, and E. Hirota, J. Chem. Phys. **81**, 122 (1984).

[15] P. G. Carrick, S. D. Brossard, and P. C. Engelking, J. Chem. Phys. **83**, 1995 (1985).

[16] K. Fuke, K. Ozawa, and K. Kaya, Chem. Phys. Lett. **126**, 119 (1986).

[17] T. Ebata, H. Yanagishita, K. Obi, and I. Tanaka, Chem. Phys. **69**, 27 (1982).

[18] B. S. Agrawalla and D. W. Setser, J. Phys. Chem. **90**, 2450 (1986).

[19] J. A. McCaulley, S. M. Anderson, J. B. Jeffries, and F. Kaufman, Chem. Phys. Lett. **115**, 180 (1985).

[20] S. D. Brossard, P. G. Carrick, E. L. Chappell, S. C. Hulegaard, and P. C. Engelking, J. Chem. Phys. **84**, 2459 (1986).

[21] N. L. Garland and D. R. Crosley, J. Phys. Chem. **92**, 5322 (1988).

[22] S. R. Lin, Y. P. Lee, and J. B. Nee, J. Chem. Phys. **88**, 171 (1988).

[23] S. C. Foster, P. Misra, T. Y. D. Lin, C. P. Damo, C. C. Carter, and T. A. Miller, J. Phys. Chem. **92**, 5914 (1988).

[24] S. Y. Chiang, Y. C. Hsu, and Y. P. Lee, J. Chem. Phys. **90**, 81 (1989).

[25] X. Liu, C. P. Damo, T. Y. D. Lin, S. C. Foster, P. Misra, L. Yu, and T. A. Miller, J. Phys. Chem. **93**, 2266 (1989).

[26] X. Liu, S. C. Foster, J. M. Williamson, L. Yu, and T. A. Miller, Mol. Phys. **69**, 357 (1990).

[27] P. Misra, X. Zhu, C. Hsueh, and J. B. Halpern, Chem. Phys. **178**, 377 (1993).

[28] Y. Y. Lee, G. H. Wann, and Y. P. Lee, J. Chem. Phys. **99**, 9465 (1993).

[29] B. A. Williams and J. W. Fleming, Chem. Phys. Lett. **221**, 27 (1994).

[30] D. E. Powers, M. Pushkarsky, and T. A. Miller, (to be submitted).

[31] F. Temps, (unpublished).

[32] D. R. Yarkony, H. F. Schaefer, and S. Rothenberg, J. Am. Chem. Soc. **96**, 656 (1974).

[33] C. F. Jackels, J. Chem. Phys. **76**, 505 (1982); *ibid.* **82**, 311 (1985).

[34] G. D. Bent, G. F. Adams, R. H. Bartram, G. D. Purvis, and R. J. Bartlett, J. Chem. Phys. **76**, 4144 (1982).

[35] J. T. Carter and D. B. Cook, J. Mol. Structure (Theochem) **251**, 111 (1991).

[36] L. A. Curtiss, L. D. Kock, and J. A. Pople, J. Chem. Phys. **95**, 4040 (1991).

[37] S. P. Walch, J. Chem. Phys. **98**, 3076 (1993).

[38] G. D. Bent, J. Chem. Phys. **100**, 8219 (1994).

[39] L. Pederson and D. Yarkony, (private communication).

[40] Q. Cui and K. Morokuma, (private communication).

[41] R. Englman, *The Jahn-Teller Effect in Molecules and Crystals* (Wiley, New York, 1972).

[42] A. Geers, J. Kappert, and F. Temps, J. Chem. Phys. **98**, 4297 (1993).

[43] J. T. Hougen, J. Mol. Spec. **81**, 73 (1980); T. K. Watson, J. Mol. Spec. **103**, 125 (1984); J. M. Brown, Mol. Phys. **20**, 817 (1971); and F. S. Ham, Phys. Rev. **166**, 307 (1968).

[44] D. E. Powers, M. Pushkarsky, and T. A. Miller, (to be submitted).

[45] M. W. Chase, Jr., C. A. Davies, J. R. Downey, Jr., D. J. Frurip, R. A. McDonald, and A. N. Syverud, JANAF Thermochemical Tables, 3rd ed., J. Chem. Phys. Ref. Data **14**, (1985), Suppl. No. 1.

[46] D. L. Osborn, H. Choi, D. H. Mordaunt, R. T. Bise, D. M. Neumark, and C. M. Rohlfing, J. Chem. Phys. **106** (1996).

[47] R. E. Continetti, D. R. Cyr, D. L. Osborn, D. J. Leahy, and D. M. Neumark, J. Chem. Phys. **99**, 2616 (1993).

[48] R. E. Continetti, D. R. Cyr, R. B. Metz, and D. M. Neumark, Chem. Phys. Lett. **182**, 406 (1991).

[49] D. L. Osborn, D. J. Leahy, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **104**, 5026 (1996).

[50] J. M. B. Bakker, J. Phys. E **6**, 785 (1973); *ibid.* **7**, 364 (1974).

[51] S. Gerstenkorn and P. Luc, *Atlas du Spectre d'Absorption de la Molecule d'Iode* (Centre National de la Recherche Scientifique, Paris, 1978).

[52] D. J. Leahy, D. L. Osborn, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **103**, 2495 (1995).

[53] D. P. de Bruijn and J. Los, Rev. Sci. Instrum. **53**, 1020 (1982).

[54] R. N. Zare, Mol. Photochem. **4**, 1 (1972).

[55] D. L. Osborn, H. Choi, D. H. Mordaunt, R. T. Bise, D. M. Neumark, and C. M. Rohlfing (in preparation).

[56] D. L. Osborn, D. J. Leahy, E. H. Kim, E. deBeer, and D. M. Neumark, (in preparation).

[57] C. Yamada, E. Hirota, and K. Kawaguchi, J. Chem. Phys. **75**, 5256 (1981); C. Yamada and E. Hirota, *ibid.* **78**, 669 (1983); H. W. Herman and S. R. Leone, J. Chem. Phys. **76**, 4759 (1982).

[58] It should be noted that the value for the parameter Δ scales roughly with the value of Γ; together they control the amount of rotational and spin-orbit energy in the model.

[59] T. J. Sears, J. M Frye, V. Spirko, and W. P. Kraemer, J. Chem. Phys. **90**, 2125 (1989).

[60] $\Delta H^{\circ}_{f,0}(CD_3)$ is derived as $\Delta H^{\circ}_{f,0}(CD_3) = \Delta H^{\circ}_{f,0}(CH_3)$ - [ZPE(CH_3) - ZPE(CD_3)] + 3/2[ZPE(H_2) - ZPE(D_2)], where the zero-point energies for the methyl radicals are taken from Ref. 59.

[61] R. Schinke, *Photodissociation Dynamics* (Cambridge University Press, Cambridge, 1993).

[62] Y. Amatatsu, K. Morokuma, and S. Yabushita, J. Chem. Phys. **94**, 4858 (1991); *ibid.*, **104**, 9783 (1996).

[63] C. E. Moore, *Atomic Energy Levels*, Vol. 1, NSRDS-NBS Circ. No. 467 (U. S. GPO, Washington, 1949), p. 15.

[64] D. J. Leahy, D. L. Osborn, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **103**, 2495 (1995).

[65] For comparison, a 10 ns fluorescence lifetime gives a 1/e broadening of 1 meV, a negligible value compared to the instrumental resolution.

[66] S.-C. Yang and R. Bersohn, J. Chem. Phys. **61**, 4400 (1974).

4 Fast beam photodissociation spectroscopy and dynamics of the vinoxy radical

I. Introduction

Molecular photodissociation is an important field of chemical physics as it offers a well-defined way of probing the forces acting between atoms and molecules in chemical reactions.[1] The dynamics of a dissociating molecule depend intimately on the shape of the potential energy surface of the system. Measurement of product state distributions can provide direct information on important features of this surface, and this prospect has motivated development of an impressive array of experimental techniques aimed at extracting such distributions.[2] While most experiments of this type have been performed on stable, closed-shell molecules, there is much to gain from the study of free radical systems using photodissociation. There are two qualitative differences between closed-shell and open-shell systems that should directly influence their photodissociation dynamics. First, the unpaired electron(s) of free radicals gives rise to more low-lying electronic states ($\leq$5 eV) than are typically found in stable molecules. Second, closed shell systems generally dissociate to give two radical fragments. In contrast, open-shell molecules can often dissociate to a radical and a closed-shell fragment, making this bond breaking process only mildly endothermic or even exothermic.

Experimental studies of free radical photodissociation are relatively uncommon, primarily because it is difficult to create sufficiently intense, contaminant-free sources of free radicals. To overcome this limitation, we have devised a general technique for the production of well-characterized radical samples using photodetachment of negative ion precursors. In this paper we report photodissociation spectroscopy and dynamics of the vinoxy radical, CH_2CHO. The vinoxy radical plays a key role combustion chemistry, specifically in the reactions of oxygen atoms with olefins.[3] CH_2CHO is a primary product of the reactions $O(^3P) + C_2H_4$,[4] $O(^3P) + C_2H_3$,[5] $OH + C_2H_2$,[6] and $OH + C_2H_4O$.[7] These reactions, and the CH_2CHO intermediate, are generally important since ethylene and acetylene are commonly present in combustion of larger aliphatic[8] and aromatic[9] compounds. It is therefore of considerable interest to understand the spectroscopy and dynamics of this species. This study of CH_2CHO is part of a systematic effort in our laboratory to determine how the dissociation dynamics of 'simple' alkoxy radicals depend on the nature of the carbon bonding in the radical. The results presented here show that the dissociation dynamics of CH_2CHO, the smallest alkenoxy radical, are quite different from those of the smallest alkanoxy and alkynoxy radicals, CH_3O [10] and HCCO.[11]

The vinoxy radical has been the subject of several spectroscopic[12-19] and electronic structure[20-24] investigations. The first spectroscopic detection of CH_2CHO, discrete bands with a probable origin at 28,780 cm^{-1}, was reported by Ramsay[12] in 1965. In 1981, Hunziker[13] *et. al.* reported two absorption bands of CH_2CHO: one band in the infrared originating at 8004 cm^{-1}, and a second in the ultraviolet, with its origin at 28,750 cm^{-1}. Electronic state assignments for these bands were aided by *ab initio* molecular orbital calculations by Dupuis *et. al.*,[21] who predicted an IR transition, $\tilde{A}\,(^2A') \leftarrow \tilde{X}\,(^2A'')$, at

6,900 cm^{-1}, and a UV transition, $\tilde{B}\,(^2A'') \leftarrow \tilde{X}\,(^2A'')$, at 26,000 cm^{-1}. The calculations predicted a planar geometry and an allylic resonance between the two valence bond structures CH_2=CH–O• and •CH_2–CH=O. The latter structure was favored as the ground electronic state, with an unpaired electron primarily on the terminal carbon atom. The $\tilde{A}$ and $\tilde{B}$ states were described as primarily CH_2=CH–O•, with the unpaired electron on the oxygen either in-plane or out-of-plane, respectively.

Lineberger and coworkers[19] established the electron affinity of CH_2CHO as 1.824 eV from negative ion photoelectron spectroscopy of CH_2CHO^-. Inoue and Akimoto[14] observed laser induced fluorescence (LIF) and dispersed fluorescence spectra of the $\tilde{B} \leftarrow \tilde{X}$ band. They assigned the vibrational modes of the radical using a triatomic model in which the three modes are the CO stretch, the CC stretch and the CCO bend. The first experimental information on the structure of CH_2CHO was given by DiMauro, Heaven, and Miller[15] from rotationally resolved LIF spectra of the $\tilde{B}\,(^2A'') \leftarrow \tilde{X}\,(^2A'')$ transition in a supersonic jet. Based on the relative intensities of the *a*-type vs. *b*-type rotational transitions (10:1) it was found that the transition dipole for the $\tilde{B} \leftarrow \tilde{X}$ transition points essentially along the CCO backbone. Their LIF spectrum refined the values of the three vibrational fundamentals in the $\tilde{B}$ state that were seen in Ref. 14 as ν_1(CO stretch) = 917 cm^{-1}, ν_2(CC stretch) = 1122 cm^{-1}, and ν_3(CCO bend) = 450 cm^{-1}. The microwave spectra of CH_2CHO and CD_2CDO due to Endo *et al.*[16] yielded a more definitive determination of the ground state geometry that was in good agreement with the calculation of Dupuis.[21] Later *ab initio* calculations by Yamaguchi *et al.*[23] showed that the triatomic vibrational model used by Inoue and DiMauro was insufficient to describe the vibrational structure of

vinoxy. Yamaguchi adopted a new numbering of the modes (which is used in this paper) based on all twelve vibrational degrees of freedom and reassigned the character of the 917 cm^{-1} fundamental as ν_8(CC stretch), and the 1122 cm^{-1} fundamental as ν_7(CH_2 rock).

Our interest in the vinoxy system was stimulated by the considerable differences between the laser-induced fluorescence and absorption spectra. The published LIF spectra[14,15] show no fluorescence above 30,200 cm^{-1} (i.e., only 1,400 cm^{-1} above the $\tilde{B}$ state origin), while the absorption spectrum[13] extends to 35,700 cm^{-1}, peaking at ≈32,500 cm^{-1}. The sharply truncated LIF spectrum implies that the excited state is depleted by a fast non-radiative process. The presence of such a process is confirmed by hole-burning spectroscopy on jet-cooled vinoxy, in which Gejo *et al.*[17] measured fluorescence depletion from 30,300 - 33,900 cm^{-1}, finding sharp vibronic absorptions in this region where fluorescence is quenched.

Three non-radiative processes could be responsible for fluorescence quenching: direct predissociation via a repulsive state, intersystem crossing (ISC) to a quartet state, and internal conversion (IC) to the ground electronic state. The latter two processes can also lead to photodissociation. There is relatively little experimental and theoretical information on the dynamics following excitation of the $\tilde{B}$ state. Yamaguchi recently reported calculations[24] on CH_2CHO that explored the energy dependence of the $\tilde{X}$, $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ ($^2A'$) states on the C-C torsional coordinate, finding that torsional motion may couple the $\tilde{B}$ state with the $\tilde{A}$ and/or $\tilde{C}$ states. Jacox found evidence for production of CH_3 + CO following UV irradiation of CH_2CHO in an Ar matrix,[25] but with a threshold near 34,500 cm^{-1}, much higher than the fluorescence cutoff energy. Previous attempts to

observe photodissociation products in the gas phase using LIF have been unsuccessful.[17,18]

Our experiment directly detects the presence of photodissociation, and identifies and characterizes the photodissociation products. We can therefore readily determine if dissociation occurs following excitation of the $\tilde{B}$ state and learn about the detailed mechanism. There are several product channels which are thermodynamically feasible following UV excitation of vinoxy:

$$CH_2CHO \xrightarrow{h\nu} CH_3 + CO \qquad \Delta H_0^0 = 0.08 \pm 0.09\ \text{eV}^{26,27} \qquad (1)$$

$$H + CH_2CO \qquad 1.49 \pm 0.09\ \text{eV} \qquad (2)$$

$$H_2 + HCCO \qquad 1.56 \pm 0.09\ \text{eV} \qquad (3)$$

$$C_2H_2 + OH \qquad 2.56 \pm 0.09\ \text{eV} \qquad (4)$$

$$CH_2 + HCO \qquad 4.16 \pm 0.13\ \text{eV} \qquad (5)$$

We observe photodissociation of vinoxy to both Channels 1 and 2 over the range $28{,}700 \leq h\nu \leq 33{,}330\ \text{cm}^{-1}$ ($3.56 \leq h\nu \leq 4.13$ eV). Channels 3, 4, and 5 are not observed in this study. Our results indicate that excitation of the $\tilde{B}$ state is followed by internal conversion, eventually populating the $\tilde{X}$ state, which then dissociates via Channels 1 and 2.

The experimental apparatus and data collection methods for the present experiments are summarized in Section II. In Section III we present the photofragment yield spectrum, translational energy, and angular distribution data for CH_2CHO and

CD_2CDO, in addition to *ab initio* molecular orbital calculations on this system. Section IV covers the analysis of the experimental data. Progressions in eight of the twelve vibrational modes are assigned with the aid of *ab initio* frequencies. The data analysis for Channel 2 is also presented, and we obtain the branching ratio between the two observed channels. In Section V, the overall dissociation mechanism is discussed. For CH_3 + CO, the translational energy distributions are compared with a hybrid statistical/impulsive dissociation model. RRKM rates for the two observed channels are calculated and compared with the experimental branching ratio. Finally, the mechanism of internal conversion from the $\tilde{B}$ state is discussed. A preliminary account of these experiments has been published previously.[28]

II. Experimental

A. The fast radical beam translational spectrometer

It is generally difficult to produce a well-characterized source of free radicals using traditional molecular beam sources due to the high reactivity of open-shell species. Our experimental approach takes advantage of this instability by recognizing that most free radicals have positive electron affinities and form stable negative ion analogues. Any contaminant ions present can be separated by mass spectrometry, after which the desired neutral free radicals are generated by laser photodetachment of the corresponding mass-selected anion. The packet of neutral radicals produced in the detachment step is then dissociated with a second laser, giving the overall scheme:

$$CH_2CHO^- \xrightarrow{h\nu_1,\ -e^-} CH_2CHO \xrightarrow{h\nu_2} \text{fragments} \qquad (6)$$

The apparatus constructed in our laboratory, the Fast Radical Beam Photofragment Translational Spectrometer, has been described previously in detail,[29,30,31] and only a brief description will be given here. The apparatus is divided into two distinct regions, as shown in Fig. 1. In the first section, a clean source of free radicals is generated from a negative ion precursor. The anion precursor, CH_2CHO^-, is formed in a 60 - 100 Hz repetition rate pulsed supersonic expansion of 1 torr of acetaldehyde (-78 °C) seeded in 3 atm of O_2. Acetaldehyde-d_4 is the precursor of CD_2CDO^-. A pulsed electric discharge[32] creates ions in the high pressure region before the expansion, allowing the ions to cool in the jet to rotational temperatures of typically 35 - 50 K. Molecular oxygen was used as a backing gas because it sustains a steady glow discharge at lower voltages than helium, neon, or argon, thereby keeping vibrational excitation of the anions to a minimum. The ions pass through a 3 mm skimmer, are accelerated to 8000 eV laboratory energy, and are mass selected using the Bakker time-of-flight method,[33] which imparts negligible energy spread to the ion beam. As the various ions separate according to their *m/e* ratio, a 25 ns light pulse from an excimer-pumped dye laser is fired at the appropriate time to photodetach only CH_2CHO^-. The detachment wavelength (663 nm for CH_2CHO^- and 667 nm for CD_2CDO^-) is only slightly above threshold,[19] producing radicals in their ground vibrational state. All remaining ions are deflected from the beam leaving an internally cold, high velocity beam of vinoxy radicals.

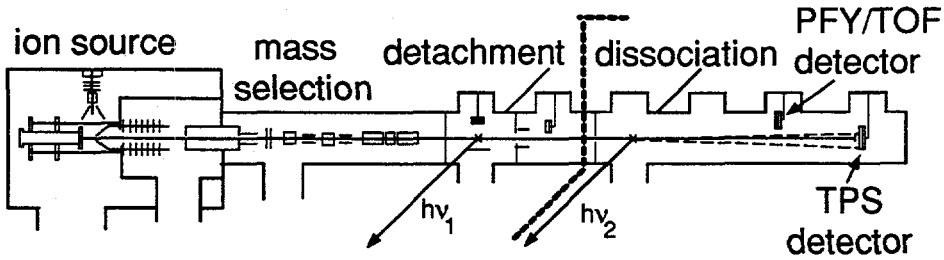


Figure 1: The experimental apparatus. The dotted lines separate the radical production region on the left from the dissociation region on the right.

In the second section of the apparatus, after a final 1 mm collimation, the vinoxy radicals are dissociated by a frequency-doubled excimer-pumped pulsed dye laser operating between 300 - 350 nm, with a bandwidth of 0.3 cm^{-1}. Neutral photofragments recoiling out of the parent radical beam are detected directly, without ionization, using one of two microchannel plate (MCP) detectors. The time-of-flight (TOF) and time- and position-sensitive (TPS) detectors (described below) are located on the radical beam axis 68 and 100 cm downstream from the dissociation laser port, respectively. An aluminum strip is placed across the center of each detector to prevent undissociated parent radicals from striking the MCPs. The fragments are detected with high efficiency (typically 50%) due to their high laboratory kinetic energy. However, the MCPs are considerably less sensitive to very light fragments, such as H or D atoms, because these fragments are formed with low laboratory kinetic energies.

B. Data collection and modes of operation

Three types of experiments are performed to characterize the photodissociation of CH_2CHO. First, the energy of the photodissociation laser is scanned and the total flux of fragments is detected giving a photofragment yield (PFY) spectrum. It should be emphasized that signal is observed *only* if the radical dissociates. This experiment uses the

TOF detector which can be lowered into the beam on a manipulator arm. The TOF detector has a 3 mm beam block and a simple metal anode which collects the total charge from the MCPs. The fragment signal is normalized to dissociation laser power, and also to the yield of electrons from the photodetachment of CH_2CHO^-, correcting for any variations in the radical production scheme. For vinoxy, a laser step size of 0.009 nm is used, corresponding to a step size in energy of 0.8 to 1.0 cm^{-1}. The data are calibrated at many different frequencies throughout the scanned range against the absorption spectrum of I_2,[34] with an absolute accuracy of 1 cm^{-1} or better.

In the second type of experiment, the dissociation laser is tuned to a specific photon energy of interest, and the dynamics of the dissociation process are studied. In this case, both fragments from a single parent radical are detected in coincidence using the TPS detector;[29,35] this is based on the detection scheme developed by de Bruijn and Los.[36] In this detector an 8 mm wide beam block is placed in front of a stack of three 46 mm diameter optical quality MCPs. The two electron clouds produced from the impact of the two fragments contain both spatial and temporal information describing where and when the fragments collided with the first MCP. The electron clouds are collected using a wedge and strip anode.[37,38] The spatial resolution (in the plane of the detector) is 100 μm while the relative timing resolution between the two fragment arrivals (perpendicular to the plane of the detector) is 500 ps.

From the timing and position information, we obtain the masses of the fragments, their relative translational energy E_T, and the polar angle θ between their relative velocity vector and the electric vector of the polarized dissociation laser (parallel to the ion beam axis) :

$$\frac{m_1}{m_2} = \frac{r_2}{r_1}\left(1 - \frac{v_0\tau}{L}\right) \tag{7}$$

$$E_T = E_0 \cdot \frac{m_1 m_2}{M} \cdot \frac{\left\{(v_0\tau)^2 + R^2\right\}}{L^2} \cdot \left(1 + 2\frac{m_1 - m_2}{M}\frac{v_0\tau}{L}\right) \tag{8}$$

$$\theta = \arctan\left(\frac{R}{v_0\tau}\right) \tag{9}$$

Here r_1 and r_2 are the distances from the center of the radical beam to each fragment on the detector face, $R = r_1 + r_2$ is the distance between the photofragments, and τ (= t_1-t_2) is the difference in the arrival times of the fragments. M, m_1, and m_2, are the masses of the parent and fragments respectively, v_0 and E_0 are the laboratory velocity and energy of the parent radical, and L is the distance between the dissociation laser and the detector face. Although the relative recoil $R = r_1 + r_2$ of the fragments is determined with high precision ($R/\Delta R \approx 100$) in the coincidence measurement, the individual recoils r_1 and r_2 are less precisely known, resulting in a fragment mass resolution $m/\Delta m \approx 15$.

Translational energy resolution of $\Delta E_T/ E_T = 0.6\%$ has been demonstrated under ideal conditions,[35] although in these experiments the resolution is somewhat coarser, typically $\Delta E_T/ E_T = 2.2\%$. The finite geometric acceptance of the detector is corrected by dividing the raw data by a detector acceptance function,[29] which gives the probability of observing a coincidence event at each combination of E_T and θ.

Due to the geometry of the detector, the coincidence detection scheme is only practical when the mass ratio of the fragments $m_1/m_2 \leq 5$. For mass ratios larger than 5,

there is very little probability of detecting both particles in coincidence; if the heavy fragment has enough energy to clear the beam block in front of the MCPs, the light fragment recoils past the edge of the detector. Channels 2 and 3 fall into this category where the mass disparity is too large for the coincidence experiment.

In order to detect Channels 2 and 3, a third type of experiment is performed that is sensitive to product channels even if $m_1/m_2 > 5$. This is a less detailed, non-coincidence experiment[31] in which the flight time of the fragments from the dissociation laser to the TOF detector is recorded using a transient digitizer with 1 ns bin width. The narrow beam block, (3 mm) on this detector allows improved collection of heavy fragments compared to the TPS detector described above. The resulting TOF distributions are a one-dimensional projection of the photofragment velocity distribution in the laboratory frame along the beam axis. These results are more averaged than the $P(E_T,\theta)$ distributions obtained with the TPS detector, but in return are sensitive to all product channels because two-particle coincidence detection is not required. This method is particularly useful for detecting light atom fragments such as H or D atoms, because these will have a much larger spread in arrival times at the detector and should thus be readily distinguishable from the heavier fragments. Pairs of TOF distributions are acquired at each photon energy with the dissociation laser polarized either parallel or perpendicular to the radical beam axis. The analysis of these data is discussed in Section IV.

III. Results

Our results from the three types of experiments are given in this section, followed by a description of the electronic structure calculations on the vinoxy system. Before presenting spectra of the vinoxy radical, we point out that the anion we photodetach at *m/e* 43 is in fact CH_2CHO^-, and not the isomer CH_3CO^-. While the two species are indistinguishable in the parent ion mass spectrum, they have very different photodetachment spectra. CH_2CHO^-, the more stable ion of the two, has an electron affinity (EA) of 1.824 eV,[19] while CH_3CO^- has an EA of 0.423 eV.[39] The photoelectron spectrum of *m/e* 43 ions measured on a different apparatus in our laboratory (using the identical ion source as in the photodissociation study) is in quantitative agreement with the literature spectrum of CH_2CHO^-.[19] Therefore we are confident that all photodissociation spectra reported here arise unambiguously from the CH_2CHO radical. Our photoelectron spectrum shows a small amount of vibrationally excited radicals produced from photodetachment of vibrationally excited anions, but their concentration is too low to substantially affect the results presented here.

A. Photofragment Yield Spectra

The PFY spectra obtained from excitation of the $\tilde{B}\,(^2A'') \leftarrow \tilde{X}\,(^2A'')$ transition are shown in Figs. 2b and 2c for CH_2CHO and CD_2CDO, respectively. The spectra show extended vibrational progressions in the excited electronic state with narrow linewidths, indicative of fragmentation by predissociation rather than by direct excitation to a repulsive potential energy surface. Peak positions and intensities for CH_2CHO and

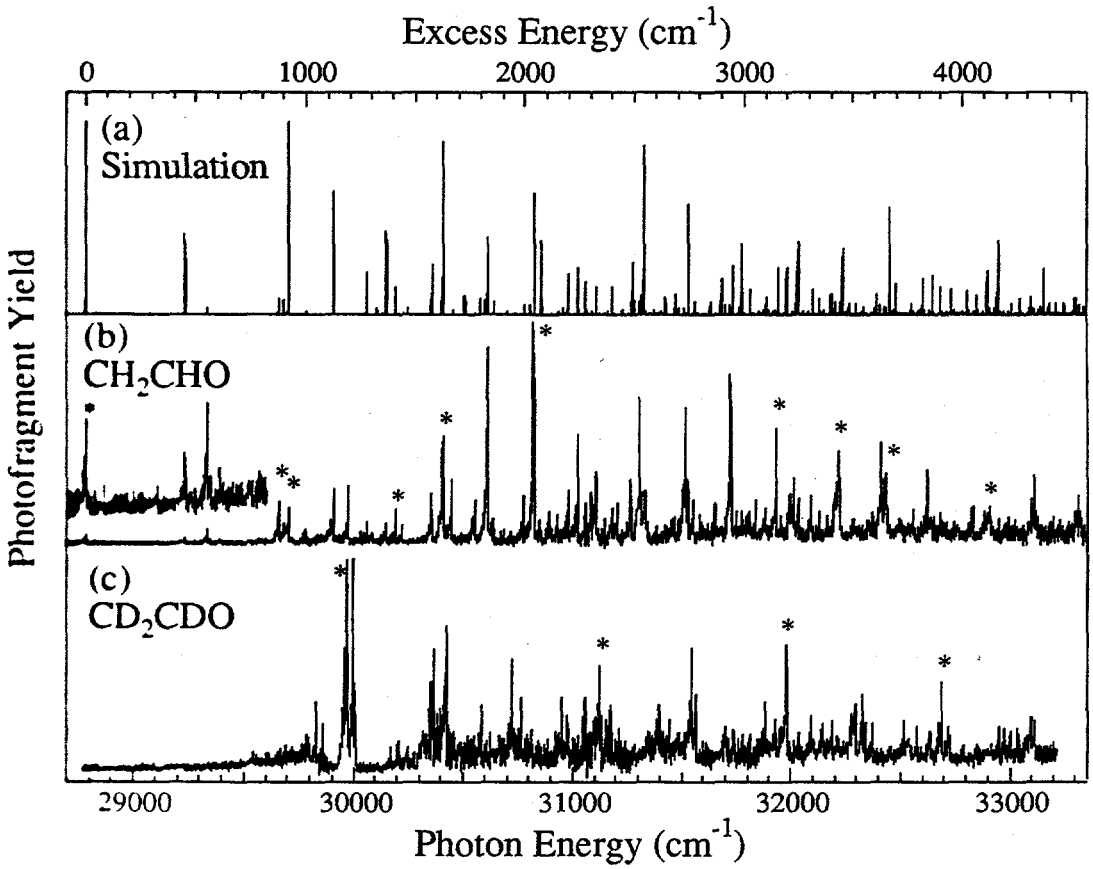


Figure 2: Photofragment yield spectra of the vinoxy radical. Energy in excess of the $\tilde{B} \leftarrow \tilde{X}$ 0_0^0 transition is given along the top abscissa. (a) simulation of CH_2CHO, (b) CH_2CHO experimental data, (c) CD_2CDO experimental data. Translational energy and angular distributions in Figs. 4 and 5 were acquired at the vibronic transitions marked with *.

CD_2CDO are given in Tables I and II. Our transition energies agree quantitatively with those of DiMauro *et. al.*[15], but disagree with the measurements of Inoue and Akimoto[14] by as much as 45 cm^{-1}.

The published LIF spectra[14,15] of CH_2CHO show features up to 1,400 cm^{-1} above the 0_0^0 transition. All these fluorescent levels are observed in our PFY spectra, indicating a competition between fluorescence and predissociation. The origin transition is the strongest line in the LIF spectrum,[15] but very weak in the PFY spectrum, indicating that the quantum yield for dissociation (Φ_{diss}) of 0_0^0 is small. Nonetheless, measurable dissociation occurs across the entire band, including the origin. The intensities in the PFY spectrum increase rapidly as the photon energy is raised, even though the Franck-Condon factors for absorption increase only slightly, indicating a steep rise in Φ_{diss}. Above 1,400 cm^{-1} of excess energy, where the LIF signal is strongly quenched, we expect that $\Phi_{diss} \cong 1$, and our overall intensities are in accord with low-resolution absorption[13] and high resolution hole-burning[17] data. Each vibrational transition in Fig. 2 is in fact composed of many rovibronic transitions. The sharp spike that dominates each vibrational transition represents a band head in the R-branch of the rotational contour, as first analyzed by DiMauro *et al.*[15] However, due to the higher rotational temperature in our supersonic expansion ($\approx$ 30 - 50 K) and limited laser resolution, it is not possible to discern whether lifetime broadening occurs in the PFY spectra above 1,400 cm^{-1}.

ν	ν - ν₀	*I*	assignment	ν	ν - ν₀	*I*	assignment	ν	ν - ν₀	*I*	assignment
28785.6	0.0	55	0_0^0	30822.3	2036.7	1000	$7_0^1\,8_0^1$	31931.0	3145.4	523	$7_0^2\,8_0^1$
29234.6	449.0	34	9_0^1	30850.2	2064.6	112	$4_0^1\,9_0^1$	31957.3	3171.7	131	
29333.3	547.7	66	11_0^2	30893.4	2107.8	156		31995.1	3209.5	235	
29659.1	873.5	195	10_0^2	30926.9	2141.3	140	$6_0^1\,10_0^2$	32002.8	3217.2	184	
29684.1	898.5	99	9_0^2	30976.7	2191.1	250	$6_0^1\,8_0^1$	32011.6	3226.0	302	
29703.2	917.6	166	8_0^1	31024.3	2238.7	496	7_0^2	32032.0	3246.4	213	
29781.0	995.4	65	$9_0^1\,11_0^2$	31058.9	2273.3	187	$8_0^2\,9_0^1$	32091.4	3305.8	227	
29908.2	1122.6	251	7_0^1	31084.5	2298.9	242		32127.6	3342.0	150	
29976.0	1190.4	265	$8_0^1\,11_0^1$?	31104.6	2319.0	311		32167.6	3382.0	130	
30030.7	1245.1	62		31111.0	2325.4	331		32200.1	3414.5	240	
30059.8	1274.2	107	6_0^1	31144.8	2359.2	117		32217.8	3432.2	428	
30085.4	1299.8	52		31181.5	2395.9	170		32413.2	3627.6	461	
30150.3	1364.7	97	$8_0^1\,9_0^1$	31205.3	2419.7	186		32426.2	3640.6	301	
30192.5	1406.9	158	5_0^1	31266.1	2480.5	296		32435.2	3649.6	323	
30221.8	1436.2	90		31307.7	2522.1	665	$4_0^1\,8_0^1$	32557.3	3771.7	157	
30355.1	1569.5	233	$7_0^1\,9_0^1$	31331.8	2546.2	248		32622.0	3836.4	338	
30407.4	1621.8	491	4_0^1	31519.4	2733.8	620	$4_0^1\,7_0^1$	32832.1	4046.5	173	
30448.4	1662.8	298	$7_0^1\,11_0^2$	31527.9	2742.3	295		32907.9	4122.3	169	
30546.0	1760.4	135		31553.2	2767.6	202		33111.1	4325.5	315	
30553.0	1767.4	201		31653.2	2867.6	165		33318.0	4532.4	228	
30612.3	1826.7	885	8_0^2	31725.2	2939.6	772	$7_0^1\,8_0^2$				
30633.6	1848.0	118	$5_0^1\,9_0^1$	31840.8	3055.2	205					
30773.9	1988.3	224	$7_0^1\,10_0^2$	31883.0	3097.4	166					

Table I: Peak positions (cm^{-1}), intensities, and vibrational assignments for the PFY spectrum of CH_2CHO, Fig. 2b.

For CD_2CDO no LIF spectrum has been published, although a spectrum was acquired by Inoue and Akimoto,[14] who reported only the positions of three peaks at 28,840, 29,608, and 29,820 cm^{-1}. The values for excess energy in the $\tilde{B}$ state given in Table II are calculated under the assumption that the first peak observed by Inoue is the 0_0^0 transition of CD_2CDO. In the PFY spectrum of CD_2CDO in Fig. 2c, the lowest transitions show no detectable signs of predissociation, and the shape of the overall spectrum, punctuated by the strong feature near 30,000 cm^{-1}, is quite different from the CH_2CHO data,. A gradual increase in the intensity of unstructured signal with photon energy is present in the spectra of both isotopes, but is much more pronounced for CD_2CDO. The unstructured component is not present in background scans taken with the photodissociation laser blocked, demonstrating that this broad signal is due to photofragments of the parent radical.

ν	$\nu - \nu_0$[a]	I	ν	$\nu - \nu_0$	I	ν	$\nu - \nu_0$	I
29828.4	988.4	320	30583.1	1743.1	310	32092.4	3252.4	264
29858.0	1018.0	230	30726.6	1886.6	523	32144.2	3304.2	228
29945.1	1105.1	293	30767.7	1927.7	343	32185.4	3345.4	244
29957.8	1117.8	575	30949.6	2109.6	344	32294.2	3454.2	315
29970.6	1130.6	987	30975.0	2135.0	264	32323.9	3483.9	361
29977.4	1137.4	216	31056.2	2216.2	344	32337.8	3497.8	234
29997.4	1157.4	1000	31122.0	2282.0	496	32371.5	3531.5	224
30006.8	1166.8	416	31397.0	2557.0	305	32504.1	3664.1	245
30173.5	1333.5	116	31546.8	2706.8	578	32573.0	3733.0	209
30200.2	1360.2	130	31559.0	2719.0	284	32635.2	3795.2	190
30207.1	1367.1	148	31568.4	2728.4	363	32689.8	3849.8	422
30247.7	1407.7	118	31882.5	3042.5	326	32719.2	3879.2	212
30369.7	1529.7	569	31927.6	3087.6	250	33096.3	4256.3	260
30423.0	1583.0	679	31979.6	3139.6	588	33111.9	4271.9	239

Table: II Peak positions (cm^{-1}) and intensities for the PFY spectrum of CD_2CDO, Fig. 2c.

[a] Energy in excess of the 0_0^0 transition measured by Inoue and Akimoto at 28,840 cm^{-1}.

B. Coincidence photofragment distributions: $CH_3 + CO$

Using the TPS detector, we perform coincidence measurements to probe for dissociation via Channels 1, 4, and 5. For this set of experiments, the photodissociation laser is fixed on one of the peaks marked in Fig. 2 and both fragments are detected in coincidence. The photofragment mass spectrum is easily found by conservation of linear momentum in the center of mass frame according to Eq. 7, and is presented in Fig. 3 for both CH_2CHO and CD_2CDO. Despite the limited fragment mass resolution of $m/\Delta m \approx 15$, the use of both isotopes of vinoxy demonstrates that the heavy fragment contains no hydrogen atoms. Therefore the only product channel observed *in coincidence* is Channel 1, $CH_3 + CO$. We see no evidence for production of C_2H_2 + OH or CH_2 + HCO. The results for Channel 2, H + CH_2CO, will be discussed in Section III.C.

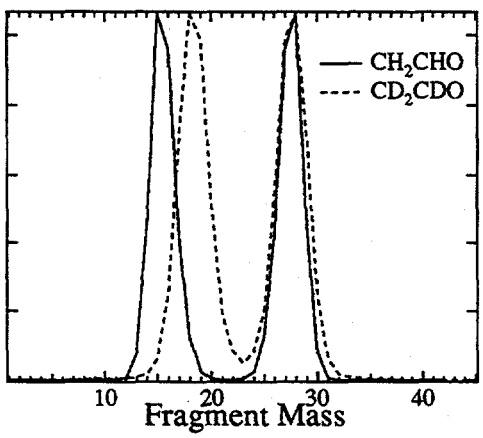


Figure 3: Photofragment mass spectrum of CH_2CHO and CD_2CDO. Note that the higher mass peak (m = 28) does not shift upon deuteration and must be assigned to CO. The cofragments are CH_3 and CD_3, respectively.

Once the identity of the fragments are known, the coupled translational energy and angular distribution, $P(E_T,\theta)$, is determined by direct inversion of the data using Eqs. 8 and 9 along with the calculated detector acceptance function.[29] The two dimensional $P(E_T,\theta)$ distribution can be separated into an angle-independent translational energy distribution $P(E_T)$, and an energy-dependent anisotropy parameter,[40] that describes the angular distribution of the fragments:

$$P(E_T,\theta) = P(E_T)\cdot\{1 + \beta(E_T)\cdot P_2(\cos\theta)\}. \tag{10}$$

The limiting cases of $\sin^2\theta$ and $\cos^2\theta$ angular distributions are given by β = -1 and +2, respectively. Together, the $P(E_T)$ and $\beta(E_T)$ distributions can yield detailed information on the dynamics of the dissociation process.

In analyzing the current experiment, E_T is binned in 30 meV increments, while θ is binned in 1° increments. The general data analysis scheme remains the same as in previous papers, with the exception of one refinement in the analysis of the angular distributions that yields more smoothly varying functions of the anisotropy parameter β with E_T. The data are divided into translational energy windows of width 0.4 eV, where E' is a running parameter such that $E' \leq E_T \leq (E' + 0.4\text{ eV})$. β is assumed to be constant over this 0.4 eV window and is determined from a linear least squares fit of the distribution $P(E_T,\theta)$ to Eq. 10 with $P(E_T)$ and β as the fitting parameters. The quantity E' is then incremented in 30 meV steps beginning with E' = 0 and the fitting procedure is repeated until the entire energy range of the data is covered. In other words, $\beta(E_T)$ is constructed as a moving average over 0.4 eV windows in E_T. Using these values of $\beta(E_T)$, a second linear least squares fit in the single parameter $P(E_T)$ is determined for each individual 30 meV wide energy bin, completing the separation of $P(E_T,\theta)$ into two one dimensional distributions, i.e., $P(E_T)$ and $\beta(E_T)$.

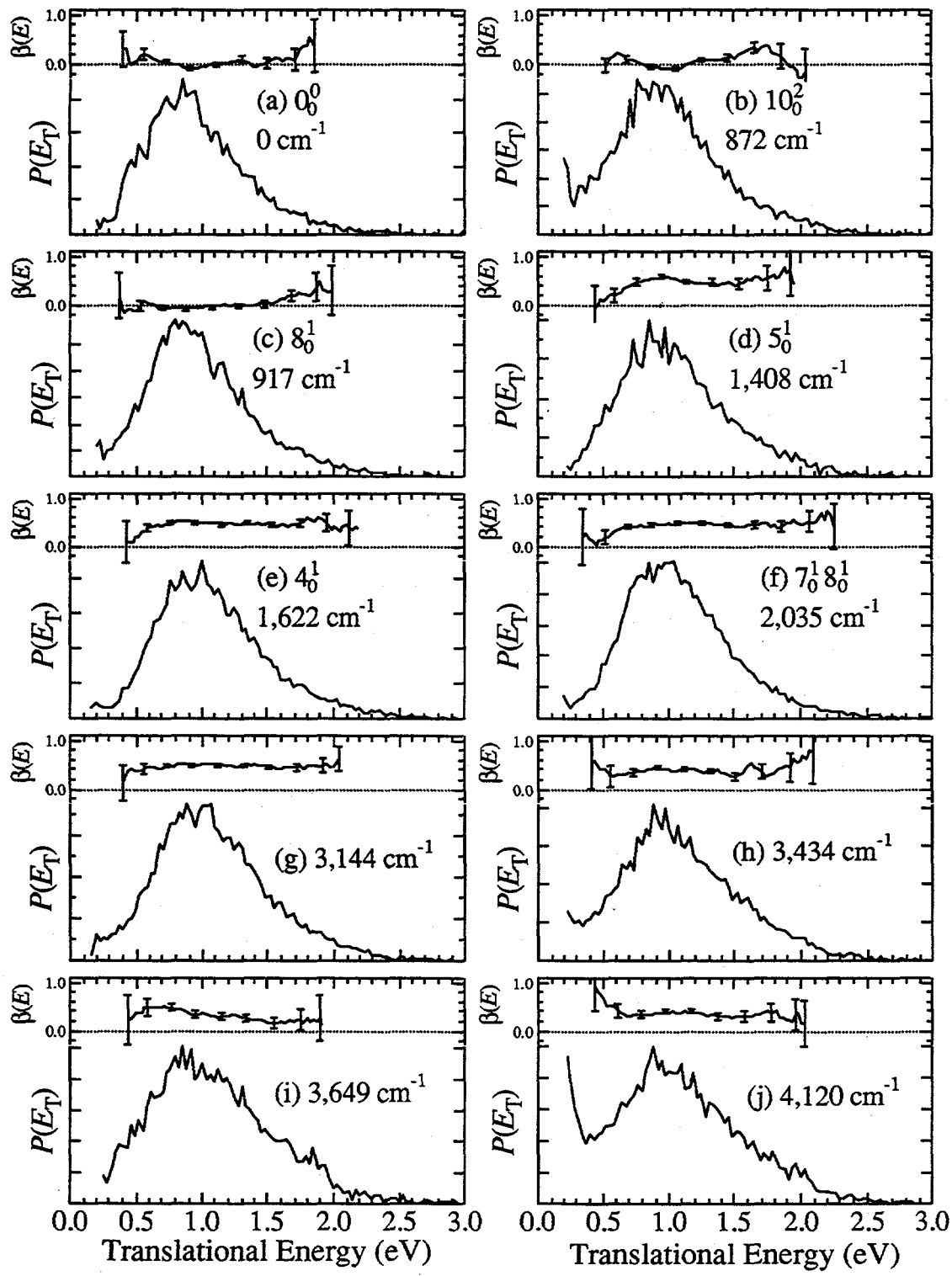


Figure 4: Photofragment translational energy distributions $P(E_T)$ and angular distributions $\beta(E_T)$ for CH_3 + CO. Statistical error bars on $\beta(E_T)$ represent $\pm 1\sigma$. Vibronic assignments are given in addition to the excess energy for each transition when possible.

The resulting $P(E_T)$ and $\beta(E_T)$ distributions for photodissociation of vinoxy yielding CH_3 + CO and CD_3 + CO are shown in Figs. 4 and 5, respectively, for the vibrational levels noted in Fig. 2. Each data set consists of 10,000 to 40,000 coincident events. The translational energy distributions are similar at all photon energies for both isomers, peaking at ≈ 0.9 eV. Both the width of the distributions and the most probable translational energy increase slightly as the photon energy is increased, but the changes are minor. The angular distributions are isotropic ($\beta \cong 0$) at the lowest three photon energies. However, anisotropic distributions with $\beta(E_T) > 0$ are observed at all energies 1,408 cm^{-1} and higher above the $\tilde{B}$ state origin. This apparently abrupt change coincides with the large drop in fluorescence quantum yield, and indicates that the dissociation rate increases markedly between 917 and 1,408 cm^{-1} excess energy.

C. Non-coincidence photofragment distributions

To determine whether the $\tilde{B}$ state of vinoxy also decays through Channels (2) and (3), non-coincident TOF distributions for CD_2CDO dissociation are obtained using the third type of experimental configuration described in Section II.B. The data are shown as circles in Fig. 6, and Fig. 7 shows one of these spectra in greater detail. The four photon energies chosen correspond to the four coincidence spectra in Fig. 5. TOF distributions are presented only for CD_2CDO, because the detection efficiency of the MCPs for D atoms is substantially greater than for H atoms due to the higher laboratory kinetic energy of the heavier isotope. Even for D atoms, the detection efficiency appears to be ≈ 8% (see Section IV.B) of that for heavy fragments such as CD_3 or CO.

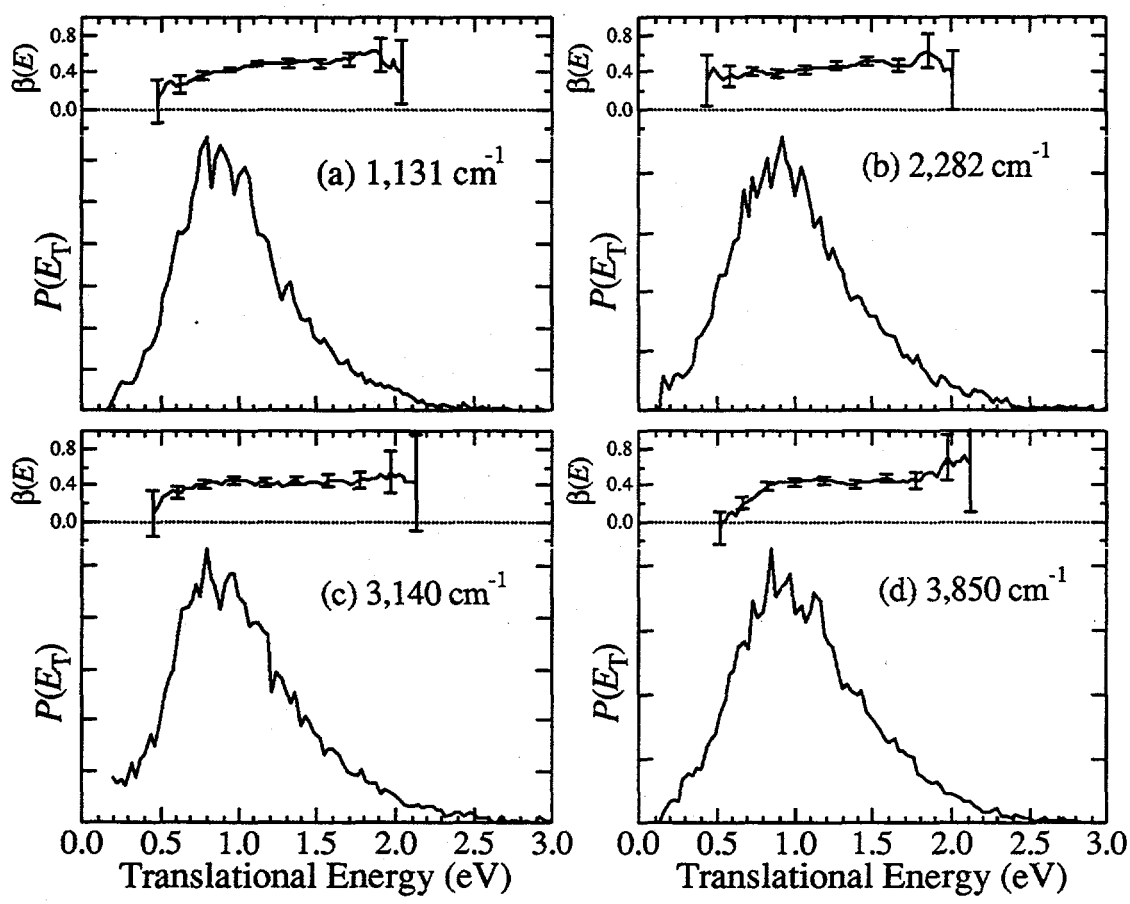


Figure 5: Photofragment translational energy distributions $P(E_T)$ and angular distributions $\beta(E_T)$ for CD_3 + CO. Statistical error bars on $\beta(E_T)$ represent $\pm 1\sigma$. Excess energy above the origin is referenced to Inoue and Akimoto's value[14] for 0_0^0 = 28,840 cm^{-1}.

The appearance of all the TOF spectra is roughly the same. Each consists of a sharp spike superimposed on a broader peak that is about 100 ns wide. There is also a very broad, weak feature from 3.5 to 3.9 μs, better seen at 20x magnification, which has a FWHM of ≈10 times that of the sharp spike. The sharp spike is only slightly broader than the TOF distribution of the undissociated parent radicals (measured by moving the TOF detector off-center by several mm; see Fig. 6f, implying that these fragments have very little velocity imparted to them by the dissociation process. The broad, weak feature arises from fragments which are significantly forward/backward scattered by the

dissociation event. It is therefore a reasonable conclusion that these two features represent the heavy and light fragments, respectively, of a dissociation channel whose mass ratio is at least 10:1. The analysis in Section IV.B shows that these two features are from the D + CD_2CO channel, and that the 100 ns wide peak under the spike is from the CD_3 + CO channel.

D. *Ab initio* electronic structure calculations

To supplement the experimental data and aid in interpreting our results, we have performed *ab initio* molecular orbital calculations using the Gaussian 92[41] and Gaussian 94[42] suite of programs. Our main goal is to characterize stationary points on both ground and excited state potential energy surfaces. The calculated energies, structures, and vibrational frequencies at various minima and transition states are very useful in assigning the PFY spectra in Fig. 2 and understanding the photodissociation dynamics of the system. The absolute and relative energetics of the various minima, transition states, and asymptotic states relevant to the vinoxy system are shown in Fig. 8 and tabulated in Table III, while geometries and vibrational frequencies are listed in Tables IV and V. Fig. 8 is a schematic representation of the surfaces involved in this system, sections of which have been previously investigated by Donaldson *et al.*[5] and Deshmukh *et al.*[43] The zero of energy is chosen to be the zero-point level of the CH_2CHO $\tilde{X}$ ($^2A''$) potential well. The only experimentally determined barrier height is that for dissociation of acetyl (CH_3CO) to CH_3 + CO, (TS3), found to be 0.75 ± 0.02 eV [44,45]. This agrees well with our calculated barrier height 0.44 + 0.31 = 0.75 eV, with respect to the CH_3CO well.

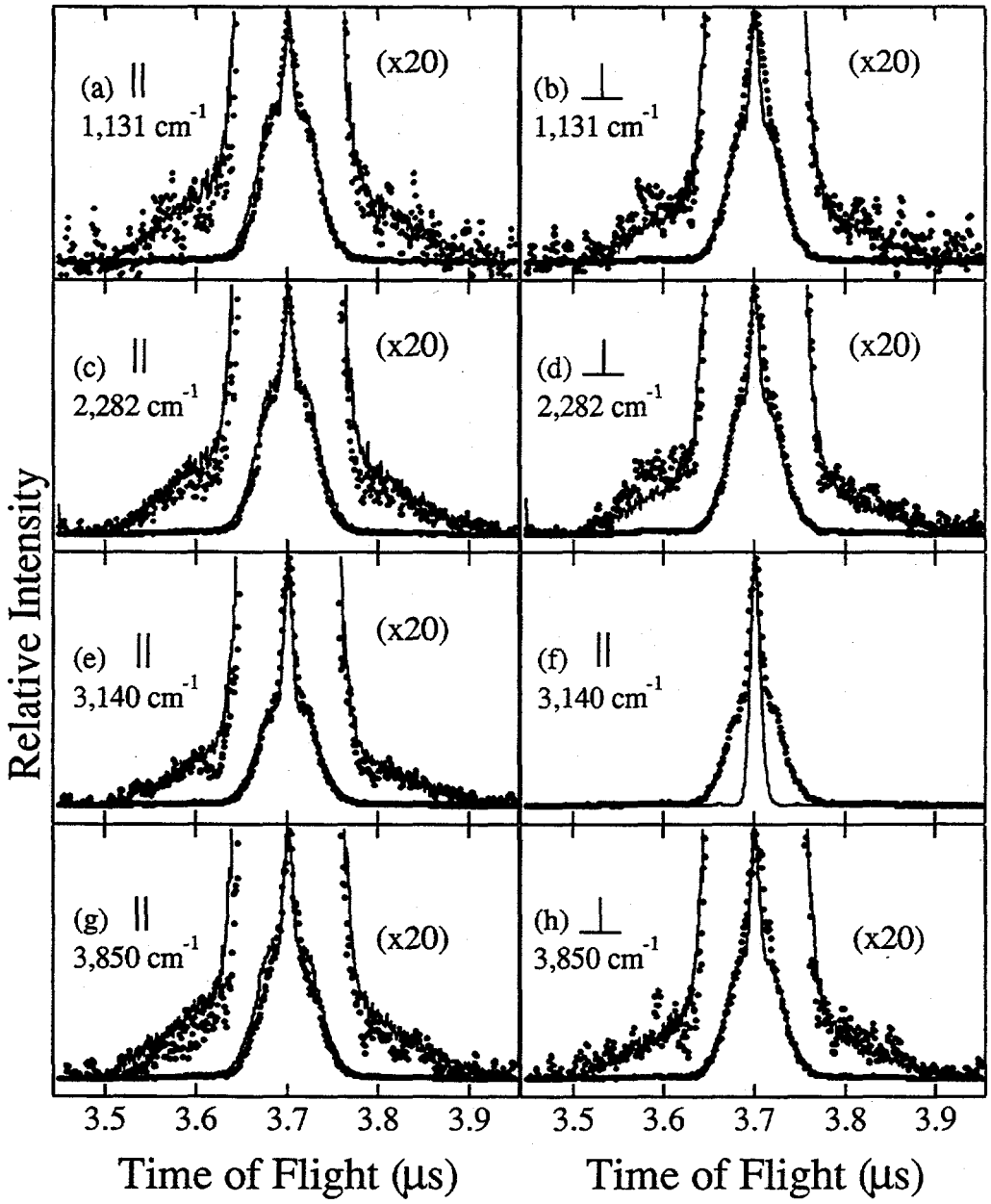


Figure 6: Photofragment time-of-flight distributions for all photon energies used in Fig. 5. Laser polarization is either parallel (| |) or perpendicular (⊥) to the radical beam axis. Circles are the experimental data. For all panels except (f), the solid lines give total simulation. In panel (f), the solid line is the experimental TOF distribution of parent radicals (CD_2CDO) taken with the detector displaced 10 mm vertically.

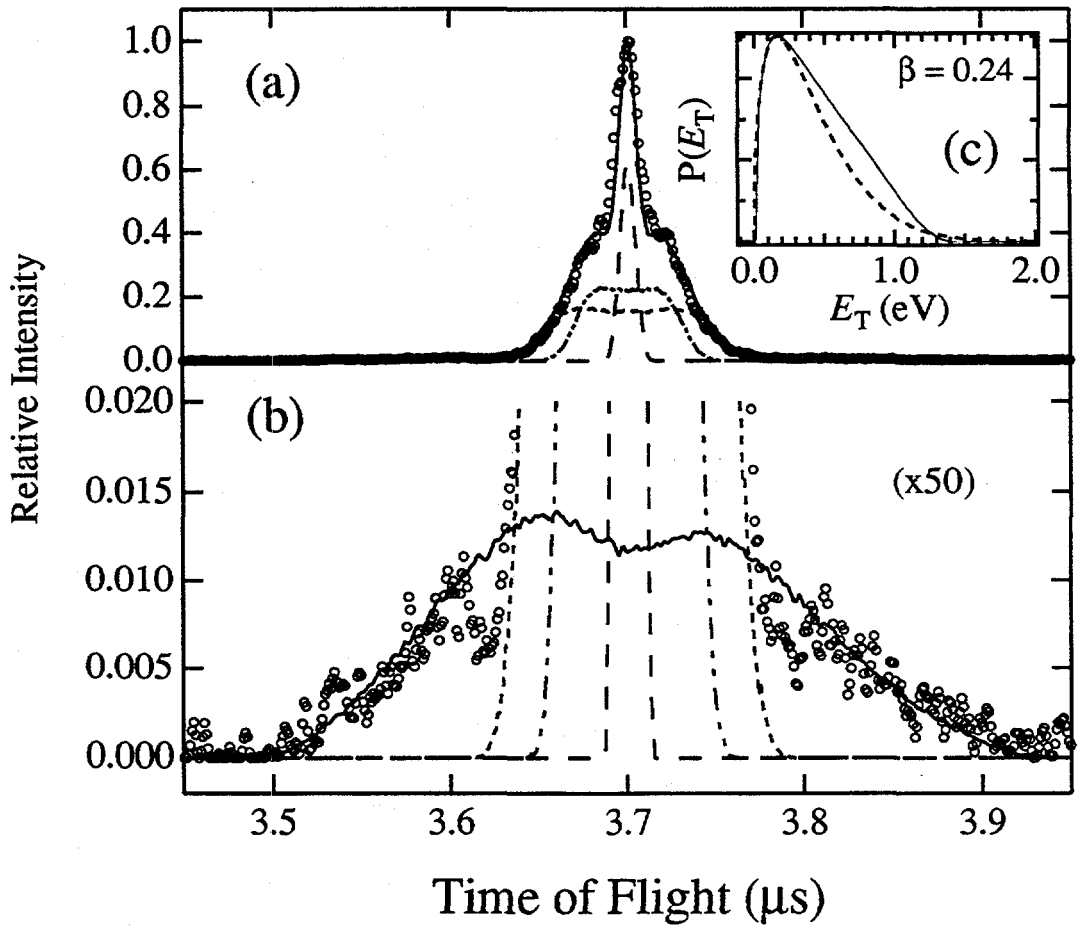


Figure 7: CD_2CDO TOF distribution for 3,140 cm^{-1} excess energy, showing individual fragment contributions. Experimental data (°), CD_2CO (— —), CO (— - - —), CD_3 (- - -). (a) sum of all contributions (——); (b) Magnification to show D atom contribution (——); (c) P(E_T) distribution (——) for this fit, with prior distribution also shown for comparison (- - -).

We have used the G2[46] and G2Q[47] methodologies for the ground state surface. These methods have been compared extensively with experimental thermodynamic values and give accurate (±0.1 eV) estimations of dissociation energies. Because the single determinant wavefunctions in the G2 treatment do not work well for excited states, we have also performed multi-configurational complete active space self consistent field (CASSCF) calculations[48] for the $\tilde{X}$, $\tilde{A}$, and $\tilde{B}$ electronic states of vinoxy with the

6-31G** basis set. Building on earlier work[21,23,24], we choose an active space which consists of three electrons in the three a'' orbitals [CAS(3,3)] for the $\tilde{X}$ and $\tilde{B}$ states, while for the $\tilde{A}$ state the a'' lone pair on oxygen is excluded and the singly occupied a' orbital on oxygen is included in the active space. The agreement with earlier calculations[21,23,24] is very good with respect to energies, geometries, and frequencies. We refine the CASSCF energetics using single point CASSCF-MP2[49] to include dynamic electron correlation.

Method	Species	Relative Energy[a]
G2 / G2Q	CH_2CHO $\tilde{X}$ ($^2A''$)	0.0
	CH_3CO $\tilde{X}$ ($^2A'$)	-0.31
	TS 1	1.78
	TS 2	1.74
	TS 3	0.44
	$CH_3 + CO$	0.04
	$H + CH_2CO$ $\tilde{X}$ ($^1A_1$)	1.50
	$H + CH_2CO$ $\tilde{a}$ ($^3A''$)	3.90
	$C_2H_2 + OH$	2.57
	$CH_2 + HCO$	4.26
CASSCF(3,3)-MP2 /6-31G**	CH_2CHO $\tilde{X}$ ($^2A''$)	0.0
	CH_2CHO $\tilde{A}$ ($^2A'$)	0.73
	CH_2CHO $\tilde{B}$ ($^2A''$)	3.52
CASSCF(5,5)-MP2 /6-31G**	CH_2CHO $\tilde{X}$ ($^2A''$)	0.0
	CH_2CHO $\tilde{A}$ ($^2A'$)	0.79
	TS 4	2.09
Experiment	CH_2CHO $\tilde{A}$ ($^2A'$)	0.99[b]
	CH_2CHO $\tilde{B}$ ($^2A''$)	3.569[c]

Table III: Ab Initio energies of stationary points in Fig. 8.

[a] Total energy relative to the zero-point level of CH2CHO, in eV.
[b] Ref. 13
[c] Ref. 15

$CH_2CHO\ \tilde{X}$ [a,b]	$CH_2CHO\ \tilde{A}$ [c]	CH_3CO [a,b]	TS1 [d]	TS2 [d]	TS3 [d]	TS4 [c]	TS5 [e]
			GEOMETRIES				
$r(C_aC_b) = 1.456$	$r(C_aC_b) = 1.333$	$r(C_aC_b) = 1.513$	$r(C_aC_b) = 1.410$	$r(C_aC_b) = 1.340$	$r(C_aC_b) = 2.113$	$r(C_aC_b) = 1.324$	
$r(C_aO) = 1.200$	$r(C_aO) = 1.348$	$r(C_aO) = 1.196$	$r(C_aO) = 1.204$	$r(C_aO) = 1.163$	$r(C_aO) = 1.147$	$r(C_aO) = 1.195$	
$r(C_bH_c) = 1.082$	$r(C_bH_c) = 1.074$	$r(C_bH_c) = 1.092$	$r(C_bH_c) = 1.094$	$r(C_bH_c) = 1.083$	$r(C_bH_c) = 1.087$	$r(C_bH_c) = 1.068$	
$r(C_bH_b) = 1.082$	$r(C_bH_b) = 1.072$	$r(C_bH_b) = 1.092$	$r(C_bH_b) = 1.083$	$r(C_bH_b) = 1.083$	$r(C_bH_b) = 1.087$	$r(C_bH_b) = 1.073$	
$r(C_aH_a) = 1.110$	$r(C_aH_a) = 1.098$	$r(C_bH_a) = 1.093$	$r(C_bH_a) = 1.481$	$r(C_bH_a) = 1.888$	$r(C_bH_a) = 1.085$	$r(C_aH_a) = 1.66$	
$\angle(C_bC_aO) = 123.6$	$\angle(C_bC_aO) = 124.7$	$\angle(C_bC_aO) = 127.5$	$\angle(C_bC_aO) = 139.3$	$\angle(C_bC_aO) = 172.1$	$\angle(C_bC_aO) = 115.1$	$\angle(C_bC_aO) = 157.5$	
$\angle(H_cC_bC_a) = 118.7$	$\angle(H_cC_bC_a) = 121.8$	$\angle(H_cC_bC_a) = 108.6$	$\angle(H_cC_bC_a) = 122.0$	$\angle(H_cC_bC_a) = 116.7$	$\angle(H_cC_bC_a) = 99.6$	$\angle(H_cC_bC_a) = 121.2$	
$\angle(H_bC_bC_a) = 121.3$	$\angle(H_bC_bC_a) = 119.6$	$\angle(H_bC_bC_a) = 108.6$	$\angle(H_bC_bC_a) = 117.8$	$\angle(H_bC_bC_a) = 116.7$	$\angle(H_bC_bC_a) = 99.6$	$\angle(H_bC_bC_a) = 118.3$	
$\angle(H_aC_aO) = 121.5$	$\angle(H_aC_aO) = 111.4$	$\angle(H_aC_bC_a) = 111.1$	$\angle(H_aC_bC_a) = 52.6$	$\angle(H_aC_bC_a) = 111.3$	$\angle(H_aC_bC_a) = 103.1$	$\angle(H_aC_aO) = 98.5$	
		$\angle(H_cC_bC_aO) = 121.6$	$\angle(H_cC_bC_aO) = 117.0$	$\angle(H_cC_bC_aO) = 105.0$	$\angle(H_cC_bC_aO) = 59.5$		
		$\angle(H_bC_bC_aO) = -121.6$	$\angle(H_bC_bC_aO) = -68.5$	$\angle(H_bC_bC_aO) = -105.0$	$\angle(H_bC_bC_aO) = -59.5$		
		$\angle(H_aC_bC_aO) = 0.0$f	$\angle(H_aC_bC_aO) = 0.0$	$\angle(H_aC_bC_aO) = 0.0$ [f]	$\angle(H_aC_bC_aO) = 180.0$ [f]		
			FREQUENCIES				
3081	3431	2950	3276 (V) [g]	3304	3276	3465	3100
2985	3336	2946	3098 (V)	3202	3260	3349	3100
2901	3107	2871	1966 (V)	2194	3105	2026	1700
1441	1773	1913	1846 (V)	1425	2044	1584	1450
1392	1550	1433	1465 (V)	1118	1453	1190	1300
1267	1352	1432	1229 (T)	1033	1446	1100	1200
1089	1183	1356	1126 (T)	799	939	714	1150
947	1025	1037	1036 (V)	571	611	592	700
910	956	938	836 (R)	567	547	542	700
704	864	827	633 (V)	426	277	432	700
482	714	454	438 (R)	309	43	169	400
474	484	85	1675*i* (T)	801*i*	466*i*	1504*i*	

Table IV: *Ab Initio* geometries and frequencies of stationary points in Fig. 8.

[a] Geometry from G2 method: MP2/6-31G*.
[b] Frequencies from G2 method: UHF/6-31G* scaled by 0.8929.
[c] Geometry and frequencies are CASSCF(5,5)/6-31G**.
[d] Geometry and frequencies from G2Q method: QCISD/6-311G**.
[e] Frequencies estimated from Ref. 65.
[f] fixed
[g] Motion into which vibrations develop asymptotically. See Section V.B.2.

The direct bond cleavage $CH_2CHO \rightarrow H(^2S) + CH_2CO$ requires a CASSCF wavefunction to properly describe the spin coupling, since the ground state of vinoxy in C_s symmetry correlates with CH_2CO $\tilde{a}$ ($^3A''$), while the $\tilde{A}$ state of vinoxy correlates with CH_2CO $\tilde{X}$ ($^1A_1$). For dissociation along the reaction coordinate leading to triplet ketene, five electrons are distributed among five orbitals [CAS(5,5)], which are chosen to be the C–H_a σ-bond (10*a′*), the C–O π-bond (1*a″*), the unpaired electron on the methylene C (2*a″*), the C–O π* orbital (3*a″*), and the C–H_a σ* orbital (11*a′*). This active space contains the most important configurations needed to describe the breaking of the C–H_a bond along with the resonance character of the CCO backbone as described by Dupuis *et al.*[21] The C–H_a cleavage of CH_2CHO $\tilde{A}$ ($^2A'$) to singlet ketene is investigated by a similar extension of the active space. Again, the energetics are refined using CASSCF-MP2.

A comparison of calculated and experimental excited state energies in Table III shows that the CASSCF-MP2 treatment gives a fairly good description of the vinoxy energetics. However, Table V shows that the CASSCF vibrational frequencies are systematically too high compared to experiment, and that the calculated C-C and C-O bond lengths for the $\tilde{X}$ ($^2A''$) state differ considerably from the experimental values. A CASSCF calculation of the geometry by Yamaguchi[24] using a larger (DZP) basis set yields results similar to ours, indicating that the good agreement found by Dupuis[21] between the experimental and *ab initio* geometry of the $\tilde{X}$ ($^2A''$) state (using a smaller 3-21G basis) was probably fortuitous. The disagreement arises mainly because CASSCF neglects dynamic electron correlation. To obtain a benchmark for the CASSCF error in the excited state geometries and frequencies, we have performed a QCISD[50]/6-31G** geometry

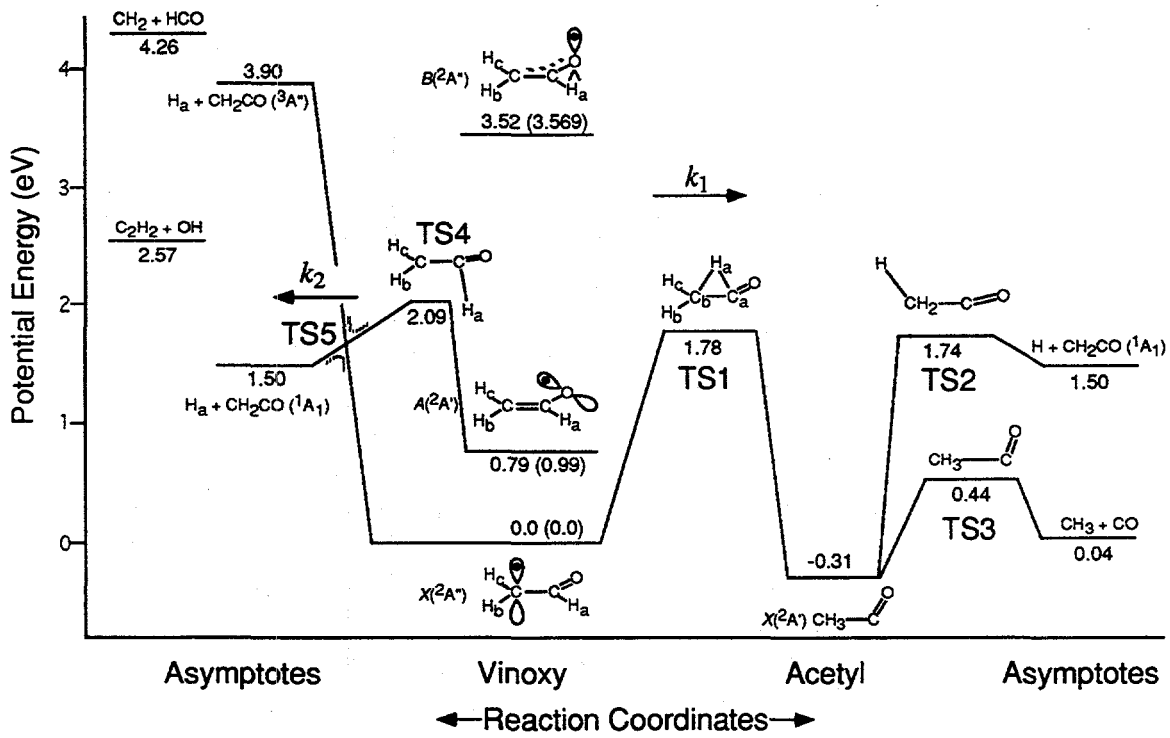


Figure 8: Schematic of potential energy surfaces of the vinoxy radical system. All energies are in eV, include zero-point energy, and are relative to CH_2CHO ′ ($^2A''$). Calculated energies are compared, when possible, with experimentally determined values in parentheses. Transition states 1-5 are labeled, along with the rate constant definitions from RRKM calculations. The solid potential curves to the left of vinoxy retain C_s symmetry. The avoided crossing (dotted lines) which forms TS5 arises when C_s symmetry is broken by out-of-plane motion.

optimization and frequency calculation on CH_2CHO $\tilde{X}$ ($^2A''$). From this comparison we have an estimate of the error in the calculated CASSCF vibrational frequencies for the $\tilde{B}$ state, which cannot be calculated using QCISD.

Method	$\tilde{X}$ (^{2}A″) CAS(3,3)	$\tilde{X}$ (^{2}A″) QCISD	experiment	$\tilde{B}$ (^{2}A″) CAS(3,3)	experiment
$r(C_aC_b)$ [a]	1.436	1.434	1.408[b]	1.450	1.466[c,d]
$r(C_aO)$	1.224	1.241	1.261[b]	1.360	1.337
$r(C_bH_c)$	1.073	1.080	1.081[d]	1.072	1.069[d]
$r(C_bH_b)$	1.073	1.081	1.080[d]	1.071	1.069[d]
$r(C_aH_a)$	1.089	1.100	1.088[d]	1.075	1.069[d]
$\angle(C_bC_aO)$	122.6	122.6	122.4[b]	121.5	129.5
$\angle(H_cC_bC_a)$	119.4	119.2		120.1	
$\angle(H_bC_bC_a)$	121.0	121.2		119.9	
$\angle(H_aC_aO)$	119.7	120.4		116.8	
$\nu_1(C_bH_cH_b$ asym. str.) a'	3447	3353		3460	
$\nu_2(C_bH_cH_b$ sym. str.) a'	3333	3233		3348	
$\nu_3(C_aH_a$ stretch) a'	3207	3057		3362	
$\nu_4(C_aO$ stretch) a'	1666	1593	1540[e]	1881	1623[f]
$\nu_5(C_bH_cH_b$ scissors) a'	1598	1522	1454	1561	1407
$\nu_6(OC_aH_a$ bend) a'	1519	1439	1361	1438	1274
$\nu_7(C_bH_cH_b$ rock) a'	1216	1179	1141	1212	1123
$\nu_8(C_aC_b$ stretch) a'	1036	993	949	998	918
$\nu_9(C_bC_aO$ bend) a'	537	507	498	480	449
$\nu_{10}(C_aH_a$ wag) a''	981	977		453	437
$\nu_{11}(C_aC_b$ torsion) a''	467	446		245	274
$\nu_{12}(C_bH_cH_b$ wag) a''	666	713		267	

Table V: Comparison of CASSCF/6-31G**, QCISD/6-31G**, and experimental geometries and frequencies.[g]

The results given in Table V demonstrate that the bond lengths are in better agreement with the microwave values than previous multi-configurational studies. The agreement is not yet quantitative, as the QCISD C–C bond length is 0.026 Å longer and

[a]See Fig. 8 for atom label definitions.
[b]Ref. 16.
[c]Ref. 15.
[d]Fixed.
[e]Ref. 15, AIP-document No. PAPS JCPSA 81-2339-8 (all frequencies in column).
[f]Current work (all frequencies in column).
[g]Polarization functions on H with CAS(3,3) accomplish a similar lowering of the a″ vibrational frequenices found from a larger CAS in Ref. 24.

the C–O bond length 0.020 Å shorter than experiment, respectively. Compared to the CASSCF frequencies, the QCISD frequencies are consistently lower for the 9 a' modes and roughly equal for the 3 a'' modes, and are therefore in improved agreement with experimental values from dispersed fluorescence.[15] The systematic lowering of frequencies and improved geometry is evidence that QCISD more accurately describes the electronic structure of CH_2CHO $\tilde{X}$ ($^2A''$). Presumably, basis set extension for QCISD would further improve the comparison of theory to experiment. The comparison between QCISD and CASSCF for the ground state suggests that the CASSCF frequencies for the $\tilde{B}$ state are also too high. This will be useful in interpreting the photofragment yield spectrum in Section IV.A.

IV. Analysis

In this section we analyze the vibrational structure in the CH_2CHO photofragment yield spectrum and present Monte Carlo simulations of non-coincident photofragment TOF distributions.

A. Photofragment yield spectrum

The vinoxy radical has 12 vibrational modes which have been assigned principal characters by Yamaguchi[23] as given in Table V: 3 out-of-plane a'' modes, 6 in-plane a' modes involving the heavy atoms, and 3 high frequency a' modes which are primarily hydrogen stretches. The large number of totally symmetric modes makes the assignment of the vibrational structure in the PFY spectrum nontrivial. We follow a systematic procedure, simulating the spectrum with the aid of *ab initio* frequencies. Starting with the

lowest peaks in the spectrum of Fig. 2b, the fundamental vibrations are assigned to a mode, the spectrum is simulated using approximate Franck-Condon factors (FCF) from Yamaguchi,[23] the vibrational constants are refined, and certain peaks at higher energy in the experimental spectrum are assigned as combination bands. Higher frequency vibrational modes are then added, one by one, and the process is repeated until the majority of the peaks in the spectrum are predicted by the simulation.

The simulated CH_2CHO spectrum is shown in Fig. 2a, the peak assignments are listed in Table I, and the frequencies for the $\tilde{B}$ state obtained from our analysis are listed in Table V. Since the intensities in the simulation mimic the FCFs for absorption, the predicted intensity for several of the lowest energy transitions is too great. The photodissociation yield Φ_{diss} for these transitions is small, so they are not as intense in the PFY spectrum as in the absorption and LIF spectra. In the region where no fluorescence is observed, the absorption and photofragment yield intensities should be identical, since essentially all the excited molecules dissociate rather than fluoresce.

Using this iterative method we assign progressions in all 6 of the heavy atom a' modes, ν_4 - ν_9, and 2 of the a'' modes, ν_{10} and ν_{11}. The vinoxy radical is planar in both the $\tilde{X}$ and $\tilde{B}$ states,[15] so that (in the absence of vibronic coupling), excitations in non-totally symmetric a'' vibrations are allowed only for $\Delta\nu = 0, \pm 2, \pm 4\ldots$ Due to the large change in *ab initio* frequencies of the a'' modes between the $\tilde{X}$ and $\tilde{B}$ states, it is not surprising that these modes are active. Our vibrational assignments agree with those of Gejo *et al.*[17] except for their assignment of ν_6 = 1,406 cm^{-1} and ν_5 = 1,433 cm^{-1}, which contrast with our assignments of ν_6 = 1,274 cm^{-1} and ν_5 = 1,406 cm^{-1}. Our assignment of the ν_6

frequency seems to be more in line with the CASSCF *ab initio* frequencies (Table V) which are consistently 7 - 11% higher than experiment for the $\tilde{X}$ state. Overall we find a more consistent match of the simulated and experimental PFY spectrum with our assignments. Note that not all of the experimental peaks listed in Table I are assigned, and that the assignment of the peak at 29,976 cm^{-1} to the $8_0^1 11_0^1$ transition requires vibronic coupling that presumably involves the $\tilde{A}$ (^{2}A$'$) state.

A comparison of the photofragment yield spectrum of CH_2CHO in Fig. 2b with the simulation in Fig. 2a and the LIF spectrum[15] shows mode specific competition between fluorescence and dissociation for the lower vibrational levels of the $\tilde{B}$ state, with excitation of a'' vibrational modes resulting in higher Φ_{diss} than nearby levels in which only a' modes are excited. For example, excitation of the 11_0^2 and 10_0^2 transitions (a'') at excess energies of 547.7 and 873.5 cm^{-1}, respectively, results in a significantly higher Φ_{diss} than the nearly isoenergetic 9_0^1 and 8_0^1 transitions (a') at 449.0 and 917.6 cm^{-1}. The implications of this effect will be discussed further in Section V.E.

The photofragment yield spectrum of CD_2CDO in Fig. 2c appears substantially different than that of CH_2CHO. The differences between these spectra arise for two reasons. First, the vibrational levels in the $\tilde{B}$ state are shifted to lower frequency upon deuteration; their *ab initio* values have been calculated by Yamaguchi.[23] Therefore deuteration will change both the positions and the FCFs of the $\tilde{B}$ (^{2}A$''$) $\leftarrow$ $\tilde{X}$ (^{2}A$''$) absorption spectrum in CD_2CDO. Secondly, the quantum yield for dissociation as a function of photon energy is different than that of CH_2CHO. In particular, we cannot detect any resolved dissociation signal for photon energies less than 29,828 cm^{-1}, implying

that the dissociation rate at low excess energies is slower than that for CH_2CHO. However, the intense cluster of six peaks around 29,975 cm^{-1} implies a substantial increase in the dissociation rate ≈1,100 cm^{-1} above the origin level. This energy is similar in magnitude to the energy at which fluorescence is quenched and the photofragment angular distributions become anisotropic in CH_2CHO. Due to the lack of structure in the first 1,000 cm-1 of the PFY spectrum of CD_2CDO, we have not attempted to assign the vibrational modes in this spectrum. The data from a full LIF spectrum of the deuterated species would make assignments more tractable.

B. Monte Carlo Forward Convolution for CD_2CDO TOF distributions

The non-coincidence TOF distributions in Fig. 6 are due to *all* the fragments produced in the dissociation of the vinoxy radical. A direct inversion of this data to give the translational energy and angular distributions for all product channels is clearly not feasible. However, because the energy and angular distributions of Channel 1 (CD_3 + CO) have already been determined, any new features in the observed non-coincidence TOF distributions must be due to a second product channel. We can extract information on the dynamics of this channel by employing a forward convolution technique to simulate the experimental TOF data. In this analysis, a set of product channels, each with its own $P(E_T)$ distribution and anisotropy parameter β, serves as inputs to a Monte Carlo forward convolution program[29] which generates the 1-D TOF distributions that would be observed in our experiment based on the apparatus geometry and finite spatial acceptance of the detector. By comparing the predicted TOF distributions with the experimental

distributions, the initial estimate for channels other than CD_3 + CO can be refined as needed until the simulated and observed TOF distributions agree.

For Channel 1, the $P(E_T)$ distributions in Fig. 5 are used, and $\beta = 0.4$ is taken as the energy-independent value of the anisotropy parameter; this is the average value of β at the excitation energies where TOF measurements were taken. The contribution of CD_3 and CO to the total TOF is shown in Fig. 7a. The remaining features in the TOF data are the narrow spike in the middle of the distribution and the broad wings at the edges of the spectrum. Only Channels 2 and 3, CD_2CO + D and DCCO + D_2, are consistent with these features. The thermodynamics for the two channels differ by only 0.06 eV,[11,26,27] making them both energetically feasible. However, the kinematics are quite different; if D_2 + DCCO is assumed to be the sole heavy-light channel, the $P(E_T)$ distributions required to simulate the broad wings extend beyond the maximum E_T allowed by thermodynamics. Moreover, the D + CD_2CO channel results from simple bond fission and should have at most a small exit barrier (< 0.3 eV), while elimination of D_2 would pass through a more strained four-center transition state, causing a substantial barrier on the potential energy surface. Therefore elimination of D_2 should be kinetically disfavored compared to simple bond fission, and we propose that Channel 2 is the only channel other than Channel 1 contributing to the TOF distributions.

Because we expect a rather small exit barrier for Channel 2, a plausible initial estimate for the $P(E_T)$ distribution is a prior distribution,[51] i.e., the approximate distribution expected for statistical dissociation with no barrier, shown in Fig. 7c (dashed line). However, the experiment is most sensitive to translational energies $E_T > 0.5$ eV, and agreement with the data is much better with a slower fall-off at higher E_T shown by the

solid line in this figure. We find that $\beta = 0.25$ gives the best overall fit using this $P(E_T)$ distribution; larger values of β cannot fit data from both polarizations simultaneously. An additional parameter in the simulations is the detection efficiency of D atoms. This should be independent of the photon energy or polarization, and only values in the range of $\approx 8\%$ efficiency can simultaneously reproduce the data at all four photon energies and both laser polarizations.

The resulting overall fits including the contributions for Channels 1 and 2 are shown as solid lines in Figs. 6 and 7a and are in good agreement with the data. The $P(E_T)$ distributions for Channel 2 at all measured photon energies are similar to the distribution in Fig. 7c, peaking close to zero and with a maximum E_T of around 1.3 eV. This approximate independence from the photon energy was also seen in Channel 1.

The branching ratio of Channels 2 to 1 derived from the simulation of the TOF distributions is 4:1, with negligible dependence on the photon energy. However, because of the insensitivity of the simulation to low translational energies for Channel 2, we estimate that the true branching ratio could range from 2:1 to 6:1. In any event, we believe that $CD_2CO + D$ is the major product channel at all photon energies studied.

V. Discussion

A. Implications of the experimental data

The most important experimental results are as follows. (1) The photofragment yield spectrum shows that CH_2CHO predissociates across the entire $\tilde{B}\,(^2A'') \leftarrow \tilde{X}\,(^2A'')$ band. Predissociation and fluorescence are competing processes, with predissociation

dominating at energies > 1,400 cm^{-1} above the $\tilde{B}$ state origin. (2) Below 1,400 cm^{-1}, there is some mode-specificity in the competition between fluorescence and dissociation, with vibrational levels in which a'' modes are excited dissociating more rapidly than nearby a' levels. (3) The CH_3+CO photofragment angular distribution is isotropic at low energies but becomes anisotropic with $\beta(E_T) > 0$ above 1,400 cm^{-1}. (4) The photofragment translational energy distributions $P(E_T)$ are relatively independent of excitation energy for Channels 1 and 2, and peak at energies well below the maximum allowed by energy conservation.

The last observation is the most important as it implies that dissociation does not occur by coupling of the $\tilde{B}$ state to a repulsive surface correlating directly to products. Dynamics occurring on a repulsive excited state typically deposit a large fraction of the available energy into relative translation of the fragments, and one expects the $P(E_T)$ distributions to shift towards higher E_T as the excitation energy is increased; both of these effects were observed in photodissociation of CH_3O,[10] for example. The $P(E_T)$ distributions for CH_2CHO imply a different mechanism, in which Channel 1 (CH_3 + CO) results from the dissociation of a strongly bound state over a substantial barrier, whereas Channel 2 (H + CH_2CO), for which the $P(E_T)$ distributions peak near $E_T = 0$, results from dissociation of such a state over a much smaller barrier.

In the following discussion, this mechanism is considered in more detail. We will show that the set of results obtained here is consistent with the initial electronic excitation being followed by internal conversion (IC). This process eventually populates the ground electronic state where energy randomization occurs, followed by decomposition to the two observed product channels. We first consider the asymptotic final state distributions

for the two channels and the branching ratios that would be expected from such a mechanism; most of this discussion focuses on Channel 1, since considerably more detailed data are available than for Channel 2. We then consider how IC might connect the initially excited electronic state to the ground state on which dissociation occurs.

B. Dissociation mechanism and product state distributions: CH_3 + CO

1. Barrier to dissociation of Channel 1

The $P(E_T)$ distributions for Channel 1 all peak in the range of E_T = 0.9-1.0 eV. If this channel resulted from statistical dissociation on a surface with no exit barrier (i.e., no barrier with respect to products), one would expect a translational energy distribution described by phase space theory or a prior distribution, peaking very near $E_T = 0$. This clearly disagrees with our experimental evidence. On the other hand, the distributions for Channel 1 are consistent with statistical dissociation over an exit channel barrier. Such a process is statistical in the sense that energy randomization occurs prior to dissociation, but passage over the barrier results in rapid release of energy which is *not* statistically distributed among the product degrees of freedom. Instead, a substantial fraction of the barrier height is converted to fragment translation, resulting in a $P(E_T)$ distribution peaking at 40-80% of the barrier height. [52] Therefore, our experimental $P(E_T)$ distributions for Channel 1 imply a barrier with respect to products of between 1.1 and 2.2 eV in the CH_3 + CO exit channel.

The nature of this barrier can be inferred from our *ab initio* calculations. Fig. 8 shows that the reaction coordinate to Channel 1 on the ground state surface involves passage over two barriers: TS1, the barrier to isomerization (via a 1,2 hydrogen shift)

between CH_2CHO and the acetyl radical, CH_3CO, and TS3, the barrier to elimination of CO from CH_3CO. TS1 is the higher energy transition state and is found to be a nearly planar CH_2CO moiety with the out-of-plane migrating H atom shared between the two carbon atoms. This isomerization is reasonably facile because the a'' unpaired electron on the methylene carbon in the $\tilde{X}$ state is available for bonding with the migrating H atom.

The calculated energies of TS1 and TS3 are 1.74 eV and 0.40 eV, respectively, above the CH_3+CO products. The latter value agrees with the experimentally determined exit barrier to CH_3CO dissociation, 0.36 eV,[44,45] but is too low to account for our $P(E_T)$ distributions for CH_2CHO dissociation. On the other hand, the calculated barrier height for TS1 is in the range expected based on the $P(E_T)$ distributions. The simplest mechanism consistent with our data and the PES is that after traversing TS1, dissociation occurs so rapidly that the small barrier TS3 does not trap the radicals in the CH_3CO well, nor does TS3 control the final product states. Therefore we can envision, for the purpose of understanding the product state distributions, a simplified PES with only a single barrier (TS1) between CH_2CHO $\tilde{X}\,(^2A'')$ and CH_3 + CO. This hypothesis can be tested by comparing the $P(E_T)$ distributions we observe (Fig. 4) against the predictions of photodissociation dynamics models (see Section V.B.2).

As an aside, we note that if CH_2CHO $\tilde{A}\,(^2A')$ were formed by IC, the 1,2 hydrogen shift would be much less favorable because the there is a C=C double bond in this state.[21,24] Forming the transition state analogous to TS1 therefore requires significant torsional distortion of the C=C double bond to a non-planar geometry, in addition to the

strain energy associated with the tight transition state, resulting in a high energy transition state. It therefore unlikely that CH_3 + CO production is feasible on the $\tilde{A}$ surface.

2. *Statistical adiabatic impulsive model*

In this section, we present a simple model to describe the dissociation dynamics of a molecule undergoing statistical decomposition over a barrier. Simple models predicting the $P(E_T)$ distribution for molecular photodissociation can be divided into two categories. *Statistical models* predict the $P(E_T)$ distribution for unimolecular dissociations through a loose transition state (TS). Examples include prior distributions,[51] phase space theory,[53] separate statistical ensembles method,[54] and the statistical adiabatic channel model.[55] *Sudden approximation models*, such as Franck-Condon (FC) mapping[56] and the impulsive model,[57] are more suitable for predicting the $P(E_T)$ distribution for direct dissociation on a repulsive potential or over a barrier associated with a tight TS.

For excitation energies in proximity to the barrier height, the repulsive nature of the potential dominates the dissociation process, and the sudden approximation can often predict the product state distributions. However, for excitation energy in significant excess of the barrier height, which we believe is the case for Channel 1, a more general approach is required combining both statistical and sudden models. As an example of such an approach, North *et al.*[58] have developed the barrier impulsive model which predicts the *average* energy in product translation, rotation, and vibration for dissociation over a potential barrier. We have developed a qualitatively similar approach, the statistical adiabatic impulsive (SAI) model, in order to predict the $P(E_T)$ *distributions* for Channel 1. The SAI model is described briefly here and will be presented in more detail elsewhere.[59]

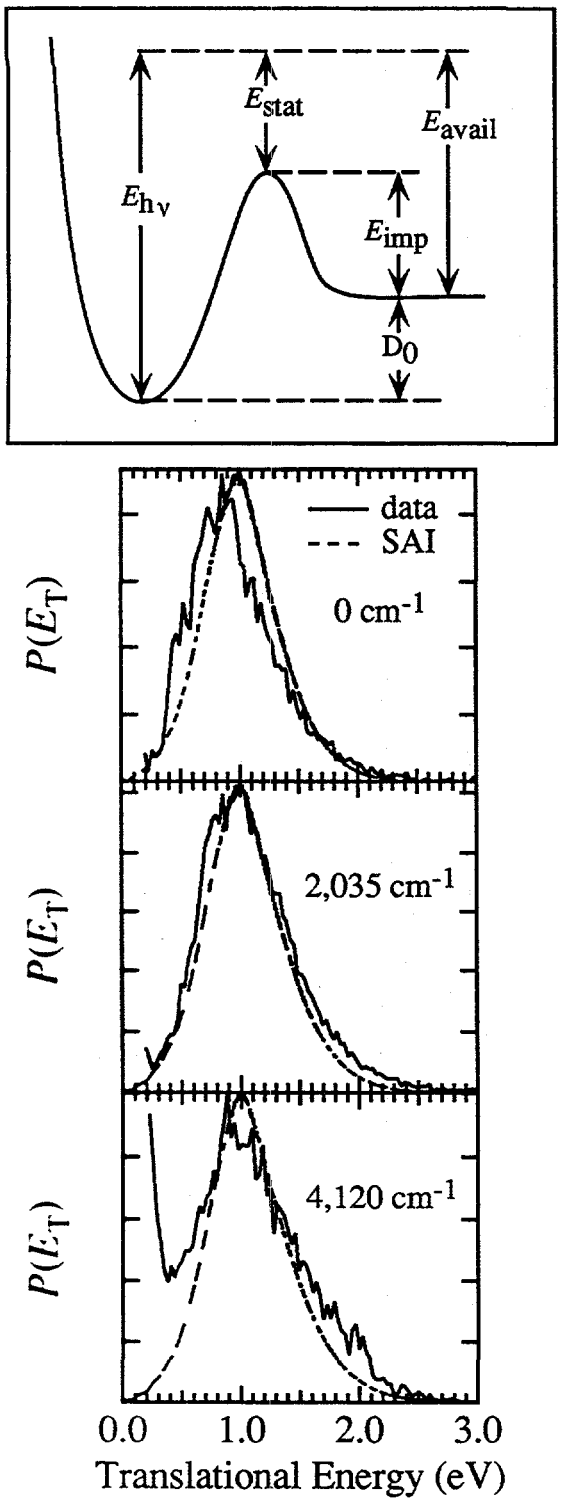


Figure 9: (a) Definition of terms in the Statistical Adiabatic Impulsive (SAI) model. (b) Comparison of SAI (- - -) and experimental (——) $P(E_T)$ distributions for CH_3+CO at three selected energies.

As shown in Fig. 9a, the total energy available to the products (E_{avail} = hν − D_0) is divided into two separate energy reservoirs: a statistical reservoir (the energy from the TS up to the total available energy, i.e., $E_{stat} = E_{avail} - E_{TS}$) and an impulsive reservoir (the energy of the TS relative to the ground state energy of the products, i.e., $E_{imp} = E_{TS}$). The energy of the TS, E_{TS}, corresponds to the zero-point level of TS1 with respect to the CH_3 + CO products. The SAI model requires knowledge of the TS geometry, frequencies, and normal modes, which in the present case are taken from our *ab initio* calculations.

We assume that energy is randomized among all vibrational modes at the transition state (i.e., the top of the barrier). For the statistical reservoir, each vibrational mode at the transition state is correlated[58] with a particular *product*

vibrational (V), rotational (R), or translational (T) degree of freedom. For example, in the CH_2CHO system at TS1, two skeletal bending modes correlate with product R, and two stretching modes and the reaction coordinate correlate with product T. The remaining seven vibrational modes at the TS are the only modes that asymptotically correlate with product V. These designations are given next to vibrational frequencies for TS1 in Table IV. Next we make the assumption that beyond the TS the internal motions evolve adiabatically to products. This assumption seems justified because the fragments will separate rapidly after traversing the transition state region. Hence, for each possible vibrational state at the TS, the quantum numbers in each mode are preserved throughout the dissociation. The vibrational energy deficit or surplus for each state is assumed to flow freely between V and T degrees of freedom, while energy at the TS in the two skeletal bends goes entirely into product R. This yields the internal energy $P_{\mathrm{stat}}(E'_{\mathrm{int}})$ distribution for the statistical reservoir. The product state distribution of the fragments from this reservoir is not strictly statistical in the sense of a prior or PST distribution, in which the energy is randomized among fragment states. Rather, energy is statistically distributed at the tight TS and evolves adiabatically to the products.

The impulsive reservoir's contribution to the product energy distributions is determined by the TS geometry. For this reservoir, the vibrational energy distribution of the products is determined by Franck-Condon projection of the zero-point TS wavefunction on to the asymptotic product wavefunctions. For each final vibrational state, the remaining energy in the impulsive reservoir is partitioned between translation and rotation using a rigid impulsive model based on the TS geometry. The $P_{\mathrm{imp}}(E''_{\mathrm{int}})$ distribution for the impulsive reservoir is the sum of the impulsive model result for each

combination of vibrational product states, weighted by their FC projection factors. For vinoxy dissociation through TS1, approximately 55% of the exit barrier height evolves into E_T. The $P_{imp}(E''_{int})$ distribution is independent of excitation energy, as the energy of this reservoir remains constant. This provides an explanation for the maximum intensity at ≈ 0.9 eV in the experimental $P(E_T)$ distributions and its relative independence on excitation energy.

Finally, the overall $P(E_{int})$ distribution is created by a convoluting the statistical and impulsive distributions:

$$P(E_{int}) = \int_0^{E_{stat}} P_{stat}(E'_{int}) \cdot P_{imp}(E_{int} - E'_{int}) dE'_{int} \tag{11}$$

The translational energy distribution $P(E_T)$ is then obtained by plotting $P(E_{int})$ vs. $E_T = E_{avail} - E_{int}$. We emphasize that the SAI model contains no adjustable parameters and conserves energy in each separate reservoir and therefore for the overall process.

The predictions of the SAI model and experimental $P(E_T)$ distributions are shown in Fig. 9b for three excitation energies. The reasonable agreement between the model and the data lends credence to the idea that the $P(E_T)$ distributions for Channel 1 are determined largely by the height and geometry of the vinoxy/acetyl isomerization barrier.

3. *Comparison with CH_3CO decomposition*

In experiments related to the vinoxy system, several groups have examined the fragmentation of the acetyl radical, $CH_3CO \rightarrow CH_3 + CO$. In these experiments, CH_3CO is not laser-excited to a state of specific energy, as in the present work on CH_2CHO, but is instead prepared with a distribution of internal energies by either (i) photodissociation of

acetyl halides CH_3COX (X = Cl[45,60], Br[60], I[61]) and acetone,[58] or (ii) charge neutralization of the acetyl cation, CH_3CO^+.[6263] In the photodissociation experiments, CH_3CO is formed with average internal energy $\langle E_{int} \rangle$ = 0.8, 1.3, 1.5, or 1.7 eV, depending on the photon energy and the identity of the primary photofragment (Cl, Br, I, or CH_3). In *all* cases the $P(E_T)$ distributions for CH_3 + CO peak at 0.22 - 0.3 eV. These experiments offer convincing evidence that the translational energy distributions are determined by the 0.36 eV exit barrier (TS3) and depend only weakly on $\langle E_{int} \rangle$. In addition, the significant differences in the $P(E_T)$ distributions in these experiments compared to ours supports our hypothesis that the dissociation dynamics of $CH_2CHO \rightarrow CH_3$ + CO are not determined by TS3, but rather by TS1.

In the second set of experiments, Körnig *et al.*[62] and Hop and Holmes[63] have studied the dissociation of CH_3CO produced by charge neutralization of CH_3CO^+ with Na, K, or Cs atoms. The work by Körnig *et al.* is particularly relevant to our results, as they measured the translational energy release of the dissociation to CH_3 + CO in a fast beam dissociation experiment similar to ours. They estimate that excited acetyl is created with $\langle E_{int} \rangle \approx$ 1.9, 2.7, or 3.1 eV of internal energy, when formed by the three alkali atoms, respectively. Their $P(E_T)$ distributions depend little on the alkali atom used, and are remarkably similar to our distributions in Fig. 4, peaking at 0.8 eV. However, since CH_3CO^+ has a linear[64] CCO backbone, whereas neutral CH_3CO $\tilde{X}\,(^2A')$ has $\angle$(CCO) = 127°, the Franck-Condon overlap of the ion with the acetyl ground state is expected to be poor. Nimlos *et al.*[39] predict that an excited state of acetyl, the linear CH_3CO $\tilde{A}\,(^2A'')$ state, lies $\approx$ 1.3 eV above the ground state. We therefore surmise that in charge

neutralization experiments, CH_3CO is formed in the excited $\tilde{A}$ state, due to better FC overlap with the linear cation. Furthermore, the $P(E_T)$ distributions of Körnig are not consistent with the work described above by North[45,58,60] and Kroger,[61] in which dissociation definitely occurs on the ground state PES. Therefore we can only conclude that the dissociation to CH_3 + CO studied by Körnig either occurs on an excited state surface, or that they grossly underestimate the value of $\langle E_{int} \rangle$ of the acetyl radicals. In either case, the resemblance of their $P(E_T)$ distributions to ours seems to be purely coincidental.

C. Dissociation mechanism and product state distributions: H + CH_2CO

The $P(E_T)$ distributions for Channel 2 peak near $E_T = 0$. Using the same arguments as in Section V.B.1, this channel likely results from statistical dissociation over at most a small exit barrier along the reaction coordinate. From Fig. 8, we can determine if such a mechanism is plausible. There are three possible routes to H + CH_2CO. The first two possibilities are direct C–H bond fission from CH_2CHO $\tilde{X}$ ($^2A''$) or CH_2CHO $\tilde{A}$ ($^2A'$) as shown on the left side of Fig. 8. The third route is direct C–H bond fission from the isomerization product CH_3CO, shown on the right side of Fig. 8.

All three routes have barriers with respect to products, because the reverse reaction of H (^2S) + CH_2CO $\tilde{X}$ ($^1A_1$) involves a radical reacting with a closed-shell molecule. TS2, the barrier to C-H bond fission from CH_3CO, lies only 0.24 eV above the product asymptote. However, TS2 lies 2.05 eV above the CH_3CO minimum, whereas the barrier to C-C bond fission, TS3, lies only 0.75 eV above CH_3CO. Once isomerization to CH_3CO occurs, Channel 1 should therefore be strongly favored over Channel 2.

However, we observe that the Channel 2:Channel 1 branching ratio is approximately 4:1. It is therefore unlikely that passage over TS2 contributes significantly to Channel 2.

CH_2CHO $\tilde{X}\,(^2A'')$ correlates adiabatically (in C_s symmetry) with the $\tilde{a}\,(^3A'')$ excited state of ketene, while the first excited vinoxy state, CH_2CHO $\tilde{A}\,(^2A')$, correlates adiabatically (over barrier TS4) with ground state ketene $\tilde{X}\,(^1A_1)$. At non-planar geometries the two surfaces may couple through an avoided crossing, allowing the dissociation pathway CH_2CHO $\tilde{X}\,(^2A'') \rightarrow CH_2CO$ $\tilde{X}\,(^1A_1)$ + H. The exit barrier TS5 along this path is suggested in Fig. 8, although we have not definitively located a non-planar transition state in our calculations. We adopt an exit barrier height of 0.13 eV[65] advocated by Benson for the addition reaction H + CH_2CO on the ground state surface. The transition state frequencies for TS5 (Table IV) are also estimated using values given by Benson.[66] TS5 lies considerably lower in energy than TS4 with respect to the H + CH_2CO products, and the $P(E_T)$ distributions for Channel 2 are more consistent with dissociation on the ground state surface over TS5.

D. Product branching ratio

In this section we consider the relationship between the product branching ratio and the proposed mechanism for Channels 1 and 2. TS5, the barrier along the most likely reaction path for Channel 2, lies slightly lower in energy (1.63 eV) with respect to the CH_2CHO $\tilde{X}\,(^2A'')$ minimum than TS1, the isomerization barrier (1.78 eV). In addition, TS5 is a "looser" transition state than TS1, since it involves bond fission rather than rearrangement. One would therefore expect statistical dissociation on the ground state to

favor Channel 2, in agreement with the experimental results that show this channel to be favored by about 4:1.

To put this argument on a more quantitative footing, we can compute the RRKM rate constants $k = \frac{W(E^{+})}{h\rho(E)}$ for the two elementary reaction steps:

$$CH_2CHO \xrightarrow{k_1} CH_3CO \tag{12}$$

$$CH_2CHO\,\tilde{X}(^2A') \xrightarrow{k_2} H + CH_2CO\,\tilde{X}(^1A_1) \tag{13}$$

where $W(E^{+})$ is the total number of available vibrational states at the transition state, h is Planck's constant, and $\rho(E)$ is the density of reactant vibrational states. Both $\rho(E)$ and $W(E^{+})$ are calculated using the Beyer-Swinehart algorithm and the vibrational frequencies for the reactant and transition states from Table IV. The resulting rate constants at a few representative photon energies are given in Table VI. The dissociative lifetimes on the ground electronic surface are in the range of 1 - 10 ps.

Photon Energy	$k_1(E)$ ($s^{-1}/10^{11}$)	$k_2(E)$ ($s^{-1}/10^{11}$)
$h\nu = 28785$	1.0	4.2
$h\nu = 31000$	1.6	6.5
$h\nu = 33300$	2.4	9.3

Table VI: RRKM dissociation rate constants.

Based on the discussion in Section V.B, isomerization should be the rate-limiting step for Channel 1, so the branching ratio can be computed from these two elementary reactions alone. The ratio k_2:k_1 is $\approx$ 4:1, a value which matches the experimental branching ratio. Although this excellent agreement may be fortuitous, it further supports our

proposed mechanism in which Channels 1 and 2 result from dissociation of CH_2CHO on its ground state potential energy surface, with the dynamics and branching ratio largely determined by the transition states TS1 and TS5, for isomerization and C-H bond fission, respectively.

E. Internal Conversion

The preceding discussion strongly indicates that Channels 1 and 2 result from dissociation on the ground state potential energy surface, but has not considered how electronic excitation of the $\tilde{B}$ state is converted to vibrational excitation of the $\tilde{X}$ state. There are two likely pathways for this internal conversion process. The simplest is direct internal conversion from the $\tilde{B}\,(^2A'')$ to $\tilde{X}\,(^2A'')$ states, with the radiationless transition rate determined by Fermi's Golden Rule. On the other hand, the vinoxy radical has two low-lying excited states in addition to the $\tilde{B}\,(^2A'')$ state, namely the $\tilde{A}\,(^2A')$ and $\tilde{C}\,(^2A')$ states, with term values of 0.99 eV (experimental[13]) and 4.60 eV (predicted[24]), respectively, at planar geometries. One must therefore consider whether energy flows to the $\tilde{X}\,(^2A'')$ state via one or both of these states.

Experimental data on the fluorescence and dissociation rates are relevant to the nonradiative relaxation mechanism. Barnhard *et al.*[18] and Rohlfing[67] have measured fluorescence lifetimes on the order of 100 ns for several vibrational levels of the $\tilde{B}$ state with less than 1400 cm^{-1} excess energy. These values are entirely consistent with the photofragment angular distributions for Channel 1 in this energy range, all of which are isotropic, indicating that the excited radical lives for many rotational periods before dissociation occurs. The non-radiative rates in this energy range must be comparable to

the fluorescence rates since fluorescence and dissociation are competitive. Given that the RRKM rate constants (Table VI) are at least 10^{11} s^{-1}, it is clear that non-radiative relaxation is the rate-limiting step for dissociation below 1,400 cm^{-1}.

However, above 1,400 cm^{-1}, the fluorescence yield drops abruptly[14,15] and the photofragment angular distributions become anisotropic. This indicates a rapid increase in the non-radiative rate. We can quantify this increase by recognizing that the anisotropic angular distributions above 1,400 cm^{-1} imply a lifetime of the excited radical with respect to dissociation of less than a rotational period. The appropriate rotational constant for calculating the rotational period of this nearly prolate symmetric top is $(B + C)/2 =$ 0.35 cm^{-1}; using $J = 10$ ($K = 0$) as an average value, the rotational period about the b and c axes is approximately 5 ps. In order for dissociation to occur on this time scale, the overall internal conversion process populating the ground state must be at least this fast, so its rate must increase by about four orders of magnitude in a relatively narrow interval around 1400 cm^{-1} excess energy.

Based on Fermi's Golden Rule, one would expect the internal conversion rate from the initially excited levels of the $\tilde{B}$ ($^2A''$) state to highly excited levels of the $\tilde{X}$ ($^2A''$) state to increase with excess energy in the $\tilde{B}$ state.[68] However, it is unlikely that this effect alone could produce the abrupt increase in the internal conversion rate implied by the experiments. The abrupt drop in fluorescence quantum yield is similar to benzene, where the fluorescence quantum yield of the S_1 ($^1B_{2u}$) state drops considerably at excess energies above 3000 cm^{-1}.[69] The mechanism of the nonradiative process responsible for this effect, known as "channel three," is still under discussion, but there appears to be a consensus that it is due to a combination of intramolecular vibrational redistribution (IVR) to dark

levels of the S_1 state followed by $S_1 \rightarrow S_0$ internal conversion, with the onset of fast IVR coinciding with the drop in fluorescence quantum yield.[70,71,72] While such a mechanism might explain the fluorescence cutoff in vinoxy, it does not explain the enhanced dissociation rate, because there is no reason to believe that the relaxed levels of the $\tilde{B}$ state populated by IVR should be so much more strongly coupled to the $\tilde{X}$ state than the originally populated state.

An alternative explanation for the abrupt fluorescence cutoff and increased dissociation rate is the possibility of a curve-crossing between the $\tilde{B}$ state and another low-lying state at an excess energy in the range of 1,400 cm^{-1}. The calculations by Yamaguchi[24] predict that the energy of the $\tilde{A}$ state rises by about 2.2 eV if the vinoxy radical is twisted by 90° about the C-C bond due to the double-bonded character of the C-C bond in the $\tilde{A}$ state, whereas the energies of the $\tilde{B}$ and $\tilde{X}$ states are relatively insensitive to torsional distortion. The net result is that the $\tilde{B}$ and $\tilde{A}$ states become nearly degenerate at a dihedral angle of 90°, suggesting that a low energy crossing may occur. This calculation also predicts that the $\tilde{C}$ ($^2A'$) state, which lies 1.1 eV above the $\tilde{B}$ state in the planar geometry, drops in energy as the radical as twisted and becomes nearly degenerate with the $\tilde{B}$ state. Thus, low energy curve crossings with either the $\tilde{A}$ or $\tilde{C}$ states appear to be feasible.

The possible role of the $\tilde{A}$ or $\tilde{C}$ states is indirectly supported by the mode-specific effects seen below 1400 cm^{-1} where fluorescence and dissociation compete. As pointed out in Section IV.A, excitation of a'' vibrational modes in the $\tilde{B}$ state enhances the dissociation yield compared to nearby levels in which only a' modes are excited. This

observation is consistent with a radiationless transition through vibronic coupling between the $\tilde{B}\,(^2A'')$ state and an electronic state of A' symmetry,[68,73] with the $\tilde{A}\,(^2A')$ or $\tilde{C}\,(^2A')$ states as the most likely candidates.

The above arguments support the idea that non-radiative relaxation process above 1,400 cm^{-1} involves coupling between the $\tilde{B}$ state and either the $\tilde{A}$ or $\tilde{C}$ states, rather than direct coupling with the $\tilde{X}$ state. However, dissociation does appear to occur on the ground state surface, so energy flow from the $\tilde{A}$ or $\tilde{C}$ state to the $\tilde{X}$ state must quite rapid for this mechanism to be valid. The $\tilde{B} \rightarrow \tilde{A}$ radiationless transition will result in highly excited species with substantial energy in the torsional vibration, whereas the transition to the $\tilde{C}$ state will produce radicals with relatively little vibrational energy. The $\tilde{B} \rightarrow \tilde{A}$ state is therefore more favorable based on the higher density of vibrational levels in the $\tilde{A}$ state compared to the $\tilde{C}$ state. In addition, since internal conversion rates generally increase with vibrational excitation in the initial state,[73] it seems a transition to the $\tilde{A}$ state is more likely to result in subsequent rapid internal conversion to the $\tilde{X}$ state than a transition to the $\tilde{C}$ state. Thus, the overall internal conversion scheme most consistent with the experimental results and *ab initio* calculations is a two-step mechanism, $\tilde{B}\,(^2A'') \rightarrow \tilde{A}\,(^2A') \rightarrow \tilde{X}\,(^2A'')$, with the first step being rate-limiting below 1400 cm^{-1} excess energy.

VI. Conclusions

The spectroscopy and dissociation dynamics of the vinoxy radical presented in this paper have been obtained under well-defined experimental conditions. Internally cold radicals are laser-excited to exact total energies and the nascent photofragments are probed as a function of photon energy, product channel, translational energy, and angular distribution. In concert with *ab initio* electronic structure calculations presented in this paper, we have explored the critical configurations on the potential energy surfaces, and presented a simple dissociation mechanism which is consistent with all the experimental and theoretical results. After initial excitation via the $\tilde{B}\,(^2A'') \leftarrow \tilde{X}\,(^2A'')$ transition, all observed vibrational levels of CH_2CHO predissociate. We assign transitions in the photofragment yield spectrum to eight vibrational progressions, two involving non-totally symmetric vibrational modes. At lower photon energies, the rate of non-radiative decay is mode-specific, depending strongly on the symmetry of the vibrational mode excited in the $\tilde{B}$ state. The photofragment energy and angular distributions indicate that dissociation occurs by internal conversions to the ground state potential energy surface upon which dissociation proceeds to H + CH_2CO and CH_3 + CO with a branching ratio of roughly 4:1. The dramatic increase in the dissociation rate above 1400 cm^{-1} suggests a curve crossing of the $\tilde{B}$ state with the $\tilde{A}$ or $\tilde{C}$ states, followed by rapid internal conversion to the ground potential energy surface,

For the CH_3 + CO channel, we show that the translational energy distribution of the fragments is determined by the $CH_2CHO \rightarrow CH_3CO$ isomerization barrier on the ground state potential energy surface. For this system, both the exit barrier height and the

excess energy at the transition state are substantial. We have therefore developed the statistical adiabatic impulsive model,[59] which successfully predicts the observed $P(E_T)$ distributions with no adjustable parameters. Our results for this channel are discussed in light of previous investigations of unimolecular decomposition of the acetyl radical.

The hydrogen atom loss channel is probed by the time-of-flight distributions arising from CD_2CDO photodissociation. Although our information on this product channel is more limited, we find the simplest explanation consistent with our results is direct dissociation from vinoxy over a small barrier. This barrier is formed by an avoided crossing between the CH_2CHO $\tilde{X}$ ($^2A''$) state, which correlates for planar geometries with H + CH_2CO $\tilde{a}$ ($^3A''$), and the CH_2CHO $\tilde{A}$ ($^2A'$) state, which correlates with H + CH_2CO $\tilde{X}$ ($^1A_1$).

Together with detailed previous results of traditional optical spectroscopy, the present work on photodissociation of vinoxy sheds new light on the electronic states and dynamics of this prototypical alkenoxy free radical.

Acknowledgments

This research is supported 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. I thank Professor Robert Continetti for advice on forward convolution simulations, Professor Bradley Moore for discussions of internal conversion mechanisms, and Dr. Eric Rohlfing for sharing his results prior to publication.

References

[1] R. Schinke, *Photodissociation Dynamics* (Cambridge University Press, Cambridge, 1993).

[2] L. J. Butler and D. M. Neumark, J. Phys. Chem. **100**, 12801 (1996).

[3] R. J. Cvetanovic and D. L. Singleton, Rev. Chem. Int. **5**, 183 (1984).

[4] A. M. Schmoltner, P. M. Chu, R. J. Brudzynski, and Y. T. Lee, J. Chem. Phys. **91**, 6926 (1989).

[5] D. J. Donaldson, I. V. Okuda, and J. J. Sloan, Chem. Phys. **193**, 37 (1995).

[6] V. Schmidt, G. Y. Zhu, K. H. Becker, and E. H. Fink, Ber. Bunsen-Ges. Phys. Chem. **89**, 321 (1985).

[7] K. Lorenz and R. Zellner, Ber. Bunsen-Ges. Phys. Chem. **88**, 1228 (1984); T. J. Wallington, R. Liu, P. Dagaut, and M. J. Kurylo, Int. J. Chem. Kinet. **20**, 41 (1988).

[8] C. K. Westbrook and F. L. Dryer, *Eighteenth Symposium (International) on Combustion* (Combustion Institute, Pittsburgh, 1981), p. 749.

[9] K. Brezinsky, Prog. Energy Combust. Sci. **12**, 1 (1986).

[10] D. L. Osborn, D. J. Leahy, E. M. Ross, and D. M. Neumark, Chem. Phys. Lett. **235**, 484 (1995).

[11] D. H. Mordaunt, D. L. Osborn, H. Choi, R. T. Bise, and D. M. Neumark, J. Chem. Phys. **105** (in press).

[12] D. A. Ramsay, J. Chem. Phys, **43**, S18 (1965).

[13] H. E. Hunziker, H. Kneppe, and H. R. Wendt, J. Photochem. **12**, 377 (1981).

[14] G. Inoue and H. Akimoto, J. Chem. Phys. **74**, 425 (1981).

[15] L. F. DiMauro, M. Heaven, and T. A. Miller, J. Chem. Phys., **81**, 2339 (1984).

[16] Y. Endo, S. Saito, and E. Hirota, J. Chem. Phys. **83**, 2026 (1985); Y. Endo and E. Hirota, J. Mol. Spec. **127**, 535 (1988).

[17] T. Gejo, M. Takayanagi, T. Kono, and I. Hanazaki, Chem. Letters 2065 (1993).

[18] K. I. Barnhard, M. He, and B. R. Weiner, J. Phys. Chem. **100**, 2784 (1996).

[19] G. B. Ellison, P. C. Engelking, and W. C. Lineberger, J. Phys. Chem. **86**, 4873 (1982); R. D. Mead, K. R. Lykke, W. C. Lineberger, J. Marks, and J. I. Brauman, J. Chem. Phys. **81**, 4883 (1984).

[20] N. C. Baird, R. R. Gupta, and K. F. Taylor, J. Am. Chem. Soc. **101**, 4531 (1979).

[21] M. Dupuis, J. J. Wendoloski, and W. A. Lester, Jr., J. Chem. Phys., **76**, 488 (1982).

[22] E. S. Huyser, D. Feller, W. T. Borden, and E. R. Davidson, J. Am. Chem. Soc. **104**, 2956 (1982).

[23] M. Yamaguchi, T. Momose, and T. Shida, J. Chem. Phys. **93**, 4211 (1990).

[24] M. Yamaguchi, Chem. Phys. Lett., **221**, 531 (1994).

[25] M. E. Jacox, Chem. Phys., **69**, 407 (1982).

[26] M. W. Chase, Jr., C. A. Davies, J. R. Downey, Jr., D. J. Frurip, R. A. McDonald, and A. N. Syverud, JANAF Thermochemical Tables, 3rd ed., J. Chem. Phys. Ref. Data **14**, (1985), Suppl. No. 1.; S. G. Lias, J. E. Bartmess, J. F. Liebman, J. L. Holmes, R. D. Levin, and W. G. Mallard, Gas-Phase Ion and Neutral Thermochemistry, J. Chem. Phys. Ref. Data **17**, (1988), Suppl. No. 1.

[27] J. T. Niiranen, D. Gutman, and L. N. Krasnoperov, J. Phys. Chem. **96**, 5881 (1992); C. W. Bauschlicher, Jr., J. Phys. Chem., **98**, 2564 (1994).

[28] D. L. Osborn, H. Choi, and D. M. Neumark, Adv. Chem. Phys., (in press).

[29] R. E. Continetti, D. R. Cyr, D. L. Osborn, D. J. Leahy, and D. M. Neumark, J. Chem. Phys. **99**, 2616 (1993).

[30] R. E. Continetti, D. R. Cyr, R. B. Metz, and D. M. Neumark, Chem. Phys. Lett. **182**, 406, 1991.

[31] D. R. Cyr, R. E. Continetti, R. B. Metz, D. L. Osborn, and D. M. Neumark, J. Chem. Phys. **97**, 4937 (1992).

[32] D. L. Osborn, D. J. Leahy, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **104**, 5026 (1996).

[33] J. M. B. Bakker, J. Phys. E **6**, 785 (1973); **7**, 364 (1974).

[34] S. Gerstenkorn and P. Luc, *Atlas du Spectre d'Absorption de la Molecule d'Iode* (Centre National de la Recherche Scientifique, Paris, 1978).

[35] D. J. Leahy, D. L. Osborn, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **103**, 2495 (1995).

[36] D. P. de Bruijn and J. Los, Rev. Sci. Instrum. **53**, 1020 (1982).

[37] C. Martin, P. Jelinsky, M. Lampton, R. F. Malina, and H. O. Anger, Rev. Sci. Instrum. **52**, 1067 (1981).

[38] D. F. Ogletree, G. S. Blackman, R. Q. Hwang, U. Stark, G. A. Somorjai, and J. E. Katz, Rev. Sci. Instrum. **63**, 104 (1992).

[39] M. R. Nimlos, J. A. Soderquist, and G. B. Ellison, J. Am. Chem. Soc. **111**, 7675 (1989).

[40] R. N. Zare, Mol. Photochem. **4**, 1 (1972).

[41] GAUSSIAN 92, Revision C, M. J. Frisch, G. W. Trucks, M. Head-Gordon, P. M. W. Gill, M. W. Wong, J. B. Foresman, B. G. Johnson, H. B. Schlegel, M. A. Robb, E. S. Replogle, R. Gompets, J. L. Andres, K. Raghavachari, J. S. Binkley, C. Gonzalez, R. L. Martin, D. J. Fox, D. J. Defrees, J. Baker, J. J. P. Stewart, and J. A. Pople, Gaussian, Inc., Pittsburgh, Pennsylvania, 1992.

[42] GAUSSIAN 94, Revision C.3, M. J. Frisch, G. W. Trucks, H. B. Schlegel, P. M. W. Gill, B. G. Johnson, M. A. Robb, J. R. Cheeseman, T. Keith, G. A. Petersson, J. A. Montgomery, K. Raghavachari, M. A. Al-Laham, V. G. Zakrzewski, J. V. Ortiz, J. B. Foresman, J. Cioslowski, B. B. Stefanov, A. Nanayakkara, M. Challacombe, C. Y. Peng, P. Y. Ayala, W. Chen, M. W. Wong, J. L. Andres, E. S. Replogle, R. Gomperts, R. L. Martin, D. J. Fox, J. S. Binkley, D. J. Defrees, J. Baker, J. P. Stewart, M. Head-Gordon, C. Gonzalez, and J. A. Pople, Gaussian, Inc., Pittsburgh PA, 1995.

[43] S. Deshmukh, J. D. Myers, S. S. Xantheas, and W. P. Hess, J. Phys. Chem. **98**, 12535 (1994).

[44] K. W. Watkins and W. W. Word, Intern. J. Chem. Kinetics **7**, 855 (1974).

[45] S. W. North, D. A. Blank, and Y. T. Lee, Chem. Phys. Lett. **224**, 38 (1994).

[46] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople, J. Chem. Phys. **94**, 7221 (1991).

[47] J. L. Durant and C. M. Rohlfing, J. Chem. Phys. **98**, 8031 (1993).

[48] D. Hegarty and M. A. Robb, Mol. Phys. **38** 1795 (1979); R. H. E. Eade and M. A. Robb, Chem. Phys. Lett. **83** 362 (1981).

[49] J. J. McDouall, K. Peasley, and M. A. Robb, Chem. Phys. Lett. **148**, 183 (1988).

[50] J. A. Pople, M. Head-Gordon, and K. Raghavachari, J. Chem. Phys. **87**, 5968 (1987).

[51] R. D. Levine and J. L. Kinsey, in *Atom-Molecule Collision Theory–A Guide for the Experimentalist,* edited by R. B. Bernstein (Plenum, New York, 1979).

[52] A. M. Wodtke, E. J. Hintsa, and Y. T. Lee, J. Phys. Chem. **90**, 3549 (1986) ; X. Zhao, R. E. Continetti, A. Yokoyama, E. J. Hintsa, and Y. T. Lee, J. Chem. Phys. **91**, 4118 (1989).

[53] P. Pechukas and J. C. Light, J. Chem. Phys. **42**, 3281 (1965).

[54] C. Wittig, I. Nadler, H. Reisler, J. Catanzarite, and G. Radhakrishnan, J. Chem. Phys. **83**, 5581 (1985).

[55] M. Quack and J. Troe, Ber. Bunsenges. Phys. Chem. **79**, 469 (1975).

[56] M. D. Morse, K. F. Freed, and Y. B. Band, J. Chem. Phys. **70**, 3604 (1979).

[57] G. E. Busch and K.R. Wilson, J. Chem. Phys. **56**, 3626 (1972); A. F. Tuck, J. Chem. Soc., Faraday Trans. **73**, 689 (1977).

[58] S. W. North, D. A. Blank, J. D. Gezelter, C. A. Longfellow, and Y. T. Lee, J. Chem. Phys. **102**, 4447 (1995).

[59] D. H. Mordaunt, D. L. Osborn, and D. M. Neumark, in preparation.

[60] S. W. North, Ph.D. thesis, University of California, Berkeley, 1995.

[61] P. M. Kroger and S. J. Riley, J. Chem. Phys. **67**, 4483 (1977).

[62] S. Körnig, J. H. M. Beijersbergen, and J. Los, J. Phys. Chem. **94**, 611 (1990).

[63] C. E. C. A. Hop and J. L. Holmes, Int. J. of Mass Spectrom. Ion Proc. **104**, 213 (1991).

[64] R. H. Nobes, W. J. Bouma, and L. Radom, J. Am. Chem. Soc. **105**, 311 (1983).

[65] S. W. Benson, *Thermochemical Kinetics* (John Wiley & Sons, New York, 1968), p. 140.

[66] *ibid*, pp. 205-207.

[67] E. A. Rohlfing, private communication.

[68] P. Avouris, W. M. Gelbart, and M. A. El-Sayed, Chem. Rev. **77**, 793 (1977).

[69] J. H. Callomon, J. E. Parkin, and R. Lopex-Delgado, Chem. Phys. Lett. **13**, 125 (1972).

[70] L. Wunsch, H. J. Neusser, and E. W. Schlag, Z. Naturforsch. **36a**, 1340 (1981); U. Schubert, E. Riedle, H. J. Neusser, and E. W. Schlag, J. Chem. Phys. **84**, 6182 (1986).

[71] T. Suzuki and M. Ito, J. Chem. Phys. **91**, 4564 (1989).

[72] A. Helman and R. A. Marcus, J. Chem. Phys. **99**, 5011 (1993).

[73] E. S. Yeung and C. B. Moore, J. Chem. Phys. **60**, 2139 (1974).

5 Photodissociation spectroscopy and dynamics of the HCCO radical

I. Introduction

The ketenyl radical (HCCO) has long been recognized as an important hydrocarbon combustion intermediate.[1] Ketenyl is the major product in the oxidation of acetylene by oxygen atoms:[2-4]

$$HCCH + O \rightarrow HCCO + H \qquad 80 \pm 15\% \qquad (1)$$

$$\rightarrow CH_2\,(\tilde{X}\,^3B_1) + CO \qquad (2)$$

Reaction (1) is a critical step in combustion chemistry because acetylene is a common intermediate in combustion of both aliphatic[5] and aromatic[6] hydrocarbons, and is removed primarily by reaction with oxygen atoms.[7] It is therefore of considerable interest to characterize the spectroscopy and thermochemistry of HCCO, so that its concentration can be monitored in combustion environments, and its reactions can be predicted with greater confidence in combustion kinetics models. Recently, we reported the first unambiguous observation of an ultraviolet spectrum of the ketenyl radical in a fast radical beam photodissociation experiment;[8] prior to this work, spectroscopic information on HCCO had been limited to the microwave region[9,10] and a single vibrational band in the infrared.[11] In this article we present a full discussion of the electronic spectroscopy and photodissociation dynamics of the HCCO radical and compare this system with the closely related CH_2CHO[12] and CH_3O[13] intermediates.

The importance of the HCCO radical in combustion has motivated several kinetics[1-4,14-24] and spectroscopy experiments,[8-11,25-29] as well as theoretical investigations[30-40] over the years. Fenimore and Jones[41] first proposed that HCCO is formed in the reaction of oxygen atoms with C_2H_2, a prediction verified by Jones and Bayes using photoionization mass spectrometry.[14] A crossed molecular beam study of the C_2H_2 + O reaction by Schmoltner *et al.*[19] showed unambiguously that combustion of C_2H_2 proceeds via reactions (1) and (2) under collisionless conditions, although the branching ratio derived does not agree with the currently accepted value.[4] The many kinetic studies involving both the production of HCCO from C_2H_2 + O, and the destruction of HCCO through various reactions have been summarized by Peeters *et al.*[4] Rate constants have been measured for the destruction of HCCO by reaction with O,[22] O_2,[20,21] H,[3,22] C_2H_2,[20,21] NO,[11,15,20] and NO_2.[20] An extensive table of reaction rate constants involved in this system is given in Ref. 16. Taatjes has recently conducted ^{13}C-labeled kinetic studies of the reaction CH(CD) + CO, finding evidence for isomerization of the HCCO adduct via an oxiryl intermediate.[23] Of particular interest to our work, Herbert *et al.*[24] have conducted kinetics measurements from 23 - 584 K on vibrational relaxation of CH ($X(^2\Pi)$; $\nu = 1$) by CO; the fast rates observed are indicative of formation of an HCCO complex.

The ground electronic state of the HCCO radical has been characterized by several spectroscopic methods. Oakes *et al.* obtained the electron affinity of HCCO and determined the heat of formation of the radical by photoelectron spectroscopy of $HCCO^-$.[25] The microwave spectra of HCCO and DCCO due to Endo *et al.*[9] showed that HCCO is bent, although it is effectively described as quasilinear. From this work the ground electronic state was determined to be of $^2A''$ symmetry. The large *a*-axis spin-rotation

constant and its anomalous dependence on K_a were attributed to perturbation of the ground state by a low-lying $^2A'$ state. In the infrared region, Jacox and Olsen[29] assigned a matrix absorption at 2,020 cm^{-1} to HCCO. This band was later identified at 2,023 cm^{-1} in the gas phase and the rotational structure analyzed in detail by Unfried *et al.*[11]

The characterization of the excited electronic states of HCCO has been more problematic. Two electronic absorption bands of HCCO were reported by Krishnamachari and Venkatasubramanian,[26] with band origins at 27,262 and 29,989 cm^{-1}. The carrier of these bands was produced by flash photolysis of oxazole and isoxazole, and their assignment to the HCCO radical has been questioned.[38] A laser induced fluorescence (LIF) spectrum of HCCO, produced by reaction of O atoms with C_2H_2, was also reported,[27] although the fluorescence was later shown to have arisen entirely from CH_2O.[28]

The experimental results on the ground state of HCCO radical are consistent with *ab initio* electronic structure calculations by Goddard,[32] Kim and Shavitt,[34] and Szalay *et al.*,[35,38,39] providing considerable insight into the excited state structure. According to these calculations, the bent ground state, $\tilde{X}$ ($^2A''$), and the linear $\tilde{A}$ ($^2A'$) [$^2\Pi$] state are a Renner-Teller (RT) pair derived from a linear $^2\Pi$ configuration. The *ab initio* splitting between these two states is 981 cm^{-1},[34] in reasonable agreement with that estimated from the microwave analysis. Kim and Shavitt[34] predicted a higher-lying RT pair, the bent $\tilde{B}$ ($^2A'$) state at 33,000 cm^{-1} and the linear $\tilde{C}$ ($^2A''$) [$^2\Pi$] state at 33,300 cm^{-1} above the ground state. Szalay *et al.* [38,39] have recently completed a study of the same electronic states and found both members of the upper RT pair to be linear ($^2\Pi$) at their highest level of theory, with a spin-orbit splitting A_{SO} = -78 cm^{-1}, and an excitation energy relative to

the $\tilde{X}$ state of 36,000 cm^{-1}. Furthermore, Nguyen *et al.*[37] predicted a linear $^2\Sigma^+$ state at $\approx$ 53,000 cm^{-1}. Hu *et al.*[36] have characterized the lowest quartet state of HCCO, a *cis*-bent $\tilde{a}$ (^{4}A″) state, which is predicted to lie 19,000 cm^{-1} above the $\tilde{X}$ (^{2}A″) state. A recent article by Yarkony explores the intersection of the $\tilde{a}$ state with the two low-lying doublet states.[40] Figure 1 shows a qualitative picture of the relevant potential energy surfaces of HCCO based on the calculations in Refs. 36 and 38.

In our experiment, a *mass-selected* source of neutral HCCO radicals is prepared by laser photodetachment of the HCCO$^-$ ion. A second laser then probes the dissociative electronically excited states of the radical, giving us confidence that all our data arises solely from the ketenyl radical. Our experiment directly assesses whether photodissociation occurs; it also identifies and characterizes the nascent photofragments.

For the ketenyl radical, three dissociation channels are energetically allowed for photon energies in this study:

$$\mathrm{HCCO}\,\tilde{X}(^2\mathrm{A}'') \xrightarrow{h\nu} \mathrm{CH}\,X(^2\Pi) + \mathrm{CO}\,X(^1\Sigma^+) \quad \Delta H_0 = 3.14 \pm 0.03\ \mathrm{eV}^{8} \qquad (I)$$

$$\mathrm{CH}\,a(^4\Sigma^-) + \mathrm{CO}\,X(^1\Sigma^+) \quad \Delta H_0 = 3.88 \pm 0.03\ \mathrm{eV}^{8} \qquad (II)$$

$$\mathrm{H}\,(^2S) + \mathrm{CCO}\,\tilde{X}(^3\Sigma^-) \quad \Delta H_0 = 4.36 \pm 0.05\ \mathrm{eV}^{42,43} \qquad (III)$$

Channels *I* and *II* are observed in our experiment. We observe structure in the photofragment yield spectrum of HCCO, and present a rotational analysis of the $\tilde{B}$ ($^2\Pi$) $\leftarrow$ $\tilde{X}$ (^{2}A″) transition. From translational energy and angular distributions of the fragments, we obtain information on the rotational and vibrational product state distributions of the fragments. The experimental results and data analysis are presented in sections III and IV. Based on this data, we develop a dissociation mechanism in section

V, and discuss the dependence of intersystem crossing and internal conversion on photon energy. We also comment on the possibility of utilizing electronic spectroscopy as an *in situ* probe of HCCO in combustion environments.

II. Experimental

The experimental apparatus shown in Fig. 2 has been described in detail

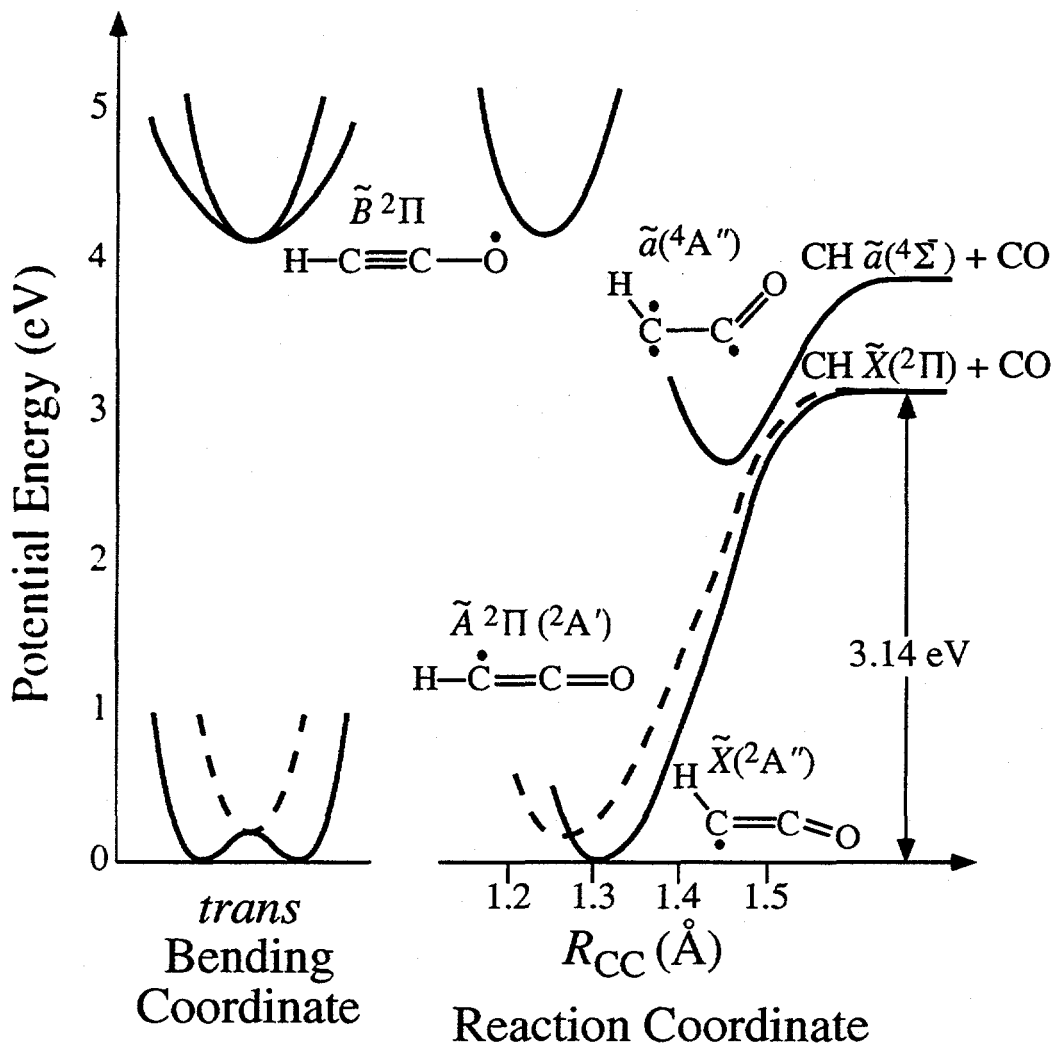


Figure 1: Schematic of potential energy surfaces for the HCCO radical.

elsewhere,[44,45] and only a brief description of the technique is given here. In the first section of the apparatus, a pulsed free jet expansion of the gas mixture C_2H_2:N_2O:O_2:Ne (mole fraction 1:3:6:90) is formed in the source region. $HCCO^-$ is generated by a pulsed electric discharge[46] in the throat of this expansion. The core of the free jet passes through a 3 mm diameter skimmer and the anions are accelerated to 8 keV. Mass-separation of the ions is accomplished using a collinear Bakker[47] type time-of-flight mass spectrometer. After collimation of the ion beam by a 1 mm pinhole, the output of a pulsed dye laser is timed such that it photodetaches only m/e = 41, producing a packet of mass-selected neutral free radicals. Any remaining ions are deflected from the beam. The photodetachment energy ($h\nu$ = 19,230 cm^{-1}) is just above threshold,[25] producing radicals in their zero-point vibrational level.

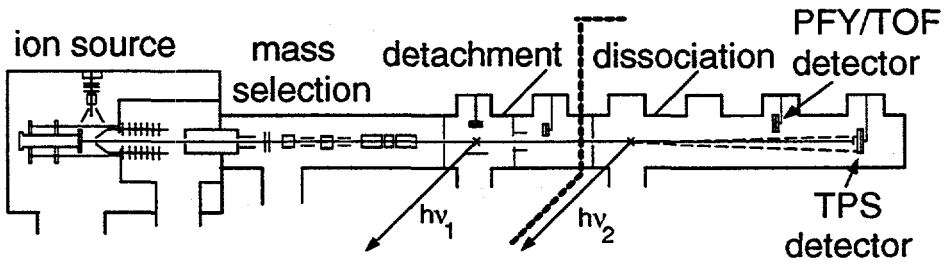


Figure 2: Fast radical beam translational spectrometer. The dotted line separates the radical production section on the left from the actual photodissociation experiment on the right.

In the second section of the apparatus, the radicals are collimated by a final 1 mm pinhole and intersect a second pulsed dye laser operating in the ultraviolet. If photodissociation occurs, fragments recoiling out of the parent radical beam are detected with high sensitivity, *without* an ionization step, using microchannel plate detectors. Photodissociation occurs under collisionless conditions (10^{-9} torr), and two types of

experiments are performed. First, the photofragment yield (PFY) spectrum is obtained by integrating the total fragment signal as a function of photodissociation laser wavelength. Second, at various fixed photon energies, the dissociation dynamics are investigated by detecting both fragments in coincidence from a single parent radical dissociation event. We measure the photofragment translational energy and angular distributions using a time- and position-sensitive (TPS) wedge and strip anode detector.[48,49] The identity of the photofragments is determined by measuring the individual recoils of each fragment from the radical beam axis to give the fragment mass ratio, m_1/m_2, by conservation of linear momentum. Angular distributions of the photofragments are defined with respect to the $\boldsymbol{E}$ vector of the linearly polarized dissociation laser. In the present experiments the translational energy resolution is given by $\Delta E_T / E_T = 2.2\%$.

The doubled output of the photodissociation laser has a bandwidth of ≈ 0.3 cm^{-1} when operated with a grating as its only tuning element. In this mode the PFY spectra are calibrated at many different frequencies throughout the scanned range against the absorption spectrum of I_2,[50] with an absolute accuracy of 1 cm^{-1} or better. The bandwidth of the laser is reduced to ≈ 0.08 cm^{-1} with the incorporation of an intracavity etalon. PFY spectra acquired in this configuration are calibrated against simultaneously collected I_2 absorption spectra yielding an absolute accuracy of 0.05 cm^{-1}.

As discussed previously,[12] the coincidence detection scheme is useful only when $m_1/m_2 \leq \sim 5$. In order to probe for channel *III* products, we utilize a non-coincidence time-of-flight detection scheme when dissociating the isotope DCCO of the ketenyl radical. Unlike the coincidence detection method described above, this experiment would be sensitive to the product channel D + CCO, and has been used to identify a D atom loss

channel in the photodissociation of the vinoxy radical. No evidence for channel *III* is observed. For experimental reasons it is difficult to place an upper bound on the importance of this channel, but it is almost certainly not a major pathway in the ultraviolet dissociation of HCCO.

III. Results

A. Photofragment yield spectra

The PFY spectrum of HCCO from 33,000 to 48,000 cm^{-1}, shown in Fig. 3, indicates the energies at which HCCO absorbs a photon and dissociates. In the limit that the quantum yield for dissociation $\Phi_{Diss} = 1$, the PFY spectrum is identical to the absorption spectrum. This spectrum shows unambiguously that dissociation occurs after UV excitation of HCCO. The lowest energy at which we observe dissociation is $h\nu \approx 33{,}400$ cm^{-1}, which is assigned below as the origin of this electronic band.

The PFY spectrum shows clear vibrational structure towards the red, as seen more clearly in Fig. 4, which displays the first 4,000 cm^{-1} of the spectrum. In addition, rotational structure is resolved at the origin and in a few vibrational bands; two such spectra are shown in Fig. 5. As seen in Fig. 3, the photofragment yield increases significantly in intensity with increasing photon energy, concurrent with broadening of the vibrational structure. The rise in the spectrum's baseline seen in Fig. 3 is not an artifact of the experiment; this signal, in addition to the structured peaks, arises from primary photofragments of HCCO. Above 41,000 cm^{-1}, no sharp vibrational transitions are

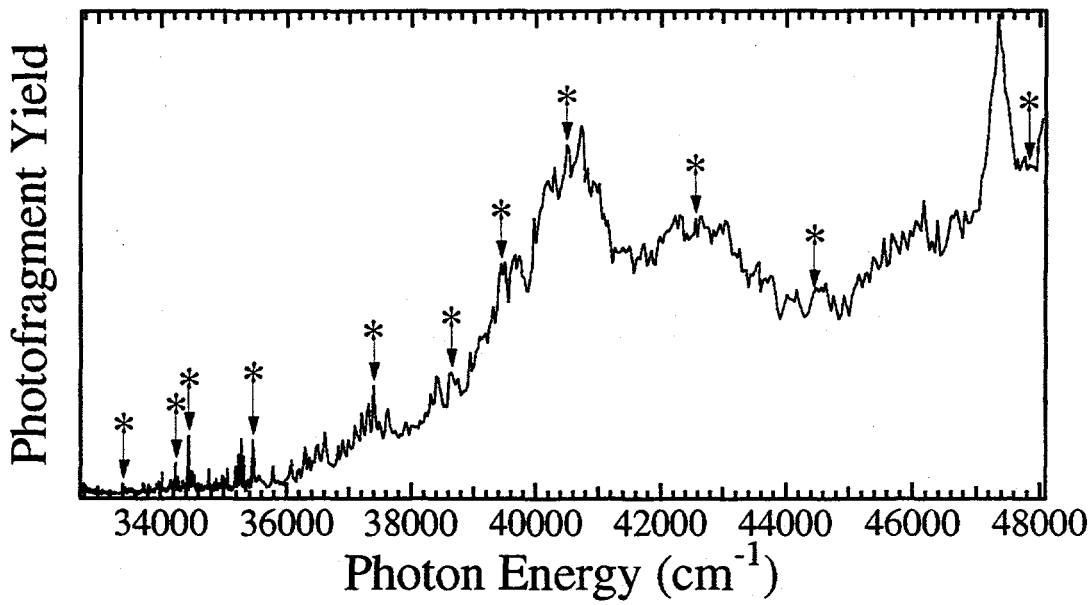


Figure 3: Overall photofragment yield (PFY) spectrum of HCCO. Photodissociation dynamics data are acquired at the peaks marked with an asterisk. The lowest energy asterisk marks the origin transition.

apparent, although broad undulations in the spectrum continue to the highest energies studied.

The spectroscopic information we obtain from HCCO is derived from the structured region in the first 4,000 cm^{-1} of the PFY spectrum, shown in Fig. 4. The structure represents extended vibrational progressions in the excited electronic state. In this spectrum, the first 1,900 cm^{-1} were acquired with a laser step size of 0.3 cm^{-1}, equal to the laser bandwidth. Beyond this region the step size is increased to 3 cm^{-1}. The coarser step size expedites data acquisition with little loss of vibrational structure information. The two lowest energy peaks (marked by $*$ in Fig. 4) are split by $\approx$ 40 cm^{-1}, and can be tentatively identified at this resolution as two R-branch band heads. The large number of transitions in the first 2,000 cm^{-1} indicates that several vibrational modes are active in the excited electronic state.

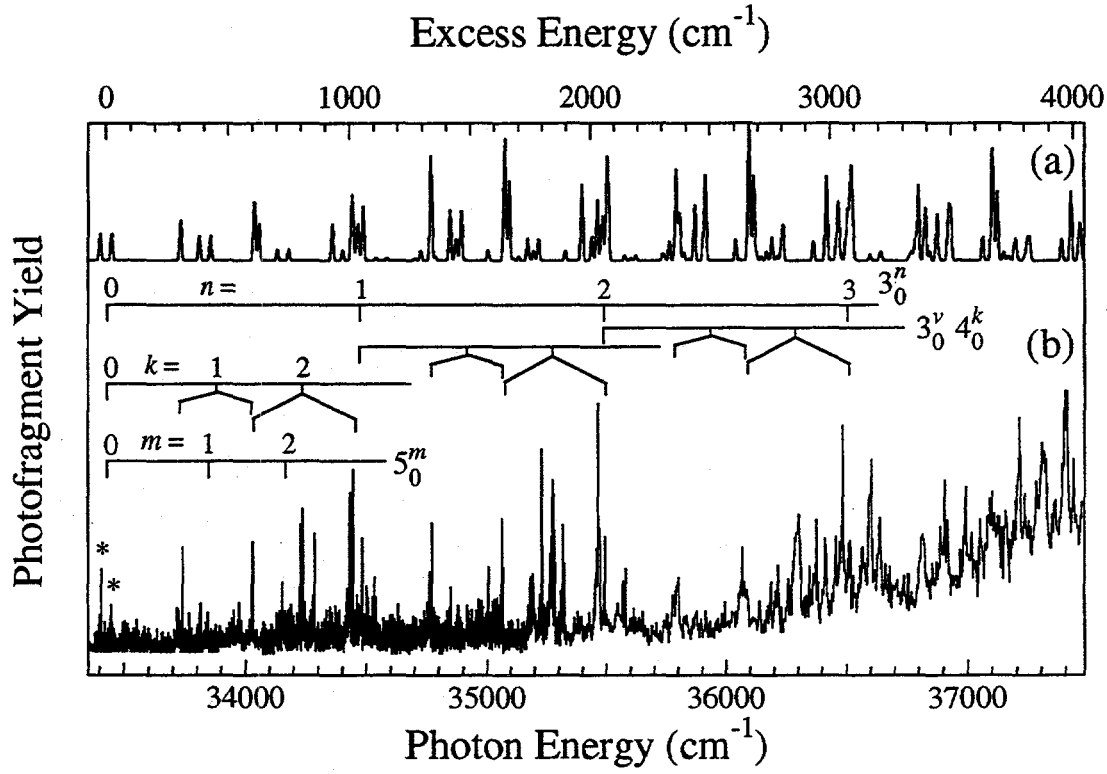


Figure 4: Vibrational structure in the first 4,000 cm^{-1} of the PFY spectrum. Excess energy is given relative to the rovibronic origin at 33,423.92 cm^{-1}. The vibrational progression in the ν_3 CCO symmetric stretch and the two bending modes ν_4 and ν_5 are indicated by the combs, with the Renner-Teller splitting shown for ν_4.

A few of the transitions in Fig. 4 were examined at higher resolution (0.08 cm^{-1} laser bandwidth). Two such rotationally resolved spectra are shown in Fig. 5. The spectrum in Fig. 5a is assigned in section IV-A as the electronic origin 0_0^0 transition, and is the source of the most detailed spectroscopic information we obtain on the HCCO radical. Upon inspection of the lower frequency band centered at 33,403 cm^{-1}, one can easily recognize *P*, *Q*, and *R* branches with a prominent band head occurring in the *R* branch. A splitting of the lines in the *P* branch is also visible below 33,397 cm^{-1}. The nature of the higher frequency branch with a band head at 33,447 cm^{-1} is perhaps not as easily recognized at first glance and will be discussed later. The experimental linewidths are instrumentally narrow (i.e., 0.08 cm^{-1}) in Fig. 5a.

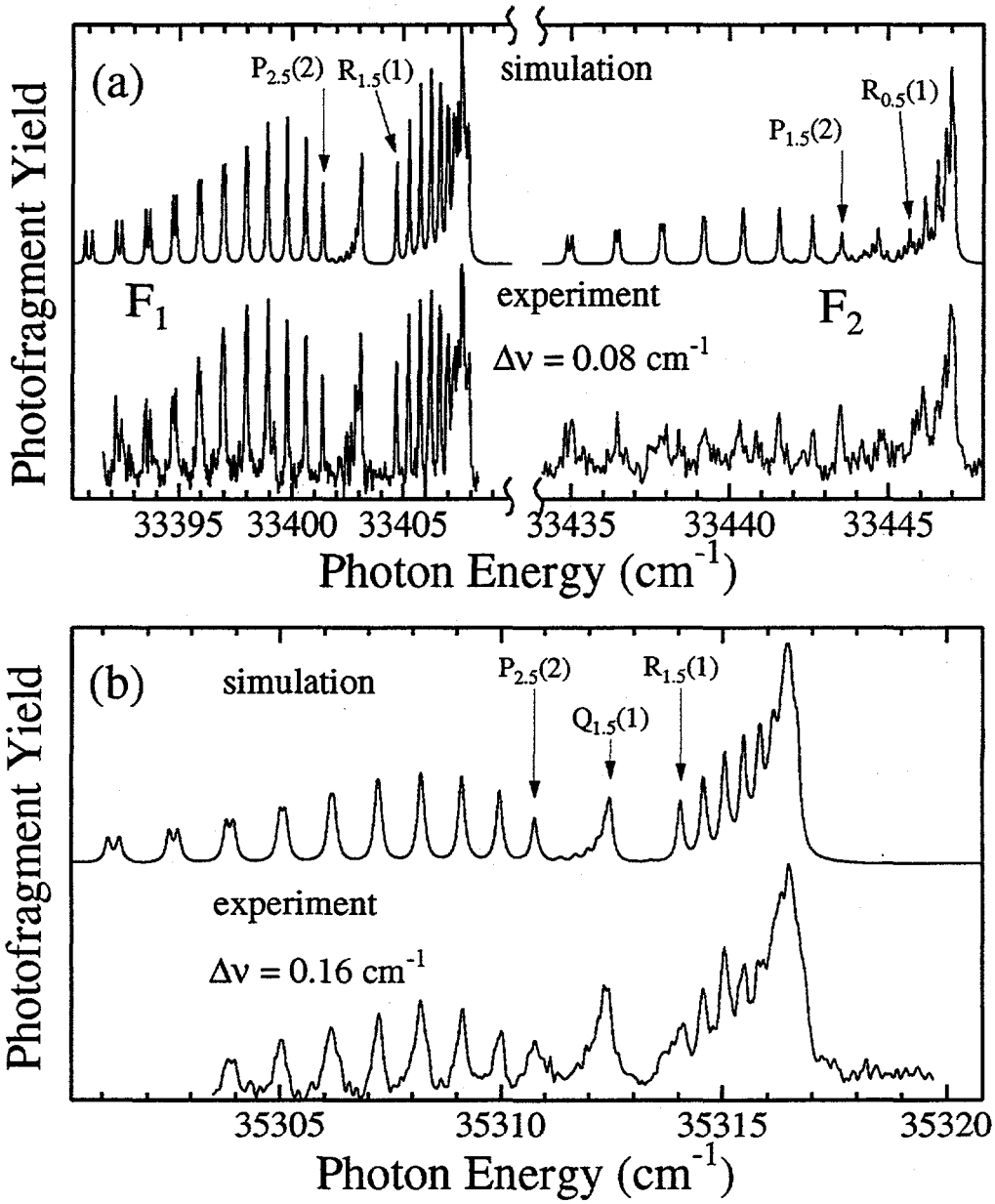


Figure 5: Rotationally resolved PFY spectra of HCCO. (a) The origin transition shows splitting into both F_1 and F_2 branches, with asymmetry doubling resolved in the F_1 branch. The experimental linewidth is instrument limited at 0.08 cm^{-1}. (b) The F_1 branch of an unassigned vibrational transition ≈ 1,900 cm^{-1} above the origin. The experimental linewidth is broadened to 0.16 cm^{-1}.

While many of the vibrational bands do not reveal rotational structure at our resolution, a few vibrationally excited bands were resolved, one of which is shown in Fig. 5b, ≈ 1,900 cm^{-1} above the origin transition. Again a simple *P*, *Q*, *R*, structure is apparent,

although in this band the FWHM of the narrowest line is 0.16 cm^{-1}, twice the width of the photodissociation laser.

B. Translational energy and angular distributions

Once the dependence of photodissociation on photon energy has been mapped out for the ketenyl radical, its photodissociation dynamics can be explored at selected photon energies using the TPS detector. First, the identity of the photoproducts is ascertained by conservation of linear momentum. At all photon energies, the only fragments observed are CH + CO, as expected for the coincidence detector configuration. With knowledge of the fragment masses, we obtain the coupled translational energy and angular distribution, $P(E_T,\theta)$, by direct inversion of the data.[44] The two dimensional $P(E_T,\theta)$ distribution can then be separated into an angle-independent translational energy distribution $P(E_T)$, and an energy-dependent anisotropy parameter $\beta(E_T)$,[51] that describes the angular distribution of the fragments:

$$P(E_T,\theta) = P(E_T)\cdot\{1 + \beta(E_T)\cdot P_2(\cos\theta)\} \tag{3}$$

where $P_2(\cos\theta)$ is the second Legendre polynomial. The limiting cases of $\sin^2\theta$ and $\cos^2\theta$ angular distributions are given by β = -1 and +2, respectively. Together, the $P(E_T)$ and $\beta(E_T)$ distributions can yield detailed information on the dynamics of the dissociation process. The procedure for extracting the $P(E_T)$ and $\beta(E_T)$ distributions from the data have been described in detail elsewhere[44] Photodissociation dynamics experiments were performed at each photon energy marked with (*) in Fig. 3. The resulting $P(E_T)$ and $\beta(E_T)$ distributions are shown in Fig. 6.

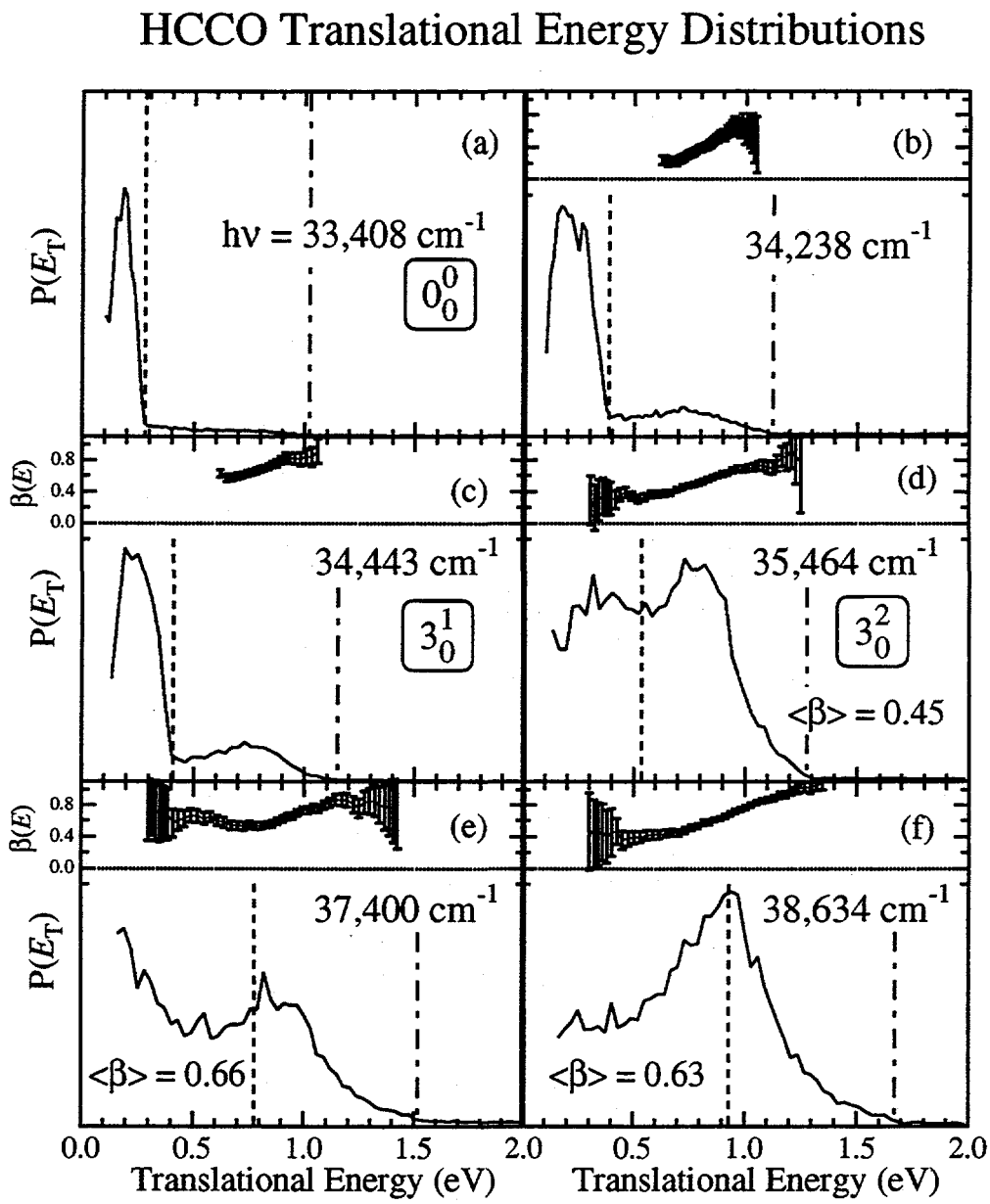


Figure 6: Translational energy distributions, $P(E_T)$, and anisotropy parameter distributions, $\beta(E_T)$, for the marked transitions in Fig. 3. In each case the photon energy is given. The $\beta(E_T)$ distributions are not determined for the spin-forbidden products (see text). For greater clarity, $\beta(E_T)$ is not plotted on its full range of -1 to 2. Average anisotropies $\langle\beta\rangle$ are indicated at each photon energy.

The $P(E_T)$ distributions show several trends with increasing photon energy. Two features are visible at the lowest excitation energies in Figs. 6a-c: a dominant peak centered at E_T = 0.20 - 0.25 eV, and a weak, broad feature centered at E_T = 0.75 eV that

gains intensity with photon energy. As discussed in our previous paper, and in more detail in Section IV-C below, these features correspond to channels *II* and *I*, respectively. However, by $h\nu$ = 35,464 cm^{-1}, the two features appear to have approximately the same intensity and are difficult to separate. As the photon energy increases further, the appearance of the spectra change, and they are dominated by a single peak at $E_T \approx 0.8$ - 0.9 eV superimposed on a broad background. This peak position is essentially independent of excitation energy over a 10,000 cm^{-1} range. At the highest energy studied, 47,831 cm^{-1}, the width of the $P(E_T)$ distribution broadens considerably. Another important trend in the $P(E_T)$ distributions is that each increase in $h\nu$ is accompanied by an equivalent increase of the highest translational energy observed, E_T^{max}. This result is a direct consequence of energy conservation, governed by the relation $E_T^{max} = h\nu - D_0$(HC–CO).

The energy dependent anisotropy parameter, $\beta(E_T)$, also shows interesting trends as a function of both E_T and $h\nu$. First, it should be noted that in all cases $\beta(E_T) \geq 0$, implying a parallel electric dipole transition.[51] Unfortunately, for the spin-forbidden process of channel *II*, the limited range of laboratory scattering angles which can be detected for these small recoils prevents a confident determination of the anisotropy, so $\beta(E_T)$ is not reported for this feature. At low photon energy, $\beta(E_T)$ increases with E_T, whereas at higher photon energy (Figs. 6g-j), there is a broad maximum in $\beta(E_T)$ whose center approximately coincides with the peak at 0.9 eV in the $P(E_T)$ distribution. For $h\nu \geq$ 35,464 cm^{-1}, the energy-averaged anisotropy $\langle\beta\rangle$ (Fig. 6) tends to be an increasing function of photon energy, with individual values of β as high as $\beta(E_T = 0.9 \text{ eV}) = 1.0$ for

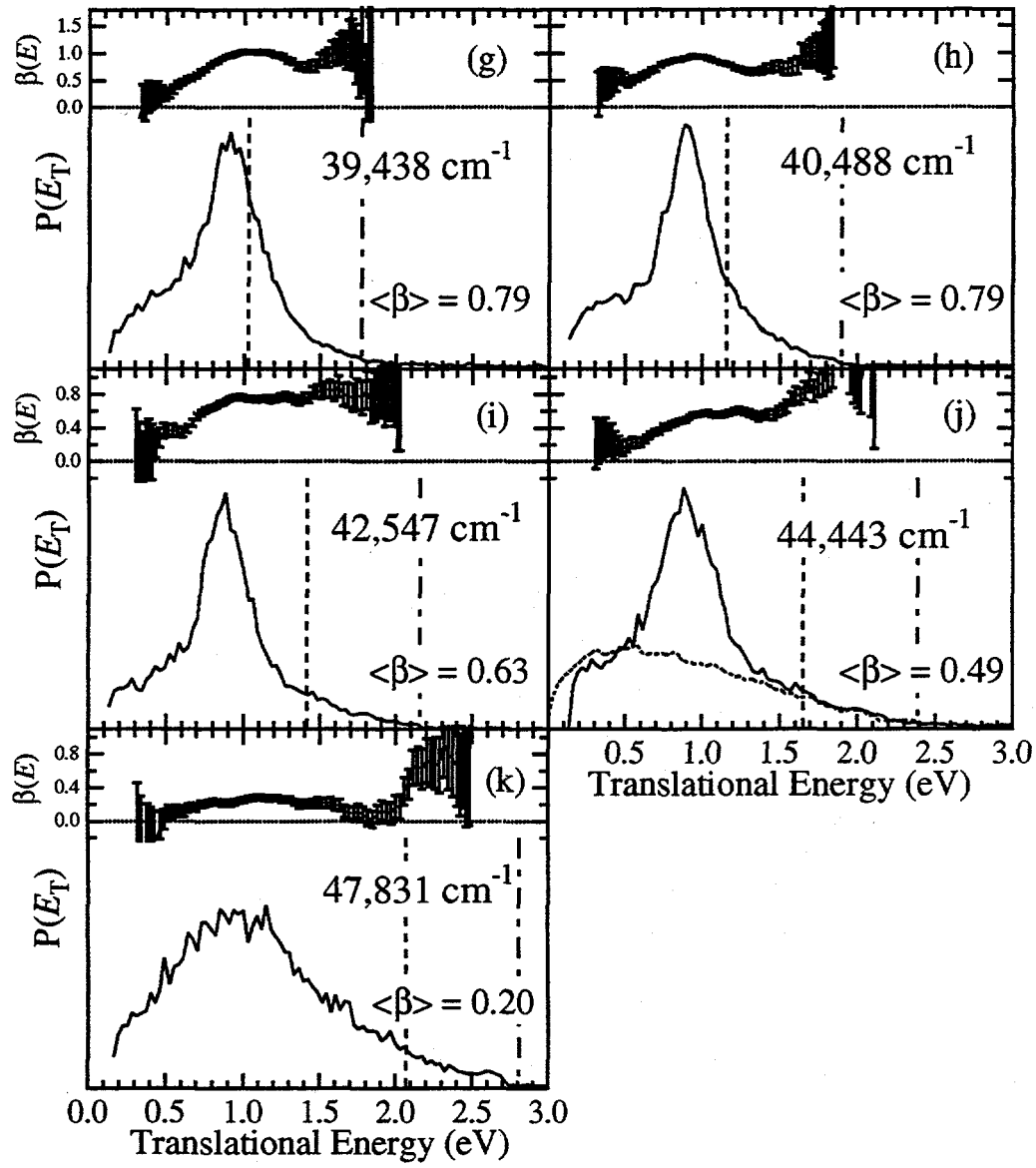


Figure 6, continued.

$h\nu$ = 40,488 cm^{-1}. At still higher photon energies $\langle\beta\rangle$ decreases in magnitude again. These trends are discussed further in Section V.

C. Electronic structure calculations

One issue which has not been addressed in previous *ab initio* calculations on the HCCO radical is the nature of the $\tilde{X}$ (^{2}A″) state as the C–C bond is elongated and finally

dissociated. We have performed *ab initio* calculations to determine whether a barrier exists on the way to product formation. For reference, the molecular orbital configurations of the relevant valence states of HCCO are given in Table I. In order to properly describe the dissociation into CH + CO, we have used the multi-configurational complete active space self consistent field[52] (CASSCF) method from the Gaussian 94 package[53] of programs. An active space of nine electrons in eight orbitals [CAS(9,8)] was chosen as the three $a''p_\pi$ orbitals, the three $a'p_\pi$ orbitals, and the C–C σ and σ^* orbitals. The 6-31G* basis set was used for this work.

The r_{CC} bond length was increased incrementally, optimizing at each increment the remaining degrees of freedom, in order to determine the minimum energy pathway to products. Referring to Fig. 1, we find no evidence for a barrier on the $\tilde{X}\,(^2A'')$ surface

Orbital Character	$\tilde{X}$ $(^2A'')$	$\tilde{A}$ $(^2A')$ $[^2\Pi]$	$\tilde{a}$ $(^4A'')$	$\tilde{B}$ $(^2\Pi)^a$
O (1s)	$(1a')^2$	$(1\sigma)^2$	$(1a')^2$	$(1\sigma)^2$
C (1s)	$(2a')^2$	$(2\sigma)^2$	$(2a')^2$	$(2\sigma)^2$
C (1s)	$(3a')^2$	$(3\sigma)^2$	$(3a')^2$	$(3\sigma)^2$
O (2s)	$(4a')^2$	$(4\sigma)^2$	$(4a')^2$	$(4\sigma)^2$
CH (σ)	$(5a')^2$	$(5\sigma)^2$	$(5a')^2$	$(5\sigma)^2$
CO (σ)	$(6a')^2$	$(6\sigma)^2$	$(6a')^2$	$(6\sigma)^2$
CC (σ)	$(7a')^2$	$(7\sigma)^2$	$(7a')^2$	$(7\sigma)^2$
CO $(\pi_x)^b$	$(8a')^2$	$(1\pi_x)^2$	$(8a')^2$	$(1\pi_x)^1$ $(1\pi_x)^2$
CO (π_y)	$(1a'')^2$	$(1\pi_y)^2$	$(1a'')^2$	$(1\pi_y)^2$ $(1\pi_y)^1$
CCO (π_x)	$(9a')^2$	$(2\pi_x)^1$	$(9a')^1$	$(2\pi_x)^2$
CCO (π_y)	$(2a'')^1$	$(2\pi_y)^2$	$(2a'')^1$	$(2\pi_y)^2$
CCO (π_x^*)	$(10a')^0$	$(3\pi_x)^0$	$(10a')^1$	$(3\pi_x)^0$
CCO (π_y^*)	$(3a'')^0$	$(3\pi_y)^0$	$(3a'')^0$	$(3\pi_y)^0$

Table I: Molecular orbital configurations of HCCO electronic states.

[a] At linearity the π_x and π_y orbitals of the $\tilde{B}$ state are degenerate. In the case of strong Renner-Teller coupling such that the lower RT state is nonlinear, as found in Ref. 34, the two resulting electronic states are described by the left and right configurations, respectively, of the CO (π) orbitals in this column.

[b] The π orbitals of the CCO backbone are analogous to the three π orbitals of linear C_3, which have zero, one, and two nodes in the *xy* plane, respectively, as a function of increasing energy.

along the minimum energy path out to the longest C–C bond length calculated, r_{CC} = 2.7 Å. If a barrier of significant magnitude existed, it would almost certainly occur for r_{CC} < 2.7 Å. The absence of an exit barrier on the ground state is supported experimentally by rate constant measurements over a wide range of temperatures for the relaxation of CH($X(^2\Pi)$; v = 1) by CO and N_2.[54] Because the $\tilde{X}\,(^2A'')$ and the $\tilde{A}\,(^2A')$ states are a Renner-Teller pair, they correlate to the same doubly degenerate asymptote CH $X(^2\Pi)$ + CO $X(^1\Sigma^+)$, and we do not expect a barrier on the $\tilde{A}$ state surface because the RT splitting should decrease as r_{CC} increases.

A second issue addressed through *ab initio* calculations is the possibility of an intersection of the HCCO $\tilde{a}\,(^4A'')$ PES with the $\tilde{B}$ $(^2\Pi)$ PES. These calculations probe regions of the $\tilde{a}\,(^4A'')$ surface far from equilibrium, where excitations other than doubles are known to be more important for an accurate description of the wavefunction. Therefore we have chosen the quadratic configuration interaction method with perturbative treatment of triple excitations[55] QCISD(T) / 6-31G**. In agreement with Hu *et al.*,[36] we find the lowest state of HCCO $\tilde{a}\,(^4A'')$ is a *cis*-bent structure with equilibrium geometry ∠HCC = 131.37°, ∠CCO = 125.14°, r_{CC} = 1.450 Å, r_{CO} = 1.206 Å, r_{CH} = 1.085 Å. At the (linear) equilibrium geometry of the $\tilde{B}$ state, the intersection with the $\tilde{a}\,(^4A'')$ surface occurs 2.4 eV above the minimum energy of the $\tilde{B}$ state, although this energy gap decreases dramatically with bending motion away from linearity.

IV. Analysis

A. Rotational structure

Analysis of the rotationally resolved PFY spectra in Fig. 5 is the first step in characterizing the electronic transition we observe in HCCO. Fortunately, the $\tilde{X}\,(^2A'')$ ground state of ketenyl is characterized from the microwave spectroscopy of Endo *et al.*[9] According to this work, the ground state has a linear or nearly linear CCO backbone, but $\angle$HCC = 138.7°, placing the H atom well away from the *a*-axis. Due to the low mass of the H atom, HCCO $\tilde{X}\,(^2A'')$ is very nearly a prolate symmetric top, with asymmetry parameter κ_{asym} = -0.9998. The microwave rotational spectra were analyzed using a Hund's case (b) Hamiltonian, although Unfried *et al.* point out that for $K_a \neq 0$ the coupling is closer to case (a).[11] However, it is less convenient to use a case (a) Hamiltonian since the ground state is non-linear, and we therefore follow the precedent of Endo *et al.*,[9] utilizing a case (b) Hamiltonian to analyze the rotational structure of ketenyl.

With this background, we can examine in more detail the PFY spectrum of the band centered at 33,403 cm^{-1} in Fig. 5a, whose line positions are given in Table II. Based on the Hönl-London factors for a symmetric top,[56] the fact that a *Q* branch is observed but does not dominate the spectrum implies a parallel, *a*-type rotational transition for which $\Delta K_a = 0$. Furthermore, the electronic transition cannot be $K_a' = 0 \leftarrow K_a'' = 0$ because: (i) in this case the *Q* branch would be vanishingly small, (forbidden in the symmetric top limit), and (ii) asymmetry doubling, which we resolve in the *P* branch, is not allowed for $K_a = 0$ levels.

$N'_{K'_aK'_c}$	$N''_{K''_aK''_c}$	branch	observed[a]	obs. - calc.
1_{11}	2_{12}	F_1	33401.39	-0.01
1_{10}	2_{11}	F_1	33401.39	-0.00
2_{12}	3_{13}	F_1	33400.63	0.01
2_{11}	3_{12}	F_1	33400.63	0.02
3_{13}	4_{14}	F_1	33399.80	0.00
3_{12}	4_{13}	F_1	33399.80	0.03
4_{14}	5_{15}	F_1	33398.94	0.02
4_{13}	5_{14}	F_1	33398.94	0.06
5_{15}	6_{16}	F_1	33398.00	-0.01
5_{14}	6_{15}	F_1	33398.00	0.06
6_{16}	7_{17}	F_1	33396.96	-0.07
6_{15}	7_{16}	F_1	33396.96	0.03
7_{17}	8_{18}	F_1	33395.99	0.01
7_{16}	8_{17}	F_1	33395.86	-0.00
8_{18}	9_{19}	F_1	33394.87	-0.01
8_{17}	9_{18}	F_1	33394.71	-0.00
9_{19}	$10_{1,10}$	F_1	33393.71	0.01
9_{18}	10_{19}	F_1	33393.51	0.01
$10_{1,10}$	$11_{1,11}$	F_1	33392.46	-0.00
10_{19}	$11_{1,10}$	F_1	33392.22	0.00
1_{10}	1_{11}	F_1	33403.08	-0.01
1_{11}	1_{10}	F_1	33403.08	-0.01
3_{12}	3_{13}	F_1	33402.87	-0.00
3_{13}	3_{12}	F_1	33402.87	0.01
4_{13}	4_{14}	F_1	33402.67	-0.03
4_{14}	4_{13}	F_1	33402.67	0.00
5_{14}	5_{15}	F_1	33402.45	-0.02
5_{15}	5_{14}	F_1	33402.45	0.03
2_{12}	1_{11}	F_1	33404.69	-0.01
2_{11}	1_{10}	F_1	33404.69	-0.01
3_{13}	2_{12}	F_1	33405.23	-0.02
3_{12}	2_{11}	F_1	33405.23	-0.01
4_{14}	3_{13}	F_1	33405.75	-0.01
4_{13}	3_{12}	F_1	33405.75	0.00
5_{15}	4_{14}	F_1	33406.22	-0.00
5_{14}	4_{13}	F_1	33406.22	0.02
6_{16}	5_{15}	F_1	33406.62	-0.02
6_{15}	5_{14}	F_1	33406.62	0.02
7_{17}	6_{16}	F_1	33406.98	-0.03
7_{16}	6_{15}	F_1	33406.98	0.04
1_{11}	2_{12}	F_2	33443.5	-0.04
1_{10}	2_{11}	F_2	33443.5	-0.03
2_{12}	3_{13}	F_2	33442.63	0.02
2_{11}	3_{12}	F_2	33442.63	0.04
3_{13}	4_{14}	F_2	33441.56	-0.01
3_{12}	4_{13}	F_2	33441.56	0.01
5_{15}	6_{16}	F_2	33439.22	0.00
5_{14}	6_{15}	F_2	33439.22	0.06
7_{17}	8_{18}	F_2	33436.47	-0.04
7_{16}	8_{17}	F_2	33436.47	0.08
8_{18}	9_{19}	F_2	33435.03	0.01
8_{17}	9_{18}	F_2	33434.81	-0.05

Table II: Rotational assignments and frequencies for 0_0^0.

[a] For transitions where the asymmetry doubling is not resolved, both $K_c = N$ and $K_c = N - 1$ components are assigned to the single observed line, so that these transitions have the proper statistical weight.

It is not possible to obtain a self-consistent set of combination differences using the integral quantum number N ($= J - S$) to describe the rotational term values. In contrast, self-consistency is obtained using half-integral J values in the combination differences, identifying the first lines in the P and R branches as $P_{2.5}(2)$ and $R_{1.5}(1)$, where the notation is $\Delta J_{J''}(N'')$. All the transitions in the 33,403 cm^{-1} branch have $J = N + 1/2$ in both ground and excited states, i.e., they represent $F_1 \leftarrow F_1$ components. Although it is perhaps not clear on inspection, the feature in Fig. 5a with a band head at 33,447 cm^{-1} corresponds to the $F_2 \leftarrow F_2$ components of this same transition.

The combination difference analysis implies that the spin is coupled to the internuclear axis, following Hund's case (a) coupling. Nevertheless, the (bad) quantum number N'' is a useful label, since it takes on all integer values $N'' \geq K_a''$. It follows, based on the identity of the first rotational transitions, that the band in Fig. 5a must be assigned as $K_a' = 1 \leftarrow K_a'' = 1$. The fact that we can observe any asymmetry doubling at all (given our limited laser resolution) for such a nearly prolate asymmetric top supports this assignment, since asymmetry doubling is strongest for $K_a = 1$.

At this point we need to determine if the upper electronic state is linear or whether it, too, is a nearly prolate top analogous to the $\tilde{X}\,(^2A'')$ state. As discussed by Herzberg,[56] there are several characteristics which differentiate the rotational spectrum of a bent $\leftarrow$ bent vs. a linear $\leftarrow$ bent electronic transition. The two most important features in the HCCO spectrum are (a) the absence of a $K_a' = 0 \leftarrow K_a'' = 0$ sub-band at the vibrational band origin, and (b) the very large apparent spin-rotation splitting in the upper state, as evidenced by the 47 cm^{-1} separation between the $F_1 \leftarrow F_1$ and $F_2 \leftarrow F_2$

components in Fig. 5a. If the upper state were an asymmetric rotor, with an A rotational constant similar to the ground state, one would expect to see a strong $K_a' = 0 \leftarrow K_a'' = 0$ sub-band lying very close to the observed $K_a' = 1 \leftarrow K_a'' = 1$ sub-band in Figure 5. If, however, the upper state were a linear $^2\Pi$ state, then $K_a' \equiv K' = 1$ for *all* rotational levels associated with the vibrational ground state, where $K' = |\Lambda \pm l|$ is the projection of the electronic orbital and vibrational angular momenta on the symmetry axis, with $\Lambda = 1$ and $l = 0$ for the vibrational ground state. In other words, the $K_a' = 0 \leftarrow K_a'' = 0$ sub-band does not exist at the vibrational origin if the upper state is $^2\Pi$.

Moreover, the large upper state "spin-rotation" splitting ($\approx$ 47 cm^{-1}) observed in HCCO would be difficult to rationalize if the excited state were bent. As a point of reference, we note that in the vinoxy radical (CH_2CHO) $\tilde{B}\,(^2A'') \leftarrow \tilde{X}\,(^2A'')$ transition, where both states are nearly prolate asymmetric tops, the spin-rotation splittings are 0.057 and 0.029 cm^{-1} for the upper and lower states, respectively, much smaller than in HCCO.[57,58] In fact, for vinoxy the F_1 / F_2 splittings are not even resolved in the $K_a'=1 \leftarrow K_a''=1$ transition using a laser bandwidth similar to ours (0.06 cm^{-1}).[57] However, if the excited state in HCCO were a linear $^2\Pi$ state, the first-order orbital angular momentum would not be quenched, and the very large splitting between the F_1 and F_2 components is naturally explained as a spin-orbit splitting, A_{SO}, between the $^2\Pi_{1/2}$ and $^2\Pi_{3/2}$ levels in the upper state.

Based on this evidence, we analyze the spectra in Fig. 5 as transitions from $K_a'' = 1$ levels of a slightly asymmetric rotor to the $K' = 1$ levels of a linear $^2\Pi$ state. For the extraction of spectroscopic constants, we choose the same Hamiltonian as Unfried *et al.*:[11]

$$\mathbf{H} = \mathbf{H}_{\text{rot}} + \mathbf{H}_{\text{sr}} \tag{4}$$

where $\mathbf{H}_{\text{rot}}$ is a Watson *S*-reduced asymmetric rotor Hamiltonian,[59] and $\mathbf{H}_{\text{sr}}$ accounts for the spin-rotation interaction.[60,61] The matrix elements for this Hamiltonian using a case(b) symmetric top basis are given in Table III. Due to the lower resolution of the present work compared to microwave spectroscopy, several of the higher order terms for the ground state are not required.

Once it is recognized that the band at 33,403 cm^{-1} corresponds to the $F_1 \leftarrow F_1$ transitions, while the band at 33,445 cm^{-1} corresponds to the $F_2 \leftarrow F_2$ transitions, quantum numbers can be assigned to all the unblended lines and the spectroscopic parameters extracted. The molecular constants are fit to the data using a non-linear least-squares method, with the results given in Tables II and IV. In all cases we fix the values of the ground state parameters to their microwave values,[9] because a simultaneous fit of both electronic states gives ground state constants which agree with microwave values to within our statistical uncertainty. The least-squares fit provides our best values for the rotational constant B', the spin-rotation constants κ' ($\equiv \varepsilon'_{aa} - \varepsilon'_{bb}$) and μ' ($\equiv \varepsilon'_{bb}$), and the band origin ν_0, of the linear excited state. It should be noted that the line doubling in the *P* branch of the F_1 component is entirely accounted for by the asymmetry doubling term $(B'' - C'')/4$ of the $\tilde{X}\,(^2A'')$ state. We cannot discern any evidence for Λ-doubling in the excited state at our resolution.

matrix element	$\mathbf{H}_{rot} = \left(\frac{B+C}{2}\right)\mathbf{N}^2 - D_N\mathbf{N}^4 + \left(\frac{B-C}{4} + d_1\mathbf{N}^2\right)\left[(N_+)^2 + (N_-)^2\right]$	$\mathbf{H}_{sr} = \frac{1}{2}\sum_{\alpha,\beta}\varepsilon_{\alpha\beta}(N_\alpha S_\beta + S_\beta N_\alpha)$
$\langle N,K \mid \mathbf{H} \mid N,K \rangle^a$	$\left(\frac{B+C}{2}\right)N(N+1) - D_N[N(N+1)]^2$	$\frac{-\Gamma}{2N(N+1)}\alpha_s K^2 - \frac{\Gamma}{2}\beta_s$
$\langle N,K\pm 2 \mid \mathbf{H} \mid N,K \rangle$	$\left\{\left(\frac{B-C}{4}\right) + d_1 N(N+1)\right\}\delta$	$-\frac{\Gamma\gamma_s}{2N(N+1)}\delta$
$\langle N-1,K \mid \mathbf{H} \mid N,K \rangle$	--	$-\frac{K}{2N}\sqrt{N^2-K^2}\alpha_s$
$\langle N-1,K\pm 2 \mid \mathbf{H} \mid N,K \rangle$	--	$\mp\frac{\gamma_s}{2N}\sqrt{(N\mp K-1)(N\pm K+1)(N\mp K-2)(N\mp K)}$

Table III: Rotation and spin-rotation matrix elements

$\alpha_s = \kappa = \{\varepsilon_{aa} - (\varepsilon_{bb} + \varepsilon_{cc})/2\}$

$\beta_s = \mu = (\varepsilon_{bb} + \varepsilon_{cc})/2$

$\gamma_s = (\varepsilon_{bb} - \varepsilon_{cc})/4$

$\delta = \sqrt{[N(N+1) - K(K\pm 1)][N(N+1) - (K\pm 1)(K\pm 2)]}$

[a] The rotational term $\{A - (B+C)/2\}K^2$ has been subsumed into the band origin since we only detect $\Delta K = 0$ transitions. In this case the linear molecule Hamiltonian is obtained from the asymmetric rotor Hamiltonian by setting $B = C$ and $\varepsilon_{bb} = \varepsilon_{cc}$.

A simulation based on the constants in Table IV is shown in the upper panel of Fig. 5a. The intensities of the simulated lines were obtained from the Hönl-London factors for a symmetric top. The best match with the data is found for a rotational temperature T = 35 K. Taking into account the large rotational constant A'' = 41.46 cm^{-1}, only those ground state levels with K_a'' = 0 or 1 should be significantly populated in the radical beam, since k_bT = 24 cm^{-1}. The simulated stick spectrum is convoluted with a 0.08 cm^{-1} FWHM Gaussian lineshape representing the laser bandwidth with the etalon installed, and the agreement with the measured linewidths is quantitative.

By the same procedure, the band in Fig. 5b with ≈ 1,900 cm^{-1} of vibrational energy can be assigned, the spectroscopic constants extracted , and the spectrum simulated, with the results given in Table IV. In this case only the spectrum of the F_1 branch was obtained. The most striking feature of this band is the increase in linewidth to $\Delta\nu$ = 0.16 cm^{-1}, which is twice the instrumental resolution.

constant	$\tilde{X}(^2A'')$ [a]	$\tilde{B}(^2\Pi)\ 0_0^0$	$\tilde{B}(^2\Pi)$ 1,900 cm^{-1} band
A	41.46	--	--
B	0.3634777	0.3242 ± 0.0002	0.3201 ± 0.0003
C	0.3591306	--	--
κ [b]	-8.266	-46.76 ± 0.04	--
μ [c]	-0.000489	-0.06 ± 0.03	--
ν_0	--	33423.92 ± 0.02	35332.82 ± 0.02 [d]

Table IV: Spectroscopic constants

[a] Ground state constants were fixed at these values from Ref. 9.
[b] $\kappa = \varepsilon_{aa} - (\varepsilon_{bb} + \varepsilon_{cc})/2$
[c] $\mu = (\varepsilon_{bb} + \varepsilon_{cc})/2$
[d] The value of κ was assumed to be identical to the value obtained in the origin transition.

In summary, the rotationally resolved PFY spectra of HCCO indicate that the excited electronic state is a linear $^2\Pi$ state. Our results are entirely consistent with the molecular constants derived from microwave spectroscopy for the $\tilde{X}\,(^2A'')$ state. A large *a*-axis spin-rotation constant, ε'_{aa} = -46.82 ± 0.05 cm^{-1}, is observed; this is more naturally interpreted as the upper state spin-orbit splitting, A_{SO} = -46.82 cm^{-1}. For the vibrationally excited rotational band in Fig. 5b, the linewidths are twice those of the origin transition.

The molecular structure of the excited $^2\Pi$ state is determined by three parameters, the H–C, C–C, and C–O bond lengths. However, only one structural parameter, the rotational constant B', has been measured. Therefore the determination of the experimental molecular structure requires measurements of two additional isotopomers. Unfortunately, preliminary PFY spectra of DCCO do not show dissociation below 33,900 cm^{-1}, i.e., 500 cm^{-1} above the origin transition of HCCO. It appears that either DCCO does not dissociate at the origin, or that the rate of dissociation is so small as to preclude its observation in our experiment. The determination of an upper state geometry therefore must await further characterization of this electronic transition, possibly by laser-induced fluorescence (see below).

B. Vibrational analysis

The next step in characterizing the excited electronic state is an analysis of the vibrational structure in the PFY spectrum shown in Fig. 4. A four-atom linear molecule has seven vibrational degrees of freedom: three stretching modes of Σ symmetry and two doubly degenerate bending modes of Π symmetry. The extended nature of the PFY spectrum in Fig. 3 implies a large geometry change between ground and excited states,

with excitations in several modes. In particular one expects bending excitation since the ground state is non-linear while the excited state is linear.

Assuming that the band at 33,403 cm^{-1} belongs to the 0_0^0 transition, the most important pattern to recognize in the spectrum of Fig. 4 is a 1,035 cm^{-1} progression of which the first member appears at 34,443 cm^{-1}. If the entire PFY spectrum is displaced 1,035 cm^{-1} to the blue, a one-to-one correlation is found with the unshifted spectrum for almost all the peaks, which are combination bands between modes of lower frequency and this main progression. Of the three Σ vibrational modes, the highest frequency mode, ν_1, is the CH stretch. The remaining two Σ modes can in a rough approximation be labeled as a higher frequency asymmetric CCO stretch (ν_2) and a lower frequency symmetric CCO stretch (ν_3). Since the asymmetric CCO stretch in the ground state has a frequency[11] of 2,023 cm^{-1}, the 1,035 cm^{-1} progression is assigned to ν_3, the symmetric CCO stretch, in reasonable agreement with the harmonic value, 1097 cm^{-1}, calculated by Szalay *et al.* [38] The full 3_0^n progression is indicated in Fig. 4.

Given that ν_3 is the lowest frequency Σ mode, it follows that all the transitions below 34,400 cm^{-1} should arise from bending modes. Note that in a linear $\leftarrow$ bent transition, odd as well as even quanta may be excited in the bending modes ν_4 and ν_5. However, these transitions do not follow a simple harmonic pattern because the vibrational term values are perturbed through the Renner-Teller effect.[62]

As a consequence of this interaction, the degeneracy of the electronic potential can be lifted along either or both of the bending coordinates. The *ab initio* work of Szalay[38] predicts a large Renner splitting for only one of the two bending modes (ν_4, the *trans*

bend) in the excited state of ketenyl. Therefore we have attempted a simulation of the PFY spectrum (Fig. 4), incorporating progressions in ν_3, ν_4 and ν_5, in which both the spin-orbit and RT effects are incorporated[63] with ω_4 = 416 cm^{-1}, ε_{RT} = -0.36 and ω_5 = 365 cm^{-1}, ε_{RT} = 0. The corresponding *ab initio* values are ω_4 = 328 cm^{-1}, ε_{RT} = -0.573 and ω_5 = 525 cm^{-1}, ε_{RT} = -0.017. Although some transitions are not predicted (most notably the peaks with 800 - 900 cm^{-1} excess energy), there is reasonable agreement between the data and this preliminary vibronic simulation.

C. Translational energy and angular distributions

In this section we discuss the assignment of the various features in the $P(E_T)$ distributions. In our previous communication, we pointed out that using the heat of formation of HCCO determined in Ref. 25, the maximum translational energies associated with the low and high energy features in the $P(E_T)$ distributions in Figs. 6a-c correspond to the maximum allowed values for channels *II* and *I*, respectively. On this basis, the feature at higher translational energy was assigned to the CH ($X\ ^2\Pi$) + CO channel, and the lower energy feature to the CH ($a\ ^4\Sigma^-$) + CO channel. Having made this assignment, we can determine the C–C bond strength directly from the $P(E_T)$ distributions, with the goal of obtaining an improved value of the bond strength and heat of formation of the radical.

The values of E_T^{max}, the highest observed E_T for each channel at each photon energy, are tabulated in Table V, along with $h\nu - E_T^{max}$. The relatively small variation of the latter quantity implies that the values for E_T^{max} represent the translational energy when

both diatomics are produced in their ground vibration/rotation levels. The values of ΔH_0 (at 0 K) for the C—C bond cleavage are obtained by conservation of energy,

$$\Delta H_0 = h\nu - E_T^{max} \tag{5}$$

$h\nu$ cm^{-1} (eV)	E_T^{max} (*II*)	$h\nu$ - E_T^{max} (*II*) (eV)	E_T^{max} (*I*)	$h\nu$ - E_T^{max} (*I*) (eV)
33,403 (4.142)	0.28 ± 0.03	3.86	1.08 ± 0.1	3.06
34,238 (4.245)	0.39 ± 0.03	3.85	1.16 ± 0.1	3.08
34,443 (4.270)	0.40 ± 0.03	3.87	1.21 ± 0.1	3.06
35,464 (4.397)	--	--	1.30 ± 0.1	3.10
37,400 (4.637)	--	--	1.54 ± 0.1	3.10
38,634 (4.790)	--	--	1.69 ± 0.1	3.10
39,438 (4.890)	--	--	1.93 ± 0.1	2.96
40,488 (5.020)	--	--	1.96 ± 0.1	3.06
42,547 (5.275)	--	--	2.17 ± 0.1	3.10
44,443 (5.510)	--	--	2.44 ± 0.1	3.07
47,831 (5.930)	--	--	2.74 ± 0.15	3.19
mean values	--	⟨ $h\nu$ - E_T^{max} (*II*) ⟩ 3.86 ± 0.03	--	⟨ $h\nu$ - E_T^{max} (*I*)⟩ 3.1 ± 0.2

Table V: Maximum observed translational energy for each product channel

and are listed in Table V. In practice the value of ΔH_0 for channel *II* is determined with greater precision than that for channel *I* because of the sharper cutoff at E_T^{max} for channel *II*. One then obtains the heat of reaction for the spin-allowed process from $\Delta H_0(I) = \Delta H_0(II) - T_0[\text{CH a}(^4\Sigma^-)]$, where the last term is the doublet-quartet splitting in the CH radical of 0.742 ± 0.008 eV.[64,65] Finally, we deconvolute the experimental resolution from the thermodynamic limit to obtain the final values given in the introduction for $\Delta H_0(I)$ and

$\Delta H_0(II)$. The thresholds ($h\nu$ - ΔH_0) for both channels are indicated by the dashed lines at each photon energy in Fig. 6.

Together with $\Delta H^\circ_{f,0}$(CH) = 6.149 ± 0.013 eV[66,67] and $\Delta H^\circ_{f,0}$(CO) = -1.180 ± 0.002 eV[67] we obtain $\Delta H^\circ_{f,0}$(HCCO) = 1.83 ± 0.03 eV. This value differs slightly from the heat of formation reported in our previous paper[8] (1.82 ± 0.03) due to the use of a more accurate value for $\Delta H^\circ_{f,0}$(CH) in the thermochemical cycle. From calculated vibrational frequencies of HCCO we obtain the $H^\circ_{298} - H^\circ_0$ value for ketenyl, which, together with tabulated values for CH and CO,[67] gives $\Delta H^\circ_{f,298}$(HCCO) = 1.83 ± 0.03 eV. Note that the numerical value is the same as at absolute zero purely by coincidence. Our heat of formation is similar to the previous determination by photoelectron spectroscopy of $\Delta H^\circ_{f,298}$(HCCO) = 1.84 ± 0.09 eV,[25] but disagrees with the value of $\Delta H^\circ_{f,298}$(HCCO) = 1.25 ± 0.09 eV obtained from mass spectrometric appearance potentials.[68] The direct nature of the measurement of the C–C bond strength in our experiment gives us confidence that the value reported in this paper represents the most accurate heat of formation for the ketenyl radical to date.

We next consider the broad peak at E_T = 0.9 eV. At the lowest photon energies where this peak is seen (Figs. 6d,e), it must be due to spin-allowed products, as the spin-forbidden channel is not accessible at this translational energy. At still higher excitation energy shown in Figs. 6f - k, one cannot make an unambiguous assignment, but because the peak remains at E_T = 0.9 eV it is likely that this feature still represents primarily spin-allowed dissociation, which we assume in the following discussion.

Finally, we consider the detailed partitioning of available energy among the fragment modes. The vibrational term values for CH $X(^2\Pi)$ (0.35 eV), CH $a(^4\Sigma^-)$ (0.39 eV), and CO (0.27 eV) are significantly larger than our experimental translational energy resolution of 0.022 eV (for a 1 eV recoil), but no vibrational resolution in observed in the $P(E_T)$ distributions. For the spin-allowed channel, many vibrational states of both CH and CO are allowed at all photon energies, but we cannot discern any pattern in the data which reflects the known vibrational spacings. This observation implies that the CH and/or CO fragments contain substantial amounts of rotational energy, so that the rotational envelope built on each vibrational threshold smoothes any vibrational information in the $P(E_T)$ distributions. For the spin-forbidden channel, conservation of energy shows that both CH and CO are produced in their $\upsilon = 0$ state in Fig. 6a, with the available energy partitioned between product translation and rotation. With increasing photon energy (Figs. 6b-c), CO ($\upsilon = 1$) and even CH $a(^4\Sigma^-)$ ($\upsilon = 1$) become allowed, although our low energy cutoff of $E_T = 0.12$ eV prevents us from detecting these products.

V. Discussion

A. Spectroscopy of the ketenyl radical.

1. Nature of the excited electronic state.

In Section IV-A, the HCCO excited state probed in these experiments was assigned as a $^2\Pi$ state. From the molecular orbital configurations given in Table I, the ultraviolet transition in HCCO corresponds to promotion of an electron from the lowest π_x

or π_y orbital (with zero nodes in the xy plane) to the next pair of π_x / π_y orbitals (one node in the xy plane). Because the zero-node π orbitals are strongly CO bonding, while the one-node π orbitals are somewhat CC bonding, but CO antibonding, the ultraviolet transition should result in the decrease in r_{CC} and increase in r_{CO} depicted in Fig. 1.

Our spectroscopic results can be compared to two *ab initio* calculations that reached somewhat different conclusions about the geometrical nature of the electronically excited doublet states of HCCO. Using a multi-reference configuration interaction (MR-CI) method, Kim and Shavitt[34] predicted that the excited $^2\Pi$ state is split by Renner-Teller coupling, resulting in a bent $\tilde{B}$ (^{2}A′) state and a linear $\tilde{C}$ (^{2}A″) [$^2\Pi$] state. The adiabatic energies for the $\tilde{B}$ (^{2}A′) $\leftarrow$ $\tilde{X}$ (^{2}A″) and $\tilde{C}$ (^{2}A″) [$^2\Pi$] $\leftarrow$ $\tilde{X}$ (^{2}A″) were predicted to occur at 33,000 cm^{-1} and 33,300 cm^{-1}, respectively. More recently, Szalay *et al.*[38] have performed calculations on the excited states of HCCO using the equation-of-motion ionization-potential coupled cluster singles and doubles (EOMIP-CCSD) method. They show for the ultraviolet states that the prediction of bent vs. linear equilibrium geometry for the lower of the two RT components is dependent on the choice of basis set. Even when a bent geometry was obtained it was found that the RT splitting between the bent and linear geometries was only a few cm^{-1}, implying a very flat potential near linearity. They conclude that the states labeled $\tilde{B}$ and $\tilde{C}$ by Kim and Shavitt are not separate states, but share an identical linear equilibrium geometry and therefore propose the label 2($^2\Pi$) for this electronic configuration.

Our experiment does not show evidence for a transition to a bent $\tilde{B}$ (^{2}A′) state. The $\tilde{B}$ (^{2}A′) $\leftarrow$ $\tilde{X}$ (^{2}A″) transition would be identified by: (i) rotational structure

characteristic of a perpendicular band,[56] i.e., *c*-type $\Delta K = \pm 1$ transitions, and (ii) a negative value of the anisotropy parameter, $\beta \leq 0$.[51] Therefore, in our previous communication,[8] we assigned the band in Figure 3 to the $\tilde{C}\,(^2A'')\,[^2\Pi] \leftarrow \tilde{X}\,(^2A'')$ transition based on the work of Kim and Shavitt.

However, from the analysis in section IV-B, we are able to fit many of the vibronic bands assuming that the upper state is a $^2\Pi$ state with spin-orbit coupling and a Renner parameter $\varepsilon_{RT} = 0.36$. A Renner parameter less than unity denotes that both components of the $^2\Pi$ state are linear, in agreement with the calculation by Szalay *et al.* Therefore we have elected to change the spectroscopic label of the excited state we access, and propose the designation $\tilde{B}$ $(^2\Pi)$ which describes the potential in Fig. 1 as the second distinct electronically excited state of HCCO. The definitive experiment to determine the shape of the excited potential energy surfaces would be a fully rotationally resolved study of the low-lying vibronic levels in the UV spectrum. The experimental values of the Renner parameter then allows one to map out the electronic potential along the bending coordinates. We comment below on the possibilities for such an investigation.

2. *Comparison with previous experimental investigations.*

None of the results presented here on the ultraviolet spectroscopy of the HCCO radical are consistent with the absorption spectrum attributed to HCCO by Krishnamachari and Venkatasubramanian.[26] They assigned two band origins at 27,262 cm^{-1} and 29,989 cm^{-1} respectively. Based on gross rotational structure, they proposed for the lower energy band that at least one of the two combining states has a linear geometry,

while the second band is assigned to a combination of two non-linear states. The energetics of these absorption bands are clearly not in agreement with our data.

Another explanation of the absorption results can be attempted by analogy with the isoelectronic NCO radical. The first two band systems for NCO are the $\tilde{A}\,(^2\Sigma^+) \leftarrow \tilde{X}\,(^2\Pi)$[69] and the $\tilde{B}\,(^2\Pi) \leftarrow \tilde{X}\,(^2\Pi)$[70] bands at 22,754 and 31,753 cm^{-1}, respectively. The band we assign in the PFY spectrum of HCCO corresponds to the $\tilde{B}\,(^2\Pi) \leftarrow \tilde{X}\,(^2\Pi)$ transition in NCO, leading one to question whether the spectra observed by Krishnamachari and Venkatasubramanian to the red of our spectrum might be due to a band analogous to the $\tilde{A}\,(^2\Sigma^+) \leftarrow \tilde{X}\,(^2\Pi)$ transition of NCO. However, *ab initio* calculations[37] predict the lowest $^2\Sigma^+$ state of HCCO at 53,000 ± 4,000 cm^{-1} above the ground state, much too high to account for the bands observed in absorption. Therefore, we can offer no explanation for the bands observed by Krishnamachari and Venkatasubramanian. It is likely that the carrier of these bands is not the HCCO radical. We emphasize again that our experiments are performed on a *mass-selected* beam of radicals, ensuring that the species we probe is indeed HCCO.

3. Prospects for further rovibronic spectroscopy

The rotationally-resolved bands shown in Fig. 5 suggest that it may be possible to study HCCO with laser-induced fluorescence (LIF) now that an HCCO electronic transition has been identified. This requires that predissociation is sufficiently slow so that fluorescence is competitive. We can gain some insight to the excited state lifetimes from the observed linewidths in Fig. 5. The two significant contributions to our lineshapes are a Gaussian component due to the laser bandwidth and a Lorentzian component due to

predissociation induced lifetime broadening. The Lorentzian width can be deconvoluted from the instrumental width to give a rough lower limit on the lifetime of the 0_0^0 transition (characterized by instrumentally narrow lines) of $\tau \geq 300$ ps. The other rotational band (Fig. 5b) shows a decrease in the excited state lifetime to $\tau \approx 90$ ps at an energy 1,900 cm^{-1} above the origin.

It is certainly possible that $\Phi_{Diss} << 1$ at the origin, as it is for the vinoxy radical.[12] If so, the lifetime might be significantly longer than 300 ps, increasing the likelihood of detecting HCCO by laser induced fluorescence. If the 0_0^0 transition does fluoresce, the rotational structure of higher vibrational levels can be probed by fluorescence depletion spectroscopy, as recently applied to the vinoxy[71] and methoxy[72] (CH_3O) radicals. If the fluorescence is simply too weak for LIF-based experiments, cavity ringdown absoption spectroscopy or non-linear techniques such as four-wave mixing spectroscopy[73] would still be possible. In any case, the PFY spectra presented here give an accurate survey for the pursuit of further spectroscopic studies of ketenyl, and advance the possibility of developing an ultraviolet diagnostic for HCCO detection in combustion research.

B. Photodissociation dynamics

From the potential energy surfaces of the ketenyl radical as shown in Fig. 1, there are at least two paths by which the electronically excited radicals might decay to products. After excitation to the $\tilde{B}$ ($^2\Pi$) state, intersystem crossing (ISC) may occur to the HCCO $\tilde{a}$ ($^4A''$) surface, followed by spin-forbidden dissociation. Calculations by Hu *et al.*[36] predict that the quartet surface correlates to channel *II* products with either no exit barrier

or at most a small barrier no larger than 0.1 eV, although a recent calculation by Yarkony finds a 0.16 eV barrier.[40]

A second decay pathway for the initially excited radicals is by internal conversion (IC) to the ground electronic state followed by dissociation to spin-allowed channel *I* products. The translational energy distributions show that ISC dominates at the three lowest photon energies, but that IC to channel *I* dominates at higher photon energies. Dissociation proceeding via IC appears to have two components, one leading to the peak at E_T = 0.9 eV corresponding to spin-allowed dissociation, which does not shift in translational energy for a wide range of excitation energies, and the broad feature underneath this peak which appears to extend across the entire $P(E_T)$ distribution, up to the value of E_T^{max} for the spin-allowed channel. An overall mechanism should be able to explain the competition between spin-allowed and spin-forbidden dissociation as well as the two components of the spin-allowed process.

The competition between ISC and IC has been considered at length in the context of radiationless transitions from the first excited singlet state (the S_1 state) in aromatic molecules.[74] Each rate is determined primarily by the product of an electronic matrix element between the initial and final electronic state and a series of Franck-Condon factors between the initial and final vibrational levels. The electronic matrix element for IC is generally substantially larger than for ISC because the spin multiplicity is unchanged. However, in aromatic molecules, the geometries of S_1 and T_1 states are quite similar, resulting in considerably higher Franck-Condon factors for ISC than for IC. For naphthalene, the resulting competition between these two compensating effects is that ISC and IC are equally important near the origin of the $S_1 \leftarrow S_0$ transition, but IC becomes

progressively more important with increasing vibrational excitation in the S_1 states as the Franck-Condon factors for IC increase more rapidly than for ISC.[75]

In HCCO, the energetic ordering of electronic states is analogous to that in aromatic molecules, with $\tilde{B}\,(^2\Pi) > \tilde{a}\,(^4A'') > \tilde{X}\ (^2A'')$. However, the Franck-Condon overlap between the vibrational levels of the three states is more difficult to asses due to large geometry differences between these states. The competition between ISC and IC appears to be similar to that in aromatics but more extreme; in HCCO there is very little IC near the $\tilde{B} \leftarrow \tilde{X}$ origin, and IC rapidly becomes dominant with increasing excitation in the $\tilde{B}\,(^2\Pi)$ state. It is possible that the relatively low energy intersection between the $\tilde{a}\,(^4A'')$ and $\tilde{B}\,(^2\Pi)$ surfaces, calculated to lie 2.4 eV above the $\tilde{B}$ state minimum (Section III-C), results in a particularly large ISC electronic matrix element. In any case, it is likely that the same general principles govern the relative rates of ISC and IC in HCCO as in larger aromatic molecules. It remains to be seen whether a reasonable set of electronic matrix elements and Franck-Condon factors can reproduce the experimental trends, and further characterization of the relevant potential energy surfaces is required.

We next consider the dynamics of the spin-allowed channel. Based on the potential energy surfaces in Figure 1, one might expect internal conversion to the ground state to be followed by statistical dissociation to products. Our *ab initio* calculations and the CH($\upsilon = 1$) + CO vibrational relaxation results of Herbert *et al.*[24] indicate that there is no barrier to dissociation on the ground state surface, so that the translational energy distribution may be predicted using phase space theory[76] or, at a simpler level, a prior distribution.[77] In fact, the broad feature in the higher photon energy $P(E_T)$ distributions

(Figs. 6f-k) can be satisfactorily reproduced by a prior distribution, as shown explicitly in Fig. 6j, where the prior distribution

$$P^{\circ}(E_T) \propto \sum_{\upsilon_{CO},\upsilon_{CH}=0}^{\upsilon_{max}\,;\,E_\upsilon < E_{avail}-E_T} \sqrt{E_T(\upsilon_{CO},\upsilon_{CH})} \tag{6}$$

is shown superimposed on the experimental $P(E_T)$ distribution at $h\nu$ = 44,443 cm^{-1}. The prior distribution takes this simple form because both fragments are diatomic molecules, whose classical rotational density of states, $\rho(E_{ROT})$, is independent of energy.

The second feature associated with the spin-allowed channel, the peak at E_T = 0.9 eV, is more problematic. This peak does not shift much with excitation energy, and therefore resembles what would be expected for statistical dissociation over a barrier. In such cases, the most probable energy in the $P(E_T)$ distribution tends to occur at some fraction of the exit barrier height and depends little on the excitation energy.[78] This peak therefore does not appear to result from statistical dissociation on the barrierless ground state surface. For reasons presented in section III-C, we also expect no barrier to dissociation on the low-lying $\tilde{A}\ ^2A'(^2\Pi)$ state.

Although it is true that in a diabatic picture HCCO cannot correlate to CH $X(^2\Pi)$ + CO $X(^1\Sigma^+)$ because ketenyl has seven π electrons, while the products have only five π electrons, the experiments of Herbert *et al.*[24] and the *ab initio* results presented here argue strongly in favor of the adiabatic picture of the doublet states shown in Fig. 1. Because only two molecular states of HCCO can correlate to ground state products, both of which are barrierless potential wells in the adiabatic picture, it is difficult to rationalize any potential energy surface with a barrier which leads to these products.

One explanation for the peak at 0.9 eV is that it results from non-statistical dissociation dynamics on the ground state potential energy surface. From a classical point of view, this could happen if some of the ground state molecules formed by internal conversion from the $\tilde{B}\ ^2\Pi$ state follow trajectories that are not trapped in the potential energy well on the $\tilde{X}\ ^2A''$ state long enough for energy randomization to occur.[79] Given that the well depth is 3.14 eV, the total energy is 4.5-5.5 eV in the range where this peak is most prominent, and the fact that HCCO is a relatively small molecule, it is possible that some non-statistical dynamics occur on the ground state potential energy surface.

There is some support for such a non-statistical process from the $\beta(E_T)$ distributions in Figure 6. The distributions at several excitation energies show a local maximum in the anisotropy near $E_T = 0.9$ eV, the same energy where the peak in the $P(E_T)$ distribution occurs. The greater anisotropy associated with this peak suggests that it arises from a more rapid dissociation process, consistent with the hypothesis that it results from trajectories not trapped in the potential energy well. We also note that the presence of a well on the PES does not always result in energy randomization. For example, in our recent study of the photodissociation of the vinoxy radical (CH_2CHO), the $P(E_T)$ distributions for the CH_3 + CO channel show that randomization in the potential well corresponding to the CH_3CO configuration does not occur. Although this well is considerably shallower (0.75 eV) than the HCCO ground state well, the vinoxy results provide more indirect support for our hypothesis.

An alternative explanation of the peak in question is that there could be more than one excited electronic state of HCCO contributing to the PFY spectrum in Fig. 3. This spectrum shows two intense, broad features centered around 41,000 and 42,000 cm^{-1}; this

is the range of excitation energy where the peak at 0.9 eV is most intense. However, the next excited state above the $\tilde{B}$ state which could contribute to the PFY spectrum is the $^2\Sigma^+$ state, calculated to occur[37] at 53,000 cm^{-1}. We would then observe a perpendicular transition to this state from the ground state, resulting in a negative anisotropy parameter, which disagrees with the observed $\beta(E_T)$ distributions in which $\beta(E_T)$ increases near 0.9 eV.

Although much of the above discussion has focused on the $P(E_T)$ distributions, the trends in the average anisotropies $\langle\beta\rangle$ as a function of photon energy provide additional insight into the photodissociation dynamics. As discussed above, the positive values of $\langle\beta\rangle$ at all photon energies are consistent with our assignment of the PFY spectrum to a parallel transition. For a linear molecule such as the $\tilde{B}$ state of HCCO, the classical treatment of dissociation predicts that for a parallel transition, $0.5 \leq \beta \leq 2$, with the lower and upper limits corresponding to complete rotational averaging and instantaneous dissociation, respectively.[80] Fig. 6 shows that $\langle\beta\rangle$ increases with photon energy up to a value of 0.79 at 40,488 cm^{-1}, but decreases as the photon energy is raised further. The rising value of $\langle\beta\rangle$ implies that the excited state lifetime decreases between the band origin and 40,488 cm^{-1}, a trend consistent with the disappearance of resolved structure in the PFY spectrum.

The maximum values of β observed are considerably less than 2, which at first glance might suggest rotational averaging at all photon energies. However, the extraction of quantitative lifetimes from the measured anisotropies is not straightforward.[80] HCCO in its excited electronic state is not a rigid linear molecule because the bending frequencies

are low, and the electronic transition results in considerable bending excitation, as demonstrated by Fig. 4. Furthermore, the $P(E_T)$ distributions discussed in section IV-C imply considerable product rotational excitation, a signature of non-linear HCCO geometries prior to dissociation. Even if dissociation occurs on the timescale of a vibrational period (i.e. without rotational averaging), such large amplitude bending motions will introduce a distribution of angles between the transition dipole and asymptotic recoil vectors, resulting in a reduced value of the anisotropy parameter.

Therefore, two effects will have a counteracting influence on the anisotropy: $\langle\beta\rangle$ rises as the dissociation lifetime decreases, but falls due to increasing bending amplitude. The rise and subsequent fall in $\langle\beta\rangle$ mentioned earlier may indicate that the increase in dissociation rate with photon energy reaches a plateau around 40,000 cm^{-1}, while increasing bending amplitude (and the concomitant reduction in $\langle\beta\rangle$) continues unabated to 48,000 cm^{-1}. Overall, the trends in β imply that there is rotational averaging at low photon energies, consistent with the rotational resolution seen in some bands in the PFY spectrum, but the relationship between dissociative lifetime and anisotropy are more complicated at those photon energies where β is largest (Figs. 6e-i).

Finally, we compare the photodissociation dynamics of HCCO with two close relatives, the vinoxy radical, CH_2CHO, and the methoxy radical, CH_3O. All three radicals undergo predissociation, but the dissociation dynamics subsequent to electronic excitation are quite different. The $P(E_T)$ distributions for $CH_3O \rightarrow CH_3 + O$ show that the $\tilde{A}\,(^2A_1)$ excited state of CH_3O is predissociated to these products by one or more repulsive states, and that no internal conversion to the ground state occurs.[13] In contrast, the CH_2CHO

radical appears to dissociate entirely by internal conversion to the ground state, followed by statistical dissociation to CH_3 + CO and H + CH_2CO products.[12] The HCCO radical falls between these extremes, with some dissociation occurring on the $\tilde{a}\,(^4A'')$ excited state, yielding CH($a\,^4\Sigma^-$) + CO products, some occurring statistically on the ground state, yielding a feature in the $P(E_T)$ distribution that can be fit by a prior distribution, and some possibly occurring by a non-statistical process on the ground state surface. A detailed understanding of these trends will require a more thorough investigation of the ground and excited states of these radicals. This is likely to occur through a combination of high resolution spectroscopy of the predissociating transitions and *ab initio* calculations of the surfaces involved in the dissociation dynamics.

VI. Conclusions

We have investigated the spectroscopy and photodissociation dynamics of the $\tilde{B}\,(^2\Pi) \leftarrow \tilde{X}\,(^2A'')$ transition of HCCO, utilizing the method of fast beam photofragment translational spectroscopy on a mass-selected, internally cold sample of ketenyl radicals. Rotational analysis of the photofragment yield spectrum shows that the excited state is linear, with $^2\Pi$ symmetry. Linewidth measurements give a lower bound on the lifetime of the 0_0^0 transition of $\tau \geq 300$ ps. A limited vibrational analysis allows us to assign the symmetric CCO stretch, ω_3 =1,035 cm^{-1}. It is also clear from the complications in the PFY spectrum that at least one of the bending modes in the excited state is split by the Renner-Teller interaction.

We also probe the photodissociation dynamics of the HCCO radical by measuring the translational energy and angular distributions of the CH + CO fragments. The spin-forbidden product channel, producing CH $a(^4\Sigma^-)$ + CO, dominates for excitation energies within the first 1,000 cm^{-1} above the origin. These products are produced primarily in their ground vibrational states. The spin-allowed channel, CH $X(^2\Pi)$ + CO, appears to dominate at all higher excitation energies, and it is clear that these products emerge with substantial vibrational and rotational excitation.

We propose a three part mechanism for the decay of $\tilde{B}$ $(^2\Pi)$ HCCO. Intersystem crossing to the $\tilde{a}$ $(^4A'')$ surface via spin-orbit coupling gives rise to spin-forbidden products at low photon energies. Dissociation proceeds on this excited state surface to yield CH $a(^4\Sigma^-)$ + CO. Internal conversion to the $\tilde{X}$ $(^2A'')$ state, which correlates without a barrier to CH $X(^2\Pi)$ + CO, is followed by two distinct decay mechanisms. We propose that the first component is statistical in nature, as implied by reasonable agreement between the broad feature extending across the entire $P(E_T)$ distribution and a prior distribution. A second, non-statistical component of ground state dissociation is required to described the feature at E_T = 0.9 eV in the higher energy $P(E_T)$ distribution. The dominance of the internal conversion process for excess energies 2,000 cm^{-1} and higher is an indication that the Franck-Condon overlap between the $\tilde{B}$ and $\tilde{X}$ states increases faster with photon energy than the overlap between the $\tilde{B}$ and $\tilde{a}$ states. The development of an accurate potential energy surface for HCCO and subsequent reaction dynamics calculations will be required to test this proposed mechanism.

In relation to past experiments on the ketenyl radical, we find no agreement between our spectroscopic work and the absorption spectrum attributed to HCCO by Krishnamachari and Venkatasubramanian.[26] There have been many unsuccessful attempts at detection of an ultraviolet transition in HCCO by laser induced fluorescence. The difficulty in attaining an LIF spectrum of ketenyl likely arises from three main factors: (i) contamination species present which obscure signal from HCCO, (ii) the origin transition of ketenyl's closest relative, CH_2CHO (ν_0 = 28,784.09 cm^{-1}),[57] along with all previous experimental data, significantly underestimate the frequency of the origin transition in HCCO (ν_0 = 33,423.92 cm^{-1}), and (iii) the quantum yield for fluorescence is likely very small over most of the $\tilde{B}$ ($^2\Pi$) $\leftarrow$ $\tilde{X}$ ($^2A''$) band. With the aid of the results presented in this paper, only the last factor remains an issue. However, our bound on the 0_0^0 fluorescence lifetime indicates that LIF detection of this transition may be possible, which would represent an important advance in our ability to directly probe HCCO in combustion environments.

Acknowledgments

This research is supported 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. I acknowledge helpful conversations with Professors Robert Curl, Richard Dixon, Yasuki Endo, and Donald Levy, and thank Professors Isaiah Shavitt and Peter Szalay for sharing their results prior to publication.

References

[1] J. Warnatz, *Combustion Chemistry*, ed. W. C. Gardiner (Springer, Berlin, 1984), p. 288.

[2] J. V. Michael and A. F. Wagner, J. Phys. Chem. **94**, 2453 (1990).

[3] W. Boullart and J. Peeters, J. Phys. Chem. **96**, 9810 (1992).

[4] J. Peeters, I. Langhans, and W. Boullart, Int. J. Chem. Kinet. **26**, 869 (1994).

[5] C. K. Westbrook and F. L. Dryer, *Eighteenth Symposium (International) on Combustion* (Combustion Institute, Pittsburgh, 1981), p. 749.

[6] K. Brezinsky, Prog. Energy Combust. Sci. **12**, 1 (1986).

[7] J. A. Miller, R. J. Kee, and C. K. Brook, Ann. Rev. Phys. Chem. **41**, 245 (1990).

[8] D. H. Mordaunt, D. L. Osborn, H. Choi, R. T. Bise, and D. M. Neumark, J. Chem. Phys. **105**, 6078 (1996).

[9] Y. Endo and E. Hirota, J. Chem. Phys. **86**, 4319 (1987); Y. Ohshima and Y. Endo, J. Mol. Spec. **159**, 458 (1993).

[10] H. E. Radford, 21st International Symposium on Free Radicals, Williamsburg, MA, USA (1991).

[11] K. G. Unfried, G. P. Glass, and R. F. Curl, Chem. Phys. Lett. **177**, 33 (1991); K. G. Unfried and R. F. Curl, J. Mol. Spec. **150**, 86 (1991).

[12] D. L. Osborn, H. Choi, D. H. Mordaunt, R. T. Bise, D. M. Neumark, and C. M. Rohlfing, J. Chem. Phys. **106**, (1996); D. L. Osborn, H. Choi, and D. M. Neumark, Adv. Chem. Phys. (in press).

[13] D. L. Osborn, D. J. Leahy, E. M. Ross, and D. M. Neumark, Chem. Phys. Lett. **235**, 484 (1995); D. L. Osborn, D. J. Leahy, and D. M. Neumark (in preparation).

[14] I. T. N. Jones and K. D. Bayes, J. Am. Chem. Soc. **94**, 6869 (1972).

[15] W. Boullart, M. T. Nguyen, and J. Peeters, J. Phys. Chem. **98**, 8036 (1994).

[16] J. Peeters, I. Langhans, W. Boullart, M. T. Nguyen, and Katia Devriendt, J. Phys. Chem. **98**, 11988 (1994).

[17] J. Peeters, W. Boullart, and Katia Devriendt, J. Phys. Chem. **99**, 3583 (1995).

[18] D. G. Williamson and K. D. Bayes, J. Phys. Chem. **73**, 1232 (1969).

[19] A. M. Schmoltner, P. M. Chu, and Y. T. Lee, J. Chem. Phys. **91**, 5365 (1989).

[20] F. Temps, H. Gg. Wagner, and M. Wolf, Z. Phys. Chem. **176**, 27 (1992).

[21] K. K. Murray, K. G. Unfried, G. P. Glass, and R. F. Curl, Chem. Phys. Lett. **192**, 512 (1992).

[22] C. Vinckier, M. Schaekers, and J. Peeters, J. Phys. Chem. **89**, 508 (1985).

[23] C. A. Taatjes, private communication.

[24] L. B. Herbert, I. R. Sims, I. W. M. Smith, D. W. A. Stewart, A. C. Symonds, A. Canosa, and B. R. Rowe, J. Phys. Chem. **100**, 14928 (1996).

[25] J. M. Oakes, M. E. Jones, V. M. Bierbaum, and G. B. Ellison, J. Phys. Chem. **87**, 4810 (1983).

[26] S. L. N. G. Krishnamachari and R. Venkatasubramanian, Pramana **23**, 321 (1984).

[27] G. Inoue and M. Suzuki, J. Chem. Phys. **84**, 3709 (1986).

[28] M. A. Hanratty and H. H. Nelson, J. Chem. Phys. **92**, 814 (1990); G. Inoue and M. Suzuki, J. Chem. Phys. **92**, 815 (1990).

[29] M. E. Jacox and W. B. Olson, J. Chem. Phys. **86**, 3134 (1987).

[30] L. B. Harding, J. Phys. Chem. **85**, 10 (1981).

[31] D. L. Cooper, Astrophys. J. **265**, 808 (1983).

[32] L. B. Harding and A. F. Wagner, J. Phys. Chem. **90**, 2974 (1986).

[33] J. D. Goddard, Chem. Phys. Lett. **154**, 387 (1989).

[34] K. Kim and I. Shavitt, private communication.

[35] P. G. Szalay, J. F. Stanton, and R. J. Bartlett, Chem. Phys. Lett. **193**, 573 (1992).

[36] C. H. Hu, H. F. Schaefer, Z. Hou, and K. D. Bayes, J. Am. Chem. Soc. **115**, 6904 (1993).

[37] M. T. Nguyen, W. Boullart, and J. Peeters, J. Phys. Chem. **98**, 8030 (1994).

[38] P. G. Szalay, G. Fogarasi, and L. Nemes, (in press).

[39] P. G. Szalay and J. P. Blaudeau, (in press).

[40] D. Yarkony, J. Phys. Chem. **100**, 17439 (1996).

[41] C. P. Fenimore and G. W. Jones, J. Chem. Phys. **39**, 1514 (1963).

[42] V. Zengin, B. J. Persson, K. M. Strong, and R. E. Continetti, J. Chem. Phys. **105**, 1996.

[43] H. Choi, D. H. Mordaunt, R. T. Bise, and D. M. Neumark, (in preparation).

[44] R. E. Continetti, D. R. Cyr, D. L. Osborn, D. J. Leahy, and D. M. Neumark, J. Chem. Phys. **99**, 2616 (1993).

[45] R. E. Continetti, D. R. Cyr, R. B. Metz, and D. M. Neumark, Chem. Phys. Lett. **182**, 406, 1991.

[46] D. L. Osborn, D. J. Leahy, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **104**, 5026 (1996).

[47] J. M. B. Bakker, J. Phys. E **6**, 785 (1973); **7**, 364 (1974).

[48] D. J. Leahy, D. L. Osborn, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **103**, 2495 (1995).

[49] D. P. de Bruijn and J. Los, Rev. Sci. Instrum. **53**, 1020 (1982).

[50] S. Gerstenkorn and P. Luc, *Atlas du Spectre d'Absorption de la Molecule d'Iode* (Centre National de la Recherche Scientifique, Paris, 1978).

[51] R. N. Zare, Mol. Photochem. **4**, 1 (1972).

[52] D. Hegarty and M. A. Robb, Mol. Phys. **38** 1795 (1979); R. H. E. Eade and M. A. Robb, Chem. Phys. Lett. **83** 362 (1981).

[53] GAUSSIAN 94, Revision C.3, M. J. Frisch, G. W. Trucks, H. B. Schlegel, P. M. W. Gill, B. G. Johnson, M. A. Robb, J. R. Cheeseman, T. Keith, G. A. Petersson, J. A. Montgomery, K. Raghavachari, M. A. Al-Laham, V. G. Zakrzewski, J. V. Ortiz, J. B. Foresman, J. Cioslowski, B. B. Stefanov, A. Nanayakkara, M. Challacombe, C. Y. Peng, P. Y. Ayala, W. Chen, M. W. Wong, J. L. Andres, E. S. Replogle, R. Gomperts, R. L. Martin, D. J. Fox, J. S. Binkley, D. J. Defrees, J. Baker, J. P. Stewart, M. Head-Gordon, C. Gonzalez, and J. A. Pople, Gaussian, Inc., Pittsburgh PA, 1995.

[54] L. B. Herbert, I. R. Sims, I. W. M. Smith, D. W. A. Stewart, A. C. Symonds, A. Canosa, and B. R. Rowe, J. Phys. Chem. **100**, 14928 (1996).

[55] J. A. Pople, M. Head-Gordon, and K. Raghavachari, J. Chem. Phys. **87**, 5968 (1987).

[56] G. Herzberg, *Molecular Spectra and Molecular Structure III. Electronic Spectra and Electronic Structure of Polyatomic Molecules* (Van Nostrand, Princeton, 1966).

[57] L. F. DiMauro, M. Heaven, and T. A. Miller, J. Chem. Phys., **81**, 2339 (1984).

[58] Y. Endo, S. Saito, and E. Hirota, J. Chem. Phys. **83**, 2026 (1985); Y. Endo and E. Hirota, J. Mol. Spec. **127**, 535 (1988).

[59] J. K. G. Watson, *Vibrational Spectra and Structure*, edited by J. R. Durig (Elsevier, Amsterdam, 1977), Vol. 6, pp. 2-89.

[60] W. T. Raynes, J. Chem. Phys. **41**, 3020 (1964).

[61] E. Hirota, *High Resolution Spectroscopy of Transient Molecules* (Springer, Heidelberg, 1985).

[62] R. Renner, Z. Physik **92**, 172 (1934).

[63] J. A. Pople, Mol. Phys. **3**, 16 (1960); J. T. Hougen, J. Chem. Phys. **36**, 519 (1962).

[64] S. G. Lias *et. al.*, J. Chem. Phys. Ref. Data **17**, Suppl. No. 1 (1988)

[65] A. Kasdan, E. Herbst, and W. C. Lineberger, Chem. Phys. Lett. **31**, 78 (1975).

[66] M. Danielsson, P. Erman, A. Hishikawa, M. Larsson, E. Rachlew-Källne, and G. Sundström, J. Chem. Phys. **98**, 9405 (1993).

[67] M. W. Chase, Jr., C. A. Davies, J. R. Downey, Jr., D. J. Frurip, R. A. McDonald, and A. N. Syverud, JANAF Thermochemical Tables, 3rd ed., J. Chem. Phys. Ref. Data **14**, (1985), Suppl. No. 1.

[68] J. L. Holmes and F. P. Lossing, Int. J. Mass. Spectrom. Ion Process. **58**, 113 (1984).

[69] P. S. H. Bolman, J. M. Brown, A. Carrington, I. Kopp, and D. A. Ramsay, Proc. R. Soc. Lond. A. **343**, 17 (1975).

[70] R. N. Dixon, M. J. Trenouth, and C. M. Western, Mol. Phys. **60**, 779 (1987).

[71] T. Gejo, M. Takayanagi, T. Kono, and I. Hanazaki, Chem. Letters 2065 (1993).

[72] D. E. Powers and T. A. Miller, (in preparation)

[73] S. Williams, J. D. Tobiason, J. R. Dunlop, and E. A. Rohlfing, J. Chem. Phys. **102**, 8342 (1995), and references therein.

[74] P. Avouris, W. M. Gelbart, and M. A. El-Sayed, Chem. Rev. **77**, 793 (1977); *Radiationless Transitions*, S. H. Lin, ed. (Academic Press, New York, 1980).

[75] G. S. Beddard, G. R. Fleming, O. L. J. Gijzeman, and G. Porter, Chem. Phys. Lett. **18**, 481 (1973).

[76] J. C. Light, Discuss. Faraday Soc. **44**, 14 (1977).

[77] R. D. Levine and J. L. Kinsey, in *Atom-Molecule Collision Theory–A Guide for the Experimentalist,* edited by R. B. Bernstein (Plenum, New York, 1979).

[78] A. M. Wodtke, E. J. Hintsa, and Y. T. Lee, J. Phys. Chem. **90**, 3549 (1986) ; X. Zhao, R. E. Continetti, A. Yokoyama, E. J. Hintsa, and Y. T. Lee, J. Chem. Phys. **91**, 4118 (1989).

[79] M. Stumpf, A. J. Dobbyn, D. H. Mordaunt, H. M. Keller, R. Schinke, H. J. Werner, and K. Yamashita, Faraday Disc. **102,** (in press).

[80] C. Jonah, J. Chem. Phys. **55**, 1915 (1971); S.-C. Yang and R. Bersohn, J. Chem. Phys. **61**, 4400 (1974).

6 Photodissociation spectroscopy and dynamics of the $N_2O_2^-$ anion

I. Introduction

In recent years, photodissociation experiments have been performed on a considerable number of neutral molecules and positive ions. These experiments have proved invaluable in understanding the electronic spectroscopy and dissociation dynamics of these species. In contrast, there have been relatively few photodissociation studies on negative ions. Anion photodissociation is problematic because, in most negative ions, bond dissociation energies are higher than the threshold for electron detachment, so that the dissociative anion state one might hope to access is generally embedded in the neutral + e^- continuum. Nonetheless, photodissociation studies have been carried out on open shell anions such as Cl_2^- [1] or CO_3^- [2,3] for which the smallest bond dissociation energy is less than the electron binding energy, and on species such as O_3^- [4] and O_4^- [5,6] in which photodissociation occurs at energies well above the threshold for electron detachment. In this paper, we use a fast ion beam experiment to investigate the photodissociation spectroscopy and dynamics of the $N_2O_2^-$ anion, an open-shell species with a high electron binding energy.

The work presented here is motivated by recent results on the gas phase spectroscopy and dynamics of the $N_2O_2^-$ anion. In a photoelectron spectroscopy study, Posey and Johnson[7] demonstrated the existence of three distinct isomers of $N_2O_2^-$ which

can each be prepared selectively in the gas phase by electron impact on a supersonic expansion of different gases. They characterized these isomers as: (I) $O_2^-\bullet N_2$, from O_2 seeded in N_2; (II) NNO_2^-, from neat N_2O; and (III) $NO^-\bullet NO$, from NO seeded in argon. Their results indicated that isomer (I) is a weakly bound cluster, while the other two isomers are more strongly bound species. Their proposed C_{2v} structure for isomer (II) is supported by subsequent *ab initio* calculations,[8] which also suggest that isomer (III) has a *trans*-$ONNO^-$ structure. Furthermore, several matrix isolation studies[9,10,11,12] have identified a C_{2v} structure consistent with isomer (II), and a species of the form $ONNO^-$ which most likely represents isomer (III).

The existence of multiple isomers is particularly interesting in light of isotopic labeling experiments on the ion-molecule reaction[13,14]

$$O^- + N_2O \rightarrow NO^- + NO \qquad \Delta H_0 = -0.15 \pm 0.01 \text{ eV}^{15} \tag{1}$$

These studies show that reaction occurs through O^- attack at both the terminal and central N atoms on N_2O. These results, when combined with the photoelectron spectra, imply that the reaction mechanism involves passing through two complexes corresponding to isomers (II) and (III), NNO_2^- and $ONNO^-$.

Posey and Johnson found that at 532 nm and 355 nm, the only peak in the photoelectron spectrum of the NNO_2^- isomer is from the photodetachment of O^- atoms. Thus, at these wavelengths, photodissociation to $O^- + N_2O$ occurs rather than photodetachment of the parent $N_2O_2^-$ anion. More recently, Arnold and Neumark[16] have measured the photoelectron spectrum of this isomer at 266 nm and 213 nm. They found

that at these higher photon energies, photodetachment of $N_2O_2^-$ does occur, populating several previously unknown electronic states of neutral N_2O_2. From this work alone, however, one cannot determine at what energy the observed N_2O_2 states lie relative to the various asymptotic channels (O + N_2O, N + NO_2, NO + NO, etc.). One of the goals of this study is to answer this question by determining the dissociation energy of the $N_2O_2^-$ anion.

In this paper, we present recent work on the excited state spectroscopy and photodissociation dynamics of the C_{2v} isomer (II), which will henceforth be referred to simply as $N_2O_2^-$. Using the technique of fast beam translational spectroscopy, we have measured the relative photodissociation cross section of $N_2O_2^-$ as a function of wavelength, and have performed more detailed dynamical measurements at selected wavelengths. We observe fragmentation to two channels, O^- + N_2O and NO^- + NO, and have measured kinetic energy and angular distributions for both channels. These measurements yield bond dissociation energies and a wealth of information concerning the detailed photodissociation mechanism. The interpretation of our results is aided by *ab initio* calculations on the ground and excited states of $N_2O_2^-$ which are also reported here.

Section II of this paper describes the experimental apparatus used in this work, including a newly developed pulsed discharge negative ion source used in these and other experiments in our laboratory. Section III details the experimental data and the results of *ab initio* molecular orbital calculations. Section IV contains analysis of the results while Section V discusses the mechanism for dissociation and the potential energy surfaces that mediate this process.

II. Experimental

A. Fast Beam Translational Spectrometer

The fast beam translational spectrometer used in these experiments was designed to study the photodissociation of neutral free radicals generated by photodetachment of a mass-selected anion beam, and has been described in detail elsewhere.[17,18] The anion photodissociation experiment described in this paper is somewhat simpler, as only one laser is required. The photodissociation of O_4^- using a similar instrument has recently been

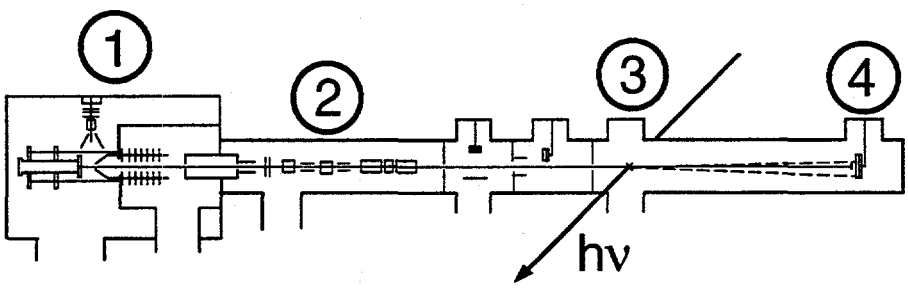


Figure 1: The fast ion beam translational spectrometer: 1) ion source region, 2) mass selection and ion optics, 3) dissociation laser port, 4) time- and position-sensitive MCP detector. A more complete explanation of the apparatus, which was designed for photodissociation of neutral free radicals, can be found in Refs. 17 & 18.

reported by Continetti *et. al.*[6] The experimental apparatus is shown in Figure 1 and is briefly described as follows.

A pulsed supersonic expansion of 10% N_2O in argon is either crossed with a 1 keV electron beam or subjected to a pulsed electric discharge to form an internally cold beam of $N_2O_2^-$. The pulsed discharge negative ion source developed in our laboratory was mentioned in a previous investigation,[19] and a full description will be given in Section IIB.

Anions formed in the source are accelerated to a laboratory energy of 8 keV, separated according to mass by a collinear beam modulation time-of-flight mass spectrometer,[20] axially compressed with a pulsed electric field, and slightly focused with an einzel lens. The dissociation laser, an excimer-pumped dye laser polarized perpendicular to the ion beam direction, is fired at the appropriate time to intersect only ions of mass 60. Photofragments which recoil out of the parent ion beam are detected with high efficiency using microchannel plate detectors located either 1 or 2 meters from the photodissociation laser.

Two types of experiments are performed to characterize the photodissociation of $N_2O_2^-$. First, the total flux of photofragments is detected as a function of laser wavelength, mapping out the total dissociation cross section. In the second type of experiment the laser is operated at a fixed wavelength and the dissociation dynamics are investigated. For each dissociation event, the two photofragments are detected in coincidence. The position of both photofragments on the detector face and the time difference between their arrival are simultaneously recorded. From this experiment we obtain three important quantities that describe the photodissociation: product mass ratios, translational energy distributions, and angular distributions. The masses of the product fragments for each dissociation event can be determined by conservation of momentum in the center-of-mass frame. After correcting for the finite geometric acceptance of the detector, the center-of-mass recoil energy and recoil angle (with respect to the electric vector of the polarized laser) are analytically determined for each event.[18] The calibration of the detector by photodissociation of the Schumann-Runge bands in O_2 is discussed in detail elsewhere.[21] We record approximately one pair of fragments every 30 laser shots. A

complete data set consisting of approximately 30,000 coincident events can be collected in 3 to 5 hours. The energy resolution under the conditions in these studies is 25 meV.

The dynamics experiment requires that both fragments be detected. However, the front of the detector is biased at a high negative potential, such that only neutral fragments can be detected. Although the dissociation of $N_2O_2^-$ yields an ion and a neutral, the experimental conditions are such that the anion fragment is typically neutralized prior to reaching the detector. This fragment can be either photodetached by an additional photon from the dissociation laser, e.g.

$$O^- + h\nu \rightarrow O + e^- \quad ; \quad NO^-(\upsilon = 0) + h\nu \rightarrow NO + e^- \tag{2}$$

or it can autodetach, since vibrationally excited NO^- will lose an electron via[22]

$$NO^-(\upsilon > 0) \rightarrow NO + e^-. \tag{3}$$

(The electron affinity of NO is only 0.024 eV). Hence, photodissociation of $N_2O_2^-$ followed by (2) or (3) results in two neutral fragments, enabling us to perform the coincidence experiment. We emphasize that even though only neutral particles are detected, this experiment nevertheless probes the dynamics of ion dissociation, because the velocity vector of the nascent ionic fragment is not affected to any appreciable degree by the loss of an electron.

B. Pulsed Discharge Negative Ion Source

One of the more successful pulsed negative ion sources developed in recent years is the design of Johnson *et al.*,[23] in which a pulsed molecular beam is crossed by a continuous beam of high energy (≈ 1,000 eV) electrons. In this source, slow secondary electrons generated by ionization initiate ion-molecule chemistry which can be used to

produce a wide variety of negative ions and negative ion clusters. A disadvantage of this source is that the secondary electrons are thermalized rapidly in the high number density of the pulsed free jet, so that only those processes which can be initiated by very low energy (1 eV) electrons can occur. Thus, the production of F^- from NF_3 is very efficient, because this dissociative attachment process occurs for near zero energy electrons.[24] On the other hand, the production of OH^- from H_2O or CH_3O^- from CH_3OH does not work well in this source because dissociative attachment requires higher electron energies (> 5 eV). An ion source in which higher energy processes can be efficiently driven is clearly desirable. To this end, we have developed a pulsed electric discharge negative ion source that is straightforward to build and use.

One can find examples of many pulsed supersonic expansion discharge sources in the literature for the production of neutral free radicals and cations. To our knowledge, the only published examples of negative ion production by these methods are given by Bondybey *et. al.*,[25] and Meiwes-Broer *et. al.*[26] Our pulsed discharge negative ion source is similar in design to that used by Ohshima and Endo[27] for production of internally cold transient neutral species.

Figure 2 shows the details of the design. The faceplate (2) of the pulsed valve (1) is separated from the cathode (4) by a Teflon insulator (3). Discharge occurs through the channel of a second insulator (5), to the anode (7). For flexibility, nozzles (6) of various inner diameter, shape, and composition can be inserted into the anode to enhance or suppress clustering and/or cooling of the negative ions. Teflon screws secure the entire assembly to the pulsed valve faceplate. We have used the discharge source with either a

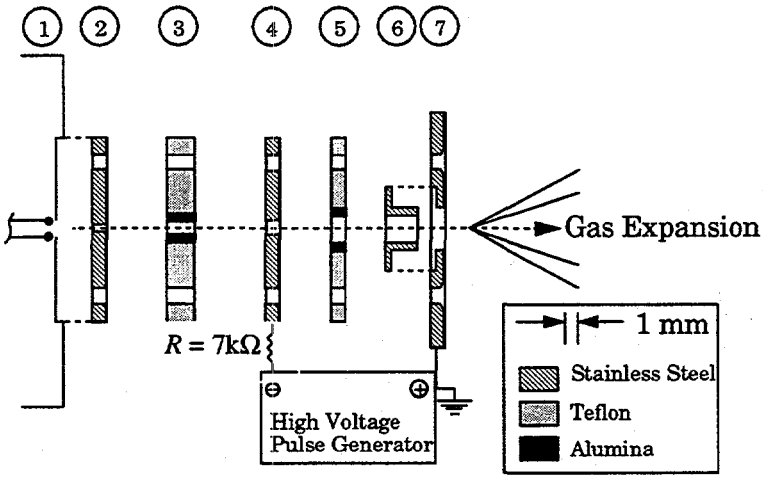


Figure 2: The pulsed discharge negative ion source. Each component is described below, with its thickness (T) and gas channel diameter (D) given in mm for components 2 - 6. (1) Pulsed valve body, (2) pulsed valve faceplate [T = 2, D = 0.6], (3) Teflon insulator with alumina insert along gas channel [T = 4, D = 2], (4) cathode [T = 2, D = 2], (5) Teflon/alumina insulator [T = 2, D = 3], (6) Nozzle insert [T = 5, D = 3], (7) Anode.

piezoelectric pulsed valve of the type developed by Trickl[28] or a General Valve solenoid valve.

The increasing channel diameter along the gas flow direction proved to be the crucial element in attaining intense, internally cold negative ion beams. An advantage of this design is that ion formation occurs within the flow channel, *before* the expansion, maximizing the number of ion-neutral collisions that occur during the free jet expansion into the source chamber.

A pulse of ≈ -500 to -700 V is generated by a MOSFET switch[29] and applied to the cathode (3). The pulse is typically 15 μs in width, a fraction of the 150 μs gas pulse. The stability of the ion beam is increased if a very weak (<1 μA) electron beam intersects the gas flow as it emerges from the nozzle. This effect is most likely due to creation of positive ions which move against the gas flow towards the negatively charged cathode, acting as seeds to begin breakdown of the gas pulse. Ion stability and intensity is also increased when the discharge pulse comes very near the beginning of the gas pulse.

The duration of the discharge is 1 to 8 μs. The current through the discharge is followed by monitoring the voltage at the cathode both before and during breakdown of the gas. Typical discharge currents range from 5 to 30 mA depending on the composition of the gas mixture and the value of a ballast resistor (typically 7 kΩ) in series with the cathode. As the current through the discharge is increased, the voltage across the discharge increases only slightly, indicating that the discharge is essentially of the "normal glow" variety.[30]

Most gaseous precursor molecules are diluted to 1% to 10% concentration with a carrier gas (e.g. Ar, Ne, or O_2) and expanded with a backing pressure of 2 - 8 atm. For liquid precursors the carrier gas is bubbled through the liquid. Substantial rearrangement of the precursor molecules is often observed. For example, a discharge of neon bubbled through methanol produces not only CH_3O^-, but also C_2^-, C_2H^-, and $C_2H_2^-$. Despite the high energy environment in which ions are formed, their internal degrees of freedom can still be effectively cooled in the supersonic expansion. Photofragment translational spectroscopy of predissociation from the O_2 Schumann-Runge band shows that an O_2^- rotational temperature of ≈ 50 K can be attained.[31] For the case of CH_3O^- produced in a discharge, photoelectron spectroscopy shows that 99% of these ions are in their vibrational ground state. These temperatures are as cold or colder than ions produced with the electron-gun source in our laboratory. In summary, we have found that the pulsed discharge negative ion source combines many of the strengths of other available sources with some significant new advantages. The source requires very little maintenance and is easily built at modest expense.

III. Results

A. Experimental Data

Figure 3 shows the total dissociation cross section spectrum of $N_2O_2^-$ from 525 - 580 nm, normalized to dissociation laser power. The broad, partially resolved features in this spectrum are reproducible, although the finer structure (< 1 nm) is not. The peak

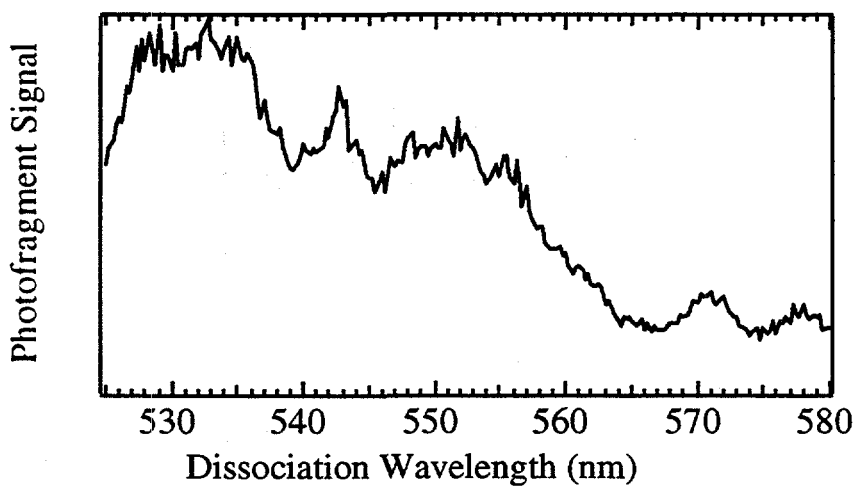


Figure 3: Total photodissociation cross section spectrum of $N_2O_2^-$. Intensity is normalized to dissociation laser power. Laser step size is 0.25 nm.

spacings range from ≈ 150 to 500 cm^{-1}, and are presumably due to excitation of multiple vibrational modes in the upper electronic state.

The main information we obtain about $N_2O_2^-$ photodissociation is derived from dissociation dynamics data acquired at three wavelengths; 570, 532, and 500 nm. Although wavelength resolved cross section data were not obtained below 525 nm, the cross section appears to be higher at 500 nm than at 532 nm. The first result from this

dynamics data is the photofragment mass spectrum, shown in Figure 4 for 532 nm dissociation. It is immediately seen that there are two product channels occurring at this wavelength:

$$N_2O_2^- + h\nu \rightarrow O^- + N_2O \text{ (channel 1)} \tag{4}$$

$$\rightarrow NO^- + NO \text{ (channel 2)} \tag{5}$$

with similar product mass spectra obtained at 500 and 570 nm.

For each channel, the joint translational energy and angular distribution is given by[18,32]

$$P(E_T,\theta) \propto P(E_T)\cdot(1+\beta(E_T)P_2(\cos\theta). \tag{6}$$

Here, E_T is the translational energy release, θ is the angle between the photofragment recoil velocity vector and the laser polarization, and β is the anisotropy parameter. Although β can depend on the translational energy, no such dependence was observed here.

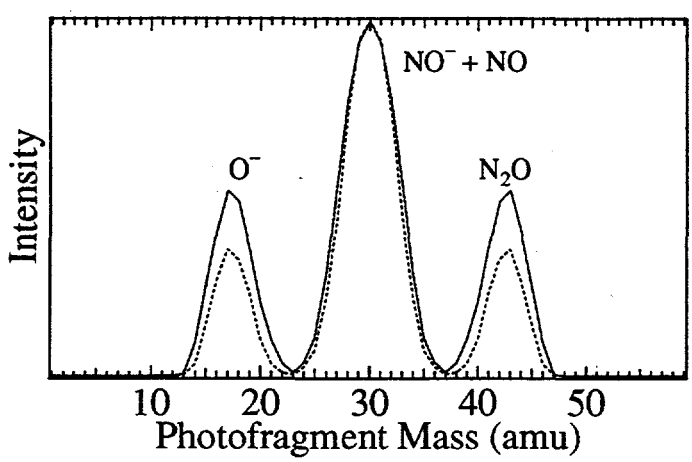


Figure 4: Product mass spectrum at 532 nm as a function of laser fluence. The central peak corresponds to the channel (2) fragments, NO^- + NO. The other two peaks correspond to the channel (1) fragments, O^- + N_2O. Laser fluences are (—) 3.2 J/cm^2, (- - -) 1.6 J/cm^2.

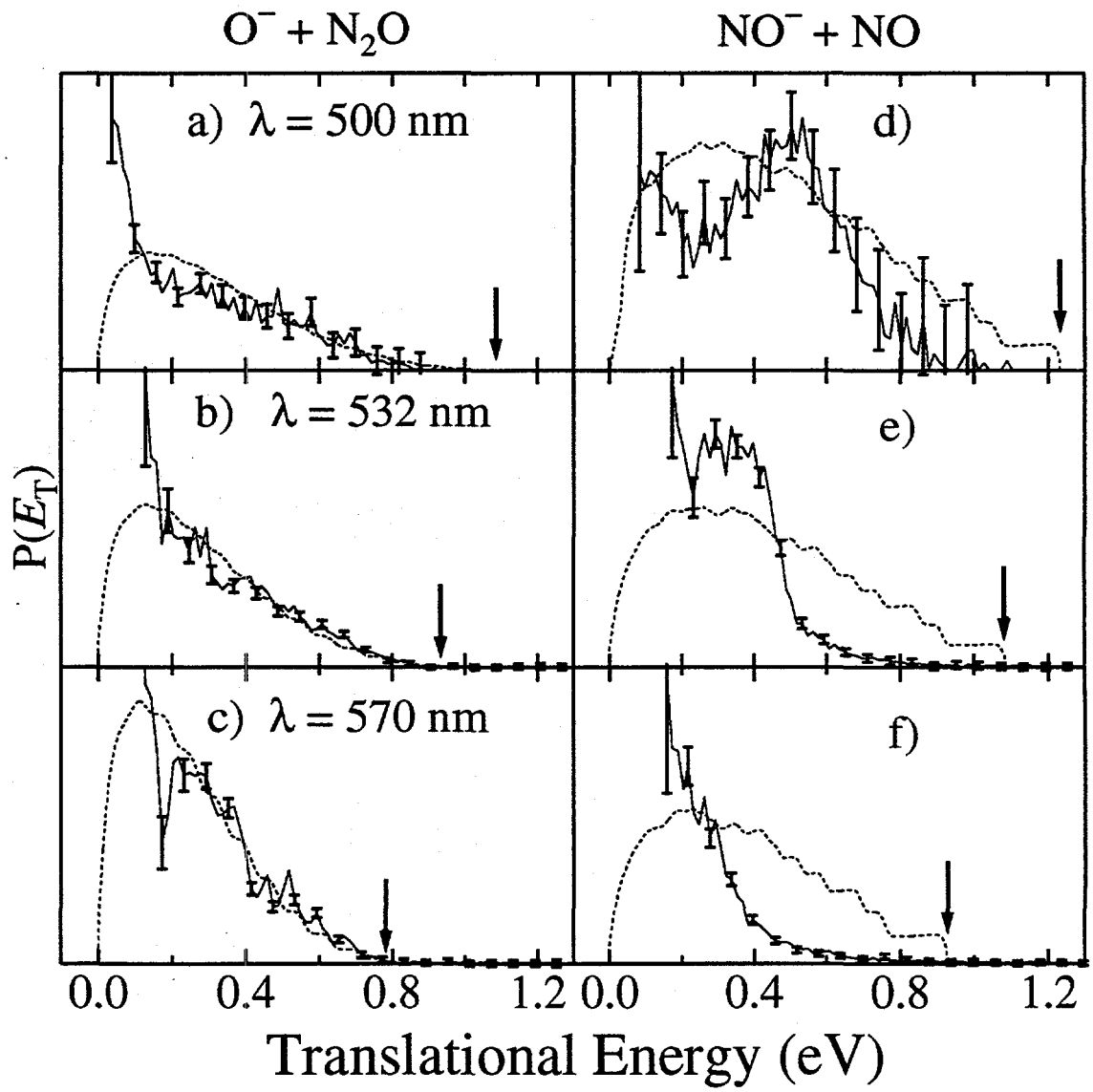


Figure 5: Translational energy distributions $P(E_T)$ at three excitation energies: (a) - (c): O^- + N_2O. (d) - (e): NO^- + NO. The solid lines connect the data points, which have error bars (± 1σ) on every fourth point. The dashed line is the prior distribution discussed in the text. The arrows denote the maximum energy available to translation for each photon energy/product channel combination.

Figure 5 shows the translational energy release distribution $P(E_T)$ for both channels at each wavelength. Statistical error bars (±1σ) are drawn for every fourth data point. All three translational energy distributions for channel 1 (Figures 5a-c) are maximal at the lowest translational energies we can detect, decreasing steadily towards

higher E_T. For the data at 570 nm in Figure 5c, which is replotted as Figure 6, vibrational structure is apparent in the $P(E_T)$ distribution. The observed spacing corresponds to excitation of the bending mode (ω_2 = 588 cm^{-1}) in the N_2O fragment.[37] This structure is less evident as the photon energy is increased in figures 5b and 5a. For the NO^- + NO

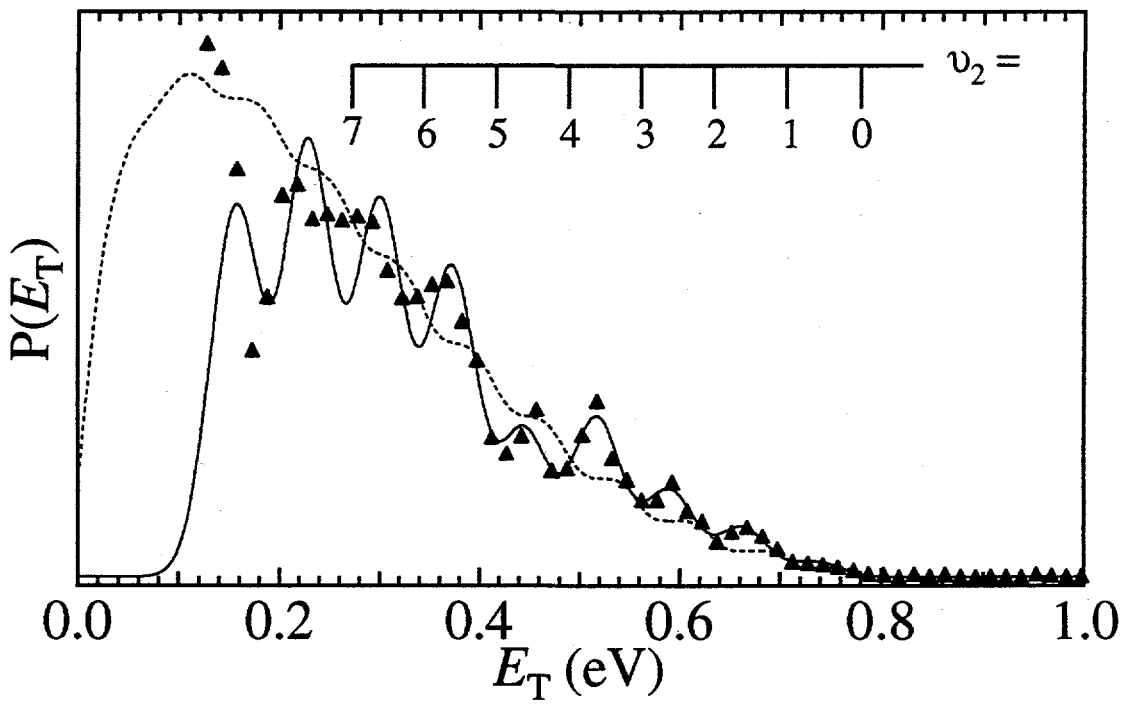


Figure 6: Translational energy distribution for O^- + N_2O at 570 nm. Triangles denote the experimental data, the solid line is the empirical fit [peak heights given by α_n of eqn. (10)] assuming bending excitation accounts for all N_2O vibrational excitation. A vibrational comb is shown for the υ_2 bending mode progression in the N_2O fragment. The dashed line is the prior distribution given by eqn. (11).

channel in figures 5d - 5f, no vibrational structure is evident. Unlike channel (1), the most probable translational energy becomes larger as the photon energy is increased.

For the NO^- + NO channel, the angular distributions are isotropic ($\beta = 0$) at all three wavelengths. For O^- + N_2O, the angular distributions are slightly anisotropic: β(570 nm) = -0.15 ± 0.2, β(532 nm) = -0.3 ± 0.2, and β(500 nm) = -0.6 ± 0.3. The anisotropy becomes more negative with increasing excitation energy. The uncertainty in the anisotropy parameter is a combination of a well-defined statistical error and an estimated systematic error which reflects the aberrations in our detector response for low translational energies. The values of the anisotropy parameter are related to both the symmetry and the lifetime of the excited anion states.

Because the data in this paper use the newly developed pulsed discharge source to produce $N_2O_2^-$, it is important to characterize the ions in this beam by comparing their photoelectron spectra, shown in Figure 7, to that obtained by Arnold[16] using an electron gun source. We present this data (from a separate apparatus in our laboratory) because photoelectron spectroscopy is a sensitive probe of ion source conditions, as is evident from the work of Posey and Johnson[7] described earlier. The well-resolved vibrational structure seen in Figure 7 at low electron kinetic energy (<1.2 eV) arises from bound neutral states of N_2O_2 (C_{2v}). The weak electron signal from 1.3 to 2.2 eV is from photodetachment of isomer (III), $ONNO^-$, based on comparison with the spectra obtained when ions are produced from NO seeded in argon.[33] This isomer is not present under the gentler conditions of an electron gun ion source as used in references 7 and 16

The photon energies used to dissociate $N_2O_2^-$ are indicated by arrows in Figure 7, demonstrating that direct photodetachment of the C_{2v} isomer is not possible at these energies. However, the photoelectron spectrum in Figure 7 does raise the possibility that the NO^- + NO product channel comes from photodissociation of the small amount of the

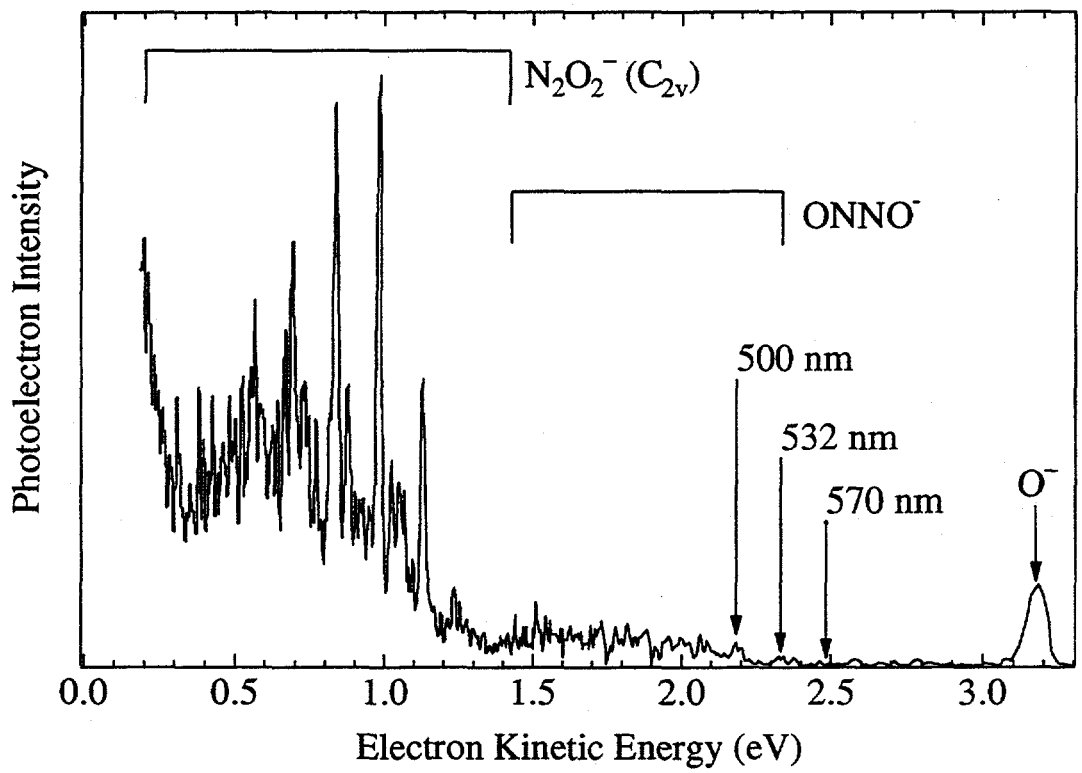


Figure 7: 266 nm photoelectron spectrum of $N_2O_2^-$ using the pulsed discharge source. The sharp features at 1.13 eV and below are due to the (C_{2v}) N_2O_2 neutral. The signal from 1.3 - 2.2 eV arises from an impurity of the ONNO$^-$ isomer (see text). The energies corresponding to the three dissociation wavelengths used in the present study are also shown.

ONNO$^-$ isomer produced in the pulsed discharge source. To examine this possibility we have performed the same photodissociation dynamics experiments described in this section using an electron gun ion source. The NO$^-$ + NO photofragments are still observed in similar amounts as when $N_2O_2^-$ is produced in the pulsed discharge source, confirming that this is a primary product channel of (C_{2v}) $N_2O_2^-$ photodissociation. However, the $P(E_T)$ distributions also show that $N_2O_2^-$ produced by the electron gun ion source is not as efficiently cooled in the supersonic expansion; for example, no vibrational structure is apparent in the O$^-$ + N_2O $P(E_T)$ at 570 nm. For this reason we present data exclusively from the pulsed discharge source in this paper.

B. *Ab Initio* Calculations

The product state information presented above is a direct result of the potential energy surfaces involved in the dissociation of $N_2O_2^-$. In order to gain more insight into the complicated electronic structure of the $N_2O_2^-$ system, we have performed *ab initio* molecular orbital calculations using the GAUSSIAN 92 package of programs.[34] In the recent work of Arnold and Neumark,[16] the ground state of $N_2O_2^-$ was assigned as $^2B_2$ with the orbital occupation ($\ldots 4b_2^2\ 2b_1^2\ 8a_1^2\ 1a_2^2\ 5b_2^1$), based on *ab initio* calculations (UHF and MP2/6-31+G*) and comparison with the isoelectronic species NO_3 and FCO_2. We have extended these calculations to include the lowest excited states of each spatial symmetry within the C_{2v} and C_s point groups. The results of the C_{2v} calculations are summarized in Table I.

Optimized geometries and vibrational frequencies were obtained at the UHF/ 6-31+G* level of theory. Five unique electronic states are found,[35] which, in order of increasing energy, are $^2B_2$, $^2A''$, $^2B_1$, $^2A_2$, and $^2A_1$ states. All of these configurations, with the exception of $^2A_2$, are bound along all vibrational coordinates. At the UHF level, the $^2A_2$ state is unbound along the υ_4 (NO_2 asymmetric stretch) normal coordinate.

When electron correlation is taken into account at the MP2/6-31+G* level of theory, geometry optimizations reveal that the highest energy $^2A_1$ state has no stationary point within C_{2v}. In addition, the SCF wavefunction for the $^2A''$ (C_s) state is no longer convergent after a few steps in the (MP2 energy gradient) geometry optimization. For this reason we discuss no other calculations at C_s geometries. At the MP2 level, therefore, the only electronic states which have stationary points along all coordinates are the ground state, $^2B_2$, and the first two excited states $^2B_1$, and $^2A_2$. Frequency calculations at these

$\tilde{X}$ ($^2B_2$)	Energy[b] (Hartrees)	$<S^2>$	T_e (eV)	R_{NN} (Å)	R_{NO} (Å)	∠NNO	ω_1 (a_1) (NN str.)	ω_2 (a_1) (NO str.)	ω_3 (a_1) (NO_2 bend)	ω_4 (b_2) (NO_2 a. s.)	ω_5 (b_2) (N_2O a. b.)	ω_6 (b_1) (umbrella)
UHF	-258.494 495	0.920	0.0	1.253	1.253	119.2	1582	1086	692	1434	654	836
MP2	-259.165 961	0.926	0.0	1.270	1.253	120.3	1411	1241	679	1538	609	998
QCISD	-259.190 341	1.022	0.0	1.267	1.296	120.0	...	...	...	...	...	...
QCISD(T)[a]	-259.215 226	1.022	0.0	...	...	...	...	...	...	...	...	...
$\tilde{A}$ ($^2B_1$)												
UHF	-258.438 789	0.778	1.52	1.352	1.220	120.1	1029	1404	735	1659	783	837
MP2	-259.124 816	0.797	1.12	1.354	1.274	120.6	990	1219	687	1389	800	713
QCISD	-259.132 971	0.798	1.56	1.345	1.274	120.7	...	...	...	...	...	...
QCISD(T)[a]	-259.161 024	0.798	1.47	...	...	...	...	...	...	...	...	...
$\tilde{B}$ ($^2A_2$)												
UHF	-258.407 553	1.404	2.36	1.255	1.320	126.6	1364	1018	707	1668i	685	470
MP2	-259.066 859	1.161	2.70	1.131	1.479	130.3	2177	613	472	1559i	666	716
QCISD	-259.101 678	1.306	2.41	1.194	1.417	128.7	...	...	...	...	...	...
QCISD(T)[a]	-259.133 796	1.306	2.24	...	...	...	...	...	...	...	...	...
$\tilde{C}$ ($^2A_2$)												
UHF	-258.325 524	0.882	4.60	1.205	1.347	122.3	1113	884	659	2264	655	689

Table I: *Ab initio* results: Optimized geometries, energies, and vibrational frequencies (frequencies in cm^{-1}).

[a] QCISD(T) energies evaluated at the QCISD/6-31+G* geometry
[b] All calculations use the standard 6-31+G* basis set

stationary points reveal that, just as at the UHF level, the first two states are bound along all coordinates, while the $^2A_2$ state has one imaginary frequency. In order to treat electron correlation more rigorously, geometries for the lowest three states were further optimized at the QCISD/6-31+G* level of theory, with a single point QCISD(T) calculation at this geometry. No vibrational frequencies were calculated at this level of theory. Finally, vertical excitation energies from the ground state are also shown at each level of theory in Table II.

State	UHF	MP2	MP4SDTQ	QCISD	QCISD(T)
$\tilde{C}$ ($^2A_2$)	4.90	3.90	4.11	3.78	3.52
$\tilde{B}$ ($^2A_2$)	3.07	3.86	4.08	3.30	3.24
$\tilde{A}$ ($^2B_1$)	1.72	1.46	1.40	1.67	1.59
$\tilde{X}$ ($^2B_2$)	0.	0.	0.	0.	0.

Table II: Vertical excitation energies in eV

The results in Table I show marked differences in the structures of the $\tilde{X}$, $\tilde{A}$, and $\tilde{B}$ states. The ground state structure (R_{NN} = 1.27 Å, R_{NO} = 1.30 Å) is best described as an N–N double bond (R_{NN} = 1.254 Å in $(CH_3)_2N_2$ [36]) and an N–O distance intermediate between single (R_{NO} = 1.398 Å in CH_3O–NO) and double (R_{NO} = 1.212 Å in HNO) bond character. In the $\tilde{A}$ ($^2B_1$) state, the N–N bond length increases to R_{NN} = 1.35 Å, indicating that it too is now intermediate between single (R_{NN} = 1.447 Å in N_2H_4) and double bond character. The N–O bond length is very similar to that of the ground state, differing by only 0.02 Å. In both the $\tilde{X}$ and $\tilde{A}$ states the geometries are not strongly dependent on the level of theory. In the $\tilde{B}$ ($^2A_2$) state, the bond lengths and bond angles change quite drastically when electron correlation is included with the MP2 or QCISD methods. The QCISD optimized geometry (R_{NN} = 1.19Å, R_{NO} = 1.42 Å) is characterized by an N–N bond

intermediate between double and triple (R_{NN} = 1.094 Å in N_2), and an N–O bond significantly longer than a typical single bond. It is also useful to compare these bond lengths with that of the photofragments observed in this study. In N_2O, R_{NN} = 1.128 Å, R_{NO} = 1.184 Å,[37] while for NO and NO^- the respective bond lengths are R_{NO} = 1.151 Å, and 1.25 Å.[22]

To determine the degree of confidence that one can place in the *ab initio* results, one can examine both spin contamination and the sensitivity of the calculated geometries and frequencies to the level of calculation. The unrestricted Hartree-Fock wavefunctions of all states suffer from some degree of spin contamination by states of higher multiplicity. As shown in Table I, spin contamination ($<S^2>$ = 0.75 for a pure doublet state) is quite large for the $\tilde{B}$ ($^2A_2$) state, small for the $\tilde{A}$ ($^2B_1$) state, and somewhere in between for the ground state. In terms of structural parameters, for the ground state the largest change in bond length (0.04 Å) and bond angle (1°) is relatively modest for the different treatments. In the $\tilde{A}$ ($^2B_1$) state the only significant change is in R_{NO}, which increases by 0.05 Å between UHF and the two correlated methods, the latter two calculations yielding essentially identical geometries. For the $\tilde{B}$ ($^2A_2$) state the results are less satisfying: the bond lengths and angles change by significant amounts (0.06 - 0.12 Å; 2-4°) at the three levels of theory. The percent change in each vibrational frequency ($\omega_{UHF}/\omega_{MP2}$) for the $\tilde{X}$ and $\tilde{A}$ states is at most 19%, while in the $\tilde{B}$ state four of the six frequencies change by 50 - 60 %. Therefore one must treat the *ab initio* results with some caution, particularly for the $\tilde{B}$ state, in which all three criteria show signs that the single-configuration picture may not be sufficient for a quantitative description.

Species	E_0[G1] (hartrees)	E_0[G2] (hartrees)	
$\tilde{X}$ ($^2B_2$) $N_2O_2^-$	-259.525 136	-259.525 952	
$\tilde{X}$ (^2A_g) trans-ONNO$^-$ [a]	-259.527 065	-259.527 948	
$\tilde{X}$ ($^1\Sigma_g^+$) N_2O	-184.437 103	-184.437 117	
X ($^2\Pi_g$) NO [b]	-129.739 12	-129.739 95	
X ($^3\Sigma_g^-$) NO$^-$ [b]	-129.736 89	-129.737 52	
(^2P_u) O$^-$ [b]	-75.033 85	-75.033 41	
Reaction		ΔH_0[G-1] (eV)	ΔH_0[G-2] (eV)
$N_2O_2^-$ (C_{2v}) $\rightarrow$ O$^-$ + N_2O		1.47	1.51
$N_2O_2^-$ (C_{2v}) $\rightarrow$ NO$^-$ + NO		1.34	1.32
O$^-$ + N_2O $\rightarrow$ NO$^-$ + NO		-0.14	-0.19
$N_2O_2^-$ (C_{2v}) $\rightarrow$ trans-ONNO$^-$ (C_{2h})		-0.05	-0.05

Table III: G-1 and G-2 *ab initio* energies and heats of reaction

[a] The *cis* (C_{2v}) isomer lies 0.1 eV higher at the MP2/6-31+G* level of theory.
[b] Energies taken from Ref. 39.

For comparison with the thermodynamic information obtained from the data in Section IV, we have also performed Gaussian-1and Gaussian-2 calculations[38,39] giving the total electronic energies (including zero-point energies) of $\tilde{X}$ ($^2B_2$) $N_2O_2^-$, $\tilde{X}$ (^2A_g) *trans*-ONNO$^-$, and $\tilde{X}$ ($^1\Sigma_g^+$) N_2O. These calculations are summarized in Table III. The ONNO$^-$ configuration is included because it is also strongly bound and has been implicated as a reaction intermediate in reaction (1). The G-1 and G-2 procedures, due to Pople and coworkers, are quite successful in predicting dissociation energies to within a target accuracy of ± 0.1 eV (2 kcal/mol) for a wide range of neutral and ionic species. Within the accuracy of these calculations, $N_2O_2^-$ (C_{2v}) and ONNO$^-$ (C_{2h}) are essentially thermoneutral. This differs from the lower-level calculation in Ref. 8, which predicts the C_{2v} isomer to be more stable by 4.4 kcal/mol.

IV. Analysis

A. Product branching ratio

In Figure 4, the ratio between the peak areas for channel 1 : channel 2 at the higher laser fluence is about 1:1. However, the extraction of quantitative branching ratios from this data is more complicated in the case of $N_2O_2^-$ photodissociation than in our previous studies of free radical photodissociation. As mentioned in the experimental section, only neutral fragments can be detected with our time- and position-sensitive detector. Although both product channels result from single photon dissociation, *detection* of O^- + N_2O requires a second photon (to detach the O^- fragment), while NO^- ($\upsilon > 0$) + NO detection does not require another photon due to autodetachment of the NO^- fragment. The channel 1 : channel 2 peak area ratio increases with laser fluence over the entire range available to us, indicating that the two photon process producing O + N_2O is not yet saturated. Therefore the branching ratio stated above should be taken as a lower bound to the true channel 1 : channel 2 branching ratio at 532 nm, because not all of the O^- is photodetached even at the highest laser fluence. Thus, while quantitative branching ratios cannot be obtained, it is clear that O^- + N_2O is the major product channel.

B. Dissociation energies and product energy distributions

From the translational energy distributions $P(E_T)$ in Figure 5, we can extract both thermodynamic data and information on product state energy disposal resulting from $N_2O_2^-$ photodissociation. By conservation of energy we may write:

$$h\nu = D_0(N_2O_2^- \rightarrow O^- + N_2O) + E_T + E_{int}(N_2O) \tag{7}$$

$$h\nu = D_0(N_2O_2^- \rightarrow NO^- + NO) + E_T + E_{int}(NO) + E_{int}(NO^-) \tag{8}$$

where $h\nu$ is the photon energy, D_0 is the dissociation energy for the specified channel, E_T is the energy in product translation, and E_{int} is the rotational, vibrational, and electronic energy of the molecular fragments. We assume in Eqns. 7 and 8 that the supersonically cooled $N_2O_2^-$ parent ion contains negligible internal energy. If for some translational energy E_T it can be ascertained that $E_{int} = 0$, the dissociation energy in the above equations can be deduced. Given these thermodynamic quantities, the total internal energy of the fragments corresponding to each energy E_T is determined, and in some cases this energy can be further delineated among rotational, vibrational, and electronic degrees of freedom.

The most detailed information can be obtained from Figure 6, showing an expanded picture of the $P(E_T)$ at 570 nm for $O^- + N_2O$. This distribution shows a series of peaks with spacings close to that of the υ_2 bending mode of the N_2O product. The resolved vibrational structure is significant, as this spectrum offers substantially more information than the qualitatively similar spectrum shown in Fig. 5a. At 570 nm, no signal is observed for $E_T > 0.78 \pm 0.03$ eV $\equiv E_T^{max}$. Because the disappearance of signal at E_T^{max} is quite abrupt, we assign it to the translational energy corresponding to ground state fragments, i.e. O^- ($^2P_{3/2}$) + N_2O ($\upsilon = 0, J = 0$). We therefore obtain

$$D_0(N_2O_2^- \rightarrow O^- + N_2O) = h\nu - E_T^{max} = 1.40 \pm 0.03 \text{ eV}, \tag{9}$$

where $h\nu = 2.175$ eV. This value agrees to within 0.1 eV of the calculated Gaussian-2 value given in Table III (1.51 eV). From the 0 K heats of formation of O^- (1.10 eV) and N_2O (0.88 eV),[40] we obtain the heat of formation of $N_2O_2^-$: $\Delta H^\circ_{f,0} = 0.58 \pm 0.04$ eV. From ΔH_0 for reaction (1) we find that $D_0(N_2O_2^- \rightarrow NO^- + NO) = 1.25 \pm 0.04$ eV. These

values of D_0 yield the maximum allowed values for E_T at each dissociation wavelength, which are shown by the arrows in Figure 5.

Based on the bond dissociation energy in Eqn. 9, the thresholds for $O^- + N_2O$ ($\upsilon_2 = n$) are indicated by the comb in Figure 6. The correspondence with the experimental peak spacings is quite good for $n < 6$. All the experimental peaks are offset towards lower translational energy than the thresholds for each vibrational state. This offset is presumably due to rotational excitation of each N_2O vibrational level. Hence, from this vibrationally resolved $P(E_T)$ distribution we would like to determine the vibrational and rotational energy distribution of the N_2O fragment.

The simplest way to extract such information is to assume that all of the N_2O excitation is in the υ_2 mode. We then construct an empirical model of the $P(E_T)$ distribution using a set of Gaussian peaks separated by the known vibrational spacings (588 cm^{-1}) of the υ_2 mode. A Gaussian functional form is chosen to model the rotational distribution purely on the basis of mathematical convenience. The model distribution $F(E_T)$ is given by $n' + 1$ Gaussians such that

$$F(E_T) = \sum_{n=0}^{n'} \alpha_n f_n(E_T) \ ; \ f_n(E_T) = \exp\left\{-\frac{1}{2}\left(\frac{E_T - (h\nu - D_0 - n\omega_2 - \Delta)}{\Gamma / 2.345}\right)^2\right\} \quad .(10)$$

Here n labels the number of N_2O bending quanta, Γ is the FWHM of the combined rotational envelope and instrumental resolution, and Δ is the energy difference between the maximum possible E_T for each n $\{E_T^{max}(n) = h\nu - D_0 - n\omega_2\}$ and the center of the nth Gaussian. Each $f_n(E_T)$ is truncated for $E_T > E_T^{max}(n)$, the limit corresponding to N_2O ($J = 0$). The coefficients α_n give the υ_2 product state vibrational distribution. The solid line in Figure 6 shows the best fit to the 570 nm $P(E_T)$ from a nonlinear least squares fitting

procedure, with parameters $\Delta = 0.045$ eV, $\Gamma = 0.058$ eV. Note that the same values of Δ and Γ were used for all vibrational states.

The fit is excellent for $E_T > 0.35$ eV. The less than satisfactory agreement below this energy may reflect excitation of other N_2O vibrational modes. In fact, it is difficult to tell from the 570 nm $P(E_T)$ if excitation in the stretching modes is occurring at low E_T, because the frequencies of the N_2O υ_1 and υ_3 modes are nearly integer multiples of the υ_2 frequency: $\omega_1 = 3.8\omega_2$ and $\omega_3 = 2.2\omega_2$. The possibility that there is excitation in all the N_2O modes has important implications for the dissociation mechanism and is discussed in more detail in Section V.

The vibrational structure in the experimental distributions is not as well-resolved at 532 nm as at 570 nm, and at 500 nm it is difficult to pick out any vibrational structure. This blurring can again be attributed to either increasing product rotational energy, or a breakdown in the assumption that all vibrational excitation of N_2O occurs as bending motion. In the absence of resolved structure, we do not attempt to fit the $P(E_t)$ distributions at the higher excitation energies with Eq. 10.

We next consider the translational energy release distributions for channel (2). In contrast to channel (1), for which all the $P(E_T)$ curves peak at the low energy cutoff of our instrument, in channel (2) it is clear that with increasing photon energy the peak in the translational energy distribution shifts to higher energy. Furthermore, each increase of photon energy, which corresponds to a 150 meV increment, shifts the peak of the $P(E_T)$ distributions to higher energy by approximately 150 meV. In other words, even though the 570 nm $P(E_T)$ distribution peaks at or below our low-energy cutoff, with little of the

available energy going into fragment translation, the extra energy available at 532 and 500 nm is, on average, channeled into translational energy.

The $P(E_T)$ distributions for channel (2) (figs. 5d-f) can be fit in the same manner as those for channel (1). These fits are underdetermined due to the lack of vibrational structure and the fact that there are two molecular fragments, NO and NO^-, which may both be rovibrationally excited. Even so, the density of vibrational product states is smaller for this case of two diatomics than for the products $O^- + N_2O$. The absence of resolved vibrational structure and the observation that the $P(E_T)$ distributions fall to zero well below E_T^{max} implies that both fragments have considerable vibrational and rotational excitation. At all three wavelengths there is very little probability of finding both diatomic fragments in their $\upsilon = 0$ vibrational states. We will show in Section V that a high degree of NO^- vibrational excitation is consistent with the photoelectron spectroscopy results of Posey and Johnson,[7] in which no evidence for the $NO^- + NO$ channel was evident.

V. Discussion

A. Dissociation Mechanism: $N_2O_2^- + h\nu \rightarrow O^- + N_2O$

We first consider the nature of the excited electronic state of $N_2O_2^-$ accessed in these experiments followed by a discussion of how this state dissociates to $O^- + N_2O$ products. To assign the initial electronic excitation we compare our experimental results with the *ab initio* calculations summarized in Tables I and II. The transition to the first excited state, $\tilde{A}\ (^2B_1) \leftarrow \tilde{X}\ (^2B_2)$, with a calculated $T_e = 1.47$ eV, is not electric dipole allowed. The lowest allowed transition, $\tilde{B}\ (^2A_2) \leftarrow \tilde{X}\ (^2B_2)$, has a calculated energy of

T_e = 2.24 eV (λ = 553 nm), and a vertical excitation energy of 3.24 eV (λ = 382 nm). The photodissociation cross section in Fig. 3 starts near the predicted T_e for this transition, and rises steadily towards the blue as would be expected when the vertical excitation energy is considerably larger than T_e. Furthermore, we measure a negative photofragment anisotropy parameter, β, indicating that the initial electronic excitation is a perpendicular transition. The $\tilde{B}$ ($^2A_2$) $\leftarrow$ $\tilde{X}$ ($^2B_2$) transition has the required perpendicular character, involving the promotion of an electron from the doubly occupied out-of-plane $1a_2$ orbital to the singly occupied in-plane $5b_2$ orbital. Based on these considerations, we assign the electronic excitation of the $N_2O_2^-$ ion in our experiment to the first optically allowed transition $\tilde{B}$ ($^2A_2$) $\leftarrow$ $\tilde{X}$ ($^2B_2$).

In order to consider the dissociation dynamics subsequent to electronic excitation, it is important to understand how the electronic states of $N_2O_2^-$ correlate to product channels. Figure 8 represents the adiabatic picture of the relevant surfaces, where the lowest states of each symmetry are correlated to asymptotic states to determine the evolution of the electronic wavefunctions. As dissociation begins towards the O^- + N_2O asymptote, the extension of the N—O bond lowers the ion's symmetry from C_{2v} to C_s, such that the in-plane orbitals (a_1 and b_2) become a′ while the out-of-plane orbitals (a_2 and b_1) correlate to a″. On the other hand if one approaches from the asymptotic region, $N_2O_2^-$ is formed by combining O^- (^2P_u) with N_2O ($^1\Sigma_g^+$), resulting in three electronic states of C_s symmetry: A′, A′, and A″. By adiabatically correlating the lowest states of each symmetry, the $\tilde{X}$ ($^2B_2$) ground state becomes $^2A'$, while the $\tilde{A}$ ($^2B_1$) excited state matches with A″. The third state, $\tilde{B}$ ($^2A_2$), *cannot* correlate with the remaining A′ state,

and must instead be matched with an A″ state that adiabatically leads to O (^3P_g) + N_2O^- ($^2A'$). These products are not energetically accessible at the photon energies used in the present study. The remaining A′ state arising from O^- (^2P_u) with N_2O ($^1\Sigma_g^+$) most likely correlates with the unbound $\tilde{C}$ ($^2A_1$) state, although it is possible that a lower-lying $^2B_2$ state exists.

The key feature of Figure 8 is that the $\tilde{B}$ ($^2A_2$) state, which we assign as the initial excited state in our experiment, does not correlate to ground state O^- + N_2O products. A nonadiabatic transition to another electronic surface is required in the dissociation mechanism. The question remains as to whether dissociation occurs on an excited state surface, such as the repulsive $\tilde{C}$ ($^2A_1$) state, or if internal conversion to the ground state occurs prior to dissociation. The translational energy distributions $P(E_T)$ in Figure 5 can, in principle, help distinguish between these possible mechanisms.

The excited state mechanism appears reasonable at first glance. Given that N_2O is linear whereas the N-N-O moiety in $N_2O_2^-$ is strongly bent, one would expect significant excitation in the N_2O bend from excited state dissociation on a repulsive surface, which is consistent with the vibrational structure seen in the translational energy distribution at 570 nm. This mechanism would also result in relatively little N_2O rotational excitation, since the departing O^- ion cannot apply much torque to the N_2O fragment; this is again consistent with the observation of resolved vibrational structure in the $P(E_T)$ distribution , at least at 570 nm.

However, photodissociation studies of many small molecules have shown that when dissociation occurs on a repulsive potential energy surface, the translational energy

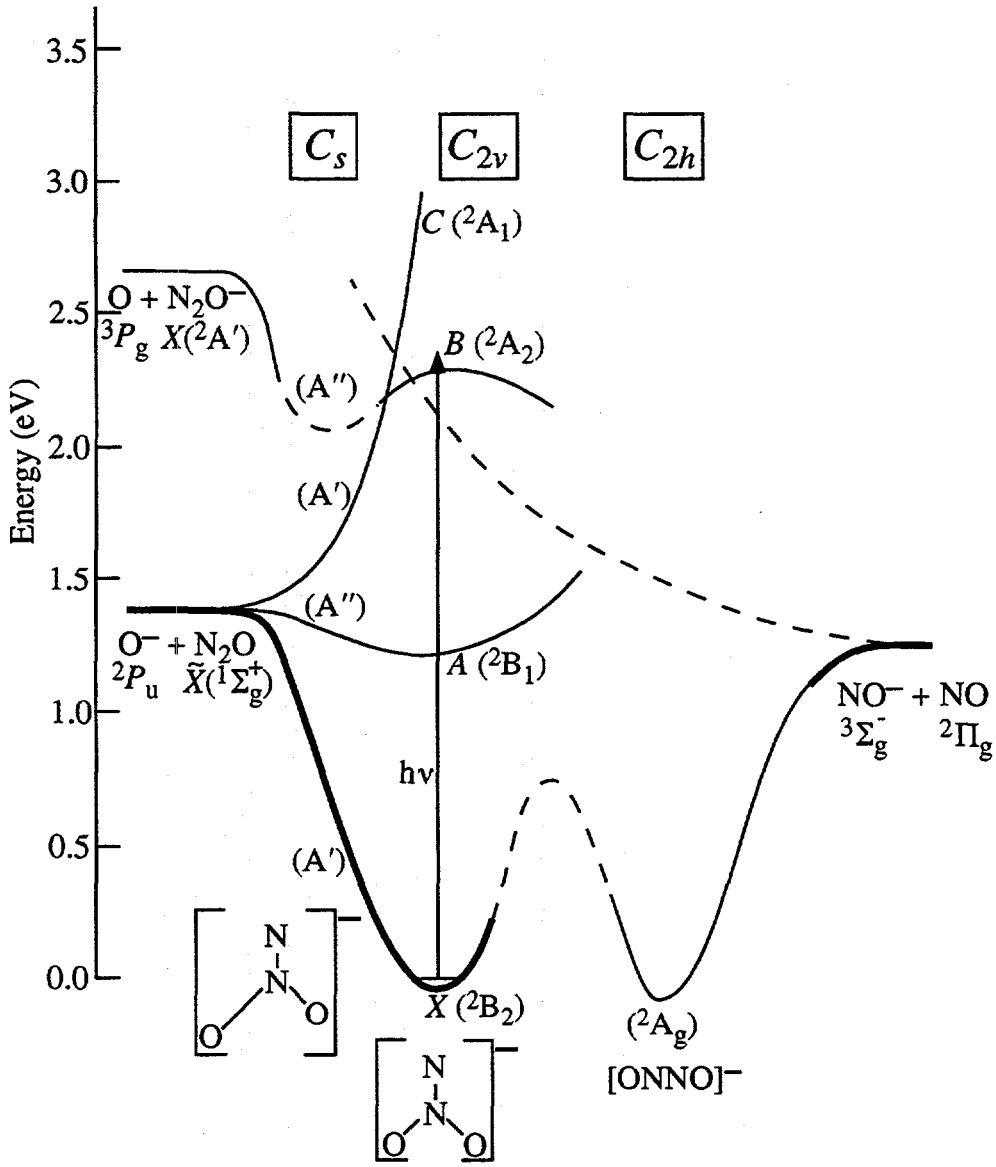


Figure 8: One dimensional projections of potential energy surfaces of the $N_2O_2^-$ system. The nuclear coordinate is approximately represented by the structures shown along the bottom of the diagram. Below the label for each symmetry group (C_s, C_{2v}, C_{2h}) are listed the electronic states of that symmetry, showing the correlation of electronic states. Energy differences between the $\tilde{X}\ (^2B_2)$ minimum and the two asymptotes are those derived from this experiment (thick lines). Thin lines are based on the *ab initio* results and molecular orbital correlations. Surfaces depicted by dashed lines are plausible guesses for which no calculations have been successful. According to our calculations (Table I), the $\tilde{B}\ (^2A_2)$ state does not have a local minimum in C_{2v} symmetry, but this may be a symmetry-breaking artifact (see text).

distribution peaks well away from $E_T = 0$. In contrast, all three experimental distributions for this channel peak near zero translational energy, and their overall shape is relatively independent of photon energy (compared to channel (2), for example). This trend is what one might expect if the dissociation mechanism were internal conversion to the ground state, followed by statistical decompostion without an exit barrier. The absence of a barrier is certainly reasonable for bond fission yielding an ion/neutral fragment pair, which will be governed by an attractive potential proportional to $-1/r^4$. It is useful for the moment to bypass the first step in this mechanism, internal conversion to the ground state, to determine if the observed translational energy distributions are consistent with statistical theory.

1. Prior distributions

The most straightforward comparison one can draw is with the prior translational energy distribution[41] for the photofragments. This model is expected to be most appropriate for a potential without an exit barrier, in which the transition state lies far enough in the exit channel that the nuclear configuration is essentially that of the products. One can then use the rotational constants and vibrational frequencies of N_2O in computing the prior distribution.

In order to compute the prior translational energy distribution, the vibrations are treated as discrete quantum states, while to good approximation the translational and product rotational degrees of freedom can be treated as continuous distributions. Based on the treatment by Levine,[41] one can show that the prior translational energy distribution $P^{\circ}(E_T)$ for the $O^- + N_2O$ channel is given by:

$$P^\circ(E_T) = C^\circ \sum_{\upsilon_1,\upsilon_2,\upsilon_3=0}^{E_\upsilon < E_{avail} - E_T} g(\upsilon_1,\upsilon_2,\upsilon_3)\cdot\sqrt{E_T} \tag{11}$$

Here E_{av} = hν - D_0 is the total energy available to the products and $g(\upsilon_1,\upsilon_2,\upsilon_3)$ is the degeneracy of the specified N_2O vibrational level; at each E_T, the sum is over all energetically allowed N_2O vibrational levels.

The prior distributions resulting from Eqn. 11, convoluted with a 0.025 eV FWHM Gaussian instrument resolution function, are superimposed on the experimental distributions in Figure 5a - 5c and Figure 6. The prior distributions peak near E_T = 0.1 eV (our experimental cutoff), decreasing towards higher E_T. Overall, the agreement between the prior and experimental distributions is quite good, except at the very lowest values of E_T where the error bars on the experimental $P(E_T)$ are large. The prior distributions match the general shape of the experimental distributions, and at each wavelength both distributions drop to zero at approximately the same value of E_T. Moreover, there is observable vibrational structure in the 570 nm prior distribution which becomes less prominent at lower wavelengths, in agreement with the experimentally observed trend.

In Figure 6 one can make a more detailed comparison between the prior and experimental distributions at 570 nm. Like the experimental data, the prior distribution contains periodic structure on an interval that matches the υ_2 bending mode of N_2O. Even though all three vibrational modes are excited, the υ_2 mode dominates the statistical distribution because it is (n + 1)-fold degenerate when n bending quanta are populated. Because of this effect, combined with the difficulty (see Section IV) in distinguishing υ_2 excitation from excitation in the υ_1 and υ_3 modes of N_2O, it is possible for even a

statistical distribution of product states to support a 588 cm^{-1} (i.e. ω_2) vibrational spacing in the translational energy distribution. We make this point because the experimental data at 570 nm may also be modeled quite convincingly by Eq. 10 (solid line in Figure 6) to represent excitation of pure N_2O bending motion, an interpretation that is irreconcilable with a statistical dissociation mechanism.

However, a closer comparison shows that structure in the prior distribution is broader than the experimental rotational envelope at 570 nm, particularly at high E_T. This broadening arises because for each N_2O vibrational level ($\upsilon_1,\upsilon_2,\upsilon_3$), the corresponding term in the summation of Eq. 11 begins abruptly at $E_T = E_{av} - E_\upsilon$. For this reason the thresholds in the prior distribution line up well with the vibrational comb in Figure 6. In addition, for each vibrational state the prior distribution extends all the way to $E_T = 0$, i.e. the maximum E_j allowed by energy conservation. This yields very broad rotational energy distributions for low vibrational states of N_2O, with narrower distributions only for the highest vibrational states. In contrast, the fitting of the experimental data to Eq. 10 indicates that the rotational envelope for each state is about the same, and is considerably narrower than that resulting from the prior distribution.

This discrepancy may simply indicate that the rotational distribution reflects the dissociation dynamics on the ground state surface, and that the constraints on N_2O rotation imposed by energy and angular momentum conservation alone are physically unreasonable. While there is some rotational excitation of the $N_2O_2^-$ in the ion beam, if we assume that that total angular momentum $J \cong 0$, then the N_2O rotational angular momentum j is approximately equal in magnitude to l, the orbital angular momentum of the separating fragments. Consider, for example, the N_2O (υ_2=4) level, for which the

maximum E_T is 0.48 eV at 570 nm. The value of $j_{max} = l_{max}$ determined by energy conservation alone is $96\hbar$, while l_{max} determined by considering the maximum l at which the translational energy is greater than the centrifugal exit barrier (assuming the appropriate $-1/r^4$ long range potential) is only slightly smaller, $93\hbar$. The corresponding impact parameters, given by $\mu v b_{max} = l_{max}$, are 1.84 Å and 1.79 Å, respectively, both of which are considerably greater than the radius of the N_2O molecule, 1.156 Å . Given that the O^- is originally bound to the central N atom of the N_2O fragment, these values appear to be unreasonably large even if there is substantial large amplitude vibrational motion on the ground state surface prior to dissociation. In this context it is not altogether surprising that the experimental rotational distributions are narrower than those predicted by statistical models.

Based on the reasonable overall agreement between the experimental results and prior distributions, we believe that the $O^- + N_2O$ channel results from decay on the ground state surface rather than dissociation on a repulsive excited state. While the latter mechanism certainly cannot be ruled out, it would be unusual for the translational energy distribution from such a process to be so similar to that predicted by a simple statistical model.

2. Angular distributions and RRKM lifetimes

The angular distributions and their dependence on excitation energy provide further insight into the dissociation dynamics. Recall that the angular distribution is essentially isotropic at 570 nm and that the anisotropy increases with energy. This indicates that the lifetime of the $N_2O_2^-$ complex is several rotational periods at 570 nm,

whereas at 500 nm the lifetime is shorter, presumably more on the order of the characteristic rotational period of the complex. If we assume a rotational temperature of 60 K and treat the complex as an oblate symmetric top with rotational constants determined from the ground state geometry, the average value of rotational quantum number J is 11. The corresponding classical rotational period about the c-axis of the complex (i.e. perpendicular to the plane) is 6.3 ps.

It is of interest to compare the rotational period to the RRKM lifetime of $N_2O_2^-$ in its ground electronic state. The RRKM dissociation rate k(E) is given by

$$k(E)=N^{\ddagger}(E^{\ddagger})/h\rho(E) \qquad (12)$$

Here $N^{\ddagger}(E^{\ddagger})$ is the number of accessible states at the transition state and $\rho(E)$ is the density of $N_2O_2^-$ states at total energy E, both of which can be calculated from the Whitten-Rabinovitch expressions[42] if the vibrational frequencies of the transition state and reactant are known. Because no barrier to dissociation is expected, we choose a loose transition state with four vibrational frequencies set equal to the N_2O frequencies and one low frequency vibration corresponding to the in-plane O^--N_2O bend (the O^--N_2O asymmetric stretch is taken to be the reaction coordinate). The calculated vibrational frequencies at the MP2 level are used to calculate $\rho(E)$ for $N_2O_2^-$. If the O^--N_2O bend frequency in the transition state is assumed to be 50 cm^{-1}, the resulting lifetime of the complex at 570 nm is 0.13 ps, considerably shorter than the rotational period. We note that the pre-exponential A-factor calculated with these transition state and reactant frequencies is only 2×10^{14} sec^{-1} (at T = 300 K). This value is somewhat low for dissociation without a barrier; more typical values are 10^{16} sec^{-1}.[43] Thus our assumed transition state is, if anything, too "tight." It might be more realistic to use a lower bend frequency or an internal rotor to

model the relative motion of the O^- and N_2O fragments at the transition state, in which case the dissociation lifetime would be even shorter.

The large discrepancy between the apparent dissociation lifetime implied by the angular distributions and the RRKM lifetime suggests that dissociation on the ground state potential energy surface is not the rate-limiting step. A possible explanation is that internal conversion from the initially excited $\tilde{B}$ ($^2A_2$) state to the ground state occurs on a time scale of several picoseconds and is therefore rate-limiting. We would then attribute the decrease in lifetime with increasing excitation energy to an increased internal conversion rate arising from more highly excited vibrational levels of the $\tilde{B}$ ($^2A_2$) state.

B. Dissociation Mechanism: $N_2O_2^- + h\nu \rightarrow NO^- + NO$

The description of product channel (2), which yields NO^- + NO, requires a discussion of several of the same issues raised in considering channel (1). However, channel (2) is clearly more complex, as it involves rearrangement and elimination, rather than the simple bond fission process of channel (1). A key issue in the mechanism of channel (2) is the role of the $ONNO^-$ isomer. Posey and Johnson[7] showed that this isomer (which they denote NO^-•NO) is a stable species on the global potential energy surface of the negative ion. There is no experimental evidence for the exact energy of this isomer relative to the $N_2O_2^-$ (C_{2v}) well. The only previously published calculation, by Hiraoka *et al.* at the ROHF/6-31+G level, finds the C_{2v} structure more stable than trans-$ONNO^-$ (C_{2h}) by 4.4 kcal/mol.[8] Our G-1 and G-2 calculations described in Section IIIB show that at this more accurate level of theory the two isomers are essentially isoenergetic, with the trans-$ONNO^-$ isomer more stable by only 0.05 eV. The depth of this well on the potential

energy surface depicted in Fig. 8 is based on the G-2 calculations. Although the photoelectron spectrum offers no clues as to the relative thermodynamics of these two isomers, we can be certain that the anion potential energy surface has a well in an $ONNO^-$ configuration.

Based on the discussion of channel (1), the first mechanism which must be considered, aside from the initial electronic excitation, is that the dynamics for channel (2) occur entirely on the global ground state potential energy surface for the $O^- + N_2O \rightarrow NO^- + NO$ reaction. Specifically, electronic excitation would be followed by internal conversion to the $N_2O_2^-$ ground electronic state, after which isomerization to $ONNO^-$ and bond fission to $NO^- + NO$ occur. The result of this mechanism would be experimental $P(E_T)$ distributions which, like those for channel (1), should reflect a largely statistical decomposition mechanism for a process with no exit channel barrier.

As one test of this mechanism, the experimental $P(E_T)$ distributions can be compared to prior distributions for channel (2), using a version of Eqn. 11 appropriate for two diatomic fragments.[41] The comparison in Figure 5d-f shows that, in contrast to channel (1), the prior and experimental distributions are very different. The experimental distributions show that the fragments have considerably less translational energy, on the average, than predicted by the prior distribution. Moreover, although the 570 nm distribution peaks near $E_T = 0$, the maxima in the other two distributions clearly move towards higher E_T as the excitation energy is raised. More specifically, as was pointed out in section IV-A, the magnitude of this shift is very nearly equal to the additional excitation energy, indicating that this additional energy is efficiently channeled into product translational energy.

Taken together, these observations cast serious doubt on a statistical decomposition mechanism for channel (2). Statistical decay should lead to translational energy distributions that are, at most, weakly dependent on the excitation energy. Moreover, the high level of product vibrational excitation suggests a rather sudden dissociation beginning from a nuclear configuration in which the N-O bond lengths are very different from those in the separated products. Extensive product rotational excitation should also be expected if the nuclear configuration changes rapidly from a C_{2v} structure to the *trans*-$ONNO^-$ intermediate on the way to products, due to the significant torque such motion exerts around the non-central N atom of $N_2O_2^-$. A possible mechanism consistent with these results is one in which the initially prepared $N_2O_2^-$ excited electronic state interacts with another excited state that correlates asymptotically to NO^- + NO products, and that dissociation to channel (2) takes place on this second excited state surface. Even though we postulate a second excited surface that is repulsive, it has been demonstrated[44] that extensive vibrational and rotational exit channel interactions, such as those qualitatively described above, can effectively channel the majority of available energy into internal states of the products.

Unfortunately, we have little information regarding what excited states may play a role in NO^- + NO production. To further explore this product channel it would be useful to form ground state $ONNO^-$ in the ion source, dissociating it at the same three wavelengths used in the current study. However, repeated attempts to create this species from the same NO / Ar mixture used in the photoelectron spectroscopy studies have not yet been successful on the fast beam translational spectrometer.

One issue raised by the mechanism proposed above is why statistical decomposition to channel (2) does not seem to be occurring, given that we believe channel (1) to proceed by dissociation from a ground state $N_2O_2^-$ (C_{2v}) complex. The statistical (i.e., ground state) mechanism for channel (2) requires passage over an isomerization barrier to form the $ONNO^-$ complex, representing a much tighter transition state than that for bond fission to $O^- + N_2O$. Thus, unless the isomerization barrier is considerably lower than D_0 (O^-–N_2O), bond fission will dominate over isomerization once the ground state complex is formed. Unfortunately, we were unable to locate the isomerization barrier between the two potential minima in our calculations, which precludes a more quantitative analysis of the possible ground state reaction path.

C. Comparison with Photoelectron Spectroscopy

The observation of both product channels shown in Figs. 4 and 5 warrants a comparison of our results to the photoelectron spectroscopy of Posey and Johnson.[7] They observe in the 532 nm photoelectron spectrum of $N_2O_2^-$ a single peak due to the detachment of O^- ions. They conclude that this isomer photofragments *exclusively* to O^- + N_2O. At first glance, our observation of NO^- + NO products at 532 nm seems to contradict their interpretation. One must question why signal from NO^- photodetachment is not observed in the photoelectron spectrum if this fragment ion is in fact produced. The explanation can be found in the translational energy release distributions for channel (2) shown in the right column of Figure 5. The small fraction of available energy in translation for this channel requires that the fragments have a high degree of vibrational excitation. Because the detachment threshold of NO^- (194 cm^{-1})[22] is small compared to its

vibrational frequency[45] of 1363 cm^{-1}, NO^- ($\upsilon > 0$) spontaneously autodetaches an electron to give primarily NO ($\upsilon' = \upsilon - 1$) + e^-. The slow electrons produced from autodetachment are below the low-energy cutoff of the time-of-flight photoelectron spectrometer used by Posey and Johnson, and would not have been observed in those experiments. Taking both the photoelectron data and our translational spectroscopy experiments into account, the only mutually consistent explanation is that photodissociation of $N_2O_2^-$ at 532 nm does produce NO^-, but none of this product is in its ground vibrational state, and is therefore not detectable in the photoelectron spectrometer.

To further support our argument, we note that Posey & Johnson *did* observe the unmistakable photoelectron spectrum of NO^- at 532 nm from photodestruction of the $ONNO^-$ isomer. As explained above, NO^- ($\upsilon = 0$) is the only product vibrational state which could produce this signal in their experiments. Therefore, if the $ONNO^-$ isomer were present in our electric discharge ion beam at any appreciable concentration, the $P(E_T)$ distributions of Fig. 5 would show some intensity at the highest possible translational energy (which corresponds to NO^- ($\upsilon = 0$) products). The complete lack of intensity at these translational energies further supports our conclusion that NO^- + NO arises unambiguously from (C_{2v}) $N_2O_2^-$ photofragmentation.

D. Comparison with photodissociation of CO_3^-

The CO_3^- ion, which is isoelectronic with $N_2O_2^-$, also dissociates in the visible region, yielding O^- + CO_2. The photodissociation dynamics of CO_3^- have been extensively examined for twenty years.[2] One might therefore expect to gain further insight into the $N_2O_2^-$ photodissociation dynamics through a comparison with the CO_3^- work. The

study by Snodgrass *et. al.*[3] is particularly relevant for comparison to our data. They measured the $O^- + CO_2$ translational energy distribution and calculated the CO_3^- geometry at the unrestricted Hartree-Fock (UHF) level, finding a $^2A_2'$ ground state with D_{3h} symmetry. Their translational energy distributions, like ours, peak near $E_T = 0$ and decrease steadily towards higher E_T, indicating a high level of CO_2 internal excitation. However, they find that the average translational energy release *decreases* as the photon energy is increased from 514 to 476 nm, finally increasing again at 458 nm, which differs from our results and the predictions of statistical models. Based on these results, they concluded that this dissociation occurs on an excited state surface, rather than by statistical decomposition of the ground state as we have proposed for $N_2O_2^- \rightarrow O^- + N_2O$.

A closer look at both ions indicates that there are important differences in the electronic states involved in the two transitions. The visible excitation in CO_3^- is assigned by Snodgrass *et al.* to the $^2E' \leftarrow {}^2A_2'$ transition. In C_{2v} symmetry, the $^2A_2'$ state correlates to $^2B_2$, implying that the ground states for CO_3^- and $N_2O_2^-$ have similar electronic structure. However, the $^2E'$ excited state of CO_3^- correlates to a $^2A_1$ and $^2B_2$ state, neither of which corresponds to the $^2A_2$ excited state that we believe provides the oscillator strength in $N_2O_2^-$ dissociation. Instead, a $^2A_2$ state correlates to $^2E''$ in D_{3h} symmetry, although the $^2E'' \leftarrow {}^2A_2'$ transition is not optically allowed. The excited states formed by electronic excitation in the two experiments are therefore quite different, which could certainly account for the different dissociation mechanisms proposed for the two anions. This comparison shows that while rough predictions about a molecule can be made based on properties of isoelectronic species, the detailed dynamics of each system can be completely dissimilar.

VI. Conclusions

The photodissociation of $N_2O_2^-$ (C_{2v}) has been investigated using fast beam translational spectroscopy. Two product channels are observed: O^- + N_2O and NO^- + NO. *Ab initio* calculations have been carried out for the lowest state of each symmetry in this ion. Based on the agreement between experiment and theory, the transition in the total dissociation cross section spectrum is assigned to $\tilde{B}$ ($^2A_2$) ← $\tilde{X}$ ($^2B_2$).

Product state distributions for both channels show that the majority of available energy goes into internal energy of the fragments. The $P(E_T)$ distributions for channel (1) demonstrate resolved vibrational structure of the N_2O fragment. From these distributions the following thermodynamic data has been derived: D_0 (O^-—N_2O) = 1.40 ± 0.03 eV, $D_0(N_2O_2^- \rightarrow NO^- + NO)$= 1.25 ± 0.04 eV, and $\Delta H^\circ_{f,0}(N_2O_2^-)$ = 0.58 ± 0.04 eV. The calculated values from G-1 and G-2 theory are in good agreement with the experimental values, which lends credibility to the value calculated by the same methods (but not experimentally known) for the isomerization reaction ΔH_0 {$N_2O_2^-$ (C_{2v}) → trans-$ONNO^-$ (C_{2h})} = -0.05 eV. Within the accuracy of the calculations, these two isomers are thermoneutral.

The formation of O^- + N_2O is most consistent with a sequential mechanism of rate-limiting internal conversion to the ground state followed by fast dissociation without a barrier. At all three excitation energies used in this study, the product translational energy distributions are in overall agreement with the calculated prior distributions. However, at the lowest excitation energy, 570 nm, vibrational structure in the experimental distribution

is resolved to the point where it is clear that the N_2O rotational distributions are colder than predicted by the prior distribution. The N_2O rotational distribution is apparently constrained by the dissociation dynamics on the ground state surface rather than by energy and angular momentum conservation. Nonetheless, because the product translational energy distributions are reasonably well predicted by a simple statistical model, we believe this channel is due to dissociation on the ground state rather than on an excited state surface.

In contrast, the translational energy distributions for the NO^- + NO channel are very different from the corresponding prior distributions. Their dependence on excitation energy suggests that this channel results from dissociation on an excited state potential energy surface that interacts with the initially excited $N_2O_2^-$ $\tilde{B}$ ($^2A_2$) state. The translational energy distributions indicate that their is significant internal excitation of the fragments. The apparent absence of this channel in the earlier study by Posey and Johnson indicates that all of the NO^- is vibrationally excited.

Acknowledgments

This research is supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences Division of the US Department of Energy under Contract No. DE-AC03-76SF00098. The authors wish to acknowledge several helpful discussions with Dr. Don Arnold. In addition we thank Cangshan Xu and Dr. Gordon Burton for their assistance in acquiring the photoelectron spectra in this study.

References

[1] L. C. Lee, G. P. Smith, J. T. Moseley, P. C. Cosby, and J. A. Guest, J. Chem. Phys. **70**, 3237 (1979).

[2] J. T. Moseley, P. C. Cosby, R. A. Bennett, and J. R. Peterson, J. Chem. Phys. **62**, 4826 (1975); J. F. Hiller and M. L. Vestal, J. Chem. Phys. **72**, 4713 (1980); D. E. Hunton, M. Hofmann, T. G. Lindeman, and A. W. Castleman, Jr., J. Chem. Phys. **82**, 134 (1985).

[3] J. T. Snodgrass, C. M. Roehl, P. A. M. van Koppen, W. A. Palke, and M. T. Bowers, J. Chem. Phys. **92**, 5935 (1990).

[4] P. C. Cosby, R. A. Bennett, J. R. Peterson, and J. T. Moseley, J. Chem. Phys. **63**, 1612 (1975); S. E. Novick, P. C. Engelking, P. L. Jones, J. H. Futrell, and W. C. Lineberger, J. Chem. Phys. **70**, 2652 (1979); J. F. Hiller and M. L. Vestal, J. Chem. Phys. **74**, 6096 (1981).

[5] C.-C. Han and M. A. Johnson, Chem. Phys. Lett. **189**, 460 (1992).

[6] C. R. Sherwood, M. C. Garner, K. A. Hanold, K. M. Strong, and R. E. Continetti, J. Chem. Phys. **102**, 6949 (1995).

[7] L. A. Posey and M. A. Johnson, J. Chem. Phys. **88**, 5383 (1988).

[8] K. Hiraoka, S. Fujimaki, K. Aruga, and S. Yamabe, J. Phys. Chem. **98**, 8295 (1994).

[9] D. E. Milligan and M. E. Jacox, J. Chem. Phys. **55**, 3404 (1971).

[10] M. E. Jacox, J. Chem. Phys. **93**, 7622 (1990).

[11] J. Hacaloglu, S. Suzer, and L. Andrews, J. Phys. Chem. **94**, 1759 (1990).

[12] M. E. Jacox, J. Chem. Phys. **93**, 7609 (1990).

[13] R. A. Moris, A. A. Viggiano, and J. F. Paulson, J. Chem. Phys. **92**, 3448 (1990).

[14] S. E. Barlow and V. M. Bierbaum, J. Chem. Phys. **92**, 3442 (1990).

[15] M. W. Chase, Jr., C. A. Davies, J. R. Downey, Jr., D. J. Frurip, R. A. McDonald, and A. N. Syverud, JANAF Thermochemical Tables, 3rd ed., J. Chem. Phys. Ref. Data **14**, (1985), Suppl. No. 1.

[16] D. W. Arnold and D. M. Neumark, J. Chem. Phys.**102**, 7035 (1995).

[17] R. E. Continetti, D. R. Cyr, R. B. Metz, and D. M. Neumark, Chem. Phys. Lett. **182**, 406 (1991).

[18] R. E. Continetti, D. R. Cyr, D. L. Osborn, D. J. Leahy, and D. M. Neumark, J. Chem. Phys. **99**, 2616 (1993).

[19] D. L. Osborn, D. J. Leahy, E. M. Ross, and D. M. Neumark, Chem. Phys. Lett. **235**, 484 (1995).

[20] J. M. B. Bakker, J. Phys. E **6**, 785 (1973); J. Phys. E **7**, 364 (1974).

[21] D. J. Leahy, D. L. Osborn, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **103**, 2495 (1995).

[22] M. W. Siegel, R. J. Celotta, J. L. Hall, J. Levine, and R. A. Bennett, Phys. Rev. A **6**, 607 (1972).

[23] M. A. Johnson, M. L. Alexander, and W. C. Lineberger, Chem. Phys. Lett. **112**, 285 (1984).

[24] L. G. Christophorou, D. L. McCorkle, and A. A. Christodoulides, in *Electron-Molecule Interactions and Their Applications, Vol. 1*, edited by L. G. Chrisotophorou (Academic Press, Orlando, 1984) pp. 477-617, and references therein.

[25] A. Thoma, B. E. Wurfel, R. Schlachta, G. M. Lask, and V. E. Bondybey, J. Phys. Chem. **96**, 7231 (1992).

[26] G. Gantefoer, H. R. Siekmann, H. O. Lutz, and K. H. Meiwes-Broer, Chem. Phys. Lett. **165**, 293 (1990).

[27] Y. Ohshima and Y. Endo, J. Mol. Spec. **153**, 627 (1992).

[28] R. Prosch and T. Trickl, Rev. Sci. Instrum. **60**, 713 (1989).

[29] R. E. Continetti, D. R. Cyr, and D. M. Neumark, Rev. Sci. Instrum. **63**, 1840 (1992). Note that only one MOSFET stage is needed for this application.

[30] S. C. Brown, *Introduction to Electrical Discharges in Gases,* (Wiley, New York 1966).

[31] D. J. Leahy, D. L. Osborn, D. R. Cyr, and D. M. Neumark, J. Chem. Phys. **103**, 2495 (1995).

[32] R. N. Zare, Mol. Photochem. **4**, 1 (1972).

[33] D. L. Osborn, D. J. Leahy, and D. M. Neumark, unpublished work.

[34] GAUSSIAN 92, Revision C, M. J. Frisch, G. W. Trucks, M. Head-Gordon, P. M. W. Gill, M. W. Wong, J. B. Foresman, B. G. Johnson, H. B. Schlegel, M. A. Robb, E. S. Replogle, R. Gompets, J. L. Andres, K. Raghavachari, J. S. Binkley, C. Gonzalez, R. L. Martin, D. J. Fox, D. J. Defrees, J. Baker, J. J. P. Stewart, and J. A. Pople, Gaussian, Inc., Pittsburgh, Pennsylvania, 1992.

[35] The lowest state of C_s symmetry ($^2A'$) converges to the same equilibrium geometry as the ground state ($^2B_2$), and is therefore not listed as a distinct species.

[36] All experimental bond lengths for comparison with calculation are taken from: J. H. Callomon. E. Hirota, T. Iijima, K. Kuchitsu, and W. J. Lafferty, *Landolt Bornstein New Series, Vol. II/15, Structure Data of Free Polyatomic Molecules*, edited by K.-H. Hellwege and A. M. Hellwege (Springer-Verlag, Berlin, 1987).

[37] G. Herzberg, *Molecular Spectra and Molecular Structure, Vol. III. Electronic Spectra and Electronic Structure of Polyatomic Molecules* (van Nostrand, New York, 1966), p. 596.

[38] J. A. Pople, M. Head-Gordon, D. J. Fox, K. Raghavachari, and L. A. Curtiss, J. Chem. Phys. **90**, 5622 (1989).

[39] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople, J. Chem. Phys. **94**, 7221 (1991).

[40] 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**, Supplement No. 1 (1988).

[41] R. D. Levine, in *Theory of Chemical Reaction Dynamics, Vol. IV*, edited by M. Baer (CRC Press, Boca Raton, 1985) pp. 1-64.

[42] G. Z. Whitten and B. S. Rabintovitch, J. Chem. Phys. **41**, 1883 (1964).

[43] R. G. Gilbert and S. C. Smith, *Theory of Unimolecular and Recombination Reactions*, (Blackwell, Oxford, 1990), Chap. 2.

[44] R. Schinke, *Photodissociation Dynamics* (University Press, Cambridge, 1993) pp. 128-132.

[45] K. P. Huber and G. Herzberg, *Molecular Spectra and Molecular Structure IV, Constants of Diatomic Molecules* (van Nostrand, New York, 1966).

Appendix A

CMLWSA:[a] Raw data analysis program for the FRBM

I. Introduction

Data sets acquired with the time- and position-sensitive detector are stored in the most basic form possible. For each coincident event, seven scalars are recorded: six charge fraction values representing W1, S1, Z1, W2, S2, Z2, and the Δt value derived from the time-to-amplitude converter (TAC). For maximum flexibility, these rather large (~ 1-2 MB) raw data files are saved for each data set and converted into the more useful form of translational energy distributions $P(E_T)$ and angular distributions $P(\theta)$ by the code CMLWSA. The practice of saving the raw data, rather than, for instance, a cartesian x,y,z file of fragment positions, allows the most flexibility in off-line analysis and the greatest confidence that no information has been discarded which might later be of use. In addition to the two important distributions mentioned above, CMLWSA produces a host of other distributions which are useful in analyzing time- and position-sensitive data. Diagnostic distributions produced by CMLWSA are:

(1) pulse height distributions: the distribution of ADC values for each of the seven raw data categories. Ideally these are narrow and peak at about half the full scale on the ADC. This data can be useful in identifying and eliminating double events.

(2) detector image: a scatter plot of the first n events showing the x,y position (i.e., in the plane of the detector) of the coincident fragment pairs.

[a] CMLWSA stands for Center of Mass to Lab conversion, Wedge and Strip Anode.

(3) TAC spectrum: the output of the TAC in ADC units

(4) centroid distributions: the distributions of the centroid for each coincident event in a variety of 2-D and 1-D projections.

(5) mass distributions: the fragment mass spectrum as determined by the parent mass M and each individual recoil distance r_1, r_2, from the center of the radical beam.

(6) one dimensional time-of-flight distributions for each particle.

(7) radial distributions for each particle.

(8) azimuthal angular distributions $P(\varphi)$.

Some of these distributions are produced within CMLWSA, and others must be viewed using the code PLOT which reads the output file `fort.8` from CMLWSA.

CMLWSA provides many methods to discriminate events according to their spatial, temporal, and pulse height distributions. This capability is crucial to exclude portions of each data set which are plagued by systematic errors of the detection scheme. Visualization of the distributions is provided by the NCAR (National Center for Atmospheric Research) graphics package which is FORTRAN callable. Therefore the code must be compiled in a FORTRAN environment which supports NCAR graphics.

The core of CMLWSA was written by Dr. Robert Continetti. The code has been extensively modified by Dr. Douglas Cyr, Dr. David Leahy, and the author. CMLWSA produces distributions which are uncorrected for the finite acceptance of the detector. The code designed to make this correction is described in Appendix B.

II. Routines comprising the CMLWSA program

CMLWSA is compiled from the following routines and subroutines. The most important routines are listed in all capitals:

CMLWSA.f: Main routine which calls all other routines. Creates the final output files which are used in later analysis steps.

CENTCAL.f: Determines the average center of the image, applies centroid discrimination, and plots the distribution of the centroids.

discr.f: Performs cartesian, radial, and azimuthal discrimination.

hist.f: Sets of the results histograms.

phaplt.f: Creates pulse heights distributions of the raw ADC data.

plotcentroid.f: Creates various 2-D and 1-D centroid distribution plots.

plotcoin.f: Plots the true coincident events image.

plotmass.f: Plots the fragment mass spectrum created in CENTCAL.f.

PSDCLCE.f: Converts x,y,z cartesian coordinates of each coincident event into the center-of-mass quantities R, θ, φ.

rawplot.f: Plots the raw image before discrimination.

savdfplt.f: Bins the 2-D projection on the detector face of the data for use by post-CMLWSA plotting routines.

srvrtn.f: Contains many service routines for CMLWSA.

tactime.f: Plots the TAC spectrum of either accepted or discriminated data.

WXZTOXY.f: Converts charge fraction and Δt values into x,y,z cartesian coordinates.

CMLWSA can be compiled on a UNIX platform by executing the `make` command:

```
make -f makecmlwsa
```

where the makefile is:

```
FFLAGS= -cg89 -dalign -O3 -Bstatic -C
cmlwsa: cmlwsa.o centcal.o discr.o hist.o plotcentroid.o plotcoin.o
plotmass.o phaplt.o ps
dclce.o rawplot.o savdfplt.o srvrtn.o tactime.o wsztoxy.o
        ncargf77 -cg89 -dalign -O3 -Bstatic -C cmlwsa.o centcal.o
discr.o hist.o plotcentr
oid.o plotcoin.o plotmass.o phaplt.o psdclce.o rawplot.o savdfplt.o
srvrtn.o tactime.o wsz
toxy.o -o cmlwsa
```

III. Usage

A. Input and Output files

Files required as input to CMLWSA are:

adc_cal.dat: Contains calibration data for the ADCs such that each channel produces identical outputs for an identical input.

namelist.dat: Contains all the parameters which are commonly changed by the user. Must be modified for each data set.

param.dat: Contains the factors discussed in Chapter 2 and Appendix D.

phd.dat: Contains upper and lower limits on the values allowed by each of the seven ADC channels and the sum of the upper and lower WSA.

raw_data.dat: Contains the seven column files of ADC scalars. The ordering of the columns is W1, S1, Z1, TAC, W2, S2, Z2.

tcal.dat: Contains calibration information which converts the observed Δt values, (which are modified by the delay cable which ensures that start always comes before stop) into the true Δt values.

title.dat: Title card for each experiment, a string of up to 40 characters.

Files produce as output to CMLWSA are:

fort.8: File of 1-D distribution which is read by the program PLOT to create NCAR graphics plots of these distributions.

fort.9: File containing the 2-D $P_{raw}(E_T,\theta)$ distribution. This is the only output of CMLWSA which is used for input to TDAF and the rest of the data analysis pipeline.

B. General usage information

CMLWSA interacts with the user via a series of interactive questions which are relatively user-friendly. Responses to most questions are either *yes* or *no*, specified by either 0 or 1. Some questions have multiple answers which are listed in the question. Some portions of the data analysis are contained within others in a branched tree structure, i.e., answering *no* to certain questions skips entire sections of the analysis.

The most important usage information is the interpretation of the parameters in the `namelist.dat` file. Therefore, a sample file is given below with annotations:

`$inp`		begin parameter input
`blocky`	`= 4e0,`	half-width of blocking strip (cm)
`ch1lm`	`= 13e0,`	channel 1 light mass (amu)
`ebeam`	`= 8125e0,`	ion beam energy (eV)
`emax`	`= 3.0,`	maximum value of E_T
`emin`	`= 0.0,`	minimum value of E_T
`debug4`	`= .false.,`	toggles on (true) or off debugging output
`dpsd`	`= 995.,`	distance from dissociation laser to detector (mm)
`iforce`	`= 1,`	force fragment masses to values given by `ch`*n*`lm`
`int_chan`	`= 1,`	analyze this product channel when `nchan` > 1
`ipolang`	`= 0,`	laser polarization angle: $\vec{E}$ parallel to radical beam axis ≡0; $\vec{E}$ perpendicular to radical beam axis ≡ 90
`iusephd`	`= 0,`	if 0, use pulse height discrimination limits in `phd.dat`; otherwise ignore this file
`iuseraw`	`= 0,`	if 0, obtain data from `raw_data.dat`; otherwise obtain data from `raw_disc.dat` (latter option seldom used)
`nbine`	`= 100,`	number of E_T bins
`nbinpha`	`= 256,`	number of pulse height analysis bins
`nbinrad`	`= 100,`	number of radial bins for centroid plots
`nchan`	`= 1,`	number of chemical product channels (up to 4)
`npp`	`= 2000,`	number of events used in image plots
`nth`	`= 180,`	number of θ (polar angle) bins
`rbeammax`	`= 1.4,`	maximum allowed centroid radius
`radical_mass`	`= 41.0,`	parent radical mass
`rfov`	`= 20.0,`	radius of circle drawn in image plots
`sxmin`	`= -20.3,`	minimum cartesian x value allowed
`sxmax`	`= 20.3,`	maximum cartesian x value allowed
`symin`	`= -20.3,`	minimum cartesian y value allowed
`symax`	`= 20.3,`	maximum cartesian y value allowed
`tcomin`	`= -250.,`	minimum coincidence time τ ($\equiv \Delta t$) allowed
`tcomax`	`= 250.,`	maximum coincidence time τ ($\equiv \Delta t$) allowed
`title`	`= 'HCCO 8 kV d01'`	title string (should be identical to `title.dat`)
`tmax`	`= 5.3,`	maximum particle time-of-flight from laser to detector
`tmin`	`= 4.75,`	minimum particle time-of-flight from laser to detector
`x0`	`= -0.35,`	center of radical beam along the x-axis relative to detector center
`y0`	`= -0.2,`	center of radical beam along the y-axis relative to detector center
`$end`		end parameter input

A few parameters deserve special mention:

(1) `ebeam`: when the compressor is in use, this mean value for the ion beam energy is typically set at the beam acceleration voltage plus half the compressor voltage.

(2) `iforce`: when this parameter is set to 1, the masses of the two fragments are forced to take on the values given in the parameter(s) `ch`*n*`lm`. When this parameter is set to 0, CMLWSA does not bias the fragment masses towards any particular value, and instead determines the mass ratio for each coincidence event. When a radical is first investigated, the latter choice should be made so that all fragment mass channels are given equal footing. Once the *n* product channels are identified in this manner, set `iforce` = 1, enter the values of the light masses in `ch`*n*`lm`, and analyze each product channel independently by choosing each value of `int_chan` in turn.

(3) `nth`: this parameter should always be set to 180 for compatibility with the TDAF and MAVABFIT codes in which it is not variable.

(4) `rbeammax`: this parameter should be tightened when fragment channels with highly disparate mass ratios $\left(\frac{m_1}{m_2} < \frac{1}{3}\right)$ are examined to prevent inclusion of false coincidences in the final output.

IV. FORTRAN code of CMLWSA

Routines are listed in alphabetical order. In addition, the code for PLOT, which is not part of CMLWSA, but is used to visualize many of the distributions produced by CMLWSA, is given for completeness at the end of this section.

CENTCAL.f

```
      Subroutine centcal
c
c          UNIX version 911113 REC
c               latest version 921207 DRC
c
c  This routine determines the average center of the image and plots
c  the distribution of the centroids
c
c          Requires: a system with known mass fragment channels
c                  a first order guess for x0,y0
c
c
      implicit real*4 (a-h,o-z)
      logical debug4
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n4 = 200    )
      parameter (n7 = 400)
c
      common / beam / ebeam,vbeam,dpsd,rbeammax,rbeammin
c
      common / centdata / cent_x(n1),cent_y(n1)
c
      common / debug / debug4
c
         common / discinf /
ispatdis,isdis,ihlf,yll,yul,isdisp,phi1,phi2,
     >                    iraddis,idisan,rlld,ruld,ttl,ttu,
     >                 nphddis,ncartdis,npoldis,nraddis,ncentdis
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                    dsy,nx,ny
c
      common / flags / iforce,iabs
c
      common / fov / xfov(n4),yfov(n4),nfov
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                 ivaltmp(n1)
cc
      common / local /nok,x_value,y_value,x_centavg(3),y_centavg(3),
     >               dist_min
cc The following 6 arrays are the parallel arrays, which are passed
cc only to centnotok.  They never go to plotting routines under
cc these names
      common / local2 /
     >          iraddisttmp(n4),ixbindisttmp(n4),iybindisttmp(n4),
     >          pimtutmp(n2)
c
      common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / psdhist / emin,emax,dse,
     >                    thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                    tmin,tmax,dwell,tcomin,tcomax,dwellbc,
     >                    nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                    ypmin,rpmin,dyp,drp
c
```

```
      common / radinf / radmin,radmax,dsrad,nbinrad,iraddist(n4),
        @               ixbindist(n4),iybindist(n4)
c
      common / results / z_corr(n1),imtu(n1),upper_mass(n1),energy(n1)
c
      common / signal / ecm(n2),th(n2),u(n2),ephetc(n2,2*n2),
        >       ethetc(n2,n2),emcc(n2),pthcmc(n2),pphcmc(2*n2),pmcc(n2),
        >       ptltlab(n2),pthvlab(n2),ptcolab(n2),ptecbin(n2),tlab(n2),
        >       ph(2*n2),tcolab(n2),ptecabin(n2),pimtu(n2),pylt(n7),
        >       pyhv(n7),prlt(n7),prhv(n7)
c
      common / tacinf / tslope,tint,tdelay,
        >                   ttim(n1),tactrue(n3),tacval(n3),ntac
c
      dimension chan_cent(4,2),x_cent(4,2),y_cent(4,2)
      dimension ivallorig(n1)
      integer selordisc,selpart
      real*4 lowy,upy
c
      data rad,pi,degree / 0.01745329, 3.14159625, 57.29577951 /
      data ch1,ch2,ch3,ch4,light,heavy / 1, 2, 3, 4, 1, 2 /
c
c  initialize values for the determination of the radial distribution
c  of centroids
c
      radmin = 0.0
      radmax = rbeammax
      dsrad = (radmax - radmin) / nbinrad
c cent_x and cent_y arrays hold centroid coords for passing to
centroid
c  plotting routines
      do i = 1,ico
        cent_x(i) = 0.0
        cent_y(i) = 0.0
      enddo
      ifind = 0
c
c
*********************************************************************
c The following iforce conditional encompasses MOST of this
subroutine
c
        if(iforce.eq.1) then
         write(*,*)
         write(*,*)'Enter 0 to find center of image ',
        >          '(1 if center is known).'
         read(*,*)ifind
         write(*,*)'Enter 0 for mass channel selected centroids (1)'
      read(*,*)iselect
      if(iselect.eq.0) then
       write(*,*)'Enter 0 to select int_chan, 1 to discriminate it'
       read(*,*) selordisc
      endif
      write(6,*)'Enter 0 to select data from parts of centroid
        @ distribution (1)'
      read(5,*)selpart
      if(selpart.eq.0) then
       write(6,*)'Enter lower and upper y values to accept data'
       read(5,*)lowy,upy
      endif       !selpart.eq.0
         do ichan = 1,nchan
          write(*,*)
          write(*,100)ichan,fragmass(ichan,light),fragmass(ichan,heavy)
 100      format('Assume channel ',i1, ' has fragments with ',
```

```
     >            'masses ',f7.4, ' amu and ', f7.4, ' amu')
      enddo
      write(*,*)
      write(*,150)int_chan
 150  format('Calculations will be performed using ',
     >        'channel ',i1,' data.')
      if(iabs.eq.1) then
       write(*,*)
       write(*,*)'Absolute forcing of channel ',int_chan,' active!'
      endif
      write(*,*)
      write(*,*)'Location of the center: ',
     >          'x0 = ', x0,' mm'
      write(*,*)'                        ',
     >          'y0 = ', y0,' mm'
c
   do i = 1, ico ! the ivallorig array allows the ival1 array to be
reset
     ivallorig(i) = ival1(i)
   end do
   icforig = icf
c
c       now begin finding the center of the data wrt the detector
c  look for center in eight iterations, starting at x0,y0.
c       Test each point for apparent mass
c       combination according to current x0,y0
c *****************
c  j is the iteration index; this loop encompasses a great deal of
code
c *****************
      do j = 1,8
       if(ifind.eq.1)  j = 8     !force the center to lie at namelist
values
c
       write(*,*)
       write(*,*)'Maximum centroid acceptance radius: ',rbeammax,'
mm'
       if(rbeammin.gt.0.0) then
        write(*,*) 'Minimum centroid acceptance radius: ',rbeammin,'
mm'
       endif
       write(*,*)
       write(*,*)'Calculating ... '
c
c initialize arrays before each centroid optimization iteration
(indexed by j)
c
        do i = 1,3
         x_centavg(i) = 0.0
         y_centavg(i) = 0.0
        enddo
        do i = 1,nbinm
         pimtu(i) = 0       ! imtu (integer upper mass) binning array
        enddo
        do i = 1,nbinrad
         iraddist(i) = 0
        enddo
   nok = 0            ! number of ok centers
   nin = 0      ! number of centroids inside rbeammin
   nout = 0     ! number of centroids outside rbeammax
   nundblock = 0      ! number of centroids under blocker
   noutyrange = 0     ! # of centroids outside y range (see selpart)
c
c ****** Master index i loops over coincident events ******
```

```
c
c Calculate where the centroids lie for each coinc event. Check each
given
c mass channel for best centroid fit before settling on point
c
   do i = 1,ico
    ival1(i) = ival1orig(i) ! restore ival1 array (superfluous for
j=1)
    ivaltmp(i) = 0          !only want ivaltmp from j = 8 loop
    if(ival1(i).eq.0) goto 2000  !skip all this for a bad event
    if(z_corr(i).lt.0.0) then
          first_hit = -1    ! fragment on bottom half hit first
    else
          first_hit = 1           ! fragment on top half hit first
         endif
c
c Master discrimination conditional: checks ttim and yu,yl for
validity
c  the endif for this if is quite a ways downstream
c
         if( (yu(i).ge.blocky).and.(yl(i).le.-blocky) ) then
c
c point OK! calculate distances from assumed centroid for each mass
channel
c  and for light particle hitting top or bottom
c x_cent and y_cent are only intermediate arrays; the centroid will
be stored
c  for posterity in the cent_x(i),cent_y(i) array a bit later
c
c z_corr(i) is in mm; dpsd in mm; epsilon is unitless
c epsilon is greater than zero when top particle arrives at detector
first
c
     epsilon = z_corr(i)/dpsd
c    epsilon = 0.0          !just test without asymmetry corr.
c    write(6,*) i,vbeam,ttim(i),epsilon
          do ichan = 1,nchan
      frac_light = fragmass(ichan,light)/radical_mass ! light=1;
      frac_heavy = fragmass(ichan,heavy)/radical_mass ! heavy=2
c assume light mass hit upper half of detector
           x_cent(ichan,light) =
xu(i)*frac_light*(1+frac_heavy*epsilon)
     >         + xl(i) * frac_heavy * (1 - frac_light*epsilon)
           y_cent(ichan,light) =
yu(i)*frac_light*(1+frac_heavy*epsilon)
     >         + yl(i) * frac_heavy * (1 - frac_light*epsilon)
           dist_to_cent = sqrt((x_cent(ichan,light) - x0)**2
     >                       + (y_cent(ichan,light) - y0)**2)
           chan_cent(ichan,light) = dist_to_cent
c now assume heavy mass hit upper half of detector
           x_cent(ichan,heavy) =
xu(i)*frac_heavy*(1+frac_light*epsilon)
     >         + xl(i) * frac_light * (1 - frac_heavy*epsilon)
           y_cent(ichan,heavy) =
yu(i)*frac_heavy*(1+frac_light*epsilon)
     >         + yl(i) * frac_light * (1 - frac_heavy*epsilon)
           dist_to_cent = sqrt((x_cent(ichan,heavy) - x0)**2
     >                       + (y_cent(ichan,heavy) - y0)**2)
           chan_cent(ichan,heavy) = dist_to_cent
          enddo
c
c iabs conditional: iabs=1 suppresses the loop over mass channels,
c        constraining the mass analysis to int_chan
c Here we determine the best mass channel, centroid coordinates
```

```
c
         dist_min = 25.0 ! dist_min will hold the optimal centroid
distance
          if(iabs.eq.0) then
      do ichan = 1,nchan
            do mass = light,heavy ! light=1; heavy=2
             if(chan_cent(ichan,mass).lt.dist_min) then
              dist_min = chan_cent(ichan,mass)
              x_value = x_cent(ichan,mass) ! x_value and y_value are
only
              y_value = y_cent(ichan,mass) !   temporary storage
variables
              imtu(i) = fragmass(ichan,mass) * first_hit
             endif ! here imtu(i) is assigned to best matched channel
            enddo
           enddo
          else ! iabs = 1
c note:  iabs = 1 always assigns imtu(i) to int_chan even if int_chan
was
c  not the best mass channel
           do mass = light,heavy
            if(chan_cent(int_chan,mass).lt.dist_min) then
             dist_min = chan_cent(int_chan,mass)
             x_value = x_cent(int_chan,mass)
             y_value = y_cent(int_chan,mass)
             imtu(i) = fragmass(int_chan,mass) * first_hit
            endif
           enddo
          endif ! end of iabs conditional
c
c  test to see if centroid is ok
c
          if((dist_min.le.rbeammax).and.(dist_min.gt.rbeammin)) then
      if(iselect.eq.0) then !only look at selected centroids
       if(selordisc.eq.0) then !only look at channel of interest
             if((abs(imtu(i)).eq.fragmass(int_chan,light)).or.
     >         (abs(imtu(i)).eq.fragmass(int_chan,heavy))) then

         call centok(i)  ! centroid is OK, *and* belongs to int_chan

        else !if((abs(imtu(i)).eq.....or) (i.e. event wasn't
int_chan)
         ivall(i) = 0
         ivaltmp(i) = 1
         call centnotok(i)
         if(dist_min.gt.rbeammax) then
               nout = nout + 1
              else
               nin = nin + 1
              endif
        endif !if((abs(imtu(i)).eq.....or)

      else !if(selordisc.eq.0) i.e. **discriminate** int_chan

             if((abs(imtu(i)).ne.fragmass(int_chan,light)).and.
     >         (abs(imtu(i)).ne.fragmass(int_chan,heavy))) then

           call centok(i)  ! centroid is OK, but is not int_chan

       else !if((abs(imtu(i)).....and.) (i.e. event was int_chan)
        ivall(i) = 0
        ivaltmp(i) = 1
        call centnotok(i)
      if(dist_min.gt.rbeammax) then
```

```
            nout = nout + 1
           else
            nin = nin + 1
           endif
        endif !if((abs(imtu(i))....and.)
       endif !if(selordisc.eq.0)

      else !if(iselect.eq.0) i.e. look at *all* centroids (trad. way)

       if(selpart.eq.0) then !use data from only some y centroids
        if((y_value - y0.ge.lowy).and.(y_value-y0.le.upy)) then
             call centok(i)      ! centroid is OK, doesn't matter
which chan
        else
         ivall(i) = 0
         ivaltmp(i) = 1
         call centnotok(i)
         noutyrange = noutyrange + 1

        endif  !no discrimination of y centroids (normal situation)
        else

         call centok(i)     ! centroid is OK, doesn't matter which
chan

       endif    !selpart.eq.0

      endif !if(iselect.eq.0)

          else ! from if(dist_min.le.rb...)  centroid is not ok
       ivall(i) = 0
       ivaltmp(i) = 1
       call centnotok(i)
       if(dist_min.gt.rbeammax) then
            nout = nout + 1
           else
            nin = nin + 1
           endif
      endif ! end of dist_min validity (wrt to rbeammax,min)
conditional
     else !woops, at least one part. must be under the beam block
      ivall(i) = 0
      ivaltmp(i) = 1
      nundblock = nundblock + 1
      call centnotok(i)
          endif ! end of centroid validity (wrt beam block)
conditional
2000    enddo ! end of coincident events loop (i index)
c
        ndisc = nin + nout + nundblock + noutyrange
   ncentdis = icf - nok    !for use in totals at the end
   if(ndisc.ne.ncentdis) then
    write(6,*)'nin = ',nin,' nout = ',nout,' nundblock = ',nundblock
    write(6,*)'noutyrange = ',noutyrange,' icf = ',icf,'nok = ',nok
    write(6,*)'nin + nout + nundblock + noutyrange = ',ndisc
    write(6,*)'icf - nok = ',ncentdis
    stop 'number of discriminated events does not match'
   endif
        frin = float(nin) / float(icf)
        frout = float(nout) / float(icf)
   frbblk = float(nundblock) / float(icf)
   frydisc = float(noutyrange) / float(icf)
        frok = float(nok) / float(icf)
        call sigmac ( x_centavg , x_centavg(1) , x_centavg(2) )
```

```
        call sigmac ( y_centavg , y_centavg(1) , y_centavg(2) )
        write(*,*)
        write(6,290)rbeammax,rbeammin
        write(6,295)frout,frin,frbblk,frydisc,frok
   write(6,*)'nundblock = ',nundblock, ' nout = ',nout
        write(6,297)ndisc,icf,(frok*100)
   if(j.eq.8) then
    icf = nok
   else
    icf = icforig !reinitialize for another centroid pass
    nundblock = 0
    nin = 0
    nout = 0
   endif
        write(*,*)

write(6,300)x_centavg(1),x_centavg(2),y_centavg(1),y_centavg(2)
 290    format(' Maximum radius of acceptance = ',f6.4,' mm ',/,
     >          ' Minimum radius of acceptance = ',f6.4,' mm ')
 295    format(' Fraction outside maximum centroid radius = ',f7.6,/,
     >          ' Fraction inside minimum centroid radius  = ',f7.6,/,
     >          ' Fraction under the beam block            = ',f7.6,/,
     >          ' Fraction discriminated in y coord.       = ',f7.6,/,
     >          ' Fraction within centroid limits          =
',f7.6,/,)
 297    format(' ',i6,' events discriminated out of ',i6,'
events.',/,
     >          '   ',f6.2,' percent of the data remains')
 300    format(t2,'Using best choice of mass channels ',
     >             'for each data point: ',/,
     >  t15,' x = ',f7.4,'   std. dev. = ',f7.4,/,
     >  t15,' y = ',f7.4,'   std. dev. = ',f7.4)
c
c       since if ifind equals 0 then we want to find the center, take
c  simple average of these two numbers for final value
c
   write(6,*)
   if(ifind.eq.0) then
    x0 = x_centavg(1)
    y0 = y_centavg(1)
    write(6,*)'            x0 = ',x0,' y0 = ',y0
   else
    write(6,*)'However, x0,y0 remain as given in namelist.dat: '
    write(6,*)'            x0 = ',x0,' y0 = ',y0
   endif
   if(j.eq.8) then ! end of centroid iterations
    write(6,*)'Enter 0 for centroid plotting.'
    read(5,*) iplot

    if(iselect.eq.0)then
     if(selordisc.eq.0) then
      write(*,*)'**Centroid plots contain only channel',int_chan
     else
      write(*,*)'**Centroid plots contain everything *except*
     >                 channel',int_chan
     endif
    endif

    if (iplot.eq.0) then
     write(*,*)
     write(*,*)'Enter 0 for centroid scatter plot (1):'
     read(*,*)iplt
          if(iplt.ne.1) then
           call plotcent(0)
```

```
      endif
      write(*,*)
      write(*,*)'Enter 0 for scatter plot of discr. centroids (1):'
      read(*,*)iplt
           if(iplt.ne.1) then
            call plotcent(3)
      endif
      write(*,*)
      write(*,*)'Enter 0 to plot centroid contours (1):'
      read(*,*)iplt
           if(iplt.ne.1) then
            call plotcont
      endif
      write(*,*)
      write(*,*)'Enter 0 to plot centroid surface (1):'
      read(*,*)iplt
           if(iplt.ne.1) then
            call plotsurf
      endif
      write(*,*)
      write(*,*)'Enter 0 to plot radial distribution. (1)'
           read(*,*)iplt
      if(iplt.ne.1) then
       call plotrad
      endif
      write(*,*)
      write(*,*)'Enter 0 to plot radial dist. of disc. centroids(1)'
           read(*,*)iplt
      if(iplt.ne.1) then
       call swaptemp(0)
       call plotrad
       call swaptemp(1)
      endif
     end if ! iplot.eq.0 conditional
     write(*,*)
     write(*,*)'Enter 0 to plot mass distribution. (1)'
     read(*,*)iplt
     if(iplt.ne.1) then
      call plotmass
     endif
     write(*,*)
     write(*,*)'Enter 0 to plot mass dist. of disc. centroids (1)'
     read(*,*)iplt
     if(iplt.ne.1) then
      call swaptemp(0)
      call plotmass
      call swaptemp(1)
     endif
         endif ! if j=8 conditional
        enddo ! *** end of the lengthy j loop (centroid optimization
loop)
c
        write(*,*)
        write(*,320)x0,y0
 320    format(t5,'Nominal Center = ',f10.4,', ',f10.4,' mm')
c
c ******** Here ends the really lengthy iforce=1 block
c
      else ! iforce = 0
c
c       Here iforce = 0;  we do not know the masses,
c       and we must discriminate against false coincidences by
c       checking that the line joining the two points passes
c       through the circle defined by rbeammax
```

```
c       To do this:
c       1) Calculate the angle of the line between the two data points
c       2) Calculate the angle of the line between the lower data
c          point and the center x0,y0
c       3) The difference of these two angles is the angle between
these
c     two lines.  Once this is known, the cosine and sine projections
of
c     the line in 2) give, respectively,
c       4) The best guess at the length of the lower moment arm of the
total
c     relative velocity vector magnitude, from which the masses are
c          found, and
c       5) The shortest distance distance from the center, (x0,y0),
c     to the line in 1), from which the validity of this event is
c     determined, (i.e. centroid disrimination)

c

   do i = 1,ico
    ivallorig(i) = ivall(i) !preserve ivall array for multiple x0,y0
    ivaltmp(i) = 0
   enddo
   icforig = icf
330      nok = 0  ! line 330 is re-entry point for new centroid guess
   do i=1,ico
    if(ivall(i).eq.0) goto 3000  !skip all this for each bad event
    if(z_corr(i).lt.0.0) then
          first_hit = -1   ! fragment on bottom half hit first
    else
          first_hit = 1          ! fragment on top half hit first
         endif
         if(abs(xu(i)-xl(i)).gt.1e-6) then
          slpdata = (yu(i)-yl(i))/(xu(i)-xl(i))
          angdata = atan(slpdata) * degree
          if(angdata.lt.0) then
           angdata = 180 + angdata
          endif
         else
     stop 'Very rare case of xu(i) = xl(i) found.  See centcal'
         endif
         if(abs(x0-xl(i)).gt.1e-6) then
          anglowr = atan((y0 - yl(i))/(x0-xl(i))) * degree
          if(anglowr.lt.0) then
           anglowr = 180 + anglowr
          endif
         else
     stop 'Very rare case of x0 = xl(i) found.  See centcal'
         endif

c     now calculate the quatities of interest for both the linoid
c     method of centroid discrimination, and the best guess at the
c     mass ratio when iforce = 0
    rlower = sqrt( (x0-xl(i))**2 + (y0 - yl(i))**2 )
    rul    = sqrt( (xu(i)-xl(i))**2 + (yu(i)-yl(i))**2 )

c     cos is an even function, so sign of it's argument doesn't matter
    anglecent = anglowr - angdata       !angle betw. 2 almost parallel
lines
    rlbest = rlower*cos(anglecent/degree)    !best guess at lower
moment arm
    rdist  = abs( rlower*sin(anglecent/degree) )! distance from x0,y0
    if(abs(anglecent).gt.89) write(*,*)'Large angle in centcal!!?!'
c     the preceding warning means something is wrong with the angular
```

```
c    determinations in finding anglecent.  These two lines should be
c    nearly parallel, so anglecent is small.  I don't think it will
happen.

     if((rdist.ge.rbeammin).and.(rdist.le.rbeammax)) then
      nok = nok + 1    ! centroid is OK!
      upper_mass(i) = radical_mass * (rlbest/rul)
      imtu(i) = nint(upper_mass(i))*first_hit
      call binmass(gmin,gmax,dsm,nbinm,upper_mass(i),imass,imassinc)
           pimtu(imass) = pimtu(imass) + imassinc
     else
      ivall(i) = 0           !centriod is not OK
      ivaltmp(i) = 1
      temp_mass = radical_mass * (rlbest/rul)
      call binmass(gmin,gmax,dsm,nbinm,temp_mass,imass,imassinc)
           pimtutmp(imass) = pimtutmp(imass) + imassinc
          endif
3000      enddo ! end of i coincidence event loop
c
   ncentdis = icf - nok
   write(6,340)icf-nok
340       format(i6,' events thrown out by centroid discrimination')
   frrej = 1.0 - float(nok)/float(icf)
   write(6,345)frrej
345       format(' Fraction rejected by centroids = ',f10.6)
   icf = nok
   frrej = 1.0 - float(nok)/float(ico)
   write(6,350)frrej
350       format(' Cumulative Fraction rejected = ',f10.6)
   write(*,*)
   write(*,*)'Enter 0 to plot mass distribution. (1)'
   read(*,*)iplt
   if(iplt.ne.1) call plotmass
   write(*,*)
    write(*,*)
    write(*,*)'Enter 0 to plot mass dist. of disc. centroids (1)'
    read(*,*)iplt
    if(iplt.ne.1) then
     call swaptemp(0)
     call plotmass
     call swaptemp(1)
    endif
   write(*,*)'Enter 0 to change x0 and y0 and repeat. (1)'
   read(*,*)ichng
   if(ichng.ne.1) then
    icf = icforig
    do i = 1,nbinm
     pimtu(i) = 0    ! imtu (integer upper mass) binning array
     pimtutmp(i) = 0
    enddo
    do i = 1,ico
     ivall(i) = ivallorig(i) !get a fresh start with the old ivall
array
     ivaltmp(i) = 0
    enddo
    write(*,*)'Enter new x0,y0 : '
         read(*,*) x0,y0
         goto 330
        endif
      endif  ! end of the else (iforce=0) block of the iforce=1
conditional
c
   write(6,*)'Enter 0 for centroid-filtered plotting.'
   read(5,*) iplot
```

```
   if (iplot.eq.0) then
    write(*,*)
    write(*,*)'Enter 0 to plot the pos. of coincident points. (1)'
    read(*,*)iptxyv
    if(iptxyv.eq.0) call rawplot(0)
c
    write(*,*)
    write(*,*)'Enter 0 to plot the pos. of non-coincident pts. (1)'
    read(*,*)iptxyn
    if(iptxyn.eq.0) call rawplot(2)
c
c  plot valid TAC distribution if desired
c
    write(*,*)
    write(*,*)'Enter 0 to plot the valid TAC spectrum. (1)'
    read(*,*)iptv
    if(iptv.ne.1) call tactime(0)
     write(*,*)
     write(*,*)'Enter 0 to plot TAC spectrum of only the ',
     >            'discriminated points. (1)'
     read(*,*)iptv
     if(iptv.eq.0) call tactime(3)
     write(*,*)

   endif ! end of iplot conditional
c
c  used to call discr here.  Now do it in cmlwsa, before centcal
c


c
c
      return
      end
*****************************************************************
******
   Subroutine centok(i)
C    This subroutine just increments some indices and bins things
c    Some of the common blocks have very few variables in this
c    subroutine.

   implicit real*4 (a-h,o-z)
   logical debug4
c
   parameter ( n1 = 100000 )
   parameter ( n2 = 300    )
   parameter ( n3 = 50     )
   parameter ( n4 = 200    )
   parameter (n7 = 400)
cc   uses all variables
   common / local /nok,x_value,y_value,x_centavg(3),y_centavg(3),
     >                dist_min
cc  uses all variables
   common / centdata / cent_x(n1),cent_y(n1)
cc  uses xu,yu,xl,yl
   common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                 ivaltmp(n1)
cc  uses radical_mass
   common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
cc  uses x0,y0
   common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
cc  uses gmin,gmax,dsm,nbinm
   common / psdhist / emin,emax,dse,
     >                    thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                    tmin,tmax,dwell,tcomin,tcomax,dwellbc,
```

```
     >                    nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                    ypmin,rpmin,dyp,drp
cc  uses all variables
   common / radinf / radmin,radmax,dsrad,nbinrad,iraddist(n4),
     @               ixbindist(n4),iybindist(n4)
cc  uses upper_mass
   common / results / z_corr(n1),imtu(n1),upper_mass(n1),energy(n1)
cc  uses pimtu
   common / signal / ecm(n2),th(n2),u(n2),ephetc(n2,2*n2),
     >      ethetc(n2,n2),emcc(n2),pthcmc(n2),pphcmc(2*n2),pmcc(n2),
     >      ptltlab(n2),pthvlab(n2),ptcolab(n2),ptecbin(n2),tlab(n2),
     >      ph(2*n2),tcolab(n2),ptecabin(n2),pimtu(n2),pylt(n7),
     >      pyhv(n7),prlt(n7),prhv(n7)

c*******Here's the real
code*********************************************

           nok = nok + 1   ! centroid is OK, doesn't matter which
chan
           cent_x(i) = x_value
           cent_y(i) = y_value
           call sigmap ( x_value , x_centavg )
           call sigmap ( y_value , y_centavg )
c bin dist_min (centroid distance from beam center) into iraddist
array
           call xbinnr2(radmin,radmax,dsrad,nbinrad,
     >                  dist_min,irad,iradinc)
           iraddist(irad) = iraddist(irad) + iradinc
c*******************
c bin radial distribution by x and y to see elipticity in centroids
      call xbinnr2(-radmax,radmax,2*dsrad,nbinrad,
     @               x_value - x0,ixbin,ixbininc)
      ixbindist(ixbin) = ixbindist(ixbin) + ixbininc
      call xbinnr2(-radmax,radmax,2*dsrad,nbinrad,
     @               y_value - y0,iybin,iybininc)
      iybindist(iybin) = iybindist(iybin) + iybininc
c***************************
           upper_mass(i) = sqrt((xl(i)-x0)**2+(yl(i)-y0)**2)
     >                 * radical_mass /
     >                (sqrt((xl(i)-x0)**2+(yl(i)-y0)**2) +
     >                 sqrt((xu(i)-x0)**2+(yu(i)-y0)**2))
c bin upper_mass(i) into pimtu array
           call binmass(gmin,gmax,dsm,nbinm,
     >               upper_mass(i),imass,imassinc)
           pimtu(imass) = pimtu(imass) + imassinc

   return
   end
*******************************************************************
*****
*******************************************************************
******
   Subroutine centnotok(i)
C    This subroutine does the same thing as the as centok, but for
the
c    events which have more recently been discriminated against.
c    Their ivaltmp(i) = 1.  It involves 4 "parallel" arrays with
c    similar names to the ones in centok.  These are always compiled.
c    When a plot is needed, these arrays are swapped into the places
c    of the true arrays, the plot generated, and then they are
c    swapped back out so as not to corrupt the real arrays


   implicit real*4 (a-h,o-z)
```

```
      logical debug4
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n4 = 200    )
cc  uses x_value,y_value,dist_min
      common / local /nok,x_value,y_value,x_centavg(3),y_centavg(3),
     >                dist_min
cc  uses xu,yu,xl,yl
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                 ivaltmp(n1)
cc  uses radical_mass
      common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
cc  uses x0,y0
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
cc  uses gmin,gmax,dsm,nbinm
      common / psdhist / emin,emax,dse,
     >                   thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                   tmin,tmax,dwell,tcomin,tcomax,dwellbc,
     >                   nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                   ypmin,rpmin,dyp,drp
cc  uses radmin,radmax,dsrad,nbinrad
      common / radinf / radmin,radmax,dsrad,nbinrad,iraddist(n4),
     @                 ixbindist(n4),iybindist(n4)

c  The following 4 arrays are the parallel arrays, which are passed
c  only to centcal.  They never go to plotting routines under
c  these names.
      common / local2 /
     >         iraddisttmp(n4),ixbindisttmp(n4),iybindisttmp(n4),
     >         pimtutmp(n2)
c*******Here's the real
code*********************************************

           cent_xtmp(i) = x_value
           cent_ytmp(i) = y_value
c bin dist_min (centroid distance from beam center) into iraddisttmp
array
           call xbinnr2(radmin,radmax,dsrad,nbinrad,
     >                  dist_min,irad,iradinc)
           iraddisttmp(irad) = iraddisttmp(irad) + iradinc
c*******************
c bin radial distribution by x and y to see elipticity in centroids
      call xbinnr2(-radmax,radmax,2*dsrad,nbinrad,
     @                x_value - x0,ixbin,ixbininc)
      ixbindisttmp(ixbin) = ixbindisttmp(ixbin) + ixbininc
      call xbinnr2(-radmax,radmax,2*dsrad,nbinrad,
     @                y_value - y0,iybin,iybininc)
      iybindisttmp(iybin) = iybindisttmp(iybin) + iybininc
c**************************
           temp_mass = sqrt((xl(i)-x0)**2+(yl(i)-y0)**2)
     >                 * radical_mass /
     >                (sqrt((xl(i)-x0)**2+(yl(i)-y0)**2) +
     >                 sqrt((xu(i)-x0)**2+(yu(i)-y0)**2))
c bin temporary mass (scalar) into pimtutmp array
           call binmass(gmin,gmax,dsm,nbinm,
     >              temp_mass,imass,imassinc)
           pimtutmp(imass) = pimtutmp(imass) + imassinc

      return
      end
c*****************************************************************
******
```

```
c*****************************************************************
******
   subroutine swaptemp(idirect)
c  This subroutine swaps the temporary arrays defined in centnotok
c  into the real arrays (that will be used in the rest of cmlwsa)
c  soley for plotting the data that has just recently been
discriminated

   implicit real*4 (a-h,o-z)
   logical debug4
c
   parameter ( n1 = 100000 )
   parameter ( n2 = 300    )
   parameter ( n3 = 50     )
   parameter ( n4 = 200    )
   parameter (n7 = 400)

c   uses all
   common / local2 /
     >          iraddisttmp(n4),ixbindisttmp(n4),iybindisttmp(n4),
     >          pimtutmp(n2)
c   uses ivall
   common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                ivaltmp(n1)
c   uses nbinrad,iraddist,ixbindist,iybindist
   common / radinf / radmin,radmax,dsrad,nbinrad,iraddist(n4),
     @               ixbindist(n4),iybindist(n4)
cc  uses nbinm
   common / psdhist / emin,emax,dse,
     >                    thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                    tmin,tmax,dwell,tcomin,tcomax,dwellbc,
     >                    nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                    ypmin,rpmin,dyp,drp
c   uses pimtu
   common / signal / ecm(n2),th(n2),u(n2),ephetc(n2,2*n2),
     >      ethetc(n2,n2),emcc(n2),pthcmc(n2),pphcmc(2*n2),pmcc(n2),
     >      ptltlab(n2),pthvlab(n2),ptcolab(n2),ptecbin(n2),tlab(n2),
     >      ph(2*n2),tcolab(n2),ptecabin(n2),pimtu(n2),pylt(n7),
     >      pyhv(n7),prlt(n7),prhv(n7)

   dimension istor_rad(n4),istor_xdis(n4)
   dimension istor_ydis(n4),stor_pmt(n2)

c  idirect = 0 swap parallel arrays into real arrays
c  idirect = 1 swap real arrays back to their permanent spots
c  ARRAY LENGTHS:
c  iraddist, ixbindist, iybindist :    nbinrad
c  pimtu :                             nbinm

   if(idirect.eq.0) then

c  copy the true arrays to temporary ones during plotting
   do i = 1,nbinrad
    istor_rad(i) = iraddist(i)    !save old arrays
    istor_xdis(i) = ixbindist(i)
    istor_ydis(i) = iybindist(i)

    iraddist(i) = iraddisttmp(i) !put bad values in old arrays
    ixbindist(i) = ixbindisttmp(i)
    iybindist(i) = iybindisttmp(i)
   enddo
   do i = 1,nbinm
    stor_pmt(i) = pimtu(i)        !save old arrays
```

```
 pimtu(i) = pimtutmp(i)        !put bad values in old arrays
enddo

else  !in this case, swap the good values back in

do i = 1,nbinrad
 iraddist(i) = istor_rad(i)    !reinstate "good" arrays
 ixbindist(i) = istor_xdis(i)
 iybindist(i) = istor_ydis(i)
enddo
do i = 1,nbinm
 pimtu(i) = stor_pmt(i)        !reinstate "good" arrays
enddo

endif
return
end
```

CMLWSA.f

```
*deck cmlwsa
      Program cmlwsa
c
c                    latest unix version 921122 DRC
c                                        from REC
c
c  This routine reads in raw coincidence data from raw_data.dat, wsa
c       conversion factors and TAC info from param.dat and tcal.dat,
c  experimental constants from namelist.dat (in namelist format)
c  and ADC calibration
c       data from adc_cal.dat.  It first uses wsztoxy to allow pulse
height
c       discrimination and also to convert the raw data to x and y
c       positions on the detector face.  A raw coincidence time can
then
c       be calculated.  At this point the program allows the center
of
c       detector to be found, or the user to assume a center of his
c       choosing via the namelist.  The routine psdclce is then
employed
c       to calculate the many quantities of interest from the raw
c       coincedence data, such as the cm translational energy, cm
angle
c       and mass distributions.
c
      implicit real*4 (a-h,o-z)
      parameter ( n1 = 100000)
      parameter ( n2 = 300)
      parameter ( n3 = 50)
      parameter ( n4 = 200)
   parameter ( n5 = 50 )
      parameter ( n6 = 8192)
   parameter (n7 = 400)
c      real*4 lctws,lctwz,lctsw,lctsz
      integer light,heavy,chanval
      logical debug4
      character*40 title

c  *************VARIABLE DEFINITIONS*************
c  ico:  number of post pulse height discriminated events (wsztoxy)
c  icf:  number of final valid events
c  icfn:  temporary value of icf in discr subroutine.  (icfn.le.icf)
```

```
c  **********END VARIABLE DEFINITIONS************

      common / adcinf / vadc(n3),adcw1(n3),adcs1(n3),adcz1(n3),
     >          adcw2(n3),adcs2(n3),adcz2(n3),
     >          nadc,sw1,w1int,ss1,s1int,sz1,
     >          z1int,sw2,w2int,ss2,s2int,sz2,z2int,
     >          cw1,cs1,cz1,cw2,cs2,cz2
c
c *** Old, Bob-style algorithm factors, saved for posterity
c
c      common / algo / dmcwu,dmcsu,dmczu,dmcw1,dmcs1,dmcz1,
c     >                dacwu,dacsu,dacw1,dacs1,
c     >                uctws,uctwz,uctsw,uctsz,
c     >                lctws,lctwz,lctsw,lctsz
c
c *** New, xyzlsq-style factors        DJL 11/93
c
   common / algo / param(16)     ! wasn't that easy?
c
      common / beam / ebeam,vbeam,dpsd,rbeammax,rbeammin
c
      common / debug / debug4
c
      common / discinf /
ispatdis,isdis,ihlf,yll,yul,isdisp,phi1,phi2,
     >                  iraddis,idisan,rlld,ruld,ttl,ttu,
     >               nphddis,ncartdis,npoldis,nraddis,ncentdis
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                   dsy,nx,ny
c
      common / flags / iforce,iabs
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ival1(n1),
     >              ivaltmp(n1)
c
      common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
c
      common / pha / puwpha(n6),puspha(n6),
     >              puzpha(n6),plwpha(n6),
     >              plspha(n6),plzpha(n6),
     >              tacpha(n6),ptacpha(n6),upha(n6),pupha(n6),
     >
lpha(n6),plpha(n6),nbinpha,phvpha(n6),pltpha(n6),
     >              iusephd
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / psdhist / emin,emax,dse,
     >                   thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                   tmin,tmax,dwell,tcomin,tcomax,dwellbc,
     >                   nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                   ypmin,rpmin,dyp,drp
c
      common / radinf / radmin,radmax,dsrad,nbinrad,iraddist(n4),
     @              ixbindist(n4),iybindist(n4)
c
   common / results / z_corr(n1),imtu(n1),upper_mass(n1),energy(n1)
c
      common / signal / ecm(n2),th(n2),u(n2),ephetc(n2,2*n2),
     >      ethetc(n2,n2),emcc(n2),pthcmc(n2),pphcmc(2*n2),pmcc(n2),
     >      ptltlab(n2),pthvlab(n2),ptcolab(n2),ptecbin(n2),tlab(n2),
     >      ph(2*n2),tcolab(n2),ptecabin(n2),pimtu(n2),pylt(n7),
```

```
     >       pyhv(n7),prlt(n7),prhv(n7)
c
      common / stat / tltavg(3),thvavg(3),tcoavg(3),ecmcavg(3),
     >          ecmcaavg(3),emltavg(3),emhvavg(3)
c
      common / tacinf / tslope,tint,tdelay,
     >                  ttim(n1),tactrue(n5),tacval(n5),ntac
      common / words / title
c
c  input namelist
c
   namelist / inp / blocky,ch1lm,ch2lm,ch3lm,ebeam,emax,emin,debug4,
     >             dpsd,iabs,iforce,int_chan,ipolang,iusephd,iuseraw,
     >             nbine,nbinpha,nbinrad,npp,nproj,nchan,nt,ntbc,
     >             nth,nx,ny,rbeammax,rbeammin,
     >             radical_mass,radmax,rfov,sxmin,sxmax,
     >             symin,symax,tcomin,tcomax,title,tmax,tmin,x0,y0
c
      data rad,pi / 0.01745329 , 3.14159265 /
      data degree / 57.29577951 /
      data ch1,ch2,ch3,ch4,light,heavy / 1, 2, 3, 4, 1, 2 /


c
c  default namelist values
c
c  the half-width of the blocker in the y direction, assumed
c       centered (mm)
      blocky = 4e0
c  the light mass of the first mass channel
      ch1lm = 16e0
c  the heavy mass of the second mass channel
      ch2lm = 30e0
c  the nominal beam energy (eV)
      ebeam    = 6000e0
c  the maximum KER bin (eV)
      emax   = 8e0
c  the minimum KER bin (eV)
      emin   = 0e0
c  prints diagnostic stuff to screen in subroutine psdclce
      debug4 = .true.
c  the distance from the center of the dissociation volume to the
c  face of the c-wsa mcp's (mm)
      dpsd   = 1000.
c  the parent mass (amu)
      radical_mass = 60e0
c       iabs = 1, absolutely force the mass to the channel specified
      iabs = 0
c  iforce = 1 force the mass, = 0 don't
      iforce = 0
c  int_chan is the channel of interest for calculations
      int_chan = 1
c       iusephd = 1 ask before using phd.dat, = 0 don't
      iusephd = 1
c  ipolang = 0 for horizontal polarization, 90 for vertical
polarization
      ipolang = 0
c       iuseraw = 1 take calculated positions and time from
raw_disc.dat
      iuseraw = 0
c  the number of energy bins
      nbine  = 100
c  the number of pulse-height distribution bins
      nbinpha= 2048
```

```
c       the number of radial centroid distribution bins
      nbinrad= 25
c       the number of mass channels
      nchan  = 1
c  the default number of points to plot for the image
      npp    = 2500
c  the number of y and r projection bins
      nproj  = 200
c  the number of flight-time bins
      nt     = 200
c  the number of coincidence-time bins
      ntbc   = 200
c  the number of cm-angle bins
      nth    = 50
c  the number of x position bins
      nx     = 240
c  the number of y position bins
      ny     = 240
c  the maximum radius of centroid acceptance (mm)
      rbeammax = 1.0e0
c  the minimum radius of centroid acceptance (mm)
      rbeammin = 0.0e0
c       the radius within which centroids are to be binned radially
(mm)
      radmax = 5.0
c  the radius of the field-of-view (mm)
      rfov   = 19e0
c  the minimum x position (mm)
      sxmin  = -25.0
c  the maximum x position (mm)
      sxmax  =  25.0
c  the minimum y position (mm)
      symin  = -25.0
c  the maximum y position (mm)
      symax  =  25.0
c  the minimum coincidence time to be binned (ns)
      tcomin = 0e0
c  the maximum coincidence time to be binned (ns)
      tcomax = 500e0
c  title
      title  = 'FRBM TPS Coincidence Data Analysis'
c  the maximum flight-time to be binned (microseconds)
      tmax   = 10e0
c  the minimum flight-time to be binned (microseconds)
      tmin   = 4e0
c  the nominal x-origin of the detector (mm)
      x0     = 0e0
c  the nominal y-origin of the detector (mm)
      y0     = 0e0
c
c  set up histogram array generator
c
      call numgen
c
c         read in raw coincidence data from
raw_data.dat
c          or phd discriminated data from                raw_disc.dat
c         read in wsa conversion factors from
param.dat
c    read in TAC-ADC calibration factors from          tcal.dat
c    read in experimental constants from               namelist.dat
c              (in namelist format)
c    output product energy distribution into           fort.8
c         product angular (theta) distrib.
```

```
c          product angular (phi) distrib.
c          product energy-angle distribu.
c            product mass spectrum
c          light particle TOF
c          heavy particle TOF
c          coincidence distrib
c
c    output product KER - theta spectra into            fort.9
c
c    read in ADC calibration data from                 adc_cal.dat
c          read in phd data from                           phd.dat
c
c          output averaged results into                    stdout
c
c    if plots are made, they are put into the following
c    files via ishell calls:
c        raw.img:    raw image
c        rawtac.img: raw tac spectrum
c        taccal.img: tac calibration curve
c        coinc.img:  image of coincident points
c        centroid.img:     image of centroid positions
c        valtac.img: valid tac spectrum
c
c  read in experimental constants from namelist.dat
c  in namelist format
c
      open(unit=4,file='namelist.dat',status='old')
      read(4,inp,iostat=ioerr)
      if(ioerr.ne.0) then
       write(*,*)'Error in namelist input!'
   write(6,*) 'Most recent changes:  ebm --> ebeam'
   write(6,*) 'ntce,tcocmin,tcocmax,gmin,gmax, eliminated'
   write(6,*) '                    DJL  4/94'
   write(6,*) 'nbinm eliminated--DJL 10/94'
       go to 999
      endif
      close(4)

c ***********************************************
      open(unit=7,file='title.dat')
      write(*,5030) title
      write(7,5030) title
c ************************************************

      if(nchan.lt.int_chan) then
       write(*,*)'Error in namelist input: int_chan > nchan!'
       write(*,*)'Exiting now!'
       stop
      endif
c
c gmin and gmax are the lower and upper limits for mass plotting
c   These are chosen so that the bins are centered on integers
   gmin = 0.5
   gmax = radical_mass - 0.5
   nbinm = radical_mass - 1
c
   vbeam = sqrt( 2*ebeam*1.60219E-19/(radical_mass*1.66056E-27) )
   vbeam = vbeam * 1000. ! convert from m/s to mm/s
c
      do i = 1,nchan
       if(i.eq.1) fragmass(ch1,light) = ch1lm
       if(i.eq.2) fragmass(ch2,light) = ch2lm
       if(i.eq.3) fragmass(ch3,light) = ch3lm
       if(i.eq.4) fragmass(ch3,light) = ch4lm
```

```
        if(fragmass(i,light).gt.(radical_mass/2)) then
         write(*,*)'Error in mass channel input, enter light masses ',
     >             'for each channel!'
         go to 999
        endif
       enddo
       do i = 1,nchan
        fragmass(i,heavy) = radical_mass - fragmass(i,light)
       enddo
c
c  now call subroutine hist to set up the results histograms
c
       call hist
c
c  read in workstation type from standard input, calibration and
c  scaling factors from various files, and then raw coincidence
c  data from raw_data.dat
c
       write(*,*)
       write(*,*)'Enter 0 for sun, 1 for TEK, 2 for X11 or 3 for
batch:'
       read(*,*)iwkst
c
   write(*,*)'Input files used by cmlwsa:'
   write(*,*)'cmlwsa parameter list ----> namelist.dat'
   write(*,*)'raw charge data       ----> raw_data.dat'
   write(*,*)'charge to pos factors ----> param.dat'
   write(*,*)'time:adc calibration  ----> tcal.dat'
   write(*,*)'adc calibration data  ----> adc_cal.dat'
   write(*,*)'pulse height limits   ----> phd.dat'
   write(*,*)'xy input data (opt.)  ----> raw_disc.dat'
c
c       now see if data has been run through previously.  If so, then
c       read in the xu,yu,xl,yl,t data from raw_disc.dat
c  If not, data is actually read in in wsztoxy.f
c
       if(iuseraw.eq.1) then !Note, raw_data.dat is read into
wsztoxy!!
        write(*,*)
        write(*,*)'!!! Reading in data from raw_disc.dat !!!'
        write(*,*)
        write(*,*)'If you have altered critical parameters, (for
example'
        write(*,*)'namelist.dat, param.dat, tcal.dat, phd.dat,',
     >             ' or adc_cal.dat), then please'
        write(*,*)'start again with iuseraw = 0 in namelist.dat ...'
        write(*,*)
        write(*,*)'Reading ... '
   open(unit=25,file='raw_disc.dat')
   ico = 0
   do i = 1,n1
    read(25,fmt=*,end=10) xu(i),yu(i),xl(i),yl(i),ttim(i)
    ico = ico + 1
   enddo
   close(25)
10 continue
   if(ico.eq.0) then
    write(*,*)
    write(*,*)'Error reading in raw_disc.dat, ',
     >             'trying raw_data.dat ...'
    iuseraw = 0
   else
    write(*,*)
    write(*,*) ico, ' events read in from raw_disc.dat'
```

```
      endif
      write(6,*)
      write(6,*)'Enter 0 to plot raw image. (1)'
      read(5,*)irawplt
      if(irawplt.eq.0) call rawplot(1)
         endif ! if iuseraw.eq.1
c
c         since there is no raw_disc.dat, or it is not to be used,
c         we will go through the whole deal. Also, set up so that if
c         factor changing is necessary it can be redone
c
         iredo = 0
c
         if(iuseraw.ne.1) then
c
c         read in adc calibration curve data from adc_cal.dat
c
          nadc = 0
          open(unit=10,file='adc_cal.dat',status='old')
          read(10,*)cw1,cs1,cz1,cw2,cs2,cz2
          do i = 1,n3
           read(10, *, end=15)vadc(i),adcw1(i),adcs1(i),adcz1(i),
     >                        adcw2(i),adcs2(i),adcz2(i)
           nadc = nadc + 1
          enddo
 15       continue
          close(10)
C
C **** Old Bob-style wedge-strip-zig factor read-in
C **** Preserved in case we want to re-analyze old N3 or CH2NO2 data
C                                  DJL 11/93
C
C  read in wsa scaling factors and TAC info from dff_tcal.dat
C
C 18      open(unit=3,file='dff_tcal.dat',status='old')
C         read(3,*)dmcwu,dmcsu,dmczu,dmcwl,dmcsl,dmczl,
C      >           dacwu,dacsu,dacwl,dacsl,
C      >           uctws,uctwz,uctsw,uctsz,
C      >           lctws,lctwz,lctsw,lctsz
C         ntac = 0
C         do i = 1,n3
C          read(3,*,end=19)tactrue(i),tacval(i)
C          ntac = ntac + 1
C         enddo
C
C 19      continue
C         close(3)
C
C **** End of commented-out Bob-style factors
C **** Now comes the new, xyzlsq-style factors    DJL 11/93
C Summary of param(i) conversion factors:
C as of 2/94: 1,3,5,7 are multiplicative factors for upper
C wedge, upper strip, lower wedge and lower strip. 2,4,6,8 are the
C corresponding additive factors.  9 and 11 are wedge-z crosstalk.
C 10 and 12 are tiltx and tilty corrections for deltaz.  13 and 14
C are multiplicative and additive factors for deltaz.  15 and 16 are
C quadratic corrections (equal for upper and lower) for the wedge and
strip.
C
18 open(10,file='param.dat',status='old')
   read(10,*) dummy
   do i=1,16
     read(10,*) dummy,param(i)
   end do
```

```
   close(10)
c Read in time-to-amplitude converter calibration numbers (and
tdelay)
   open (12,file='tcal.dat',status='old')
   read (12,*) tdelay
   ntac = 0
50 ntac = ntac + 1
   read (12,*,end=60) tactrue(ntac),tacval(ntac)
   goto 50
60 ntac = ntac - 1
   close(12)
c
c  now call subroutine wsztoxy to generate phd's, do pulse height
c       discrimination, convert raw adc data to nominal x,y positions
c  for each coincident event: xu(i),yu(i) and xl(i),yl(i)
c
       call wsztoxy


c
       write(*,*)
       write(*,*)'Enter 0 to plot raw image. (1)'
       read(*,*)irawplt
       if(irawplt.eq.0) then
        call rawplot(1)
        write(*,*)'Enter 0 if you wish to try another ',
     >            'param.dat file (1): '
        read(*,*)iredo
        if(iredo.eq.0) goto 18
       endif
c
c       here is where we ask the user if they would like to make a
c       raw_disc.dat file with the data they have calculated from the
c       phd discriminated raw data
c
       write(*,*)
       write(*,*)'Enter 0 if you wish to write to raw_disc.dat. (1)'
       read(*,*)iwrtft2
       if(iwrtft2.eq.0) then
        open(unit=25,file='raw_disc.dat')
        do i = 1,ico
         write(25,100)xu(i),yu(i),xl(i),yl(i),ttim(i)
        enddo
 100    format(5(2x,f10.5))
        close(25)
       endif
      endif    ! from if(iuseraw.ne.1)
c
c  plot tac spectrum if desired
c

   call tactime(1)


c
c  now, implement the various spatio-temporal (!!) discrimination
c  algorithms.  Used to do this in centcal, after finding the
c  all the centroid information.  The variable ico contains the
c  number of post phd-disc coincident events.  Use this as a starting
c  point for discr.f by setting icf = ico
c
   icf = ico
   call discr
```

```
c
c
c  now call centcal, to find the center of the image, if necessary,
c       and to then discriminate against false coincidences
c
   call centcal
c
c  now call savdfplt to save an array containing the binned detector
c  face image
c
c  call savdfplt     !     or not  DJL 10/94
c
c  now, given the x,y,t data, call subroutine psdclce (borrowed from
c       psdclc.f routine in cmlfrb2) to convert to energy,angle and
mass
c  distributions
c
   call psdclce
c
      call sigmac (ecmcaavg , ecmcaavg(1) , ecmcaavg(2))
      call sigmac (ecmcavg  , ecmcavg(1)  , ecmcavg(2))
      call sigmac (emltavg  , emltavg(1)  , emltavg(2))
      call sigmac (emhvavg  , emhvavg(1)  , emhvavg(2))
      call sigmac (tltavg   , tltavg(1)   , tltavg(2))
      call sigmac (thvavg   , thvavg(1)   , thvavg(2))
      call sigmac (tcoavg   , tcoavg(1)   , tcoavg(2))
      frrej = 1e0 - float(icf)/float(ico)
   do i = 1,ico
    chanval = chanval + ival1(i) !valid events in int_chan
   enddo
   accchanval = 100.*float(chanval)/float(ico)
c
c  find fwhm of KER, tlt and thv distributions
c
c      call fwhm(0.5,ecm,ptecbin,nbine,eh,epk,el,efw)
c      call fwhm(0.05,ecm,ptecbin,nbine,eh2,epk,el2,efw2)
c      call fwhm(0.5,tlab,ptltlab,nt,tlth,tltpk,tltl,tltfw)
c      call fwhm(0.5,tlab,pthvlab,nt,thvh,thvpk,thvl,thvfw)
c
c  output averaged results to standard output
c
      write(*,*)
      write(6,4011)ico,icf,frrej*100e0,chanval,accchanval
 4011 format(t10,i5,t20,' Raw coincidence events read ',/,
     >       t10,i5,t20,' Coincidence events accepted ',/,
     >       t10,g10.4,t20,' Percent rejected',/,
     >        t10,i5,t20,' Events accepted in int_chan',/,
     >        t10,g10.4,t20,' Percent accepted in int_chan',/)
      write(6,4012)nphddis,ncartdis,npoldis,nraddis,ncentdis
 4012 format(t10,i5,t20,' Events pulse height discriminated',/,
     >       t10,i5,t20,' Events cartesian discriminated ',/,
     >       t10,i5,t20,' Events azimuthal angle discriminated ',/,
     >       t10,i5,t20,' Events radially discriminated ',/,
     >       t10,i5,t20,' Events centroid discriminated ',/)
      write(6,4014)x0,y0
 4014 format(t5,'Nominal center of image = ',t35,g10.4,1x,g10.4,' mm
'
     >           ,//)
      write(6,4024)emltavg(1),emltavg(2),emhvavg(1),emhvavg(2)
 4024 format(t5,'Av light particle mass =',t40,g10.4,' amu ',/,
     >       t10,'Standard deviation = ',t40,g10.4,/,
     >       t5,'Av heavy particle mass =',t40,g10.4,' amu ',/,
     >       t10,'Standard deviation = ',t40,g10.4)
      write(6,4050) tcoavg(1),tcoavg(2)
```

```
 4050 format(t5,'Average Coincidence time = ',t40,g10.4,' ns ',/,
     > t10,'Standard deviation = ',t40,g10.4)
      write(6,4060)ecmcavg(1),ecmcavg(2),epk,efw
 4060 format (t5,'Average CM KER = ',t40,g10.4,' eV ',/,
     > t10,'Standard deviation = ',t40,g10.4,/,
     > t5,'Peak CM KER    = ',t40,g10.4,' eV ',/,
     > t5,'  CM KER FWHM  = ',t40,g10.4,' eV ')
      write(6,4070) ecmcaavg(1),ecmcaavg(2)
 4070 format(t5,' Approximate KER Calculation: ',/,
     > t5,'Average CM KER = ',t40,g10.4,' eV ',/,
     > t10,'Standard deviation = ',t40,g10.4)
c
c  now write out distributions for plotting to fort.8
c
c  title information (maximum of 40 characters)
   write(8,*)title
c  product KER spectrum
      call wrtla (8 , emin, dse , nbine)
      call wrtarr(8 , ptecbin , nbine)
c  approximate KER spectrum
      call wrtla (8 , emin, dse , nbine)
      call wrtarr(8 , ptecabin , nbine)
c        angular (theta) distribution
      call wrtla (8 , thmin, dsth, nth)
      call wrtarr(8 , pthcmc , nth)
c  angular (phi) distribution
      call wrtla (8 , thmin, dsth, 2*nth)
      call wrtarr(8 , pphcmc, 2*nth)
c  product mass spectrum
      call wrtla (8 , gmin, dsm, nbinm)
      call wrtarr(8 , pmcc, nbinm)
c  light particle TOF
      call wrtla (8 , tmin, dwell, nt)
      call wrtarr(8 , ptltlab, nt)
c  heavy particle TOF
      call wrtla (8 , tmin, dwell, nt)
      call wrtarr(8 , pthvlab, nt)
c  coincidence TOF spectrum
      call wrtla (8 , tcomin, dwellbc, ntbc)
      call wrtarr(8 , ptcolab, ntbc)
c  light particle y projection
      call wrtla (8, -ymx, dyp, nproj)
      call wrtarr(8, pylt , nproj)
c  heavy particle y projection
      call wrtla (8, -ymx, dyp, nproj)
      call wrtarr(8, pyhv , nproj)
c  light particle radial distribution
      call wrtla (8, -ymx, drp, nproj)
      call wrtarr(8, prlt, nproj)
c  heavy particle radial distribution
      call wrtla (8, -ymx, drp, nproj)
      call wrtarr(8, prhv, nproj)
      call wrtla (8, gmin, dsm, nbinm)
      call wrtarr(8, pimtu, nbinm)
c
c  Now output (energy, P(e)) to fort.55 for making daf
c  corrected KER plots
c
   call wrtarr2(55,ptecbin,nbine)
c
c       Write 2-d KER-theta distribution to fort.9
c
   write(9,*)nbine
   write(9,5000)(ecm(i),i=1,nbine)
```

```
      write(9,*)nth
      write(9,5000)(th(i),i=1,nth)
      do j = 1,nth
       write(9,5020)(ethetc(i,j),i=1,nbine)
      enddo

c
c        Write 2-d KER-phi distribution to fort.19
c
c  write(19,*)nbine
c  write(19,5000)(ecm(i),i=1,nbine)
c  write(19,*)2*nth
c  write(19,5000)(ph(i),i=1,2*nth)
c  do j = 1,2*nth
c   write(19,5020)(ephetc(i,j),i=1,nbine)
c  enddo
5000      format(6(1x,f8.4))
5020      format(6(1x,e10.5))
c

c*****************************************
5030    format(a25)
c*****************************************

c
999       continue
      end
```

discr.f

```
      subroutine discr
c
c         UNIX Version 920722 REC
c
c  This routine performs the various spatial and temporal
discrimination
c  routines.
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n4 = 200    )
c
        common / discinf /
ispatdis,isdis,ihlf,yll,yul,isdisp,phi1,phi2,
     >                    iraddis,idisan,rlld,ruld,ttl,ttu,
     >                 nphddis,ncartdis,npoldis,nraddis,ncentdis
c
        common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                ivaltmp(n1)
c
        common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / results / z_corr(n1),imtu(n1),upper_mass(n1),energy(n1)
c
        common / tacinf / tslope,tint,tdelay,
     >                   ttim(n1),tactrue(n3),tacval(n3),ntac
c
      data rad,pi / 0.01745329, 3.14159625 /
```

```
      dimension ivalorig(n1)
c
c  now, implement selective discrimination against a horizontal
c  stripe on the detector face.
c  currently, if y is within a range, check if x is and discriminate
c
c  set up the ival1(i) array to 1 for any event which passed pulse
c  height discrimination.  In this subroutine, centcal, and psdclce,
c  further deletion of events is accomplished by setting ival1(i) = 0

      icforig = icf
      do i = 1,ico
       ival1(i) = 1      !validity flag.  ival1 = 1 implies a good event
       ivaltmp(i) = 0 !temporary flag for use in plotting bad events
      enddo

      write(*,*)
      write(*,*)'Enter 0 to perform spatial or temporal ',
        >            'discrimination. (1)'
      read(*,*)ispatdis
         if(ispatdis.eq.0) then
      write(*,*)
      write(*,*)'Enter 0 to implement cartesian discrimination. (1)'
      read(*,*)isdis
      if(isdis.eq.0) then
      irspds = 1  !don't reset ival arrays if this is first time
399         continue
      if(irspds.eq.0) then
       do i = 1,ico
        ival1(i) = 1     !reset validity flag
        ivaltmp(i) = 0 !reset temp flag
       enddo
       icf = icforig      !reset number of valid events too
      endif
       write(*,*)
       write(*,*)'Enter 0 for upper half, 1 for lower, 2 for both:'
       read(*,*)ihlf
       write(*,*)'Enter lower and upper y limits for discrimination'
       write(*,*)' (absolute value of distance from the center [mm]):'
       read(*,*)yll,yul
       write(*,*)'Enter left and right x limits for discrimination:'
       read(*,*)xll,xrl
       write(*,*)'Enter lower and upper limits for abs(deltaT) in ns'
       read(*,*)ttl,ttu
c temporal discrimination
       icfn = icf
       do i = 1,ico
        if ( (ival1(i).eq.1).and.
        @          ((abs(ttim(i)).lt.ttl).or.(abs(ttim(i)).gt.ttu)) )
then
         ival1(i) = 0
         ivaltmp(i) = 1   !this will allow plots for thrown out pts.
         icfn = icfn - 1
        endif
       end do
c upper half cartesian discrimination
       if(ihlf.eq.0.or.ihlf.eq.2) then
        do i = 1,ico
          if(ival1(i).eq.1) then
      if((yu(i).lt.yll).or.(yu(i).lt.blocky).or.(yu(i).gt.yul).or.
        >          (xu(i).lt.xll).or.(xu(i).gt.xrl)) then
           ival1(i) = 0
           ivaltmp(i) = 1
           icfn = icfn - 1
```

```
        endif
       endif    ! ival.eq.1?  don't re-discriminate!
      enddo ! event loop, i index
     endif
c lower half cartesion discrimination
     if(ihlf.eq.1.or.ihlf.eq.2) then
      do i = 1,ico
       if(ivall(i).eq.1) then
        if((abs(yl(i)).lt.yll).or.(abs(yl(i)).lt.blocky).or.
      > (abs(yl(i)).gt.yul).or.(xl(i).lt.xll).or.(xl(i).gt.xrl)) then
         ivall(i) = 0
         ivaltmp(i) = 1
         icfn = icfn - 1
        endif
       endif
      enddo
     endif
     write(*,*)
     write(*,*)'No. of cart. disc. rejected points = ',icf - icfn
     ncartdis = icf - icfn
     icf = icfn
     write(6,*) 'Enter 0 for plotting.'
     read(5,*) iplot
     if (iplot.eq.0) then
      write(*,*)'Enter 0 to plot the spatially discriminated ',
      >            'TAC spectrum. (1)'
      read(*,*)iptv
      if(iptv.eq.0) call tactime(0)
      write(*,*)
      write(*,*)'Enter 0 to plot TAC spectrum of only the ',
      >            'discriminated points. (1)'
      read(*,*)iptv
      if(iptv.eq.0) call tactime(3)
      write(*,*)
      write(*,*)'Enter 0 to plot the pos. of coincident points. (1)'
      read(*,*)iptxyv
      if(iptxyv.eq.0) call rawplot(0)
      write(*,*)
      write(*,*)
      write(*,*)'Enter 0 to plot the pos. of only disc. points. (1)'
      read(*,*)iptxyv
      if(iptxyv.eq.0) call rawplot(3)
      write(*,*)
      write(*,*)'Enter 0 to plot array of coincident points. (1)'
      read(*,*)iptarr
      if(iptarr.eq.0) call plotcoin
      write(*,*)
      write(*,*)'Enter 0 to do cart. discrimination again. (1)'
      read(*,*)irspds
      if(irspds.eq.0) goto 399
     endif ! iplot.eq.0 conditional
    endif ! isdis.eq.0 conditional
    write(*,*)
    write(*,*)'Enter 0 to implement polar spatial ',
      >            'discrimination. (1)'
    read(*,*)isdisp
    if(isdisp.eq.0) then
     icfn = icf
     icforig = icf     !ok, move on to next discr., with new icforig
     do i = 1,ico
      ivalorig(i) = ivall(i)    !preserve ival array for 2nd time
     enddo
     irspds = 1
402       continue
```

```
      if(irspds.eq.0) then
       icfn = icforig
       icf = icforig
       do i = 1,ico
        ival1(i) = ivalorig(i)  !get a fresh start on 2nd time thru
       enddo
      endif
      write(*,*)
      write(*,405)
405         format(2x,'Polar spatial discrimination:',/,
     >   2x,' Enter phi1 ( acute ) and phi2 ( obtuse ) in degrees',/,
     >   2x,' These angles specify the azimuthal angles relative
to',/,
     >   2x,' the true center of the detector between which real',/,
     >   2x,' data must fall (defined in the upper half plane as',/,
     >   2x,' you view the detector).')
      read(*,*)phi1,phi2
      do i = 1,ico
       ivaltmp(i) = 0
       if(ival1(i).eq.1) then
c         check upper half, from +x axis go counterclockwise
        if(xu(i).gt.0e0) then
         phiu = atan(yu(i)/xu(i))/rad
        else
         phiu = atan(yu(i)/xu(i))/rad + pi/rad
        endif
        if((phiu.lt.phi1).or.(phiu.gt.phi2)) then
         ival1(i) = 0
         ivaltmp(i) = 1
        endif
c         check lower half, from +x axis go clockwise
        if(xl(i).lt.0e0) then
         phil = atan(abs(yl(i))/xl(i))/rad + pi/rad
        else
         phil = atan(abs(yl(i))/xl(i))/rad
        endif
        if((phil.lt.phi1).or.(phil.gt.phi2)) then
         ival1(i) = 0
         ivaltmp(i) = 1
        endif
        if(ival1(i).eq.0) icfn = icfn - 1
       endif ! if ival.eq.1
      enddo
      write(*,*)
      write(*,*)'No. of azimuth angle rejected points = ',icf - icfn
      npoldis = icf - icfn
      icf = icfn
      write(6,*) 'Enter 0 for plotting.'
      read(5,*) iplot
      if (iplot.eq.0) then
       write(*,*)'Enter 0 to plot the spatially discriminated ',
       >            'TAC spectrum. (1)'
       read(*,*)iptv
       if(iptv.eq.0) call tactime(0)
       write(*,*)
       write(*,*)'Enter 0 to plot TAC spectrum of only the ',
       >            'discriminated points. (1)'
       read(*,*)iptv
       if(iptv.eq.0) call tactime(3)
       write(*,*)
       write(*,*)'Enter 0 to plot the position of coincident ',
       >            'points. (1)'
       read(*,*)iptxyv
       if(iptxyv.eq.0) call rawplot(0)
```

```
      write(*,*)'Enter 0 to plot the pos. of only disc. points. (1)'
      read(*,*)iptxyv
      if(iptxyv.eq.0) call rawplot(3)
      write(*,*)
      write(*,*)
      write(*,*)'Enter 0 to plot array of coincident points. (1)'
      read(*,*)iptarr
      if(iptarr.eq.0) call plotcoin
      write(*,*)
      write(*,*)'Enter 0 to do polar discrimination again. (1)'
      read(*,*)irspds
      if(irspds.eq.0) go to 402
     end if ! iplot.eq.0
    endif ! isdisp.eq.0
c
    write(*,*)
    write(*,*)'Enter 0 to implement radial discrimination. (1)'
    read(*,*)iraddis
    if(iraddis.eq.0) then
     icfn = icf
     icforig = icf    !ok, move on to next discr., with new icforig
     do i = 1,ico
      ivalorig(i) = ival1(i)   !preserve ival array for 2nd time
     enddo
     irspds = 1
407       continue
     if(irspds.eq.0) then
      icfn = icforig
      icf = icforig
      do i = 1,ico
       ival1(i) = ivalorig(i)  !get a fresh start on 2nd time thru
      enddo
     endif
     write(*,*)
     write(*,*)'Enter 0 for upper half, 1 for lower, 2 for both:'
     read(*,*)ihlf
     write(*,*)
     write(*,*)'Enter 0 to discriminate an annulus, 1 to accept:'
     read(*,*)idisan
     write(*,*)
     write(*,*)'Enter lower and upper radial limits for ',
      >            'discrimination.'
     write(*,*)' (distance from the center (mm)):'
     read(*,*)rlld,ruld
     if(ihlf.eq.0.or.ihlf.eq.2) then
      do i = 1,ico
       ivaltmp(i) = 0 !must always reinitialize
       if(ival1(i).eq.1) then
             ruu = sqrt(xu(i)**2 + yu(i)**2)
             if(idisan.eq.0) then
              if(ruu.gt.rlld.and.ruu.lt.ruld) then
               ival1(i) = 0
         ivaltmp(i) = 1
               icfn = icfn - 1
              endif
             else
              if(ruu.lt.rlld.or.ruu.gt.ruld) then
               ival1(i) = 0
         ivaltmp(i) = 1
               icfn = icfn - 1
              endif
             endif
            endif
      enddo ! event loop, i index
```

```
      endif ! discrim top? conditional
      if(ihlf.eq.1.or.ihlf.eq.2) then
       do i = 1,ico
             if(ival1(i).eq.1) then
              rll = sqrt(xl(i)**2 + yl(i)**2)
              if(idisan.eq.0) then
               if(rll.gt.rlld.and.rll.lt.ruld) then
                ival1(i) = 0
          ivaltmp(i) = 1    !no need to reinitialize for lower half :-
)
                icfn = icfn - 1
               endif ! rll inside?
              else
               if(rll.lt.rlld.or.rll.gt.ruld) then
                ival1(i) = 0
          ivaltmp(i) = 1
                icfn = icfn - 1
               endif ! rll outside?
              endif ! accept inside or outside?
             endif ! ival.eq.1?
       enddo ! event loop, i index
      endif ! discrim bottom? conditional
      write(*,*)
      write(*,*)'No. of radially rejected points = ',icf - icfn
      nraddis = icf - icfn
      icf = icfn
      write(6,*)'Enter 0 for plotting.'
      read(5,*) iplot
      if (iplot.eq.0) then
       write(*,*)
       write(*,*)'Enter 0 to plot the spatial disc. TAC spectrum.(1)'
       read(*,*)iptv
       if(iptv.eq.0) call tactime(0)
       write(*,*)
       write(*,*)'Enter 0 to plot TAC spectrum of only the ',
       >            'discriminated points. (1)'
       read(*,*)iptv
       if(iptv.eq.0) call tactime(3)
       write(*,*)
       write(*,*)'Enter 0 to plot the pos. of coincident points. (1)'
       read(*,*)iptxyv
       if(iptxyv.eq.0) call rawplot(0)
       write(*,*)
       write(*,*)'Enter 0 to plot the pos. of only disc. points. (1)'
       read(*,*)iptxyv
       if(iptxyv.eq.0) call rawplot(3)
       write(*,*)
       write(*,*)'Enter 0 to plot array of coincident points. (1)'
       read(*,*)iptarr
       if(iptarr.eq.0) call plotcoin
       write(*,*)
       write(*,*)'Enter 0 to do radial discrimination again. (1)'
       read(*,*)irspds
       if(irspds.eq.0) go to 407
      end if ! iplot.eq.0
     endif ! iraddis.eq.0
        endif ! ispatdis.eq.0 (the big one)
c
        return
        end
```

hist.f

```
*deck hist.f
      Subroutine hist
c
c        unix version 911114 REC
c
c  this subroutine sets up the results histograms for the
c  cmlwsa.f program
c
      implicit real*4 (a-h,o-z)
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n4 = 200    )
      parameter ( n5 = 50     )
   parameter (n7 = 400)
      logical debug4
c
      common / beam / ebeam,vbeam,dpsd,rbeammax,rbeammin
c
      common / debug / debug4
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                   dsy,nx,ny
c
      common / fov / xfov(n4),yfov(n4),nfov
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / psdhist / emin,emax,dse,
     >                   thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                   tmin,tmax,dwell,tcomin,tcomax,dwellbc,
     >                   nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                   ypmin,rpmin,dyp,drp
c
      common / signal / ecm(n2),th(n2),u(n2),ephetc(n2,2*n2),
     >      ethetc(n2,n2),emcc(n2),pthcmc(n2),pphcmc(2*n2),pmcc(n2),
     >      ptltlab(n2),pthvlab(n2),ptcolab(n2),ptecbin(n2),tlab(n2),
     >      ph(2*n2),tcolab(n2),ptecabin(n2),pimtu(n2),pylt(n7),
     >      pyhv(n7),prlt(n7),prhv(n7)
c
      common / stat / tltavg(3),thvavg(3),tcoavg(3),ecmcavg(3),
     >          ecmcaavg(3),emltavg(3),emhvavg(3)
c
c  initialize all histograms on first trip through ( ncyc = 1 )
c
c
c  prepare for histogramming of translational energy distribution for
c  the state currently under consideration
c
      data  pi,degree / 3.14159265,57.29577951 /
      do 5 i = 1,3
       ecmcaavg(i) = 0e0
       ecmcavg(i)  = 0e0
       emltavg(i)  = 0e0
       emhvavg(i)  = 0e0
       tltavg(i)   = 0e0
       thvavg(i)   = 0e0
       tcoavg(i)   = 0e0
 5    continue
c
```

```
c  prepare for histogramming of the translational energy
distributions
c
c  first generate angular distribution arrays
c
      thmin  = 0e0
      thmax  = pi*degree
      phminc = 0e0
      phmaxc = 2e0*pi*degree
      call vsets ( nth, pthcmc, 1, 0e0  )
      call vsets ( 2*nth, pphcmc, 1, 0e0 )
      dsth    = ( thmax - thmin )/float(nth )
      call grs1 ( nth , thmin , dsth, th   )
      call grs1 ( 2*nth , phminc , dsth, ph   )!phi array of same bin
size
c
c  generate translational energy distribution array
c
      call vsets ( nbine, ptecbin,  1, 0e0 )
      call vsets ( nbine, ptecabin, 1, 0e0 )
      dse = ( emax - emin )/( nbine )
      call grs1 ( nbine, emin , dse, ecm )
c
c  position arrays for each particle
c
       dsx = ( sxmax - sxmin )/(nx)
       dsy = ( symax - symin )/(ny)
c
c  initialize the 2-d arrays P(ecm,theta) and P(ecm,phi)
c
       call asets( nbine, nth , ethetc , 0e0 )
       call asets( nbine, 2*nth , ephetc , 0e0 )
c
c        initialize mass spec array
c
       call vsets(nbinm,pmcc,1,0e0)
       call vsets(nbinm,pimtu,1,0e0)
       dsm = ( gmax - gmin )/(nbinm)
       call grs1( nbinm, gmin, dsm, emcc)
c
c  initialize time of arrival arrays for both products
c  and the coincidence time array.
c
       call vsets(nt,ptltlab,1,0e0)
       call vsets(nt,pthvlab,1,0e0)
       call vsets(ntbc,ptcolab,1,0e0)
       dwellbc = ( tcomax - tcomin )/( ntbc )
       dwell   = ( tmax - tmin )/( nt )
       call grs1( nt  , tmin  , dwell  , tlab   )
       call grs1( ntbc, tcomin, dwellbc, tcolab )
c
c  prepare detector face array for plots
c
      xmax = sxmax
      xmin = sxmin
      ymax = symax
      ymin = symin
      nfov = n4/2
      dfov = ( 2e0*pi )/float( nfov - 1 )
      do 10 i = 1,nfov
       xfov(i) = rfov*cos( float(i-1)*dfov )
       yfov(i) = rfov*sin( float(i-1)*dfov )
 10   continue
c
```

```
c  prepare r and y arrays for both light and heavy particles
c
      call vsets( nproj, pylt, 1, 0e0 )
      call vsets( nproj, pyhv, 1, 0e0 )
      call vsets( nproj, prlt, 1, 0e0 )
      call vsets( nproj, prhv, 1, 0e0 )
      ymx   = ymax
      ypmin = 0e0
      rpmin = 0e0
      dyp   = 2*ymx/(nproj)
      drp   = 2*ymx/(nproj)
c
c  I don't know why ypart and rpart arrays exist.  Must
c  be for histerical reasons.  Trash them for now.
c
c      call grs1(nproj, ypmin, dyp , ypart )
c      call grs1(nproj, rpmin, drp , rpart )
      return
      end
```

phaplt.f

```
*deck phaplt
      Subroutine phaplt
c
c        unix version 920229 REC
c
c  This subroutine plot raw tac data into pha spectrum
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n3 = 50     )
      parameter ( n6 = 8192   )
   character*40 title
c
      common / pha / puwpha(n6),puspha(n6),
     >               puzpha(n6),plwpha(n6),
     >               plspha(n6),plzpha(n6),
     >               tacpha(n6),ptacpha(n6),upha(n6),pupha(n6),
     >               lpha(n6),plpha(n6),nbin,phvpha(n6),pltpha(n6),
     >               iusephd
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / words / title

      call opngks
c  Plot Upper W PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
      call agsetc ( 'LABEL/NAME.','B')
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','Upper W PHD (ADC units)$')
      call ezxy (upha,puwpha,nbin,title)
c  Plot Upper S PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
```

```
      call agsetc ( 'LABEL/NAME.','B')
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','Upper S PHD (ADC units)$')
      call ezxy (upha,puspha,nbin,title)
c  Plot Upper Z PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
      call agsetc ( 'LABEL/NAME.','B')
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','Upper Z PHD (ADC units)$')
      call ezxy (upha,puzpha,nbin,title)
c  Plot Lower W PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
      call agsetc ( 'LABEL/NAME.','B')
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','Lower W PHD (ADC units)$')
      call ezxy (lpha,plwpha,nbin,title)
c  Plot Lower S PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
      call agsetc ( 'LABEL/NAME.','B')
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','Lower S PHD (ADC units)$')
      call ezxy (lpha,plspha,nbin,title)
c  Plot Lower Z PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
      call agsetc ( 'LABEL/NAME.','B')
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','Lower Z PHD (ADC units)$')
      call ezxy (lpha,plzpha,nbin,title)
c  Plot Upper Sum PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
      call agsetc ( 'LABEL/NAME.','B')
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','Upper Sum PHD (ADC units)$')
      call ezxy (upha,pupha,nbin,title)
c  Plot Lower Sum PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
      call agsetc ( 'LABEL/NAME.','B')
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','Lower Sum PHD (ADC units)$')
      call ezxy (lpha,plpha,nbin,title)
c  Plot TAC PHA
      call setusv('PB',2)
      call agsetc ( 'LABEL/NAME.','L')
      call agseti ( 'LINE/NUMBER.',100)
      call agsetc ( 'LINE/TEXT.','No. of Counts$')
      call agsetc ( 'LABEL/NAME.','B')
```

```
      call agseti ( 'LINE/NUMBER.',-100)
      call agsetc ( 'LINE/TEXT.','TAC PHD (ADC units)$')
      call ezxy (tacpha,ptacpha,nbin,title)
c
      call clsgks
      ierr = system( 'mv gmeta pha.img' )
      if(iwkst.eq.0) ierr = system( 'ictrans -d sunview pha.img' )
      if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 pha.img' )
      if(iwkst.eq.2) ierr = system( 'ictrans -d  X11 -bell pha.img' )
      return
      end
```

plotcentroid.f

```
c
c This file contains four subroutines, namely:
c  plotcent    creates scatter plot of centroids
c  plotcont    contour plot of centroid distribution
c  plotsurf    mesh surface plot of centroid distribution
c  plotrad           radial distribution of centroids
c                          DJL 4/94
c
c
*********************************************************************
*****
      subroutine plotcent(ivalid)
c
c        unix version 920716 DRC
c
c  this subroutine bins and plots the centroids as found
c  in the centcal routine, with options for thresholding
c  the image, etc.
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n4 = 200    )
   character*40 title
c
      common / beam / ebeam,vbeam,dpsd,rbeammax,rbeammin
c
      common / centdata / cent_x(n1),cent_y(n1)
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                   dsy,nx,ny
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >               ivaltmp(n1)
c
      common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / words / title
      dimension icentpos(n2,n2)
      dimension rminx(n4),rminy(n4)
      dimension ithreshdist(n2)
c
      data pi,degree / 3.14159265,57.29577951 /
      data ch1,ch2,ch3,ch4,light,heavy / 1, 2, 3, 4, 1, 2 /
```

```
c
c  using the x,y data in common / centdata /, bin into a position
c  array icentpos(n2,n2)
c
      xbin = 2 * rbeammax/float(nx)
      ybin = 2 * rbeammax/float(ny)
c
c  put the centroids of valid events (ivall=1) into the
c       position array within rbeammax
c
   if(ivalid.eq.0) then    !normal plot of valid centroids
      do i = 1,ico
       if (ivall(i).eq.1) then
        ixubin = nint((cent_x(i) - x0)/xbin + nx/2 + 0.5)
        iyubin = nint((cent_y(i) - y0)/ybin + ny/2 + 0.5)
        if((ixubin.ge.1).and.(ixubin.le.240).and.
     >     (iyubin.ge.1).and.(iyubin.le.240)) then
         iwin = iwin + 1
         icentpos(ixubin,iyubin) = icentpos(ixubin,iyubin) + 1
        endif
       endif
      enddo
   elseif(ivalid.eq.3) then
      do i = 1,ico
       if (ivaltmp(i).eq.1) then  !plot recently discr. centroids
        ixubin = nint((cent_x(i) - x0)/xbin + nx/2 + 0.5)
        iyubin = nint((cent_y(i) - y0)/ybin + ny/2 + 0.5)
        if((ixubin.ge.1).and.(ixubin.le.240).and.
     >     (iyubin.ge.1).and.(iyubin.le.240)) then
         iwin = iwin + 1
         icentpos(ixubin,iyubin) = icentpos(ixubin,iyubin) + 1
        endif
       endif
      enddo
   else
    stop 'check plotcentroid calling routine for argument error'
   endif !if (ivalid.eq.0)
      write(*,*)
      write(*,*)iwin,' valid events found within the ', rbeammax,
     >                ' mm test radius.'
      maxthreshdist = 1
      do i = 1,nx
       do j = 1,ny
        if((icentpos(i,j).gt.0).and.(icentpos(i,j).le.n2)) then
         ithreshdist(icentpos(i,j)) = ithreshdist(icentpos(i,j)) + 1
         if(icentpos(i,j).gt.maxthreshdist) then
          maxthreshdist = icentpos(i,j)
         endif
        endif
       enddo
      enddo
      write(*,*)
      nbinspos = nx*ny
 99   write(*,*)'Out of a possible ', nbinspos, ' bins, there are:'
      do i = 1,maxthreshdist
       if(ithreshdist(i).gt.0) then
        write(*,*)ithreshdist(i), ' bins with ', i, ' centroids'
       endif
      enddo
c
c       now plot the image
c
       d4 = ( 2e0*pi )/float( n4 - 1 )
       do 100 i = 1,n4
```

```
         rminx(i) = rbeammax*cos( float(i-1)*d4 ) + x0
         rminy(i) = rbeammax*sin( float(i-1)*d4 ) + y0
 100    continue
        xpmax = rbeammax + x0
        xpmin = xpmax - (2*rbeammax)
        ypmax = rbeammax + y0
        ypmin = ypmax - (2*rbeammax)
        call opngks
        call setusv('PB',2)
        call agseti('FRAME.',2)
        call agsetf('X/MAXIMUM.',xpmax)
        call agsetf('X/MINIMUM.',xpmin)
        call agsetf('Y/MAXIMUM.',ypmax)
        call agsetf('Y/MINIMUM.',ypmin)
c        call agseti('SET.',-1)
        call agsetc('LABEL/NAME.','L')
        call agseti('LINE/NUMBER.',100)
        call agsetc('LINE/TEXT.','Y [mm]$')
        call agsetc('LABEL/NAME.','B')
        call agseti('LINE/NUMBER.',-100)
        call agsetc('LINE/TEXT.',
     @          'X [mm] Scatter Plot of Valid Centroids$')
        call ezxy(rminx,rminy,n4,title)
        call points(rminx,rminy,n4,-1,0)
        ithresh = 1
        write(*,*)
        write(*,*)'Enter the plotting threshold:'
        read(*,*)ithresh
c        put crosshairs at x0,y0
        call line (x0, (y0+rbeammax), x0, (y0-rbeammax))
        call line ((x0-rbeammax), y0, (x0+rbeammax), y0)
c        now plot all the centroid bins with greater than 'ithresh'
        do i = 1,nx
         do j = 1,ny
          if(icentpos(i,j).ge.ithresh) then
           xpk = (float(i - nx/2) - 0.5)*xbin + x0
           ypk = (float(j - ny/2) - 0.5)*ybin + y0
           call point(xpk,ypk)
          endif
         enddo
        enddo
c
        call clsgks
        ierr = system( 'mv gmeta centroid.img' )
        if(iwkst.eq.0) ierr = system( 'ictrans -d sunview
centroid.img' )
        if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 centroid.img'
)
        if(iwkst.eq.2) ierr =
     @          system( 'ictrans -d X11 -bell centroid.img' )
        write(*,*)
        write(*,*)'Enter 0 if you want to try another threshold. (1)'
        read(*,*)iathr
        if(iathr.eq.0) go to 99
        return
       end
c
c
*****************************************************************
*****
   subroutine plotcont
c
c         unix version 930318 DRC
c
```

```
c  this subroutine bins and plots the centroids as a contour
c  plot in the centcal routine
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n4 = 200    )
      parameter ( n5 = 25     )
   character*40 title

c
      common / beam / ebeam,vbeam,dpsd,rbeammax,rbeammin
c
      common / centdata / cent_x(n1),cent_y(n1)
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                   dsy,nx,ny
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                 ivaltmp(n1)
c
      common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / words / title

      dimension centpos(n5,n5)
      dimension ithreshdist(n5)
c
      data pi,degree / 3.14159265,57.29577951 /
      data ch1,ch2,ch3,ch4,light,heavy / 1, 2, 3, 4, 1, 2 /
c
c  using the x,y data in common / centdata /, bin into a position
c  array centpos(n5,n5)
c
      nxcont = n5
      nycont = n5
c
      do i = 1,nxcont
       do j = 1,nycont
        centpos(i,j) = 0.0
       enddo
      enddo
c
      xbin = 2.0 * rbeammax/float(nxcont)
      ybin = 2.0 * rbeammax/float(nycont)
c
c  put the centroids of valid events (ivall=1) into the
c       position array within rbeammax
c
      do i = 1,ico
       if(ivall(i).eq.1) then
         ixubin = nint((cent_x(i) - x0)/xbin + float(nxcont)/2 + 0.5)
         iyubin = nint((cent_y(i) - y0)/ybin + float(nycont)/2 + 0.5)
         if((ixubin.ge.1).and.(ixubin.le.nxcont).and.
     >       (iyubin.ge.1).and.(iyubin.le.nycont)) then
          iwin = iwin + 1
          centpos(ixubin,iyubin) = centpos(ixubin,iyubin) + 1.0
         endif
       endif
      enddo
      write(*,*)
```

```
      write(*,*)iwin,' valid events found within the ', rbeammax,
     >                ' mm test radius.'
      maxthreshdist = 1
      do i = 1,nxcont
       do j = 1,nycont
        if(int(centpos(i,j)).gt.maxthreshdist) then
         maxthreshdist = int(centpos(i,j))
        endif
       enddo
      enddo
      ibinsize = int(maxthreshdist/n5) + 1
      numbins = n5
      do i = 1,nxcont
       do j = 1,nycont
        if(int(centpos(i,j)/ibinsize).gt.0) then
         ithreshdist(int(centpos(i,j)/ibinsize)) = 1 +
     >                ithreshdist(int(centpos(i,j)/ibinsize))
        endif
       enddo
      enddo
      write(*,*)
      nbinspos = nxcont*nycont
 99   write(*,*)'Out of a possible ', nbinspos, ' bins, there are:'
      do i = 1,numbins
       if(ithreshdist(i).gt.0) then
        write(*,*)'Number of bins with between ',(ibinsize*(i-1)),
     >            ' and ',(ibinsize*i),' is : ',ithreshdist(i)
       endif
      enddo
c
c       now plot the centroid image with contours
c
c  now get set for contours
c
       call opngks
       ierror = 0
c
c        select normalization transformation number 0 (?)
c
       call GSELNT(0)
c
c  now generate contours
c
       distmax = float(maxthreshdist)
       contspace = int(0.01*float(maxthreshdist)) * 10.0
       call WTSTR (0.5, 0.975,
     >         'Contour plot of centroids',2,0,0)
       call CONREC(centpos,nxcont,nycont,nycont,
     >             0.0,distmax,contspace,0,0,0)
       call FRAME
       if(ierror.ne.0) then
        write(*,*)
        write(*,*)'ERROR in CONREC!!!!!'
       endif
       do i = 1,nxcont
        do j = 1,nycont
         write(91,*)i,j,centpos(i,j)
        enddo
       enddo
c
       call clsgks
       ierr = system( 'mv gmeta centcont.img' )
       if(iwkst.eq.0) ierr = system( 'ictrans -d sunview
centcont.img' )
```

```
      if(iwkst.eq.1) ierr =
     @         system( 'ictrans -d t4010 centcont.img' )
      if(iwkst.eq.2) ierr =
     @         system( 'ictrans -d X11 -bell centcont.img' )
      return
      end
c
c
*****************************************************************
*****
   subroutine plotsurf
c
c        unix version 930321 DRC
c
c  this subroutine bins and plots the centroids as a surface
c  plot in the centcal routine
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n4 = 200    )
      parameter ( n5 = 30     )
   character*40 title
c
      common / beam / ebeam,vbeam,dpsd,rbeammax,rbeammin
c
      common / centdata / cent_x(n1),cent_y(n1)
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                   dsy,nx,ny
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >               ivaltmp(n1)
c
      common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / words / title

c  load the SRFACE common block
c
      common / srfipl / ifr,istp,irots,idrx,idry,idrz,iupper,iskirt,
     >                  ncla,theta,hskirt,chi,clo,cinc,ispval
c
      dimension xx(n5),yy(n5),zheight(n5,n5),sight(6)
      dimension scratch(2*n5*n5)
c
      data pi,degree / 3.14159265,57.29577951 /
      data ch1,ch2,ch3,ch4,light,heavy / 1, 2, 3, 4, 1, 2 /
      data sight(1), sight(2), sight(3), sight(4), sight(5), sight(6)
/
     >     -8.0, -6.0,  3.0,  0.0,  0.0,  0.0 /
c
c  using the x,y data in common / centdata /, bin into a position
c  array zheight(n5,n5)
c
      nxcont = n5
      nycont = n5
c
      do i = 1,nxcont
       do j = 1,nycont
        zheight(i,j) = 0.0
```

```
        enddo
       enddo
c
       xbin = 2.0 * rbeammax/float(nxcont)
       ybin = 2.0 * rbeammax/float(nycont)
c
       do i = 1,nxcont
        xx(i) = (i-0.5)*xbin + x0 - rbeammax
       enddo
c
       do i = 1,nycont
        yy(i) = (i-0.5)*ybin + y0 - rbeammax
       enddo
c
c  put the centroids of valid events (ivall=1) into the
c       position array within rbeammax
c
       do i = 1,ico
        if(ivall(i).eq.1) then
          ixubin = nint((cent_x(i) - x0)/xbin + float(nxcont)/2 + 0.5)
          iyubin = nint((cent_y(i) - y0)/ybin + float(nycont)/2 + 0.5)
          if((ixubin.ge.1).and.(ixubin.le.nxcont).and.
     >       (iyubin.ge.1).and.(iyubin.le.nycont)) then
           iwin = iwin + 1
           zheight(ixubin,iyubin) = zheight(ixubin,iyubin) + 1.0
          endif
        endif
       enddo
       write(*,*)
       write(*,*)iwin,' valid events found within the ', rbeammax,
     >                ' mm test radius.'
       zheightmax = 1
       do i = 1,nxcont
        do j = 1,nycont
         if(zheight(i,j).gt.zheightmax) then
          zheightmax = zheight(i,j)
         endif
        enddo
       enddo
c
c  normalize so zheightmax is now 2.0*n5
c
       do i = 1,nxcont
        do j = 1,nycont
         zheight(i,j) = zheight(i,j) * n5 / zheightmax
        enddo
       enddo
c
       sight(3) = 5.0 * n5
c
c        now plot the centroid image as a surface
c
c  now get set for surface
c
        call opngks
        ierror = 0
c
c  set surface parameters to suppres the FRAME call and draw contours
c
c       IFR = 0
c       IDRZ = 1
c
c         select normalization transformation number 0 (?)
c
```

```
      call GSELNT(0)
c
c  use text alignment to center string, set character height
c
      call GSTXAL(2,3)
      call GSCHH(0.015)
c
c  looking from bottom of front of detector
c
     sight(1) = -4.0 * rbeammax
     sight(2) = -5.0 * rbeammax
c
c  now write the text, generate surface
c
      call GTX (0.5, 0.975,
    >          'CENTROID DENSITY MAP FROM BOTTOM/LEFT/FRONT')
c
      call SRFACE(xx,yy,zheight,scratch,n5,n5,n5,sight,0.0)
c
      if(ierror.ne.0) then
       write(*,*)
       write(*,*)'ERROR in CONREC!!!!!'
      endif
c
c  looking from left of front of detector
c
     sight(1) = 5.0 * rbeammax
     sight(2) = -4.0 * rbeammax
c
c  now write the text, generate surface
c
      call GTX (0.5, 0.975,
    >          'CENTROID DENSITY MAP FROM BOTTOM/RIGHT/FRONT')
c
      call SRFACE(xx,yy,zheight,scratch,n5,n5,n5,sight,0.0)
c
      if(ierror.ne.0) then
       write(*,*)
       write(*,*)'ERROR in CONREC!!!!!'
      endif
c
     write(91,*) sight
     write(91,*)
     write(91,*) xx
     write(91,*)
     write(91,*) yy
     write(91,*)
     write(91,*) zheightmax
     write(91,*)
     write(91,*) zheight
c
      call clsgks
      ierr = system( 'mv gmeta centsurf.img' )
      if(iwkst.eq.0) ierr =
    @         system( 'ictrans -d sunview centsurf.img' )
      if(iwkst.eq.1) ierr =
    @         system( 'ictrans -d t4010 centsurf.img' )
      if(iwkst.eq.2) ierr =
    @         system( 'ictrans -d X11 -bell centsurf.img' )
      return
     end
c
```

```
c
*****************************************************************
*****
   subroutine plotrad
c
c        unix version 921201 DRC
c
c  this subroutine plots the radial histogram of the centroids
c  as found in the centcal routine, with options for
c  thresholding the image, etc.
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n4 = 200 )
   character*40 title
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / radinf / radmin,radmax,dsrad,nbinrad,iraddist(n4),
     @                ixbindist(n4),iybindist(n4)
      common / words / title

c
      dimension radialdistx(3*n4+1),gridpt(3*n4+1)
      dimension radialdisty1(3*n4+1), radialdisty2(3*n4+1)
      dimension xdist(3*n4+1),ydist(3*n4+1)
c
      data  pi,degree / 3.14159265,57.29577951 /
c
c  using the data in common / radinf /, construct and plot a
c       histogram of the results
c
      nbin = nbinrad
c
c       now plot the histogram, remembering the normalization factor
c       for the area found in a given annulus
c
      xpmax = 1.05*radmax
      xpmin = radmin-0.05*radmax
      ypmin = 0
      ypmax = 1.05
      do i = 1,(3*nbin+1)  !radialdistx is the radial coordinate
values
       radialdistx(i) = float(i/3) * radmax/float(nbin)
      enddo
c

      do i = 1,(3*nbin+1)
       if(mod(i,3).eq.1) then    !mod(i,3) returns remainder of i/3
        radialdisty1(i) = 0.0
       else  !demoninator in radialdisty1 is area of anulus!
        radialdisty1(i) = float(iraddist(((i-1)/3) + 1)) /
     >                   (radialdistx(i+1)**2 - radialdistx(i-1)**2)
       endif
      enddo
      ymaxval = 0
      do i = 3,(3*nbin),3  !find maximum in radialdisty1
       if(radialdisty1(i).gt.ymaxval) then
        ymaxval = radialdisty1(i)
       endif
      enddo
      do i = 1,(3*nbin+1)  !normalize radialdisty1
       radialdisty1(i) = radialdisty1(i) / ymaxval
      enddo
```

```
c
c*****************************************************
c  Insert arrays for radial distribution along x and y axes
c  Since we want to see the centroid distributions for both positive
c  and negative values of x and y, creat new set of grid points.
These
c  points are spaced at half the resolution of the radial points, but
c  cover twice the distance, so the points per bin will be comparable

      do i = 1,(3*nbin+1)  !radialdistx is the radial coordinate
values
       gridpt(i) = (float(-nbin) + 2*(float(i/3))) *
radmax/float(nbin)
      enddo

c  x axis first
      do i = 1,(3*nbin+1)
       if(mod(i,3).eq.1) then
        xdist(i) = 0.0
       else   !assign intensities for histogram along x axis
        xdist(i) = float(ixbindist(((i-1)/3) + 1))
       endif
      enddo
      xmaxval = 0          !find maximum
      do i = 3,(3*nbin),3
       if(xdist(i).gt.xmaxval) then
        xmaxval = xdist(i)
       endif
      enddo
c      do i = 1,(3*nbin+1) !normalize intensity to 1
c       xdist(i) = xdist(i) / xmaxval
c      enddo

c  now do y axis distribution
      do i = 1,(3*nbin+1)
       if(mod(i,3).eq.1) then
        ydist(i) = 0.0
       else
        ydist(i) = float(iybindist(((i-1)/3) + 1))
       endif
      enddo
      ymaxval = 0
      do i = 3,(3*nbin),3
       if(ydist(i).gt.ymaxval) then
        ymaxval = ydist(i)
       endif
      enddo
c      do i = 1,(3*nbin+1) !normalize to 1
c       ydist(i) = ydist(i) / ymaxval
c      enddo


   open(unit = 99)
   do i = 1,nbin
   write(99,*)iraddist(i),ixbindist(i),iybindist(i)
   enddo
   close(99)
c*************************end of x,y axes
distributions******************

c        change normalization factor
c
```

```
      do i = 1,(3*nbin+1)
       if(mod(i,3).eq.1) then
        radialdisty2(i) = 0.0
       else        !not normalized to area of anulus
        radialdisty2(i) = float(iraddist(((i-1)/3) + 1))
       endif
      enddo
      ymaxval = 0
      do i = 3,(3*nbin),3
       if(radialdisty2(i).gt.ymaxval) then
        ymaxval = radialdisty2(i)
       endif
      enddo
      do i = 1,(3*nbin+1)
       radialdisty2(i) = radialdisty2(i) / ymaxval
      enddo
c
c  now plot radial distributions
c
      call opngks
      call setusv('PB',2)
      call agseti('SET.',1)
      call agseti('FRAME.',1)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','Normalized Counts [arb]$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.',
     @          'Radius [mm] Centroid Radial Distribution$')
      call ezxy(radialdistx,radialdisty1,(3*nbin+1),title)
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','Counts [arb]$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.',
     @          'Radius [mm] Centroid Radial Distribution$')
      call ezxy(radialdistx,radialdisty2,(3*nbin+1),title)
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','Counts [arb]$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.',
     @          'Radius [mm] X AXIS CENTROID DIST$')
      call ezxy(gridpt,xdist,(3*nbin+1),title)
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','Counts [arb]$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.',
     @          'Radius [mm] Y AXIS CENTROID DIST$')
      call ezxy(gridpt,ydist,(3*nbin+1),title)
      call clsgks
      ierr = system( 'mv gmeta raddist.img' )
      if(iwkst.eq.0) ierr = system( 'ictrans -d sunview raddist.img'
)
      if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 raddist.img' )
```

```
      if(iwkst.eq.2) ierr = system( 'ictrans -d X11 -bell
raddist.img' )
      return
      end
```

plotcoin.f

```
*deck plotcoin.f
      subroutine plotcoin
c
c        unix version 911118 REC
c
c  this subroutine bins and plots the true coincidences as
c  found in the centcal routine, with options for thresholding
c  the image, etc.
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n4 = 200    )
   character*40 title
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                   dsy,nx,ny
c
      common / fov / xfov(n4),yfov(n4),nfov
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ival1(n1),
     >                 ivaltmp(n1)
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / tacinf / tslope,tint,tdelay,
     >                  ttim(n1),tactrue(n3),tacval(n3),ntac
c
      common / words / title

c  using the x,y data in common / image /, bin into a position
c
      xref = ( sxmax - sxmin )/2e0
      yref = ( symax - symin )/2e0
      xbin = ( sxmax - sxmin )/float(nx)
      ybin = ( symax - symin )/float(ny)
c
      write(*,*)'Enter 0 to plot x,y for a specific coinc.
window.(1)'
      read(*,*)ipsco
c
c  put valid events (ival1=1) into the position array
c
      if(ipsco.eq.0) then
 98    continue
       do i = 1,n2
        do j = 1,n2
         idfimg(i,j) = 0
        enddo
       enddo
       write(*,*)'Enter lower and upper coinc. times (in nsec)'
       read(*,*) ttl,ttu
       iwin = 0
```

```
        do i = 1,ico
         if( (abs(ttim(i)).ge.ttl).and.(abs(ttim(i)).le.ttu) ) then
          if(ivall(i).eq.1) then
           iwin = iwin + 1
           if(xu(i).le.sxmax.and.xu(i).ge.sxmin) then
            if(yu(i).le.symax.and.yu(i).ge.symin) then
             ixubin = int( (xu(i) + xref ) /xbin ) + 1
             iyubin = int( (yu(i) + yref ) /ybin ) + 1
             idfimg(ixubin,iyubin) = idfimg(ixubin,iyubin) + 1
            endif
           endif
           if(xl(i).le.sxmax.and.xl(i).ge.sxmin) then
            if(yl(i).le.symax.and.yl(i).ge.symin) then
             ixlbin = int( (xl(i) + xref ) /xbin ) + 1
             iylbin = int( (yl(i) + yref ) /ybin ) + 1
             idfimg(ixlbin,iylbin) = idfimg(ixlbin,iylbin) + 1
            endif
           endif
          endif
         endif
        enddo
        write(*,*)iwin,' valid events found within this window'
       else
        do i = 1,ico
         if(ivall(i).eq.1) then
          if(xu(i).le.sxmax.and.xu(i).ge.sxmin) then
           if(yu(i).le.symax.and.yu(i).ge.symin) then
            ixubin = int( (xu(i) + xref ) /xbin ) + 1
            iyubin = int( (yu(i) + yref ) /ybin ) + 1
            idfimg(ixubin,iyubin) = idfimg(ixubin,iyubin) + 1
           endif
          endif
          if(xl(i).le.sxmax.and.xl(i).ge.sxmin) then
           if(yl(i).le.symax.and.yl(i).ge.symin) then
            ixlbin = int( (xl(i) + xref ) /xbin ) + 1
            iylbin = int( (yl(i) + yref ) /ybin ) + 1
            idfimg(ixlbin,iylbin) = idfimg(ixlbin,iylbin) + 1
           endif
          endif
         endif
        enddo
       endif
c
c      now plot the image, if desired
c
       write(*,*)'Enter 0 to plot the image of valid data.(1)'
       read(*,*)ivalt
       if(ivalt.eq.0) then
 99     continue
        call opngks
        call setusv('PB',2)
        call agseti('FRAME.',2)
        call agseti('X/MAXIMUM.',sxmax)
        call agseti('X/MINIMUM.',sxmin)
        call agseti('Y/MAXIMUM.',symax)
        call agseti('Y/MINIMUM.',symin)
        call agseti('SET.',-1)
        call ezxy(xfov,yfov,nfov,'Scatter Plot of Valid Events$')
        call points(xfov,yfov,nfov,-1,0)
        thresh = 1e0
        write(*,*)'Enter the plotting threshold'
        read(*,*)ithresh
c  mark center
        call points(x0,y0,1,-2,0)
```

```
        do i = 1,nx
         do j = 1,ny
          if(idfimg(i,j).ge.ithresh) then
           xpk = (float(i) - 0.5)*xbin - xref
           ypk = (float(j) - 0.5)*ybin - yref
           call point(xpk,ypk)
          endif
         enddo
        enddo
        call clsgks
        ierr = system( 'mv gmeta coinc.img' )
        if(iwkst.eq.0) ierr = system( 'ictrans -d sunview coinc.img' )
        if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 coinc.img' )
        if(iwkst.eq.2) ierr = system( 'ictrans -d X11 coinc.img' )
        write(*,*)'Enter 0 if you want to try another threshold. (1)'
        read(*,*)iathr
        if(iathr.eq.0) go to 99
        write(*,*)'Enter 0 if you want to try another coinc. window
(1)'
        read(*,*)icowin
        if(icowin.eq.0) go to 98
       endif
       return
       end
```

plotmass.f

```
       subroutine plotmass
c
c         latest version 930316 DRC
c
c  this subroutine plots the histogram of the mass of the fragments
c  as found in the centcal routine
c
       implicit real*4 (a-h,o-z)
c
       parameter ( n2 = 300 )
   parameter (n7 = 400)
   character*40 title
c
       common / debug / debug4
c
       common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
c
       common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
       common / psdhist / emin,emax,dse,
     >                    thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                    tmin,tmax,dwell,tcomin,tcomax,dwellbc,
     >                    nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                    ypmin,rpmin,dyp,drp
c
       common / signal / ecm(n2),th(n2),u(n2),ephetc(n2,2*n2),
     >      ethetc(n2,n2),emcc(n2),pthcmc(n2),pphcmc(2*n2),pmcc(n2),
     >      ptltlab(n2),pthvlab(n2),ptcolab(n2),ptecbin(n2),tlab(n2),
     >      ph(2*n2),tcolab(n2),ptecabin(n2),pimtu(n2),pylt(n7),
     >      pyhv(n7),prlt(n7),prhv(n7)
c
       common / words / title

       dimension axis_mass(0:3*n2), top_mass(0:3*n2), all_mass(0:3*n2)
c
```

```
c  using the mass data in common / signal /, namely, in pimtu,
c  construct and plot a histogram of the results
c
   open(22,file='massdist.dat')
   do i = 1,nbinm
     write(22,*) i,(pimtu(i) + pimtu(nbinm-i+1))/2. !all data
   enddo
   close(22)
c
c The data bins are centered on the integers; bin 1 is mass=1, and so
on
c (There are no bins for mass = 0 or mass = radical_mass)
c nbinm = radical_mass - 1; the bin spacing is fixed at dsm = 1
c Histogram points come in triplets:
c   1st point is top left of bar for the ith point
c   2nd point is top right of bar
c   3rd point is bottom right of bar (i.e., y value is zero)
c There is one bonus point at the beginning for the bottom left of
the first
c   histogram bar; thus the arrays go from 0 to 3*n, for a total of
3*n+1
c
   axis_mass(0) = 0.5
   do i = 1,nbinm*3
     axis_mass(i) = 0.5 + ((i+1)/3)
   enddo
c
   top_mass(0) = 0
   all_mass(0) = 0
   do i = 1,nbinm
     top_mass(3*i-2) = pimtu(i)
     top_mass(3*i-1) = pimtu(i)
     top_mass(3*i)   = 0.0
     all_mass(3*i-2) = ( pimtu(i) + pimtu(nbinm+1-i) ) / 2
     all_mass(3*i-1) = ( pimtu(i) + pimtu(nbinm+1-i) ) / 2
     all_mass(3*i)   = 0.0
   enddo
c
c  now plot mass spec graphs: top_mass and all_mass vs. axis_mass
c
      call opngks
      call setusv('PB',2)
      call agseti('SET.',1)
      call agseti('FRAME.',1)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','Top Half of Detector Raw Counts$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.',
     @   'Mass [amu] Mass Spec, Top Half of Detector Only$')
      call ezxy(axis_mass,top_mass,nbinm*3+1,title)
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','All Raw Counts$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.',
     @         'Mass [amu] Raw Mass Spectrum, All Data$')
      call ezxy(axis_mass,all_mass,nbinm*3+1,title)
      call clsgks
      ierr = system( 'mv gmeta massdist.img' )
```

```
      if(iwkst.eq.0) ierr = system( 'ictrans -d sunview massdist.img'
)
      if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 massdist.img' )
      if(iwkst.eq.2) ierr = system( 'ictrans -d X11 massdist.img' )
      return
      end
```

PSDCLCE.f

```
   subroutine psdclce
c
c------------------------------ psdclce ----------------------------
-----------
c
c                 unix version  911114 REC
c                      updated       920810 DRC
c
c  This subroutine is adapted from the psdclc.f subroutine in the
c  cmlfrb2 program.
c  It takes results of raw coincidence data and converts to the
c  center-of-mass quantities and bins the results as desired.
c  MODIFIED 12/11/92 by DLO to use vertically polarized data.
c
c-----------------------------------------------------------------------
---------
c
c  ipolang: laser polarization angle (0 = horiz., 90 = vert.)
c  radical_mass  : parent mass
c
c  common block signal
c        ecm(nbine) is the cm trans. energy array
c        th(nth) is the vector which defines the cm angle bins
c        ethetc(nbine,nth) is the array of cm energy vs. angle
derived
c                    from the experimental observables
c        imtu(nbinm) is the integer mass dist. that is negative
c         if particle on bottom hit first, positive otherwise.  Also
c         remember that imtu and pimtu only refer to the upper half.
c         imtu is an array which gives this info for all events,
c         whereas pimtu will be used by plotmass and at any given
c         point in the code may only represent a subset of the valid
c         data.  That's why you need them both. imtu is created in
c         centcal and is never modified later.  pimtu is frequently
c         modified throughout the code.
c               pimtu(nbinm) is the mass distribution binned to nint
and
c         is always positive
c        ptltlab(nt) is the toa distribution for light particle.
c        pthvlab(nt) is the toa distribution for heavy particle.
c        ptcolab(ntbc) is the coinc. time distribution
c               ptecbin(np) is the calculated cm energy distribution
c        tlab(nt) is the toa array
c        tcolab(ntbc) is the time-of-coincidence array
c
c-----------------------------------------------------------------------
-------
c
      implicit real*4 (a-h,o-z)
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n5 = 50     )
```

```
      parameter (n7 = 400)
         logical debug4
c
         common / beam / ebeam,vbeam,dpsd,rbeammax,rbeammin
c
         common / debug / debug4
c
         common / flags / iforce,iabs
c
         common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                  ivaltmp(n1)
c
         common / multchan / radical_mass,fragmass(4,2),nchan,int_chan
c
         common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
         common / psdhist / emin,emax,dse,
     >                     thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                     tmin,tmax,dwell,tcomin,tcomax,dwellbc,
     >                     nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                     ypmin,rpmin,dyp,drp
c
      common / results / z_corr(n1),imtu(n1),upper_mass(n1),energy(n1)
c
         common / signal / ecm(n2),th(n2),u(n2),ephetc(n2,2*n2),
     >        ethetc(n2,n2),emcc(n2),pthcmc(n2),pphcmc(2*n2),pmcc(n2),
     >        ptltlab(n2),pthvlab(n2),ptcolab(n2),ptecbin(n2),tlab(n2),
     >        ph(2*n2),tcolab(n2),ptecabin(n2),pimtu(n2),pylt(n7),
     >        pyhv(n7),prlt(n7),prhv(n7)
c
         common / stat / tltavg(3),thvavg(3),tcoavg(3),ecmcavg(3),
     >           ecmcaavg(3),emltavg(3),emhvavg(3)
c
         common / tacinf / tslope,tint,tdelay,
     >                    ttim(n1),tactrue(n3),tacval(n3),ntac
c
         real*4 lower_mass,rul(4),pimtuold(n2)
      character*1 massch
      character*20 massfile
      real*4 posnum,negnum,posfrac,negfrac

c
         data rad,pi / 0.01745329 , 3.14159265 /
         data degree / 57.29577951 /
         data ch1,ch2,ch3,ch4,light,heavy / 1, 2, 3, 4, 1, 2 /
c
      write(*,*)
      write(*,*)'radical beam energy = ',ebeam,' eV '
      write(*,*)'radical beam velocity = ',vbeam/1000.,' m/s'
c
c  reset ivall to 0 for all events not belonging to int_chan
c  (the channel of interest) only if we know the mass channels
c
         if(iforce.eq.1) then

          do 35 i = 1,ico
      if(ivall(i).eq.0) goto 35  !if it's already junk, skip it!
           if((abs(imtu(i)).ne.fragmass(int_chan,light)).and.
     >        (abs(imtu(i)).ne.fragmass(int_chan,heavy))) then
            ivall(i) = 0
           endif
35        enddo

         endif    !if(iforce.eq.1)
```

```
c
c  reset the upper fragment mass spectrum array so better
c  discriminated data is accumulated into fresh array
c
      do i = 1,nbinm
       pimtu(i) = 0
      enddo
c
   do 7000 i=1,ico
         if(ivall(i).eq.0) goto 7000
c
c  If ttim(i) < 0 then the 'stop' would have come first, if not for
c  the digital delay box;
c  this means the particle hit the bottom half of the psd first.
c
    if(abs(ttim(i)).gt.tcomax) goto 7000
c
c          rul is the total distance between the two fragments ( in
mm )
c
       rul(1) = xu(i) - xl(i)
       rul(2) = yu(i) - yl(i)
       rul(3) = 0
       rul(4) = vnorm(rul,3)
       if(debug4) then
        write (*,*)
        write (*,*) '   rul(1),   rul(2),   rul(3),   rul(4)'
        write (*,*) rul(1), rul(2), rul(3), rul(4)
       endif
c********************Begin angle determination if
block********************
c    interrogate ipolang with if statement and choose appropriate
block
c
   if(ipolang.eq.0) then
c
c         thcmc is the cm polar scattering angle from 0 to 180
degrees
c         found (for this polarization)
c         by taking the inverse tangent of the line from the lower
c         fragment to the upper fragment divided by the distance
c         equal to the beam velocity multiplied by the coincidence
c         time.  ttim(i) is the array which contains the center of
c    mass time information.  Positive values correspond to a particle
c    hitting the top half of the detector first, negative values to
c    a fragment hitting the bottom half first.  This used to be an
c    absolute value quantity when theta was calculated only from
c    0 to 90.  To correspond with tdaf definitions, a vector is drawn
c    pointing from the heavy to the light mass.  This vector is
c    compared with the detector based coordinate axes to determine
the
c    values of theta and phi.
c         phcmc is the cm azimuthal scattering angle from 0 to 180
c         degrees found
c         by taking the inverse tangent of the y component of the
line
c         from the lower
c         fragment to the upper fragment divided by the x component
of
c         that same line
c
c  WARNING:  Leave the cases in this order:  horizontal:1,3,2,4
c  and vertical 1,2,3,4 so that equal mass channels are properly
c  treated.  For equal mass, heavy = light, and only two cases are
```

```
c  unique.  The current ordering makes sure that the first two unique
c  cases encountered are one with theta = 0 - 90, one with 90 - 180.
c
c  Note: if ipolang = 0 (horizontal polarization), for equal mass
c  the particle that hits the top of the detector is defined
c  as the "light" particle.
c
   thcmc = atan(rul(4)/z_corr(i)) * degree ! theta and phi in
   phcmc = atan(rul(1)/rul(2)) * degree ! 'detector frame' coords
c
c        case 1:  light frag, first, on top (imtu small and pos.)
   if((imtu(i).le.(radical_mass/2)).and.(imtu(i).ge.0.)) then
c    thcmc is OK the way it is   0 < theta < 90
     if(rul(1).lt.0.0) then      ! if not, 0 < phi < 90
       phcmc = phcmc + 360 ! 270 < phi < 360
     endif
c
c        case3:  light frag, second, on top (imtu small and neg.)
   elseif((-imtu(i).le.(radical_mass/2)).and.(imtu(i).le.0.)) then
     thcmc = thcmc + 180   ! 90 < theta < 180
     if(rul(1).lt.0.0) then      ! if not, 0 < phi < 90
       phcmc = phcmc + 360 ! 270 < phi < 360
     endif
c

c        case2:  light frag, first, on bottom (imtu large and neg.)
   elseif(-imtu(i).gt.(radical_mass/2)) then
      thcmc = -1.0 * thcmc ! 0 < theta < 90
      phcmc = phcmc + 180  ! 90 < phi < 270
c
c        case4:  light frag, second, on bottom (imtu large and pos.)
   elseif(imtu(i).gt.(radical_mass/2)) then
     thcmc = 180 - thcmc   ! 90 < theta < 180
     phcmc = phcmc + 180   ! 90 < phi < 270
c
   else  !if this else block is reached, something is really wrong.
     write(6,*) 'No category found in theta/phi conditional in
psdclce!'
     write(6,*)'Check your mass chan. of interest, among other
things.'
     stop
   endif !from imtu.compared to.radical_mass/2 block
c
   if (thcmc.lt.0) then
     write(6,*) 'thcmc less than zero?!'
     write(6,*) 'i=',i,' theta=',thcmc,' phi=',phcmc,' imtu=',imtu(i)
     @ ,' z_c=',z_corr(i),' ttim=',ttim(i)
   end if
c
c
   else       ! from if(ipolang.eq.0) statement; here, ipolang = 90
     if (ipolang.ne.90) stop 'ipolang should be 0 or 90'
c
c    thcmc is the cm polar scattering angle from 0 to 180 degrees and
is
c    found by adding in quadrature v0tau and x divided by y.  The
c    arctangent of this quotient is theta.  Remember that our
c    observables are x, y, and tau in the coordinates Bob has
c    always used.  Without considerable recoding, this must remain
c    as the basic coordinate system.
c
c  For equal mass:  As in the ipolang=0 block, only the first two
c  cases apply; 0 < theta < 180, but phi is in the first or fourth
c  quadrants only.
```

```
c
c  Note: if ipolang = 90 (vertical polarization), for equal mass
c  the particle that hits first, whether on top or bottom
c       of the detector, is defined as the "light" particle.
c
   thcmc = atan( sqrt( z_corr(i)*z_corr(i) + rul(1)*rul(1) )
     @          / rul(2) ) * degree
   phcmc = atan( rul(1)/z_corr(i) ) * degree
c
c        case 1:  light frag, first, on top (imtu small and pos.)
c
   if((imtu(i).le.(radical_mass/2)).and.(imtu(i).ge.0.)) then
c    thcmc is OK here
     if(rul(1).ge.0.0) then
       phcmc = 360 - phcmc ! 270 < phi < 360
     else
       phcmc = -phcmc      ! 0 < phi < 90
     endif
c
c        case2:  light frag, first, on bottom (imtu large and neg.)
c
   elseif(-imtu(i).ge.(radical_mass/2)) then
     thcmc = 180 - thcmc
     if(rul(1).ge.0.0) then
        phcmc = -phcmc     ! 0 < phi < 90
     else
        phcmc = 360 - phcmc ! 270 < phi < 360
     endif
c
c        case3:  light frag, second, on top (imtu small and neg.)
c
   elseif((-imtu(i).lt.(radical_mass/2)).and.(imtu(i).le.0.)) then
c    thcmc is OK here
     phcmc = 180 - phcmc   ! 90 < phi < 270
c
c        case4:  light frag, second, on bottom (imtu large and pos.)
c
   elseif(imtu(i).gt.(radical_mass/2)) then
          thcmc = 180 - thcmc
     phcmc = 180 - phcmc   ! 90 < phi < 270
c
   else
     write(6,*)'psdclce failed to assign theta and phi to '
     write(6,*)'data point',i,'.  imtu(i)=',imtu(i)
     stop
c
   end if  ! from imtu.compared to.radical_mass conditionals
   if (thcmc.lt.0) then
     write(6,*) 'thcmc less than zero?!'
     write(6,*) 'i=',i,' theta=',thcmc,' phi=',phcmc,' imtu=',imtu(i)
     @ ,' z_c=',z_corr(i),' ttim=',ttim(i)
   end if
      endif      !(from if(ipolang.eq.0) statement)
c*********************End angle determination if
block********************
c
c
c        bin the angular distribution
c
       call xbinnr2(thmin,thmax,dsth,nth,thcmc,ithc,iincth)
       call xbinnr2(thmin,2e0*thmax,dsth,2*nth,phcmc,ithpc,iincph)
       pthcmc(ithc) = pthcmc(ithc) + iincth  ! fort.8
       pphcmc(ithpc) = pphcmc(ithpc) + iincph      ! fort.8
c*************************
```

```
c  bin pimtu, the mass distribution on the upper half

   if (iforce.eq.1) then
     lower_mass = radical_mass - abs(imtu(i))
     call
binmass(gmin,gmax,dsm,nbinm,real(abs(imtu(i))),imass,imassinc)
   else
     lower_mass = radical_mass - upper_mass(i)
     call binmass(gmin,gmax,dsm,nbinm,upper_mass(i),imass,imassinc)
   end if
   pimtu(imass) = pimtu(imass) + imassinc    !fort.8
c*************************
c  determine heavy and light particle tofs
c  bin the y and r positions relative to the nominal x0,y0 center
c       for the light and heavy particles
c  Note that for equal mass, the following if statement always
c  finds that the upper mass is light.  This is exactly consistent
c  with the earlier definitions of ipolang = 0.  But for ipolang =
90,
c  it is not correct.  For now I will just leave it incorrect
c  for the ipolang = 90 case.
c
c  As of 2/9/95, the P(y) and P(r) plots will now contain negative
c  values on the x axis.  For P(y) this means negative values of
c  y.  For P(r), a "negative" radius means the probability of
c  receiving a given radius on the lower half, while a positive
c  radius means upper half.
c
       if(lower_mass.lt.(radical_mass/2.0)) then ! lower mass is
light part.
        rhv  = sqrt( (xu(i)-x0)**2 + (yu(i)-y0)**2 )
        rlt  = -1.*sqrt( (xl(i)-x0)**2 + (yl(i)-y0)**2 )
        emhv = radical_mass - lower_mass
        emlt = lower_mass
        tltlab = (dpsd/vbeam)*1e06
     @         +(emhv/radical_mass)*ttim(i)*1e-03 ! microseconds
        thvlab = (dpsd/vbeam)*1e06
     @         -(emlt/radical_mass)*ttim(i)*1e-03

c
        call xbinnr2(-ymx,ymx,dyp,nproj,(yl(i)-y0),iylt,iinc)
        call xbinnr2(-ymx,ymx,dyp,nproj,(yu(i)-y0),iyhv,iinc)
       else    ! upper fragment is the light fragment
        rhv  = -1.*sqrt( (xl(i)-x0)**2 + (yl(i)-y0)**2 )
        rlt  = sqrt( (xu(i)-x0)**2 + (yu(i)-y0)**2 )
   if( ( lower_mass.eq.(radical_mass/2.0) ).and.
     @         (imtu(i).ge.0) ) then !only for equal mass case
    rlt = -1.*rlt !forward scat. light => negative radius
    rhv = -1.*rhv !backward scat. heavy => positive radius
   endif
        emhv = lower_mass
        emlt = radical_mass - lower_mass
        tltlab = (dpsd/vbeam)*1e06
     @         -(emhv/radical_mass)*ttim(i)*1e-03
        thvlab = (dpsd/vbeam)*1e06
     @         +(emlt/radical_mass)*ttim(i)*1e-03

        call xbinnr2(-ymx,ymx,dyp,nproj,(yu(i)-y0),iylt,iinc)
        call xbinnr2(-ymx,ymx,dyp,nproj,(yl(i)-y0),iyhv,iinc)
       endif
c
       call xbinnr2(-ymx,ymx,drp,nproj,rlt,irlt,iinc)
       call xbinnr2(-ymx,ymx,drp,nproj,rhv,irhv,iinc)
c*************************
```

```
c
c       bin heavy and light masses
c
       call binmass(gmin,gmax,dsm,nbinm,emhv,imhvc,iinch)
       call binmass(gmin,gmax,dsm,nbinm,emlt,imltc,iincl)
c
c       find locations in time-of-arrival arrays
c
       call xbinnr2(tmin,tmax,dwell,nt,tltlab,itltlab,iinclt)
       call xbinnr2(tmin,tmax,dwell,nt,thvlab,ithvlab,iincht)
       call xbinnr2(tcomin,tcomax,dwellbc,ntbc,ttim(i),itco,iinctc)
     if(ttim(i).ge.0.) then
      posnum = posnum + 1.
     else
      negnum = negnum + 1
     endif
c
c       now accumulate these calculated results in the appropriate
arrays
c       including statistics only for the position arrays at this
time
c
       ptltlab(itltlab)    = iinclt + ptltlab(itltlab)     !fort.8
       pthvlab(ithvlab)    = iincht + pthvlab(ithvlab)     !fort.8
       ptcolab(itco)       = iinctc + ptcolab(itco) !fort.8
       pmcc(imltc)         = iincl + pmcc(imltc)           !fort.8
       pmcc(imhvc)         = iinch + pmcc(imhvc)           !fort.8
       pylt(iylt)          = iinc + pylt(iylt)             !fort.8
       pyhv(iyhv)          = iinc + pyhv(iyhv)             !fort.8
       prlt(irlt)          = iinc + prlt(irlt)             !fort.8
       prhv(irhv)          = iinc + prhv(irhv)             !fort.8
       call sigmap ( tltlab , tltavg  )
       call sigmap ( thvlab , thvavg  )
       call sigmap ( abs(ttim(i)), tcoavg  )
       call sigmap ( emlt   , emltavg )
       call sigmap ( emhv   , emhvavg )
c
c  newest, finally accurate equation, with imtu(i)
c  (best when mass channel is known and iforce = 1)
c
   lower_mass = radical_mass - abs(imtu(i))
       energy(i) = ebeam*( (abs(imtu(i))*lower_mass)/radical_mass**2
)*
     >   ( rul(4)*rul(4) + z_corr(i)*z_corr(i) ) /
     >   ( dpsd*dpsd + 2.0*dpsd*z_corr(i) *
     >   ( (abs(imtu(i))-lower_mass) / radical_mass ) +
     >   ( ((abs(imtu(i))-lower_mass) / radical_mass)**2 *
     >   z_corr(i)*z_corr(i) ) )
c
c  newest, finally accurate equation, with upper_mass(i)
c  less accurate when masses are, equivalent otherwise
c
   lower_mass = radical_mass - upper_mass(i)
       etrca = ebeam*( (upper_mass(i)*lower_mass) / radical_mass**2
) *
     >   ( rul(4)*rul(4) + z_corr(i)*z_corr(i) ) /
     >   ( dpsd*dpsd + 2.0*dpsd*z_corr(i) *
     >   ( (upper_mass(i)-lower_mass) / radical_mass ) +
     >   ( ((upper_mass(i)-lower_mass) / radical_mass)**2 *
     >   z_corr(i)*z_corr(i) ) )
c
       call xbinnr2(emin,emax,dse,nbine,energy(i),ietrc,iincec)
       call xbinnr2(emin,emax,dse,nbine,etrca,ietrca,iinceca)
       ethetc(ietrc,ithc) = ethetc(ietrc,ithc) + iincth*iincec
```

```
        ephetc(ietrc,ithpc) = ephetc(ietrc,ithpc) + iincph*iincec
        ptecbin(ietrc) = ptecbin(ietrc) + iincec   !fort.8
        ptecabin(ietrca) = ptecabin(ietrca) + iinceca    !fort.8
c
        if(debug4) then
         write(*,*)
         write(*,*)' Values calculated from exp. observables:'
         write(*,*)' rul(4), mutual sepn = ',rul(4),' mm '
         write(*,*)' coincidence time    = ',ttim(i),' nsec '
         write(*,*)' mass of particle 2 = ',upper_mass(i),' amu '
         write(*,*)' mass of particle 3 = ',lower_mass,' amu '
         write(*,*)' parent mass        = ',radical_mass,' amu '
         write(*,*)' theta cm           = ',thcmc,' deg. '
         write(*,*)' phi cm             = ',phcmc,' deg. '
         write(*,*)' e trans, cm.       = ',energy(i),' eV '
         write(*,*)' e trans, cm. appx  = ',etrca,' eV '
         write(*,*)
         write(*,*)'Enter 0 to leave debug mode. (1)'
         read(*,*) idebug4
         if(idebug4.eq.0) debug4 = .false.
        endif
        call sigmap ( etrca , ecmcaavg )
        call sigmap ( energy(i)  , ecmcavg  )
c
7000     continue ! finally the loop over coincident events ends (i
index)
c
   posfrac = 100*posnum / (posnum + negnum)
   negfrac = 100*negnum / (posnum + negnum)
   write(6,2500)posfrac
   write(6,2501)negfrac
2500     format('positive delta t is ',f4.1,'% of total')
2501     format('negative delta t is ',f4.1,'% of total')

c*******The next section will allow mass spectra for a section of the
c*******KER.  This is useful if you have two well-separated features
in
c*******the KER and you believe they might be two different product
channels.

   if(iforce.eq.0) then  !only allow energy windows for multiple chan
    write(*,*)
    write(*,*)'Enter 0 to examine data over an energy range. (1)'
    read(*,*)ichoice
    if(ichoice.eq.0) then
     do i = 1,nbinm  !save original pimtu array
      pimtuold(i) = pimtu(i)
     enddo
     write(*,*)
8000       write(*,*)'Enter the minimum and maximum energies to
retain'
     read(*,*)ewinmin,ewinmax

          do i = 1,nbinm          !reset pimtu to zero
           pimtu(i) = 0
          enddo
     do i = 1,ico
      if(ivall(i).eq.1)then            !only look at good data
       if((energy(i).ge.ewinmin).and.(energy(i).le.ewinmax)) then
         call binmass(gmin,gmax,dsm,nbinm,upper_mass(i),
     @                imass,imassinc)
        pimtu(imass) = pimtu(imass) + imassinc
       endif   !if (energy withing range)
      endif          !if (ivall.eq.1)
```

```
      enddo            !loop over all events

      write(6,*)'Do you want to save the mass spec to file? [y/n]'
      read(5,*) massch
      if(massch.eq.'y') then
       write(6,*)'Enter filename'
       read(5,*) massfile
       open(unit = 25, file =massfile)
       do i = 1,nbinm
        write(25,*) float(i)/float(nbinm)*radical_mass,pimtu(i) +
      @                 pimtu(nbinm + 1 - i)
       enddo
      endif
      call plotmass

cccccccccccccc
       write(*,*)
       write(*,*)'Differences in each mass bin:'
      do i = 1,nbinm
       if(pimtu(i).ne.pimtuold(i)) then
        write(*,*)'pimtu(',i,') =',pimtu(i),'  pimtuold(',i,') = ',
      @          pimtuold(i)
       endif
      enddo
cccccccccccccccccccc

      write(*,*)
      write(*,*)'Enter 0 to use another energy range. (1)'
      read(*,*)ichoice
      if(ichoice.eq.0) goto 8000
    endif       !if(ichoice.eq.0)
   endif !if(iforce.eq.0)


   write(*,*)
   write(*,*)'Enter 0 to plot up final image. (1)'
   read(*,*) iplot
   if(iplot.eq.0) call rawplot(0)

   return
   end
```

rawplot.f

```
   subroutine rawplot(ivalid)
c
c        unix version 911114 REC
c
c  this subroutine makes a scatter plot of the image
c
c        if ivalid = 0, plot true coincidences
c        if ivalid = 1, plot all data
c        if ivalid = 2, plot non-coincidences
c        if ivalid = 3, plot only points just discriminated against
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n4 = 200    )
```

```
      character*40 title
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                   dsy,nx,ny
c
      common / fov / xfov(n4),yfov(n4),nfov
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                 ivaltmp(n1)
c
      common / tacinf / tslope,tint,tdelay,
     >                  ttim(n1),tactrue(n3),tacval(n3),ntac
c
      common / words / title

      data pi / 3.14159625 /
c
      xmax = nint(sxmax)
      xmin = nint(sxmin)
      ymax = nint(symax)
      ymin = nint(symin)
      call opngks
      call setusv('PB',2)
c  draw a boundary around the edge of the plotter frame
c     call bndary
c  suppress the frame advance
      call agseti('FRAME.',2)
c  set limits for axes
      call agseti('X/MAXIMUM.',xmax)
      call agseti('X/MINIMUM.',xmin)
      call agseti('Y/MAXIMUM.',ymax)
      call agseti('Y/MINIMUM.',ymin)
c  suppress the drawing of curves by the EZ... routines
      call agseti('SET.',-1)
c  draw the background using EZXY
      call ezxy (xfov,yfov,nfov,' ')
c  put a plus sign at each of the x-y positions
      call points (xfov,yfov,nfov,-1,0)
c  put a plus sign at the center of the image
      call points (x0,y0,1,-2,0)
c       draw the blocking strip
      call line(xmin,blocky,xmax,blocky)
      call line(xmin,-blocky,xmax,-blocky)
c
      call setusv('MS',1000)
c ico is number of post-phd events : npp is num. pnts. to plt. (2500)
      np = min(ico,npp)
      if(ivalid.ne.1) then !to plot all points, move below
       ictl = 1
       ttl = 0e0
       ttu = 1000e0
       write(*,*)
       write(*,*)'Enter 0 to override 0-1000 ns coinc. time limits.
(1)'
       read(*,*)ictl
       if(ictl.eq.0) then
        write(*,*)
        write(*,*)'Enter lower and upper Coinc. Time limits in nsec:'
        read(*,*)ttl,ttu
       endif
       icond = 1
       if(ivalid.eq.2) icond = 0
```

```
      do i = 1,np
       if(ivalid.eq.0.or.ivalid.eq.2) then
        if(ival1(i).eq.icond) then
         if( (abs(ttim(i)).ge.ttl).and.(abs(ttim(i)).le.ttu) ) then
          call point (xu(i),yu(i))
          call point (xl(i),yl(i))
         endif
        endif
       else !(ivalid.eq.0.or.ivalid.eq.2) here plot recent discr. evnts.
c              this only plots from first 2500 points.  Could give
low S/N
        if(ivaltmp(i).eq.1) then
         if( (abs(ttim(i)).ge.ttl).and.(abs(ttim(i)).le.ttu) ) then
          call point (xu(i),yu(i))
          call point (xl(i),yl(i))
         endif
        endif
       endif !(ivalid.eq.0.or.ivalid.eq.2)
      end do

          else !(ivalid.ne.1) !plot all points, regardless of ival
      do i = 1,np
       call point (xu(i),yu(i))
       call point (xl(i),yl(i))
      end do
          endif
c  close gks
         call clsgks
         if(ivalid.eq.1) then
          ierr = system( 'mv gmeta raw.img' )
          if(iwkst.eq.0) ierr = system( 'ictrans -d sunview raw.img' )
          if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 raw.img' )
          if(iwkst.eq.2) ierr = system( 'ictrans -d X11 raw.img' )
         endif
         if(ivalid.eq.0) then
          ierr = system( 'mv gmeta coincs.img' )
          if(iwkst.eq.0) ierr = system( 'ictrans -d sunview coincs.img'
)
          if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 coincs.img' )
          if(iwkst.eq.2) ierr = system( 'ictrans -d X11 coincs.img' )
         endif
         if(ivalid.eq.2) then
          ierr = system( 'mv gmeta ncoincs.img' )
          if(iwkst.eq.0) ierr = system( 'ictrans -d sunview
ncoincs.img')
          if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 ncoincs.img' )
          if(iwkst.eq.2) ierr = system( 'ictrans -d X11 ncoincs.img' )
         endif
         if(ivalid.eq.3) then
          ierr = system( 'mv gmeta discncs.img' )
          if(iwkst.eq.0) ierr = system( 'ictrans -d sunview discncs.img'
)
          if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 discncs.img' )
          if(iwkst.eq.2) ierr = system( 'ictrans -d X11 discncs.img' )
         endif
c
      return
      end
```

savdfplt.f

```
*deck savdfplt.f
```

```
      subroutine savdfplt
c
c        unix version 930119 DRC
c
c  this subroutine bins and saves the detector face valid
c  data array for later plotting on graphics machines (or our
c  future color monitor!)
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 300    )
      parameter ( n3 = 50     )
      parameter ( n4 = 200    )
c
      common / display / idfimg(n2,n2),sxmin,sxmax,symin,symax,dsx,
     >                   dsy,nx,ny
c
      common / fov / xfov(n4),yfov(n4),nfov
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                 ivaltmp(n1)
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
c  using the x,y data in common / image /, bin into a position
c
      xref = ( sxmax - sxmin )/2e0
      yref = ( symax - symin )/2e0
      xbin = ( sxmax - sxmin )/float(nx)
      ybin = ( symax - symin )/float(ny)
c
c
c  put valid events (ivall=1) into the position array
c
      do i = 1,n2
       do j = 1,n2
        idfimg(i,j) = 0
       enddo
      enddo
      do i = 1,ico
       if(ivall(i).eq.1) then
        if(xu(i).le.sxmax.and.xu(i).ge.sxmin) then
         if(yu(i).le.symax.and.yu(i).ge.0) then
          ixubin = int( (xu(i) + xref ) /xbin ) + 1
          iyubin = int( (yu(i) + yref ) /ybin ) + 1
          idfimg(ixubin,iyubin) = idfimg(ixubin,iyubin) + 1
         endif
        endif
        if(xl(i).le.sxmax.and.xl(i).ge.sxmin) then
         if(yl(i).le.0.and.yl(i).ge.symin) then
          ixlbin = int( (xl(i) + xref ) /xbin ) + 1
          iylbin = int( (yl(i) + yref ) /ybin ) + 1
          idfimg(ixlbin,iylbin) = idfimg(ixlbin,iylbin) + 1
         endif
        endif
       endif
      enddo
c
c     now save the detector face image array
c
      write(*,*)
      write(*,*)'Writing detector face array ...'
      open(unit=22,file='savdfplt.out')
```

```
      do i = 1,nx
       do j = 1,ny,6
        write(22,1000) idfimg(i,j),idfimg(i,j+1),idfimg(i,j+2),
     >                 idfimg(i,j+3),idfimg(i,j+4),idfimg(i,j+5)
       enddo
      enddo
      close(22)
 1000 format(6(i10))
      return
      end
```

srvrtn.f

```
*deck srvrtn.f
c        unix version 911114 REC
c
c  service routines for the cmlwsa program
c  Modified 12/8/92 by DLO by addition of wrtarr2 routine
c  which writes the KER in a one column file
c
c
*deck Asets
      Subroutine Asets(nx,ny,axy,s)
c
c   __________        Asets __________
c
c>>>>> ASets; sets all elements of a 2d array equal to a scalar
c
c        last modified 900818 REC
c
c  nx, ny are the dimensions of the array
c  axy is the 2-d array
c  s is the scalar
c-----------------------------------------------------------------
c
      integer nx,ny
      dimension axy(nx,ny)
      do 2 i = 1,nx
       do 1 j = 1,ny
        axy(i,j) = s
 1     continue
 2    continue
      return
      end
*deck fwhm
      Subroutine fwhm(f,x,px,n,xh,xpk,xl,fw)
c
c       find peak and FW at f maximum of distribution px
c
      dimension x(n),px(n)
c
      pxmax = 0e0
      do 10 i = 1,n
       if(px(i).gt.pxmax) then
        pxmax = px(i)
        xpk   = x(i)
        imax  = i
       endif
 10   continue
      pxf = pxmax*f
c           look for low FWHM
      do 20 i = 1,imax
```

```
        if( px(i).le.pxf ) then
         xl     = x(i)
         ilfmax = i
        endif
 20    continue
c            look for high FWHM
       do 30 i = imax,n
        if( px(i).ge.pxf ) then
         xh      = x(i)
         ihfmax  = i
        endif
 30    continue
       fw = xh - xl
       return
       end
*deck grs1
       subroutine grs1 ( n, a, b, v )
c
c------------------------------- grs1 ------------------------------
----------
c
c === Latest Revision:     | 890226 at 11:39:26 |
c
c      this subroutine generates a real scaling vector v(i)
c         such that v(i) = a + b*(i-1)
c
c      the routine is optimized so that
c          the additional loop overhead for vectorization is
c          avoided for n.lt.9
c
c          as compared to generating the appropriate vector in the
c            calling loop, this routine breaks even at n=12 (calling
overhead)
c
c-----------------------------------------------------------------
----------
c
       implicit real*4 ( a - h, o - z )
c
       dimension v(n)
       common / vector / fpn(64), igr(64)
       save   / vector /
c
c--------------------------- execution begins here ------------------
---------
c
       if ( n .lt. 1 ) RETURN
c
       IF ( n .gt. 8 ) THEN
         ij = 0
         do 1 j = 1, n, 64
           jo = j - 1
           ni = min0(j+63,n) - jo
           ao = a + b*(64*ij)
           ij = ij + 1
           do 2 i = 1, ni
             v(i+jo) = ao + b*fpn(i)
 2         continue
 1       continue
       ELSE
         neven = mod(n,2)
         do 101 i = 1+neven, n-1, 2
           v(i) = a + b*(i-1)
      v(i+1) = a + b*i
```

```
 101    continue
        if ( neven .eq. 0 ) RETURN
        v(1) = a
      END IF
c
      return
      end
*deck numgen
      subroutine numgen
c
c------------------------------ numgen ------------------------------
----------
c
c >>> numgen   -- fills up common blocks with integer and floating
consts
c
c === Latest Revision:       | 890226 at 11:35:12 |  combine loops
c
c-----------------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
      common / vector / fpn(64), igr(64)
      save /vector/
c
c---------------------------- execution begins here ------------------
----------
c
      do 1 i = 1, 64
        igr(i) = i - 1
        fpn(i) = igr(i)
 1    continue
c
      return
      end
*deck sigmac
      subroutine sigmac ( xa, avg, sdev )
c
c------------------------------ sigmac ------------------------------
----------
c
c >>> sigmac   --
c
c-----------------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
      dimension xa(3)
c
c------------------------- execution begins here --------------------
----------
c
      xnit = 1e0
      if ( xa(3) .gt. 0e0 ) xnit = xnit/xa(3)
      avg  = xnit*xa(1)
      sdev = xnit*xa(2)
      sdev = sqrt(max(0e0, sdev-avg**2))
c
      return
      end
*deck sigmap
      subroutine sigmap ( x, xa )
```

```
c
c---------------------------- sigmap -------------------------------
----------
c
c >>> sigmap   --
c
c-------------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
      dimension xa(3)
c
c---------------------------- execution begins here ----------------
----------
c
      if ( xa(3).gt.0e0 ) go to 1
      xa(1) = 0e0
      xa(2) = 0e0
      xa(3) = 0e0
c
 1    continue
      xa(1) = xa(1) + x
      xa(2) = xa(2) + x**2
      xa(3) = xa(3) + 1e0
c
      return
      end
*deck vnorm
      function vnorm ( a, n )
c
c------------------------------- vnorm ----------------------------
----------
c
c >>> vnorm    -- computes the length of the input vector
c
c-------------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
      dimension a(n)
c
c------------------------- execution begins here -------------------
----------
c
      sum = 0e0
      do 1 i = 1, n
         sum = sum + a(i)**2
 1    continue
c
      vnorm = sqrt(sum)
c
      return
      end
*deck vsets
      subroutine vsets ( n, a, ia, s )
c
c------------------------------- vsets ----------------------------
----------
c
c >>> vsets    -- vector set to a (real) scalar
c
```

```
c-----------------------------------------------------------------
--------
c
      integer  n
      integer  ia
c
      real*4 a(n)
      real*4 s
c
c------------------------- execution begins here ------------------
----------
c
      IF ( ia .ne. 1 ) THEN
c
        ix = 1
        do 1 i = 1, n
          a(ix) = s
          ix = ix + ia
 1      continue
c
      ELSE
c
        do 2 i = 1, n
          a(i) = s
 2      continue
c
      END IF
c
      return
      end
*deck wrtarr
      subroutine wrtarr (iunit,a,n)
c
c--------------------------- wrtarr --------------------------------
----------
c
c >>> wrtarr   -- writes out an array to iunit
c
c-----------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
      dimension a(n)
c
 1000 format ( 1x,i8)
 1001 format ( 1x,6g13.5 )
c
      write ( iunit, 1000 ) n
      write ( iunit, 1001 ) a
c
      return
      end
*deck wrtarr2
       subroutine wrtarr2 (iunit,a,n)
c
c--------------------------- wrtarr2-------------------------------
----------
c
c >>> wrtarr2   -- writes out an array to iunit
c
c-----------------------------------------------------------------
----------
c
```

```
      implicit real*4 ( a - h, o - z )
c
      dimension a(n)
c
c1000   format ( 1x,i8)
1001   format ( 1x,g13.5 )
c
c      write ( iunit, 1000 ) n
       write ( iunit, 1001 ) a
c
       return
       end
*deck wrtla
      subroutine wrtla ( iunit, a1, da, n )
c
c---------------------------- wrtla ---------------------------------
----------
c
c >>> wrtla    -- writes out stuff
c
c-------------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
c
c----------------------------- execution begins here ----------------
----------
c
      write ( iunit, 1000 ) n
      write ( iunit, 1001 ) a1, da
 1000 format ( 1x, i7)
 1001 format ( 1x, 6g13.5 )
c
      return
      end
*deck xbinnr
      subroutine xbinnr ( xmin,xmax,dx,nx,x,weight,dis )
c
c-------------------------- xbinnr ---------------------------------
----------
c
c >>> xbinnr   -- accumulates counts into bins
c
c     accumulates counts into bins,
c         ibin=1 implies too low
c         ibin=nx implies too high
c
c       ibin=2  ==> x between xmin and xmin+dx
c       ibin=3  ==> x between xmin+dx and xmin+2*dx, etc.
c       ibin=nx-1 ==> x between xmax-dx and xmax
c
c-------------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
      dimension dis(nx)
c
c------------------------ execution begins here ---------------------
----------
c
      wgt = weight
      ibin = int( (x-xmin)/dx ) + 1
```

```
      if ( x .lt. xmin ) then
       ibin = 1
       wgt = 0e0
      endif
      if ( x .gt. xmax ) then
       ibin = nx
       wgt = 0e0
      endif
      dis(ibin) = dis(ibin) + wgt
c
      return
      end
*deck xbinnr2
      subroutine xbinnr2 ( xmin,xmax,dx,nx,x,ibin,iinc)
c
c------------------------- xbinnr2--------------------------------
----------
c
c >>> xbinnr2  -- determines bin numbers
c  this routine is used for binning with truncation (uses int)
c  e.g.  xmin = 0, xmax = 10, dx = 1, nx = 10 will put all
c  numbers from [0,1) in the first bin, [1,2) in second bin... ,
c  up to [9,10] in the 10th bin.  x = 10 used to cause ibin = 11,
c  which is not a good idea.  Now it is included as noted above.
c
c
c-----------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
c----------------------- execution begins here --------------------
----------
c
      iinc = 1
      ibin = int( (x-xmin)/dx ) + 1
      if ( x .lt. xmin ) then
       ibin = 1
       iinc = 0
      endif
      if ( x .gt. xmax ) then
       ibin = nx
       iinc = 0
      endif
      if ( x .eq. xmax ) then !added 2/9/95 to handle x = xmax (DLO)
       ibin = nx
       iinc = 1
      endif

      return
      end
c
*deck bin
      subroutine bin (xmin,xmax,dx,nx,x,ibin,iinc)
c
c------------------------- bin-------------------------------------
------
c
c >>> bin      -- determines bin numbers, binning to nearest bin
c        #, ie uses nint, which rounds 1.5 up to 2, and
c        rounds 2.4999 down to 2
c
c     accumulates counts into bins,
c         ibin=1 implies too low, iinc set to 0
```

```
c          ibin=nx implies too high, iinc set to 0
c
c        ibin=1  ==> x between xmin-dx/2 and xmin+dx/2, iinc = 0
c        ibin=1  ==> x between xmin+dx and xmin+3*dx, etc.
c        ibin=nx ==> x between xmax-dx and xmax, iinc = 0
c
c
c-----------------------------------------------------------------
----------
c
      implicit real*4 ( a - h, o - z )
c
c------------------------ execution begins here ---------------------
----------
c
      iinc = 1
      ibin = nint((x-xmin)/dx)
      if (x.lt.(xmin+(0.5*dx))) then
       ibin = 1
       iinc = 0
      endif
      if (x.gt.xmax) then
       ibin = nx
       iinc = 0
      endif
      return
      end
c
*deck binmass
      subroutine binmass (xmin,xmax,dx,nx,x,ibin,iinc)
c
c-------------------------- binmass---------------------------------
----------
c  DLO 1/14/95
c >>> binmass  -- determines bin numbers for masses, binning to
nearest bin
c         #, ie uses nint, which rounds 1.5 up to 2, and
c         rounds 2.4999 down to 2
c  remember that for mass binning, xmin = 0.5, xmax = radmass - 0.5
c  and nbinm = radmass - 1  There is no 0th bin or radmassth bin
c  and the bin width, dx, is always equal to 1 mass unit.  Therefore
c  this routine is very simple
c
c     accumulates counts into bins,
c         for x < 0.5 or x.ge.(radmass - 0.5),  iinc = 0, and nothing
c    is incremented in the calling routine.  For all other values,
c    iinc = 1, and pimtu(ibin) = pimtu(ibin) + 1
c
c  note that xmin and nx do not appear, but I leave them in anyway
c-----------------------------------------------------------------
---------
c
      implicit real*4 ( a - h, o - z )
c
c------------------------ execution begins here ---------------------
----------
c
      ibin = nint(x/dx)
      iinc = 1
   if(ibin.eq.0) iinc = 0
   if(ibin.eq.nint(xmax)) iinc = 0
      return
      end
```

tactime.f

```
*deck tactime
      Subroutine tactime(ivalid)
c
c        unix version 911114 REC
c
c  This subroutine plots tac-adc data
c  if ivalid = 0, just plot the valid events
c  if ivalid = 3, plot the recently discriminated events
c  It plots ttim(i), that is, actual signed coincidence time deltaT
c
c
      implicit real*4 (a-h,o-z)
c
      parameter ( n1 = 100000 )
      parameter ( n2 = 400 )
      parameter ( n3 = 50 )
   character*40 title
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                 ivaltmp(n1)
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
      common / tacinf / tslope,tint,tdelay,
     >                  ttim(n1),tactrue(n3),tacval(n3),ntac
c
      common / psdhist / emin,emax,dse,
     >                   thmin,thmax,dsth,umin,umax,du,gmin,gmax,dsm,
     >                   tmin,tmax,dwell,tcomin,tcomax,dwellbc,
     >                   nbine,nbinm,nt,ntbc,nth,nproj,ymx,
     >                   ypmin,rpmin,dyp,drp
c
      common / words / title

      dimension ptac(n2),tac(n2)
      dimension tc(n3),tcv(n3)
   real*4 posnum,negnum,posfrac,negfrac
c
c  write(6,*)'iplttac = ',iplttac
      if(ivalid.eq.0.or.ivalid.eq.3) go to 222
      write(*,*)
      write(*,*)'Enter 0 if you want to plot TAC spectrum. (1)'
      read(*,*)iplttac
   if(iplttac.eq.0) then
    ivalid = 1 !skip to else statement
   else
    ivalid = 2 !don't plot anything, just get out of here
   endif
c  if(iplttac.ne.1) ivalid = 0
 222  continue
c  write(6,*)'You have reached tactime!, ivalid = ',ivalid
c  write(6,*)'iplttac = ',iplttac
c      if(ivalid.eq.0) iplttac = 0
c      if(iplttac.eq.0) then
       if( (ivalid.eq.0).or.(ivalid.eq.1).or.(ivalid.eq.3) ) then
c  if(ivalid.eq.0.or.ivalid.eq.3) then
       xmax = tcomax
       xmin = tcomin
   posnum = 0.
   negnum = 0.
   posfrac = 0.
```

```
   negfrac = 0.
       call vsets(ntbc,ptac,1,0e0)
c       dwellr = (xmax - xmin)/float( n2 - 1 ) !old hard-coded way
   dwellr = dwellbc
       call grs1( ntbc, xmin, dwellr, tac )
       if(ivalid.eq.0) then
        do i = 1,ico
         if(ival1(i).eq.1) then
          call xbinnr(xmin,xmax,dwellr,ntbc,ttim(i),1e0,ptac)
     if(ttim(i).ge.0.) then
      posnum = posnum + 1.
     else
      negnum = negnum + 1
     endif
         endif
        enddo
       elseif(ivalid.eq.3) then
   write(6,*)'WARNING:  This plot contains only ''bad'' data!'
        do i = 1,ico
         if(ivaltmp(i).eq.1) then !these are the points you've just
disc.
          call xbinnr(xmin,xmax,dwellr,ntbc,ttim(i),1e0,ptac)
     if(ttim(i).ge.0.) then
      posnum = posnum + 1.
     else
      negnum = negnum + 1
     endif
         endif
        enddo
       else
        do i = 1,ico
         call xbinnr(xmin,xmax,dwellr,ntbc,ttim(i),1e0,ptac)
     if(ttim(i).ge.0.) then
      posnum = posnum + 1.
     else
      negnum = negnum + 1
     endif
        enddo
       endif

   posfrac = 100*posnum / (posnum + negnum)
   negfrac = 100*negnum / (posnum + negnum)
   write(6,2500)posfrac
   write(6,2501)negfrac
2500     format('positive delta t is ',f4.1,'% of total')
2501     format('negative delta t is ',f4.1,'% of total')
       call opngks
       call setusv('PB',2)
c
       call agsetc ( 'LABEL/NAME.','L')
       call agseti ( 'LINE/NUMBER.',100)
       call agsetc ( 'LINE/TEXT.','No. of Counts$')
c
       call agsetc ( 'LABEL/NAME.','B')
       call agseti ( 'LINE/NUMBER.',-100)
       call agsetc ( 'LINE/TEXT.',
     @         'Coincidence Time (nsec) TAC Spectrum$')
c
       call agseti('X/MAXIMUM.',xmax)
       call agseti('X/MINIMUM.',xmin)
c      call agseti('Y/MINIMUM.',0e0)
c      call agseti('Y/MAXIMUM.',1.E36)
       call agseti('Y/LO.',0)
       call agseti('FRAME.',2)
```

```
c
       call agseti('SET.',1)
       call ezxy (tac,ptac,ntbc,title)
c  uncomment the following line if you want only points plotted
c      call points (tac,ptac,ntbc,-4,0)
       call clsgks
       if(ivalid.eq.1) then
        ierr = system( 'mv gmeta tac.img' )
        if(iwkst.eq.0) ierr = system( 'ictrans -d sunview tac.img' )
        if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 tac.img' )
        if(iwkst.eq.2) ierr = system( 'ictrans -d X11 tac.img' )
       else
        ierr = system( 'mv gmeta valtac.img' )
        if(iwkst.eq.0) ierr = system( 'ictrans -d sunview valtac.img'
)
        if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 valtac.img' )
        if(iwkst.eq.2) ierr = system( 'ictrans -d X11 valtac.img' )
       endif
      endif
      if(ivalid.eq.0) go to 333
c     write(*,*)'Enter 0 if you want to plot TAC calibration curve.
(1)'
c     read(*,*)ipltcal
      ipltcal = 1
      if(ipltcal.eq.0) then
       xmax = 1000e0
       xmin = 0e0
       dt = ( xmax - xmin )/49e0
       do i = 1,n3
        tc(i) = float(i-1)*dt
        tcv(i) = tslope*tc(i) + tint
       enddo
       call opngks
       call setusv('PB',2)
c
       call agsetc ( 'LABEL/NAME.','L')
       call agseti ( 'LINE/NUMBER.',100)
       call agsetc ( 'LINE/TEXT.','No. of Counts$')
c
       call agsetc ( 'LABEL/NAME.','B')
       call agseti ( 'LINE/NUMBER.',-100)
       call agsetc ( 'LINE/TEXT.','Coinc. Time (nsec)$')
c
       call agseti('X/MAXIMUM.',xmax)
       call agseti('X/MINIMUM.',xmin)
       call agseti('Y/MINIMUM.',0e0)
       call agseti('Y/MAXIMUM.',1.E36)
       call agseti('X/LO.',0)
       call agseti('Y/LO.',0)
       call agseti('FRAME.',2)
c
       call agseti('SET.',1)
       call ezxy (tc,tcv,n3,' TAC Calibration Curve$')
c  uncomment the following line if you want only points plotted
       call points (tactrue,tacval,ntac,-4,0)
       call clsgks
       ierr = system( 'mv gmeta taccal.img' )
       if(iwkst.eq.0) ierr = system( 'ictrans -d sunview taccal.img'
)
       if(iwkst.eq.1) ierr = system( 'ictrans -d t4010 taccal.img' )
       if(iwkst.eq.2) ierr = system( 'ictrans -d X11 taccal.img' )
      endif    ! end of ipltcal conditional
 333  continue
   if(ivalid.eq.2) ivalid = 1
```

```
c NOTE:  THIS IS THE WEIRDEST THING I HAVE EVER SEEN IN FORTRAN.  IF
YOU
c EXECUTE THE CALL "CALL TACTIME(1)" IN THE MAIN PROGRAM, AND IN
TACTIME
c YOU CHANGE IVALID TO 2, UPON RETURNING TO THE MAIN PROGRAM, FORTRAN
c GETS SCREWED UP BEYOND BELIEF.  TRY IT SOMETIME FOR KICKS.
c THEREFORE I HAVE INSERTED THE PRECEDING LINE TO KEEP FORTRAN HAPPY

      return
      end
*deck linfit
      Subroutine linfit(x,y,n,em,b)
c
c  this subroutine does a linear least squares fit to
c  a set of x,y data
c
      dimension x(n),y(n)
      sigxy  = 0e0
      sigx   = 0e0
      sigy   = 0e0
      sigxsq = 0e0
      do 10 i = 1,n
       sigxy  = sigxy + x(i)*y(i)
       sigx   = sigx  + x(i)
       sigy   = sigy  + y(i)
       sigxsq = sigxsq + x(i)**2
 10   continue
c
c  now, using the well-known least squares equations
c  calculate the slope and intercept.
c
      em = (float(n)*sigxy - sigx*sigy)/(float(n)*sigxsq -
     >      sigx**2)
      b  = ((sigy*sigxsq) - (sigxy*sigx))/(float(n)*sigxsq -
     >      sigx**2)
      return
      end
```

WSZTOXY.f

```
*deck wsztoxy
      Subroutine wsztoxy
c
c        unix version 920328 REC
c          newest version 921122 DRC
c        modified December 92 by David Leahy to include
c              corrections for strip nonlinearity
c        (dmczu is quadratic wedge coefficient 'qw', and
c        dmczl is quadratic strip coefficient 'qs')
c  ***** Modified to take 'factors' as they appear in the O2
c  ***** least squares fitting program xyzlsq!  DJL 11/93
c
c  this subroutine takes raw wedge, strip and z data and
c  converts it to x,y values for upper and lower halves of
c  the detector
c
c       if iusephd = 0, this algorithm will apply the pulse height
c       discrimination parameters found in phd.dat upon initial read-
in
c
      implicit real*4 (a-h,o-z)
```

```
c
      parameter ( n1 = 100000 )
      parameter ( n5 = 50    )
      parameter ( n6 = 8192  )
   character*20 gostring
   integer readpart,lowlim,range
c
      common / adcinf / vadc(n5),adcw1(n5),adcs1(n5),adcz1(n5),
     >                  adcw2(n5),adcs2(n5),adcz2(n5),
     >                  nadc,sw1,w1int,ss1,s1int,sz1,
     >                  z1int,sw2,w2int,ss2,s2int,sz2,z2int,
     >                  cw1,cs1,cz1,cw2,cs2,cz2
c Old Bob-style, commented out  DJL 11/93
c      common / algo / dmcwu,dmcsu,dmczu,dmcw1,dmcs1,dmcz1,
c     >                dacwu,dacsu,dacw1,dacs1,
c     >                uctws,uctwz,uctsw,uctsz,
c     >                lctws,lctwz,lctsw,lctsz
c
   common / algo / param(16)     ! new xyzlsq-style params DJL 11/93
c
      common / beam / ebeam,vbeam,dpsd,rbeammax,rbeammin
c
      common / discinf /
ispatdis,isdis,ihlf,yll,yul,isdisp,phi1,phi2,
     >                   iraddis,idisan,rlld,ruld,ttl,ttu,
     >                 nphddis,ncartdis,npoldis,nraddis,ncentdis
c
      common / image / xu(n1),yu(n1),xl(n1),yl(n1),ivall(n1),
     >                ivaltmp(n1)
c
      common / pha / puwpha(n6),puspha(n6),
     >               puzpha(n6),plwpha(n6),
     >               plspha(n6),plzpha(n6),
     >               tacpha(n6),ptacpha(n6),upha(n6),pupha(n6),
     >
lpha(n6),plpha(n6),nbinpha,phvpha(n6),pltpha(n6),
     >               iusephd
c
      common / phys / blocky,rfov,iwkst,x0,y0,ico,icf,npp,ipolang
c
   common / results / z_corr(n1),imtu(n1),upper_mass(n1),energy(n1)
c
      common / tacinf / tslope,tint,tdelay,
     >                  ttim(n1),tactrue(n5),tacval(n5),ntac
c
c  call linfit to get the linear fit to the TAC calibration curve
c  As of 12/93, this linfit doesn't include any correction for the
c  pulse generation time (around 45 ns) of the digital delay
generator.
c  Indeed, the variable t0 has been nixed entirely.  Furthermore, the
c  array ttim(i) now holds the actual signed coincidence times,
rather
c  than the 'raw coinc times', which were displaced by t0+tdelay.
   DJL
c
   call linfit(tactrue,tacval,ntac,tslope,tint)
   write(6,*)'linfit results: tslope=',tslope,'  tint=',tint
c
c       call linfit to get the linear fit to all the other ADC
channels
c       note, adcscale in units of adc unit/volt
c
      adcscale = 8064.0/10.0
      call linfit(adcw1,vadc,nadc,sw1,w1int)
```

```
      call linfit(adcs1,vadc,nadc,ss1,s1int)
      call linfit(adcz1,vadc,nadc,sz1,z1int)
      call linfit(adcw2,vadc,nadc,sw2,w2int)
      call linfit(adcs2,vadc,nadc,ss2,s2int)
      call linfit(adcz2,vadc,nadc,sz2,z2int)
c
      ico    = 0
      xmin   = 0e0
      xmax   = 8192
      nbin   = nbinpha
c
c  set up pha arrays
c
      chnwid = ( xmax - xmin )/float( nbin - 1 )
      call grs1( nbin, xmin, chnwid, tacpha )
      call grs1( nbin, xmin, chnwid, upha   )
      call grs1( nbin, xmin, chnwid, lpha   )
c
c  read in data
c  or, if second time around, implement new discriminator settings.
c
 100  continue
c
      if(iusephd.eq.0) then
c
c  NEW format of phd.dat
c    normalized upper sum ( = (iuw+ius+iuz)/4 ) lower and upper level
c    normalized lower sum ( = (ilw+ils+ilz)/4 ) lower and upper level
c    tac          lower and upper level
c
       open(unit=14,file='phd.dat',status='old')
   read(14,*) idusuml,idusumu,idlsuml,idlsumu,idtl,idtu
c
       rewind(14)
       write(*,*)
       write(*,*)'Using pulse height discrimination values ',
     >           'from phd.dat ...'
       close(14)
      endif
      write(*,*)
      write(*,*)' Current discriminator settings are:'
   write(6,*) idusuml,idusumu,idlsuml,idlsumu,idtl,idtu
c
      write(*,*)
      write(*,*)'Reading data in ...'
      write(*,*)
c
c  clear pha arrays
c
      call vsets(nbin,puwpha,1,0e0)
      call vsets(nbin,puspha,1,0e0)
      call vsets(nbin,puzpha,1,0e0)
      call vsets(nbin,plwpha,1,0e0)
      call vsets(nbin,plspha,1,0e0)
      call vsets(nbin,plzpha,1,0e0)
      call vsets(nbin,ptacpha,1,0e0)
      call vsets(nbin,pupha,1,0e0)
      call vsets(nbin,plpha,1,0e0)
      call vsets(nbin,phvpha,1,0e0)
      call vsets(nbin,pltpha,1,0e0)
c
c Parameter list:    NEW as of 11/93 DJL
c 1 awedgeu  2 bwedgeu  3 astripu  4 bstripu
c 5 awedgel  6 bwedgel  7 astripl  8 bstripl
```

```
c
   awedgeu = PARAM(1)
   bwedgeu = PARAM(2)
   astripu = PARAM(3)
   bstripu = PARAM(4)
   awedgel = PARAM(5)
   bwedgel = PARAM(6)
   astripl = PARAM(7)
   bstripl = PARAM(8)
   ctwzu   = PARAM(9)
   tiltx = PARAM(10)
   ctwzl   = PARAM(11)
   tilty = PARAM(12)
   az    = PARAM(13)
   bz    = PARAM(14)
   qw      = PARAM(15)
   qs      = PARAM(16)
c
c Beginning of point-by-point data read-in loop
c
c Modified 9/8/95 to do readins of fractions of a data set in
c order to check reproducibility, etc.

   write(6,*)'Enter 0 to read only part of the data file (1)'
   read(5,*)readpart
   if(readpart.eq.0) then
    write(6,*)'Enter event number to begin data read'
    read(5,*)lowlim
    write(6,*)'Enter number of points to use'
    read(5,*)range
    write(6,*)'*****   *****    *****    *****    *****'
    write(6,*)'Using data from point ',lowlim,
     @                ' to ',lowlim + range-1
    write(6,*)'*****   *****    *****    *****    *****'
   else  !do normal read of all data
    lowlim = 1
    range = n1
   endif !ifreadpart.eq.0
      irawdata = 0
      open(unit=1,file='raw_data.dat',status='old')
      do 200 i = 1,n1      ! *** master raw data read-in loop ***
       read(1,fmt=*,end=201)iuw,ius,iuz,itac,ilw,ils,ilz
       if((i.lt.lowlim).or.(i.gt.lowlim+range-1)) goto 200
       irawdata = irawdata + 1
c
c  implement software discrimination
c
   normsumu = (iuw+ius+iuz)/4.0
   if(normsumu.lt.idusuml) goto 200
   if(normsumu.gt.idusumu) goto 200
   normsuml = (ilw+ils+ilz)/4.0
   if(normsuml.lt.idlsuml) goto 200
   if(normsuml.gt.idlsumu) goto 200
       if(itac.lt.idtl) go to 200
       if(itac.gt.idtu) go to 200
c
c  Point is OK!  Increment coincident event index
c
       ico = ico + 1
c
c  bin raw pulse height distributions
c
       call xbinnr(xmin,xmax,chnwid,nbin,float(iuw),1e0,puwpha)
       call xbinnr(xmin,xmax,chnwid,nbin,float(ius),1e0,puspha)
```

```
      call xbinnr(xmin,2*xmax,2*chnwid,nbin,float(iuz),1e0,puzpha)
      call xbinnr(xmin,xmax,chnwid,nbin,float(ilw),1e0,plwpha)
      call xbinnr(xmin,xmax,chnwid,nbin,float(ils),1e0,plspha)
      call xbinnr(xmin,2*xmax,2*chnwid,nbin,float(ilz),1e0,plzpha)
      call xbinnr(xmin,xmax,chnwid,nbin,float(itac),1e0,ptacpha)
      call xbinnr(xmin,xmax,chnwid,nbin,float(normsumu),1e0,pupha)
      call xbinnr(xmin,xmax,chnwid,nbin,float(normsuml),1e0,plpha)
c
c       correct ADC data using the appropriate calibration curves
c
      iuw =       adcscale * ( ( cw1*sw1*float(iuw) ) + w1int )
      ius =       adcscale * ( ( cs1*ss1*float(ius) ) + s1int )
      iuz = 2e0 * adcscale * ( ( cz1*sz1*float(iuz) ) + z1int )
      ilw =       adcscale * ( ( cw2*sw2*float(ilw) ) + w2int )
      ils =       adcscale * ( ( cs2*ss2*float(ils) ) + s2int )
      ilz = 2e0 * adcscale * ( ( cz2*sz2*float(ilz) ) + z2int )
c
c       convert TAC data to coincidence times ( in nsec )
c
c  This conversion relies on the calibration data having been
recorded
c  under the same conditions as the experiment.  This is to say:
c  the internal conversion time of the digital delay box (about 45
ns,
c  formerly referred to in this program as 't0') must be present
c  during both the calibration and the experiment.  If in the
experiment
c  the delay for the lower half fragment timing is generated by some
c  other means than the digital delay box, e.g., by a cable delay,
then
c  the calibration data must take this into account.  DJL 12/93
c
      ttim(ico) = (float(itac) - tint)/tslope - tdelay
c  write(6,*) ico,'  itac=',itac,'  ttim(ico)=',ttim(ico)
c  read(5,*) gostring
c
C Old Bob-style algorithm is commented out
c New algorithm follows    DJL 11/93
c
c  get upper x,y, using the 'new' algorithm
c  dmczu is the quadratic wedge coefficient in this version
c  dmczl is the quadratic strip coefficient
c
c       iuz  = iuz*dmczu  ! there are no z multiplicative factors in
this v.
c        totchgu = iuw + ius + iuz
c       wedgfr = iuw/totchgu
c       strpfr = ius/totchgu
c       zfrac  = iuz/totchgu
c       xu(ico)  = 28.399 + dacwu
c     >          - ( 113.596 * dmcwu *
c     >            ( float(iuw) - float(iuz) *
c     >            ( ( uctws - uctwz )
c     >            / ( 1.0 - 3.0 * uctwz ) )
c     >          + iuw*iuw*(dmczu-1)/totchgu )
c     >            / totchgu )
c       yu(ico)  =  - 4.563 + dacsu
c     >            + ( 63.750 * dmcsu *
c     >            ( float(ius) - float(iuz) *
c     >            ( ( uctsw - uctsz )
c     >            / ( 1.0 - 3.0 * uctsz ) )
c     >          + ius*ius*(dmczl-1)/totchgu )
c     >            / totchgu )
c
```

```
c  get lower x,y
c
c       ilz = ilz*dmczl      ! there are no z multiplicative factors
in this v.
c       totchgl = ilw + ils + ilz
c       wedgfr = ilw/totchgl
c       strpfr = ils/totchgl
c       zfrac  = ilz/totchgl
c       xl(ico)  = - 28.399 - dacwl
c     >          + ( 113.596 * dmcwl *
c     >            ( float(ilw) - float(ilz) *
c     >            ( ( lctws - lctwz )
c     >            / ( 1.0 - 3.0 * lctwz ) )
c     >          + ilw*ilw*(dmczu-1)/totchgu )
c     >            / totchgl )
c       yl(ico)  =    4.563 - dacsl
c     >            - ( 63.750 * dmcsl *
c     >            ( float(ils) - float(ilz) *
c     >            ( ( lctsw - lctsz )
c     >            / ( 1.0 - 3.0 * lctsz ) )
c     >          + ils*ils*(dmczl-1)/totchgl )
c     >            / totchgl )
C
C Here's the new algorithm  DJL 11/93
c
     tcu = iuw + ius + iuz
     tcl = ilw + ils + ilz
     fwu = iuw/tcu
     fsu = ius/tcu
     fzu = iuz/tcu
     fwl = ilw/tcl
     fsl = ils/tcl
     fzl = ilz/tcl
c
     xu(ico) = -awedgeu*(fwu+(ctwzu-1)*fzu+(qw-1)*fwu*fwu)-bwedgeu
     yu(ico) = astripu*(fsu+(qs-1)*fsu*fsu)+bstripu
     xl(ico) = awedgel*(fwl+(ctwzl-1)*fzl+(qw-1)*fwl*fwl)+bwedgel
     yl(ico) = -astripl*(fsl+(qs-1)*fsl*fsl)-bstripl
     z_corr(ico) = vbeam*(ttim(ico)/1e9)*az + bz
     @          + tiltx*(xu(ico)-xl(ico)) + tilty*(yu(ico)-yl(ico))
c
 200  continue
 201  continue
      write(*,*)'# of raw data events read in:', irawdata
      write(*,*)'# of events remaining after phd:', ico
      prcntdis = float((irawdata - ico) * 100) / float(irawdata)
   nphddis = irawdata - ico
      write(*,*)'% of events discriminated by phd:', prcntdis
      write(*,*)
      rewind(1)
      close(1)
c
c  plot pulse height distributions, and if necessary apply software
c  discriminators
c
       write(*,*)
       write(*,*)'Enter 0 to plot pulse height distributions. (1)'
       read(*,*)iphd
       if(iphd.eq.0) call phaplt
       if(iusephd.eq.0) goto 5000
       write(*,*)
       write(*,*)"Enter 0 to implement phd's from phd.dat. (1)"
       read(*,*)idisc
       if(idisc.eq.0) then
```

```
          ico = 0
          iusephd = 0
          goto 100
         endif
c
5000     return
         end
```

plot.f

```
      Program plot
c
c          Version 921212
c
c  Program to plot out the various diagnostic distributions
c  written to fort.8 by the cmlwsa program.  Plots done by
c  NCAR 3.01.
c
c  Plots:
c                P(calc cm trans E  )   vs. translational energy
c                P(app. calc Etr,cm )   vs. translational energy
c          P( theta            )   vs. theta
c            P( phi              )   vs. phi
c                P( product mass    )   vs. mass
c                P( toa 2           )   vs. toa
c            P( toa 3            )   vs. toa
c          P( toa sum          )   vs. toa
c                P( t coinc.        )   vs. tcoinc
c                P( y, 2        )   vs. y
c            P( y, 3             )   vs. y
c            P( r, 2             )   vs. r
c                P( r, 3            )   vs. r
c          P( upper frag mass )   vs. mass
c                P( all frag mass   )   vs. mass
c                P( heavy grag mass )   vs. mass
c
         implicit Real*4 (a-h,o-z)
      parameter (n1 = 512)
      parameter (n2 = 720)
      character*40 title
         dimension ecbin(n1),pecbin(n1),
     >           emcbin(n1),pmcc(n1),pimtu(n1),tbbin(n1),
     >           ptblab(n1),tcbin(n1),ptclab(n1),tcobin(n1),
     >           ptcolab(n1),pecabin(n1)
         dimension th(n1),pth(n1),ph(n2),pph(n2)
         dimension y(n1),r(n1),pyb(n1),pyc(n1),prb(n1),prc(n1)
         dimension axis_mass(0:n1),top_mass(0:n1),all_mass(0:n1)
c
c  OK,    read all this stuff in from unit 8
c
c
c  read in calculated cm translational energy distribution
c
      open(unit=88,file='cmlwsa.ker')
      open(8,status='old')
         read(8,5000)title
5000        format(A40)
         read(8,*)ncebin
         read(8,*)ecbinmin,decbin
         read(8,*)ndum
         read(8,*)(pecbin(i),i=1,ncebin)
         do i = 1,ncebin
```

```
        ecbin(i) = ecbinmin + (float(i)-0.5)*decbin
    write(88,*)ecbin(i),pecbin(i)
       enddo
    close(88)
c
c   read in approx. calculated cm translational
c   energy distribution
c
       read(8,*)ncebin
       read(8,*)ecbinmin,decbin
       read(8,*)ndum
       read(8,*)(pecabin(i),i=1,ncebin)
       do i = 1,ncebin
        ecbin(i) = ecbinmin + (float(i)-0.5)*decbin
       enddo
c
c     read in theta angular distribution
c
       read(8,*)nth
       read(8,*)thmin,dth
       read(8,*)ndum
       read(8,*)(pth(i),i=1,nth)
       do i = 1,nth
        th(i) = thmin  + (float(i)-0.5)*dth
       enddo
c
c     read in phi angular distribution
c
       read(8,*)nph
       read(8,*)phmin,dph
       read(8,*)ndum
       read(8,*)(pph(i),i=1,nph)
       do i = 1,nph
        ph(i) = phmin + (float(i)-0.5)*dph
       enddo
c
c   read in calculated mass distribution
c
       read(8,*)nmcbin
       read(8,*)embinmin,dmbin
       read(8,*)ndum
       read(8,*)(pmcc(i),i=1,nmcbin)
       do i = 1,nmcbin
        emcbin(i) = embinmin + (float(i)-0.5)*dmbin
       enddo
c
c   read in toa light particle distribution
c
       read(8,*)ntbbin
       read(8,*)tbbinmin,dtbbin
       read(8,*)ndum
       read(8,*)(ptblab(i),i=1,ntbbin)
       do i = 1,ntbbin
        tbbin(i) = tbbinmin + (float(i)-0.5)*dtbbin
       enddo
c
c   read in toa heavy particle distribution
c
       read(8,*)ntcbin
       read(8,*)tcbinmin,dtcbin
       read(8,*)ndum
       read(8,*)(ptclab(i),i=1,ntcbin)
       do i = 1,ntcbin
        tcbin(i) = tcbinmin + (float(i)-0.5)*dtcbin
```

```
      enddo
c
c  read in coinc. time distribution
c
      read(8,*)ntcobin
      read(8,*)tcobinmin,dtcobin
      read(8,*)ndum
      read(8,*)(ptcolab(i),i=1,ntcobin)
      do i = 1,ntcobin
       tcobin(i) = tcobinmin + (float(i)-0.5)*dtcobin
      enddo
c
c  read in y,r distributions
c
      read(8,*)nproj
      read(8,*)ymin,dyp
      read(8,*)ndum
      read(8,*)(pyb(i),i=1,nproj)
      do i = 1,nproj
       y(i) = ymin + (float(i)-0.5)*dyp
      enddo
c
      read(8,*)nproj
      read(8,*)ymin,dyp
      read(8,*)ndum
      read(8,*)(pyc(i),i=1,nproj)
      do i = 1,nproj
       y(i) = ymin + (float(i)-0.5)*dyp
      enddo
c
      read(8,*)nproj
      read(8,*)rmin,drp
      read(8,*)ndum
      read(8,*)(prb(i),i=1,nproj)
      do i = 1,nproj
       r(i) = rmin + (float(i)-0.5)*drp
      enddo
c
      read(8,*)nproj
      read(8,*)rmin,drp
      read(8,*)ndum
      read(8,*)(prc(i),i=1,nproj)
      do i = 1,nproj
       r(i) = rmin + (float(i)-0.5)*drp
      enddo
c
      read(8,*)nbinm
      read(8,*)gmin,dsm
      read(8,*)ndum
      read(8,*)(pimtu(i),i=1,nbinm)
c
   close(8)
c
c The data bins are centered on the integers; bin 1 is mass=1, and so
on
c (There are no bins for mass = 0 or mass = radical_mass)
c nbin = radical_mass - 1; the bin spacing is fixed at dsm = 1
c Histogram points come in triplets:
c   1st point is top left of bar for the ith point
c   2nd point is top right of bar
c   3rd point is bottom right of bar (i.e., y value is zero)
c There is one bonus point at the beginning for the bottom left of
the first
```

```
c   histogram bar; thus the arrays go from 0 to 3*n, for a total of
3*n+1
c
   axis_mass(0) = 0.5
   do i = 1,nbinm*3
     axis_mass(i) = 0.5 + ((i+1)/3)
   enddo
c
   top_mass(0) = 0
   all_mass(0) = 0
   do i = 1,nbinm
     top_mass(3*i-2) = pimtu(i)
     top_mass(3*i-1) = pimtu(i)
     top_mass(3*i)   = 0.0
     all_mass(3*i-2) = ( pimtu(i) + pimtu(nbinm+1-i) ) / 2
     all_mass(3*i-1) = ( pimtu(i) + pimtu(nbinm+1-i) ) / 2
     all_mass(3*i)   = 0.0
   enddo
c
c*********************************************************************
*
c        ok, all distributions are   read in now
c        now, plot one at a time with NCAR.
c
c        Plot translational energy distribution
c
        write(*,*)'Enter 0 for Suntools, 1 for TEK, 2 for X11 ',
     >            'or 3 for Batch'
      read(*,*)iwkst
c        Plot calc'd translational energy distribution
      call opngks
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(Et)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Calc. CM Kinetic Energy (eV)$')
      call fwhm(ecbin,pecbin,ncebin,xh,xpk,xl,fw)
        write(*,*)'CM KER peak = ',xpk,' eV; FWHM = ',fw,' eV'
      call ezxy(ecbin,pecbin,ncebin,title)
c
c        Plot approx. calc'd translational energy distribution
c
      call setusv('PB',2)
c       call agseti('SET.',1)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(Et)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Approx. Calc. CM Etrans (eV)$')
      call fwhm(ecbin,pecabin,ncebin,xh,xpk,xl,fw)
        write(*,*)'Approx CM KER peak = ',xpk,' eV; FWHM = ',fw,' eV'
      call ezxy(ecbin,pecabin,ncebin,title)
c        Plot theta angular distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(theta)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','CM Angle Distribution, Theta (deg)$')
      call fwhm(th,pth,nth,xh,xpk,xl,fw)
```

```
        write(*,*)'CM Angle Theta peak = ',xpk,' deg; FWHM = ',fw,'
deg'
      call ezxy(th,pth,nth,title)
c        Plot phi angular distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(phi)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','CM Angle Distribution, Phi (deg)$')
      call agseti('X/MAXIMUM.',350)    !plots from 0 to 360 (not a
typo)
      call fwhm(ph,pph,nph,xh,xpk,xl,fw)
        write(*,*)'CM Angle Phi peak = ',xpk,' deg; FWHM = ',fw,'
deg'
      call ezxy(ph,pph,nph,title)
c        Plot calc'd mass  distribution
      call agseti('SET.',1)      !very important, set maximum back to
default
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(M)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Mass (amu)  MASS SPECTRUM$')
      call fwhm(emcbin,pmcc,nmcbin,xh,xpk,xl,fw)
        write(*,*)'Peak Mass = ',xpk,' amu; FWHM = ',fw,' amu'
      call ezxy(emcbin,pmcc,nmcbin,title)
c        Plot light particle toa distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(t)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','TOA Light Particle  (microsec.)$')
      call fwhm(tbbin,ptblab,ntbbin,xh,xpk,xl,fw)
        write(*,*)'Peak light TOF = ',xpk,' us; FWHM = ',fw,' us'
      call ezxy(tbbin,ptblab,ntbbin,title)
c        Plot particle 3 toa distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(t)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','TOA Heavy Particle  (microsec.)$')
      call fwhm(tcbin,ptclab,ntbbin,xh,xpk,xl,fw)
      write(*,*)'Peak heavy TOF = ',xpk,' us; FWHM = ',fw,' us'
      call ezxy(tcbin,ptclab,ntcbin,title)
c
c        sum tof spectra
c
      do i = 1,ntcbin
       ptclab(i) = (ptclab(i) + ptblab(i))/2e0
      enddo
c
c        output summed tof spectrum to unit 15
c
c  do i = 1,ntcbin
c   write(15,*)tcbin(i),ptclab(i)
c  enddo
```

```
c
c        plot summed tof spectrum
c
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(t)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Summed TOA Spectrum (microsec.)$')
      call fwhm(tcbin,ptclab,ntcbin,xh,xpk,xl,fw)
      write(*,*)'Peak sum TOF = ',xpk,' us; FWHM = ',fw,' us'
      call ezxy(tcbin,ptclab,ntcbin,title)
c        Plot Coinc. time distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(t)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Coinc. Time Distribution (nsec)$')
      call fwhm(tcobin,ptcolab,ntcobin,xh,xpk,xl,fw)
      write(*,*)'Peak Coinc time = ',xpk,' ns ; FWHM = ',fw,' ns'
      call ezxy(tcobin,ptcolab,ntcobin,title)
c
c        Plot Y(b) distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(y)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Y(mm) Light Mass Y Distribution$')
      call ezxy(y,pyb,nproj,title)
c        Plot Y(c) distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(y)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Y(mm) Heavy Mass Y Distribution$')
      call ezxy(y,pyc,nproj,title)
c        Plot r(b) distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(r)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','r(mm) Light Mass Radial Dist.$')
      call ezxy(r,prb,nproj,title)
c        Plot r(c) distribution
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(r)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','r(mm) Heavy Mass Radial Dist.$')
      call ezxy(r,prc,nproj,title)
c
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
```

```
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','Top Half of Detector Raw Counts$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Mass [amu] Spectrum, Top Half$')
      call ezxy(axis_mass,top_mass,nbinm*3+1,title)
c
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','All Raw Counts$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Mass [amu] Spectrum, All Data$')
      call ezxy(axis_mass,all_mass,nbinm*3+1,title)
c
      call clsgks
c
        ierr = system(' mv gmeta dynrsl.img ')
        if(iwkst.eq.0) ierr = system(' ictrans -d sunview dynrsl.img
')
        if(iwkst.eq.1) ierr = system(' ictrans -d t4010 dynrsl.img')
        if(iwkst.eq.2) ierr = system(' ictrans -d X11 -bell
dynrsl.img')
   stop
   end
*deck fwhm
      Subroutine fwhm(x,px,n,xh,xpk,xl,fw)
c
c       find peak and FWHM of distribution px
c
      dimension x(n),px(n)
c
      pxmax = 0e0
      do 10 i = 1,n
       if(px(i).gt.pxmax) then
        pxmax = px(i)
        xpk   = x(i)
        imax  = i
       endif
 10   continue
      pxhalf = pxmax/2e0
c           look for low FWHM
      do 20 i = 1,imax
       if(px(i).le.pxhalf) then
        xl     = x(i)
        ilhmax = i
       endif
 20   continue
c           look for high FWHM
      do 30 i = imax,n
       if(px(i).ge.pxhalf) then
        xh      = x(i)
        ihhmax  = i
       endif
 30   continue
      fw = xh - xl
      return
      end
```

Appendix B

TDAF[a] and TDAFXDATA[b]: Detector acceptance function correction

I. Introduction

A general discussion of the detector acceptance function (DAF) was given in the Ph.D. dissertation of Douglas Cyr, and only a brief description is given here, along with usage information and the FORTRAN code which accomplishes the DAF treatment. Due to the finite geometric acceptance of the time- and position-sensitive detector, the $P_{raw}(E_T,\theta)$ 2-D coupled energy-angle distribution produced by CMLWSA must be corrected in order to produce the true distribution $P_{true}(E_T,\theta)$ which would be observed if the detector had a full 4π steradian active area. The correction is accomplished by calculating the geometric detector acceptance function $D(E_T,\theta)$ which describes the probability of receiving an event at each (E_T,θ) combination. This function is given by:

$$D(E_T,\theta) = \frac{1}{2\pi}\int_0^{2\pi} D'(E_T,\theta,\varphi)d\varphi$$

where $D'(E_T,\theta,\varphi)$ is the doubly-differential DAF (which is what TDAF actually calculates). Integration over the azimuthal coordinate gives the desired function $D(E_T,\theta)$. The true energy-angle distribution can then be constructed from:

$$P_{true}(E_T,\theta) = \frac{P_{raw}(E_T,\theta)}{D(E_T,\theta)}$$

[a] TDAF stands for Theta Detector Acceptance Function
[b] TDAFXDATA stands for Theta Detector Acceptance Function Times DATA.

The 2-D distribution $P_{true}(E_T,\theta)$ is the input to the code MAVABFIT which finds the two 1-D distributions which are most physically meaningful, namely the translational energy distribution $P(E_T)$ and the energy dependent anisotropy parameter $\beta(E_T)$.

A second code is also described in this appendix, TDAFXDATA, which is used to accomplish the division of $P_{raw}(E_T,\theta)$ by the detector acceptance function $D(E_T,\theta)$.

II. Usage

A. Input file

One file is required on input to TDAF, and is similar to the `namelist.dat` file used for CMLWSA. This input file can have any prefix but must have the suffix `.list`. An annotated example is given below:

`$inp`	
`blocky = 4e0,`	half-width of blocking strip (cm)
`light = 13e0,`	light fragment mass (amu)
`heavy = 28e0,`	heavy fragment mass (amu)
`cbins = 360,`	number of azimuthal angle (φ) bins
`dpsd = 0.995,`	distance from dissociation laser to detector (mm)
`ebm = 8125e0,`	ion beam energy (eV)
`emax = 3.0,`	maximum value of E_T
`emin = 0.0,`	minimum value of E_T
`radmass = 41e0,`	parent radical mass
`ipolang= 0,`	laser polarization angle: $\bar{E}$ parallel to radical beam axis ≡ 0; $\bar{E}$ perpendicular to radical beam axis ≡ 90
`nbine = 100,`	number of E_T bins
`rfov = 20.0,`	maximum allowed radius of single particle recoil[a]
`rmin = 0.0,`	minimum allowed radius of single particle recoil
`x0 = -0.2785,`	center of radical beam along the x-axis relative to detector center
`y0 = -0.5803,`	center of radical beam along the y-axis relative to detector center
`iphdisc= 1,`	if `iphdisc` = 1, use azimuthal discrimination
`phdisc = 35,`	discriminates all events which have $\varphi <$ `phdisc` or $\varphi >$ 180 - `phdisc` and the complementary angles on the lower half of the detector

[a] Note that the variable `rfov` in `namelist.dat` for CMLWSA (cf. Appendix A) has an different meaning than the variable used here. The value of `rfov` in the TDAF code should correspond to the upper radial discrimination limit given interactively in CMLWSA.

`rbeammax = 1.4,`	maximum allowed centroid radius
`fwhmx = 1.7,`	FWHM along *x*-axis of radical beam shape
`fwhmy = 1.4,`	FWHM along *y*-axis of radical beam shape
`ngridx = 10,`	number of grid points defining the *x*-axis radical beam shape
`ngridy = 10,`	number of grid points defining the *y*-axis radical beam shape

```
$end
```

All user input to TDAFXDATA is given interactively and is self-explanatory. Of course, TDAFXDATA cannot be executed without first obtaining the output of both CMLWSA (the data) and the output of TDAF (the DAF).

B. Output files

On output, TDAF produces many plots using the NCAR graphics package which give a visual interpretation of the detector acceptance function. However, the most important output is the file `*.tdaf`, where * is the same prefix used on the `*.list` input file to TDAF. The `*.tdaf` file, together with `fort.9` produced by CMLWSA (which contains the $P_{raw}(E_T,\theta)$ distribution), are the files which TDAFXDATA reads.

On output, TDAFXDATA produces the file `*.cpth`, the $P_{true}(E_T,\theta)$ 2-D distribution. This file in turn is the input to MAVABFIT, described in Appendix C.

III. FORTRAN code of TDAF and TDAFXDATA

tdaf.f

```
      Program thetadaf
c
c             unix version 920804 DRC
c
c
c     When given the orientation of the
c     polarization of the E vector relative to the radical beam,
c     masses of the photofragments of interest, radical beam kinetic
c     energy, physical characteristics of the
c     detector, and an energy window of interest, this routine
calculates
c     the appropriate theta (i.e. integrated over phi)
```

```
c     detector acceptance functions (DAF) to be used to normalize raw
data.
c     A vector for the center of mass recoil is defined which has as
its
c     base the heavy mass and as its point the light mass.  This
vector is
c     stepped through phi and theta for each energy and the total of
valid
c     events is summed for each energy and angle.  This is the most
differential
c     form of the data.  The code is ready for full theta = 1-180,
phi =
c     1-360 data from cmlwsa when that becomes available.  For now
this
c     problem is dealt with in tdafxdata, where the second half of
the tdaf
c     is ignored.    DLO 3/94
c
c  Modified 12/15/92 to include angular (phi) discrimination, but
only
c  in the horizontal (ipolang = 0 ) mode.  DLO
c  Now will also work in vertical mode.      DLO 3/94
c
c  Modified 2/2/93 to do multiple energy windows, as mentioned above.
c
c  Optimized for speed; now only writes out thdaf, and not thdafinv.
c  tdaffile now contains emin, emax, nbine
c                                        DJL 3/93
c
c  v0      : parent beam velocity (km/sec)
c  radmass : parent mass
c       light  : light fragment mass
c       heavy  : heavy fragment mass
c      ipolang : polarization of laser E vector relative to the
radical
c                beam axis (should be 0 or 90)
c  dpsd    : is the nominal flight distance from photodiss. center to
c         the center of the detector
c       x0     : the x coordinate where the undissociated radical
beam
c                would impinge on the detector if the true center of
the
c                detector were (0,0) in mm
c       y0     : similar story as x0
c  rbeammax: a radial cutoff for the centroids which matches the
rbeammax                c          in cmlwsa
c       ith    : the angle of fragment recoil axis relative to the E
c                vector of the polarized laser light
c  iphdisc: Do phi discrimination?  0 = no, 1 = yes
c  phdisc : the angle in phi to discriminate data at angles lower
c           than phi or higher than 180 - phi.
c  nsubbine: The width (in bins) of each subwindow (over which
c            theta is ~ constant) (at the moment = 1 always)
c  fwhmx  : the full width at half maximum of the neutral radical
beam
c           in the x and (fwhmy) y directions
c  ngridx : number of grids to use to approximate gaussian.  should
c           be an even number!
c
c
      implicit real*4 (a-h,o-z)

      parameter (n2 = 300  )
      parameter (n4 = 360  )
```

```
      parameter (n6 = 300  ) !NOTE:  n6 should be .ge. nbine!!

      real*4 ebinval(n2),factor(n2),cosval(2*n4),sinval(2*n4)
      real*4 thetaval(n4),thdaf(n4),lorentznorm,corrfac
      real*4 thdafex(n6,n4),thdafexsum(n6,n4),phival(2*n4)
      real*4 light,heavy,radmass,forwnum,backnum
   real*4 suma(n6)
      integer phlower,phupper,phdisc,integr(n2),cbins
   character*20 outfile,temp,listfile

      logical debug4
c
c  input namelist
c
   namelist / inp / blocky,light,heavy,cbins,
     >          ebm,emax,emin,debug4,dpsd,radmass,ipolang,nbine,
     >           rfov,rmin,x0,y0,rbeammax,iphdisc,
     >           phdisc,inorm,fwhmx,fwhmy,ngridx,ngridy
c
      data rad,pi / 0.01745329 , 3.14159265 /
      data degree / 57.29577951 /
      data amutokg / 6.022045e26 /
      data eVtoJ / 1.6021892e-19 /
c
c  amutokg is Avagadros' number times 1000 (the g->kg conversion)
c
c       eVtoJ is the eV->J conversion factor
c
c  default namelist values
c
c  the half-width of the blocker in the y direction, assumed centered
c       about true detector center (mm)
      blocky = 4e0
c  the number of angular units in a full circle (degrees = 360,grads
= 400)
      cbins = 360
c  the light mass used in determination of the center of the image
c       in (amu)
      light = 30e0
c  the heavy mass used in determination of the center of the image
c  in (amu)
      heavy = 30e0
c  the nominal beam energy (eV)
      ebm    = 6000e0
c  the maximum KER bin (eV)
      emax   = 8e0
c  the minimum KER bin (eV)
      emin   = 0e0
c  the distance from the center of the dissociation volume to the
c  face of the c-wsa mcp's (m)
      dpsd   = 0.995
c  the parent mass (amu)
      radmass = 60e0
c  the number of energy bins
      nbine  = 100
c  the number of energy bins which comprises a subwindow
c      nsubbine = 1  (at the moment force nsubbine = 1 always)
c       the polarization angle of the laser wrt the radical beam axis
      ipolang = 0
c  the radius of the field-of-view (mm)
      rfov   = 19
c  the minimum radius for radial discrimination (mm)
      rmin   =  4
c  the nominal x-origin of the detector (mm)
```

```
      x0     = 0e0
c  the nominal y-origin of the detector (mm)
      y0     = 0e0
c  the radial cutoff beyond which we will not consider events
      rbeammax = 1.5
c       do phi discrimination ?  0 = no, 1 = yes
      iphdisc = 1
c  angles in phi to do discrimination
      phdisc = 0
c       full width half max of gausian in x dimension (mm)
      fwhmx = 1.0
c       full width half max of gausian in y dimension (mm)
      fwhmy = 1.0
c       the number of (x) sections to divide the gaussian into (even
number!)
      ngridx = 10
c       the number of (y) sections to divide the gaussian into (even
number!)
      ngridy = 10
c  discriminate against all events lying outside a vertical strip
c  half width of the strip. To not discriminate, make this large
c      strdisc = 200.0   DEFUNCT 5/94
c
c    read in experimental constants from                *.list
c              (in namelist format)    (formerly fort.4)
c
c    output detector acceptance functions into          *.out
c                                (formerly fort.18)
c         output averaged results into                      stdout
(????)
c
c  read in experimental constants in namelist format
c

   corrfac = cbins/(2.0*pi)

   write(6,*)'Enter root name of namelist file'
   read(5,*)temp
   call addsuffix(temp,'.list',listfile)
   write(6,*)'Reading namelist variables from ',listfile
      open(unit = 4, file = listfile,status = 'old')
      read(4,inp,iostat=ioerr)
      if(ioerr.ne.0) stop 'Error in namelist input!'
c  if (strdisc.ne.200) stop 'strdisc not implemented in tdaf.f!!'
c
c  Following line commented out 3/2/93.  Could see no reason for it.
DLO
c  if (emin.ne.0) stop 'Yikes! emin not zero!'
   if (emin.ne.0) then
    write(6,*) 'emin is not zero:  emin = ',emin
   endif

   if( (iphdisc.eq.0).and.(phdisc.ne.0) ) then
    write(6,*)'if you do not want phi discrimination,'
    write(6,*)'then set phdisc = 0 in the namelist!'
    stop
   endif
   if( (float(ngridx/2).ne.ngridx/2.0).or.
     @         (float(ngridy/2).ne.ngridy/2.0) )
     @         stop 'ngridx and ngridy must be even numbers!'
   if(mod(cbins,4).ne.0) stop 'cbins must be divisible by 4'
   if(cbins.gt.2*n4) stop 'cbins is too large.  change param n4)'
c
   write(6,*)'Enter output file ''root'' name'
```

```
   read(5,*)temp
   call addsuffix(temp,'.tdaf',outfile)
   write(6,*)'tdaf output will be stored in:  ',outfile
c
      write(*,*)'Enter 0 for sun, 1 for TEK, 2 for X11 or 3 for
batch:'
      read(*,*)iwkst
c
c        start calculations
c
c  determine v0 in m/s
c
c
c  Since the code cannot yet handle phi discrimination with vertical
c  polarization, exit if both iphdisc = 1 and ipolang = 90
c
   if((iphdisc.eq.1).and.(ipolang.eq.90)) then
     write(6,*) ' '
     write(6,*) '*****************************************'
     write(6,*) 'The code now handles phi'
     write(6,*) 'discrimination with vertical polarization.'
     write(6,*) 'Finally you can sleep at night again!'
     write(6,*) '*****************************************'
   endif
c
   if(iphdisc.eq.1) then   !(do phi discrimination)
     phlower = phdisc
     phupper = 180 - phdisc
   else              !(no phi disc)
     phlower = 0
     phupper = 180
   endif
c
      v0   = sqrt((2.0*ebm*amutokg*eVtoJ)/radmass)
      gam   = v0/dpsd      !gam = 1/flight time.  It is an REC era
obscure term
   dpsdmm = dpsd*1000      !flight distance in mm
      write(*,*)
      write(*,*)' Incident beam energy = ',ebm,' eV'
      write(*,*)' Incident beam veloc. = ',v0,' m/sec '
      write(*,*)
      write(*,*)' Beam center at x = ',x0,' y = ',y0
      write(*,*)
c
c       begin by initializing three do loops that calculate, in turn,
c       if a given set of center of mass (com) energy, polar angle
theta
c       and azimuthal angle phi will result in a valid coincidence
event
c
c       for ipolang = 0, (the laser polarization is parallel to the
c       radical beam direction of propagation) y~sin(th)cos(ph) and
c       x~sin(th)sin(ph).  In these two expressions, y and x are in
detector
c  coordinates, while the variables th and ph are in the
c  photodissociation coordinate system.  Keep this in mind
throughout!
c  Phi is referenced from ph=0 as the light
c       mass is heading straight upward when th=90
c       pd coords : z towards the detector, x up, rh coords
c       detector coords : x to right, y up, z towards source
c       transformation : pd -> det
c                        x -> y
c                        y -> x
```

```
c                          z -> -z
c
c       for ipolang = 90, (the laser polarization is perpendicular to
c       the radical beam direction of propagation) y~cos(th)
c       and x~-sin(th)sin(ph).  Phi is referenced from ph=0 as the
light
c       mass is heading parallel to the radical beam axis when th=90
c       pd coords : x towards the detector, z up, rh coords
c       detector coords : x to right, y up, z towards source (same as
above)
c       transformation : pd -> det
c                          x -> -z
c                          y -> -x
c                          z -> y
c
c  now decide whether to use distribution beam shape or not
   write(6,*)'Enter 0 for gaussian beam shape, 1 for point source'
   write(6,*)'2 for lorentzian beam shape'
   read(5,*) igauss
   if(igauss.eq.0) then
         sigmax = fwhmx/2.345
         write(6,*)'sigmax =',sigmax
         sigmay = fwhmy/2.345
         xbinwdth = 2*rbeammax/float(ngridx)
    write(6,*) 'xbinwidth =',xbinwdth
         ybinwdth = 2*rbeammax/ngridy
   elseif(igauss.eq.2) then
    xbinwdth = 2*rbeammax/float(ngridx)      !kluge for widthrange
    ybinwdth = 2*rbeammax/float(ngridy)
   endif

      erange = emax - emin
      if ((erange.le.0).or.(nbine.gt.n2)) then
       write(*,*) 'Energy range or bin number incompatibility, ',
     >            'check namelist!!!'
       stop
      endif
      if (abs(light + heavy - radmass).gt.1e-4) then
       write(*,*) 'Parent and fragments mass incompatibility, ',
     >            'check namelist!!!'
       stop
      endif
      if((ipolang.ne.0).and.(ipolang.ne.90)) then
       write(*,*) 'Laser polarization variable incorrect, ',
     >            'check namelist!!!'
       stop
      endif
      redmass = light * heavy / radmass
      step = erange / nbine
c
c  nbine1 = (ewinmin/emax)*nbine + 1
c  nbine2 = (ewinmax/emax)*nbine
c
   write(*,*)'step size = ',step*1000,'  meV'
c
c  create xcenter,ycenter, which are the "true" centers

   xcenter = x0
   ycenter = y0
   if(igauss.eq.1) then !if point source, then disable distribution
loops
    xbinwdth = 0.0
    ybinwdth = 0.0
    ngridx = 1
```

```
     ngridy = 1
    endif
cccccccccc calculate tabulated data for main loop
cccccccccccccccccccccccc
c
    write(6,*)'Calculating tabulated arrays'
    ratio = light/heavy
c   write(6,*)'light = ',light,'heavy =',heavy,'radmass =',radmass
c   write(6,*)'ratio =',ratio,'redmass =',redmass,amutokg,eVtoJ
    do inrg = 1,nbine
          ebinval(inrg) = emin + ((inrg - 0.5) * step)
          vrel = sqrt((2*ebinval(inrg)*amutokg*eVtoJ)/redmass) !m/sec
      factor(inrg) = 1000 * heavy/radmass*vrel/gam  !([m3/M]*vrel*t in
mm)
c    write(6,*)'ebinval(',inrg,') = ',ebinval(inrg)
c    write(6,*)'vrel = ',vrel,'gam = ',gam,'factor = ',factor(inrg)
c The quantity (vrel/gam) is the ideal distance between the two
particles.
c i.e. It is the distance when the center of mass is in the plane of
c the detector.
    end do
    angstep = 360.0/cbins
    do ith = 1, cbins/2
      thetaval(ith) = (ith - 0.5)*angstep !converts index to angle
    end do
c   create trig tables
         do iph = 1,cbins
      phival(iph) = (iph - 0.5)*angstep !converts index to angle
      cosval(iph) = cos((iph - 0.5)*angstep/corrfac)
      sinval(iph) = sin((iph - 0.5)*angstep/corrfac)
c     tanval(iph) = tan((iph - 0.5)*angstep/corrfac)
    end do
c
cccccccccc Begin loop over energy
cccccccccccccccccccccccccccccccccccc

    write(6,*)'Entering main loop.'

    if(nint(light).ne.nint(heavy)) then !use full range of phi
c                              (very long conditional)

    do 80 inrg = 1,nbine

cccccc begin beam center loops cccccccccccccccccccccccccccccccc
c
    do 85 ix = 1,ngridx
        x0 = xcenter - 0.5*xbinwdth + float(ix - ngridx/2)*xbinwdth
      do 87 iy = 1,ngridy
        y0 = ycenter - 0.5*ybinwdth + float(iy - ngridy/2)*ybinwdth

c   skip out of entire calculation if center point is outside rbeammax
c   dist_from_cent = sqrt((x0-xcenter)**2 + (y0 - ycenter)**2)
c   if(dist_from_cent.gt.rbeammax) goto 87
cccccccccccc now that the binwidths are defined in terms of rbeammax,
there
cccccccccccc is no need for the preceeding statement

    if(igauss.eq.0) then
c    calculate value of distribution to use in normalization
     zx = (x0 - xcenter)/sigmax
     zy = (y0 - ycenter)/sigmay
     gausnorm = (1.0/(2.0*sigmax*sigmay*pi))
     probdist = gausnorm*exp( -0.5*(zx*zx + zy*zy) )
    elseif(igauss.eq.2) then
```

```
      zx = (x0 - xcenter)/(0.5*fwhmx)
      zy = (y0 - ycenter)/(0.5*fwhmy)
      lorentznorm = 1.0/(pi*pi)
      probdist = lorentznorm*(1.0/(1.0+(zx*zx)))*(1.0/(1.0+(zy*zy)))
     else
      probdist = 1.0
     endif

cccccccccc Begin loops over theta,energy, and phi
cccccccccccccccccccccc
c       Note that these loops are over photodissociation coordinates
c       In this section, the loop over phi and theta, which describes
c            different space in the lab frame for vertical vs.
horizontal
c       laser polarization, gives x2,x3,y2,y3 in detector coordinates,
c       which are the same regardless of polarization.  Discrimination
c       is accomplished in this frame of reference, and the tdaf is
c       accumulated in the photodissociation frame (theta and phi).
c       Eventually cmlwsa should also put out pd frame data from
c       theta = 1-180 and phi = 1 - 360.
c       A vector with base at heavy particle, arrow point at light
c       particle, is swept through the coordinates theta and phi DLO
3/94
c
c       Now, for equal mass, one would overcount using the above
definitions
c       In keeping with cmlwsa's definitions in psdclce, for equal
mass
c       in either polarization, we will use theta = 1-180, phi = 1 -
90
c       and phi = 271 - 360       DLO 6/94
c
c       Don't worry that a choice of cbins.ne.360 means that theta
c       and phi don't seem to go to 180 and 360 degrees, respectively
c       This is taken care of earlier, when the tables are created.
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccc

       do 100 ith = 1,cbins/2
        do 300 iph = 1,cbins
         x2 = factor(inrg)*sinval(ith)*sinval(iph) + x0
         x3 = -ratio*(x2 - x0) + x0
         if(ipolang.eq.0) then
          y2 = factor(inrg)*sinval(ith)*cosval(iph) + y0
         elseif(ipolang.eq.90) then
          x2 = -(x2 - x0) + x0
          x3 = -(x3 - x0) + x0
          y2 = factor(inrg)*cosval(ith) + y0
     z2 = -factor(inrg)*sinval(ith)*cosval(iph) !needed for thprime
c    z3 = -ratio*z2                    !don't need z3
         endif
         y3 = -ratio*(y2 - y0) + y0
c
c  Now do discrimination for beam block, field of view, and phi
c  angular disrimination
c  First calculate phprime, the phi angle with respect to the
c  center point of the beam block, being careful of arctangent
c  pitfalls.  Phprime is a detector coordinate frame based phi angle.
c  For vertical polarization it may seem confusing.  The phi disc
values
c  given in cmlwsa always refer to **detector coordinates** and it
c  is phprime that must be equated to these values, even if, (as is
the
```

```
c  case for vertical polarization) this is not the [center of mass]
phi
c  we normally discuss.  Therefore the tdaf plots for a vertical
c  polarization will look much funkier than those for horizontal
c  polarization since we have broken lab frame azimuthal symmetry
c        DLO 3/94

c  Now create also a detector based theta angle in order to correct
c  for the forward/backward scattering effect.  Basically we will
c  modify the x2,y2,x3,y3 values based on the angle thprime such that
c  these cartesian displacements are not referenced to their values
c  at the time the center of mass hits the detector, but their values
c  when each particle hits the detector, i.e. real life.

c  due to a bizarre case in which x2 rounded to zero, add the
following:
     if(x2.eq.0.0) x2 = x2 + 0.000005
     if(x3.eq.0.0) x3 = x3 + 0.000005
c  go figure

c  the value of thprime depends on ipolang

     if(ipolang.eq.0) then
      thprime = 180 - thetaval(ith) !thprime is zero towards source
     else                           !and 180 towards detector
      thprime = acos(z2/factor(inrg))*degree
     endif

c  Now modify x2,y2,x3,y3 as described above to reflect the real
c  positions of the particles when they strike the detector
c  For thprime < 90, the light particle (#2) is back scattered, and
c  therefore its real life recoil must increase.  For thprime > 90
c  #2 is forward scattered, and its recoil must decrease.  The
c  change in sign is from the tangent function.
c  Since the forward/backward asymmetry is accounted for in
c  detector coordinates, which are the same for horizontal and
c  vertical polarization, this section doesn't depend on ipolang.

          radius2 = sqrt(x2*x2 + y2*y2)
          radius3 = sqrt(x3*x3 + y3*y3)
    tanval = tan(thprime/degree)
    corr2 = 1+ (radius2/dpsdmm) / tanval
    corr3 = 1- (radius3/dpsdmm) / tanval

   x2 = x2*corr2
   x3 = x3*corr3
   y2 = y2*corr2
   y3 = y3*corr3

     phprime2 = atan(y2/x2)*degree
     if(phprime2.lt.0.0) then            !(avoid arctan range
problem)
       phprime2 = phprime2 + 180.0
     endif
     phprime3 = atan(y3/x3)*degree
     if(phprime3.lt.0.0) then            !(avoid arctan range
problem)
       phprime3 = phprime3 + 180.0
     endif


c
c Here it is:  the detector acceptance conditional!
c   WARNING this version no longer pays attention to strdisc!!
```

```
c
    if ( (min(phprime2,phprime3).ge.phlower).and.
     @          (max(phprime2,phprime3).le.phupper) ) then
          if ( (radius2.le.rfov).and.(radius3.le.rfov) ) then
      if ( (radius2.ge.rmin).and.(radius3.ge.rmin) ) then
       if ( (abs(y3).ge.blocky).and.(abs(y2).ge.blocky) ) then
        thdaf(ith) = thdaf(ith) + 1   !(sum over phi and energy)
         integr(inrg) = integr(inrg) + 1
       endif
      endif
     end if
    end if

300     continue  ! end of phi loop
c
   thdafex(inrg,ith) = thdaf(ith)*sinval(ith)*360/cbins !(jacobian)
c  thdafex(inrg,ith) = thdaf(ith) !(without jacobian)
c
100      continue ! end of theta loop
c
c  accumulate the thdafex sums from each x0,y0 in the distribution

     do 225 ith = 1,cbins/2
c       now multiply by probdist to normalize daf for distribution
beam shape
   thdafexsum(inrg,ith) = thdafexsum(inrg,ith)
     @          + thdafex(inrg,ith)*probdist
   thdafex(inrg,ith) = 0.0 ! set to zero for next distribution point
   thdaf(ith) = 0.0
225        continue
cccccccccccc End Main Loops
cccccccccccccccccccccccccccccccccccccccccccc

87 continue !(loop over ngridy)
85      continue !(loop over ngridx)

   do jjj = 1,cbins/2
    thdaf(jjj) = 0.0 !(this array used inside each main loop)
c  now that the entire distribution grid has been cycled through for
c  one energy bin, rename thdafexsum to thdafex.
    thdafex(inrg,jjj) = thdafexsum(inrg,jjj)
   enddo

      thdafmax = 1e-36
      thdafmin = 1e36
   suma(inrg) = 0.0
   forwnum = 0.0
   backnum = 0.0
      do 400 ith = 1,cbins/2
   suma(inrg) = suma(inrg) + thdafex(inrg,ith)
       if (thdafex(inrg,ith).ge.thdafmax) then
        thdafmax = thdafex(inrg,ith)
       endif
       if (thdafex(inrg,ith).le.thdafmin.and.
     >        thdafex(inrg,ith).gt.1e-9) then
        thdafmin = thdafex(inrg,ith)
       endif
   if(ith.le.cbins/4) then
    forwnum = forwnum + thdafex(inrg,ith)
   else
    backnum = backnum + thdafex(inrg,ith)
   endif
400   continue
c  write(6,*) 'Integrated intensity (early) = ',suma(inrg)
```

```
      suma(inrg) = suma(inrg)/(360*cbins/pi)

      write(6,*) 'Energy bin ',inrg,' of energy ',ebinval(inrg)
      write(6,*) 'Integrated intensity = ',suma(inrg)
      write(6,*) 'Light Particle recoil = ',factor(inrg),' (mm)'
      if((suma(inrg).gt.0.0).and.(ipolang.eq.0)) then
       write(6,272) 'Forward scatt.  = ',
        @          100*forwnum/(forwnum + backnum),'%'
       write(6,272) 'Backward scatt. = ',
        @          100*backnum/(forwnum + backnum),'%'
      endif
      write(6,*) ' '
c     write(6,*) 'Un-sine-thetad integrated intenisty = ',integr(inrg)
c     write(6,*) 'thdafmax = ',thdafmax
c     write(6,*) 'thdafmin = ',thdafmin
c     write(6,*) 'number of theta singularities = ', isingul
c     write(6,*)

80 continue ! end of inrg index loop

272       format(A18,F5.2,A1)
      else  !from if(light.ne.heavy), i.e. this is the equal mass case

      do 180 inrg = 1,nbine

cccccc begin beam center loops cccccccccccccccccccccccccccccccc
c

      do 185 ix = 1,ngridx
          x0 = xcenter - 0.5*xbinwdth + float(ix - ngridx/2)*xbinwdth
        do 187 iy = 1,ngridy
          y0 = ycenter - 0.5*ybinwdth + float(iy - ngridy/2)*ybinwdth

c     skip out of entire calculation if center point is outside rbeammax
c     dist_from_cent = sqrt((x0-xcenter)**2 + (y0 - ycenter)**2)
c     if(dist_from_cent.gt.rbeammax) goto 187
ccccccccccc now that the binwidths are defined in terms of rbeammax,
there
ccccccccccc is no need for the preceeding statement

      if(igauss.eq.0) then
c      calculate value of distribution to use in normalization
       zx = (x0 - xcenter)/sigmax
       zy = (y0 - ycenter)/sigmay
       gausnorm = (1.0/(2.0*sigmax*sigmay*pi))
       probdist = gausnorm*exp( -0.5*(zx*zx + zy*zy) )
      elseif(igauss.eq.2) then
       zx = (x0 - xcenter)/(0.5*fwhmx)
       zy = (y0 - ycenter)/(0.5*fwhmy)
       lorentznorm = 1.0/(pi*pi)
       probdist = lorentznorm*(1.0/(1.0+(zx*zx)))*(1.0/(1.0+(zy*zy)))
      else
       probdist = 1.0
      endif


ccccccccccc Begin loops over theta,energy, and phi
cccccccccccccccccccccc
c       Note that these loops are over photodissociation coordinates
c       In this section, the loop over phi and theta, which describes
c            different space in the lab frame for vertical vs.
horizontal
```

```
c       laser polarization, gives x2,x3,y2,y3 in detector coordinates,
c       which are the same regardless of polarization.  Discrimination
c       is accomplished in this frame of reference, and the tdaf is
c       accumulated in the photodissociation frame (theta and phi).
c       Eventually cmlwsa should also put out pd frame data from
c       theta = 1-180 and phi = 1 - 360.
c       A vector with base at heavy particle, arrow point at light
c       particle, is swept through the coordinates theta and phi DLO
3/94
c
c       Now, for equal mass, one would overcount using the above
definitions
c       In keeping with cmlwsa's definitions in psdclce, for equal
mass
c       in either polarization, we will use theta = 1-180, phi = 1 -
90
c       and phi = 271 - 360       DLO 6/94
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccc

        do 1000 ith = 1,cbins/2
         do 3000 iph = 1,cbins

   if((iph.gt.cbins/4).and.(iph.le.3*cbins/4)) goto 3000!i.e. skip 90
- 270

          x2 = factor(inrg)*sinval(ith)*sinval(iph) + x0
          x3 = -ratio*(x2 - x0) + x0
          if(ipolang.eq.0) then
           y2 = factor(inrg)*sinval(ith)*cosval(iph) + y0
          elseif(ipolang.eq.90) then
           x2 = -(x2 - x0) + x0
           x3 = -(x3 - x0) + x0
           y2 = factor(inrg)*cosval(ith) + y0
      z2 = -factor(inrg)*sinval(ith)*cosval(iph) !needed for thprime
c     z3 = -ratio*z2                    !don't need z3
          endif
          y3 = -ratio*(y2 - y0) + y0
c
c  Now do discrimination for beam block, field of view, and phi
c  angular disrimination
c  First calculate phprime, the phi angle with respect to the
c  center point of the beam block, being careful of arctangent
c  pitfalls.  Phprime is a detector coordinate frame based phi angle.
c  For vertical polarization it may seem confusing.  The phi disc
values
c  given in cmlwsa always refer to **detector coordinates** and it
c  is phprime that must be equated to these values, even if, (as is
the
c  case for vertical polarization) this is not the [center of mass]
phi
c  we normally discuss.  Therefore the tdaf plots for a vertical
c  polarization will look much funkier than those for horizontal
c  polarization since we have broken lab frame azimuthal symmetry
c         DLO 3/94

c  Now create also a detector based theta angle in order to correct
c  for the forward/backward scattering effect.  Basically we will
c  modify the x2,y2,x3,y3 values based on the angle thprime such that
c  these cartesian displacements are not referenced to their values
c  at the time the center of mass hits the detector, but their values
c  when each particle hits the detector, i.e. real life.
```

```
c  due to a bizarre case in which x2 rounded to zero, add the
following:
     if(x2.eq.0.0) x2 = x2 + 0.000005
     if(x3.eq.0.0) x3 = x3 + 0.000005
c  go figure

     if(ipolang.eq.0) then
      thprime = 180 - thetaval(ith) !thprime is zero towards source
     else                           !and 180 towards detector
      thprime = acos(z2/factor(inrg))*degree
c        arccos(x) is 0 to pi as x goes from 1 to -1
cccccccccccccccccccccccccccccccc
c  if(inrg.eq.28) write(6,*) 'Div 1'
     endif

c  Now modify x2,y2,x3,y3 as described above to reflect the real
c  positions of the particles when they strike the detector

         radius2 = sqrt(x2*x2 + y2*y2)
         radius3 = sqrt(x3*x3 + y3*y3)
    tanval = tan(thprime/degree)
cccccccccccccccccccccccccccccccc
c  if(inrg.eq.28) write(6,*) 'Div 2'
c  for 0 < thprime < 90, tanval > 0, so
c  for thprime near 0 or 180, tanval approaches +inf or -inf
c  such that corr2 approaches 1
    corr2 = 1+ (radius2/dpsdmm) / tanval
cccccccccccccccccccccccccccccccc
c  if(inrg.eq.28) write(6,*) 'Div 3'

    corr3 = 1- (radius3/dpsdmm) / tanval
cccccccccccccccccccccccccccccccc
c  if(inrg.eq.28) write(6,*) 'Div 4'


   x2 = x2*corr2
   x3 = x3*corr3
   y2 = y2*corr2
   y3 = y3*corr3

     phprime2 = atan(y2/x2)*degree
cccccccccccccccccccccccccccccccc
c  if(inrg.eq.28) write(6,*) 'Div 5'

     if(phprime2.lt.0.0) then            !(avoid arctan range
problem)
       phprime2 = phprime2 + 180.0
     endif
   if(abs(x3).lt.0.00001) then
   write(6,*) 'Div/0 coming',inrg,ith,iph,
     @ 'y3 =',y3,'x3 = ',x3, x2
   write(6,*)'factor =',factor(inrg)
   iiiii = 1
   endif
     phprime3 = atan(y3/x3)*degree
   if(iiiii.eq.1) then
    write(6,*) 'phprime3 = ',phprime3
    iiiii = 0
   endif
     if(phprime3.lt.0.0) then            !(avoid arctan range
problem)
       phprime3 = phprime3 + 180.0
     endif
```

```
          radius2 = sqrt(x2*x2 + y2*y2)
          radius3 = sqrt(x3*x3 + y3*y3)

c
c Here it is:  the detector acceptance conditional!
c   WARNING this version no longer pays attention to strdisc!!
c
c
     if ( (min(phprime2,phprime3).ge.phlower).and.
      @          (max(phprime2,phprime3).le.phupper) ) then
           if ( (radius2.le.rfov).and.(radius3.le.rfov) ) then
       if ( (radius2.ge.rmin).and.(radius3.ge.rmin) ) then
        if ( (abs(y3).ge.blocky).and.(abs(y2).ge.blocky) ) then
         thdaf(ith) = thdaf(ith) + 1    !(sum over phi and energy)
          integr(inrg) = integr(inrg) + 1
        endif
       endif
      end if
     end if

3000      continue  ! end of phi loop
c
   thdafex(inrg,ith) = thdaf(ith)*sinval(ith)*360/cbins !(jacobian)
c  thdafex(inrg,ith) = thdaf(ith) !(without jacobian)
c
1000      continue ! end of theta loop
c
c  accumulate the thdafex sums from each x0,y0 in the distribution

      do 2250 ith = 1,cbins/2
c        now multiply by probdist to normalize daf for distribution
beam shape
   thdafexsum(inrg,ith) = thdafexsum(inrg,ith)
     @           + thdafex(inrg,ith)*probdist
   thdafex(inrg,ith) = 0.0 ! set to zero for next distribution point
   thdaf(ith) = 0.0
2250        continue
cccccccccccc End Main Loops
cccccccccccccccccccccccccccccccccccccccccccc

187       continue !(loop over ngridy)
185       continue !(loop over ngridx)

   do jjj = 1,cbins/2
    thdaf(jjj) = 0.0 !(this array used inside each main loop)
c  now that the entire distribution grid has been cycled through for
c  one energy bin, rename thdafexsum to thdafex.
    thdafex(inrg,jjj) = thdafexsum(inrg,jjj)
   enddo

      thdafmax = 1e-36
      thdafmin = 1e36
   suma(inrg) = 0.0
   forwnum = 0.0
   backnum = 0.0
      do 4000 ith = 1,cbins/2
   suma(inrg) = suma(inrg) + thdafex(inrg,ith)
       if (thdafex(inrg,ith).ge.thdafmax) then
        thdafmax = thdafex(inrg,ith)
       endif
       if (thdafex(inrg,ith).le.thdafmin.and.
     >         thdafex(inrg,ith).gt.1e-9) then
```

```
          thdafmin = thdafex(inrg,ith)
         endif
      if(ith.le.cbins/4) then
       forwnum = forwnum + thdafex(inrg,ith)
      else
       backnum = backnum + thdafex(inrg,ith)
      endif
4000   continue
c  write(6,*) 'Integrated intensity (early) = ',suma(inrg)

      suma(inrg) = suma(inrg)/(360*cbins/pi)

      write(6,*) 'Energy bin ',inrg,' of energy ',ebinval(inrg)
      write(6,*) 'Integrated intensity = ',suma(inrg)
      if(suma(inrg).gt.0.0) then
       write(6,272) 'Forward scatt.  = ',
        @          100*forwnum/(forwnum + backnum),'%'
       write(6,272) 'Backward scatt. = ',
        @          100*backnum/(forwnum + backnum),'%'
      endif
      write(6,*) ' '
c  write(6,*) 'Un-sine-thetad integrated intenisty = ',integr(inrg)
c  write(6,*) 'thdafmax = ',thdafmax
c  write(6,*) 'thdafmin = ',thdafmin
c  write(6,*) 'number of theta singularities = ', isingul
c  write(6,*)

180      continue ! end of inrg index loop

      endif !from if(light.ne.heavy)

      write(6,*) 'out of calculation loops !!'

c
c  now write out detector acceptance function to outfile
c
      open(unit = 18, file = outfile)
      write(6,*)'nbine = ',nbine
      write(18,*) emin,emax,nbine   ! this is for kdaf.f
      do 590 inrg = 1,nbine
        write(18,*) inrg
        do 600 ith = 1,cbins/2
          write (18,1001) thetaval(ith), thdafex(inrg,ith)
600         continue
590      continue

      close (18)

1001     format (f7.1,f16.7)

c  Now create 1-D arrays for each energy-bin-thdaf.  This
c  is done solely for ncarg plotting reasons.  Overwrite
c  this array each time.
c
       call opngks
c
c plot 'isotropic' (beta=0) 1-D detector acceptance D(E)
c
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','D(E)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
```

```
      call agsetc('LINE/TEXT.','Energy$')
      call ezxy(ebinval,suma,nbine,
     >          'Isotropic Detector Acceptance Function$')
c
      do 2000 inrg = 1,nbine
       do 1900 ith = 1,cbins/2
        thdaf(ith) = thdafex(inrg,ith)
1900      continue
c
c        plot normalized theta detector acceptance function
c
        call setusv('PB',2)
        call agsetc('LABEL/NAME.','L')
        call agseti('LINE/NUMBER.',100)
        call agsetc('LINE/TEXT.','P(E,theta)$')
        call agsetc('LABEL/NAME.','B')
        call agseti('LINE/NUMBER.',-100)
        call agsetc('LINE/TEXT.','Theta$')
        call ezxy(thetaval,thdaf,cbins/2,
       >          'Theta Detector Acceptance Function$')
c
2000     continue
c
        call clsgks
        ierr = system(' mv gmeta tdaf.img ')
        if(iwkst.eq.0) ierr = system(' ictrans -d sunview tdaf.img ')
        if(iwkst.eq.1) ierr = system(' ictrans -d t4010 tdaf.img')
        if(iwkst.eq.2) ierr = system(' ictrans -d X11 -bell tdaf.img')
c
        end
```

tdafxdata.f

```
      program thetadafxdata
c
c  this program takes the energy and angle resolved two dimensional
array
c  from cmlwsa, and divides each row of this array in energy by
c  the thetadaf factors previously generated by tdaf.
c  nbine from tdaf output file should aggree exactly
c  with the nbine of fort.4 fort cmlwsa.  Then the daf corrected
c  angular distributions (at each energy) are added together.
c  Care must be taken to use only the section of the data (in
c  theta) which all energies contribute to.
c
c  The output file with error bars
c  (daf corrected KER) is named by the user (ch variable outfile).
c
c  The cmlwsa code is now set to run with theta = 180, and so this
c  code will now use that data too.    DLO 3/94
c
c  Previously, the threshold for the daf (or daf inverse) was
c  implemented in thdafex, throwing away information irrevocably.
c  Now we're implementing the cutoff at the dafxdata stage.  3/93
c
      implicit real*4 (a-h,o-z)
      parameter ( n6 = 150 )  !energy array range
      parameter ( n7 = 180 )  !theta array range
      dimension th(n7),rptherr(n6,n7),cptherr(n6,n7)
      dimension thdaf(n6,n7),thdafmin(n6),thdafmax(n6)
      dimension cpth(n6,n7),ecm(n6),ebinval(n6),pthfinal(n7),pthraw(n7)
      dimension varpthfinal(n7),pthfinalerr(n7),integint(n6)
```

```
      dimension varpthraw(n7),pthrawerr(n7),dafplot(n7),expplot(n7)
      integer anydata1(n6),anydata2(n6),anydaf1(n6),anydaf2(n6)
      dimension expmax(n6),dafmmax(n6)
      integer       expthmin2(n6),expthmax2(n6)
      dimension ithmin2(n6),ithmax2(n6)
      integer   expthmin1(n6),expthmax1(n6),zerocount,totalcount
      dimension ethetc(n6,n7),ithmin1(n6),ithmax1(n6)
      character*20  root,tdaffile,fort9file,cpthfile,outfile
      character*1   comp,graphq

      character*55 title
      character*3 char(174)
c
      data char / ' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9',
     @   '10','11','12','13','14','15','16','17','18','19','20','21',
     @  '22','23','24','25','26','27','28','29','30','31','32','33',
     @  '34','35','36','37','38','39','40','41','42','43','44','45',
     @   '46','47','48','49','50','51','52','53','54','55','56','57',
     @   '58','59','60','61','62','63','64','65','66','67','68','69',
     @   '70','71','72','73','74','75','76','77','78','79','80','81',
     @   '82','83','84','85','86','87','88','89','90','91','92','93',
     @   '94','95','96','97','98','99','100','101','102','103','104',
     @   '105','106','107','108','109','110','111','112','113','114',
     @   '115','116','117','118','119','120','121','122','123','124',
     @   '125','126','127','128','129','130','131','132','133','134',
     @   '135','136','137','138','139','140','141','142','143','144',
     @   '145','146','147','148','149','150','151','152','153','154',
     @   '155','156','157','158','159','160','161','162','163','164',
     @   '165','166','167','168','169','170','171','172','173','174'
/
c

c
c  Prompts for two input files and one output file
c  Also writes single-energy-bin-wide ang dist'ns to cpth.dat
c  Also writes raw data in energy interval to fort.73
c
          write(*,*)'Enter 2 for X11, 3 for batch'
          read(*,*)iwkst
      write(6,*)'Enter root name of tdaf file (from thdafex)'
      read(5,*) root
      call addsuffix(root,'.tdaf',tdaffile)
c     write(6,*)'Enter tdaf lower cutoff, in % of maximum daf'
c     read(5,*) tdafcutoff
      write(6,*)'Enter name of raw data file (i.e., fort.9)'
      read(5,*) fort9file
c
c         read in theta angular distribution, but first read through
the
c  other junk in fort.8 ahead of the theta information
c
      write(6,*)'Reading raw data from ',fort9file
      open(unit=9,file = fort9file, status='old')
      read(9,*) nbine
      write(6,*)nbine
          read(9,5000)(ecm(i),i=1,nbine)
      write(6,5000)(ecm(i),i=1,nbine)
      nbine1 = 1
      nbine2 = nbine
      ewinmin = ecm(1)
      ewinmax = ecm(nbine)+ecm(2)-ecm(1)
          read(9,*)nth
      if (nth.ne.180) then
        write(6,*)'nth=',nth
```

```
      write(6,*) 'nth not equal to 180 bozo!'
      write(6,*) 'abfit needs to have nth = 180 for its indexing'
      stop
    end if
        read(9,5000)(th(i),i=1,nth)
c   cmlwsa has th(i) from 0-179: I use instead th(i) from 0.5-179.5
    do i = 1,nth
     th(i) = th(i) + 0.5
    enddo
c


        do j = 1,nth
          read (9,5020)(ethetc(i,j),i=1,nbine)
        enddo
5000    format(6(1x,f8.4))
5020    format(6(1x,e10.5))
c
c  read in all thdaf functions from tdaffile
c
    write(6,*)'Reading thetadaf from ',tdaffile
    open (unit=18,file = tdaffile, status = 'old')
    read(18,*) emin,emax,nbineprime
    if (nint(ewinmin*1000).ne.nint(emin*1000)) stop 'emin
imcompatible!'
    if (nint(ewinmax*1000).ne.nint(emax*1000)) stop 'emax
imcompatible!'
    if (nbine.ne.nbineprime) stop 'nbine imcompatible!'
    erange = emax - emin
    step = erange/nbine
    do 80 j = 1,nbine
     read(18,*) iwin
     read(18,*)(trsh1,thdaf(iwin,ith),ith=1,nth)
80  continue
c
    close(unit = 18)  ! done with tdaffile
c
c  find tdaf cutoff for each energy bin
c
27  write(6,*)'Enter tdaf lower cutoff, in % of maximum daf'
    read(5,*) tdafcutoff
    write(6,*)'Finding thetadaf threshold for each energy bin.'
    do iwin = 1,nbine
      thdafmax(iwin) = 0.0
      anydaf1(iwin) = 0
      anydaf2(iwin) = 0
      anydata1(iwin) = 0
      anydata2(iwin) = 0
      do ith = 1,nth
        if ( thdaf(iwin,ith).gt.thdafmax(iwin) )
      @                      thdafmax(iwin)=thdaf(iwin,ith)
      end do
      thdafmin(iwin)=thdafmax(iwin)*(tdafcutoff/100)
      do ith = 1, nth/2
        if (thdaf(iwin,ith).gt.thdafmin(iwin)) anydaf1(iwin)=1
        if (ethetc(iwin,ith).gt.0.0) anydata1(iwin) = 1
      end do
      do ith = (nth/2)+1, nth
        if (thdaf(iwin,ith).gt.thdafmin(iwin)) anydaf2(iwin)=1
        if (ethetc(iwin,ith).gt.0.0) anydata2(iwin) = 1
      end do
    end do

    zerocount = 0
```

```
      totalcount = 0
      do i = 1,nbine
        do j = 1,nth
          if(ethetc(i,j).eq.0.0) zerocount = zerocount + 1
          if(ethetc(i,j).ge.0.0) totalcount = totalcount + 1
        enddo
      enddo
      write(6,*)'# of matrix elements with value zero = ',zerocount
      write(6,*) 'total counts = ',totalcount
c
c  do multiplication of each thetadaf by each row of fort.9 and
generate
c  error bars
c
c  To add the cpth(ith) values for a range of energies together,
c  you must add only the angular range in which all energy
c  values have finite probabilities.  E.g., you will never add
c  over theta = 0 --> ~16 degrees, since the blocker always
c  discriminates there (horizontal polarization).  You must pick the
c  largest block of
c  theta values for which each energy has a daf value above your
cutoff.
c
      write(6,*)'Finding theta acceptance window,'
      write(6,*)'and calculating cpth array.'
      do iwin = 1,nbine
       ithmin1(iwin) = 0
       ithmax1(iwin) = 91
       ithmin2(iwin) = 90
       ithmax2(iwin) = 181
       expthmin1(iwin) = 0
       expthmax1(iwin) = 91
       expthmin2(iwin) = 90
       expthmax2(iwin) = 181
      enddo
c
      do 90 iwin = 1, nbine
c  interrogate each row (iwin index) of thdaf(iwin,ith)
c  to find what range of theta has above-threshold probabilities.
c
c        first half of theta dist'n
c
      if (anydaf1(iwin).eq.0) then  ! if thdaf(iwin,ith)=0 for all ith
        ithmin1(iwin)=45
        ithmax1(iwin)=45
      else
96      ithmin1(iwin) = ithmin1(iwin) + 1
        if( thdaf(iwin,ithmin1(iwin)).lt.thdafmin(iwin) ) goto 96
97      ithmax1(iwin) = ithmax1(iwin) - 1
        if( thdaf(iwin,ithmax1(iwin)).lt.thdafmin(iwin) ) goto 97
      end if
c
c        second half of theta distribution
c
      if (anydaf2(iwin).eq.0) then  ! if thdaf(iwin,ith)=0 for all ith
        ithmin2(iwin) = 135
        ithmax2(iwin) = 135
      else
196         ithmin2(iwin) = ithmin2(iwin) + 1
        if( thdaf(iwin,ithmin2(iwin)).lt.thdafmin(iwin) ) goto 196
197         ithmax2(iwin) = ithmax2(iwin) - 1
        if( thdaf(iwin,ithmax2(iwin)).lt.thdafmin(iwin) ) goto 197
      end if
c
```

```
c do the same again, but looking for data
c
   if (anydata1(iwin).eq.0) then  ! if ethetc(iwin,ith)=0 for all
theta
     expthmin1(iwin)=0
     expthmax1(iwin)=0
   else
98   expthmin1(iwin) = expthmin1(iwin) + 1
     if( ethetc(iwin,expthmin1(iwin)).eq.0.0 ) goto 98
99   expthmax1(iwin) = expthmax1(iwin) - 1
     if( ethetc(iwin,expthmax1(iwin)).eq.0.0 ) goto 99
   end if
   if (anydata2(iwin).eq.0) then  ! if ethetc(iwin,ith)=0 for all
theta
     expthmin2(iwin)=0
     expthmax2(iwin)=0
   else
198        expthmin2(iwin) = expthmin2(iwin) + 1
     if( ethetc(iwin,expthmin2(iwin)).eq.0.0 ) goto 198
199        expthmax2(iwin) = expthmax2(iwin) - 1
     if( ethetc(iwin,expthmax2(iwin)).eq.0.0 ) goto 199
   end if

c
   do 100 ith = 1,nth
c
c  ethetc(iwin,ith) is the raw energy,theta array from cmlwsa
c  cpth(iwin,ith) is the thdafex corrected angular distribution
c
c  This is the most microscopic I(theta,E) information that we get
c  Because these bins don't hold very many counts, we tend to sum
c  over a few energy bins to get good statistics. However, let's
c  write out this info just for kicks.
c
c  if thdaf is zero, set it to 1e-20; none of the
c  resulting cpth elements will ever be used later in the program
c
      if (thdaf(iwin,ith).eq.0) thdaf(iwin,ith) = 1e-20
c
       rptherr(iwin,ith) = sqrt(ethetc(iwin,ith))
       cpth(iwin,ith) = ethetc(iwin,ith)/thdaf(iwin,ith)
       cptherr(iwin,ith) = rptherr(iwin,ith)/thdaf(iwin,ith)
c
100       continue   !theta loop
90 continue    !iwin (energy index) loop
c
        write(6,*)'Enter root name of .cpth file: '
   read(5,*) root
     call addsuffix(root,'.cpth',cpthfile)
     write(6,*)'Writing out ',cpthfile
     open(33,file = cpthfile)
     write(33,*) emin,emax,nbine
     do iwin = 1,nbine
       ebinval(iwin) = emin + (iwin - 0.5)*step
c Write out this bin's energy
       write(33,*) iwin,ebinval(iwin),(ithmax1(iwin)-
ithmin1(iwin)+1),
     @                (ithmax2(iwin)-ithmin2(iwin)+1)
c Write out the corrected intensities, with errors
c        first half
       do ith = ithmin1(iwin),ithmax1(iwin)
         if (cpth(iwin,ith).ne.0) then
           write(33,*) ' ',ith,cpth(iwin,ith),cptherr(iwin,ith)
         else  ! if no counts, write out one count error
```

```
          write(33,*) ' ',ith,cpth(iwin,ith),1/thdaf(iwin,ith)
        end if
      end do
c       second half
      do ith = ithmin2(iwin),ithmax2(iwin)
        if (cpth(iwin,ith).ne.0) then
          write(33,*) ' ',ith,cpth(iwin,ith),cptherr(iwin,ith)
        else  ! if no counts, write out one count error
          write(33,*) ' ',ith,cpth(iwin,ith),1/thdaf(iwin,ith)
        end if
      end do
    end do
    close(33)

c  Now add option of comparing, for a given E, the range of theta
c  which was observed from the data and the range of theta that the
c  DAF predicts is possible.

c  create angle integrated raw distribution
   do i = 1,nbine
    integint(i) = 0.0
   enddo
   do i = 1,nbine
    do j = 1,nth
     integint(i) = integint(i) + ethetc(i,j)
    enddo
   enddo

   write(6,*)'Make comparison of th ranges bwt. data and DAF? (y/n)'
650      read(5,*) comp
   if (comp.eq.'y') then
     write(6,*)'Energy    Bin#    Intens   Data        DAF
     @  Data        DAF'
     write(6,*)'                              min  max   min  max
     @  min  max  min  max'
     do iwin=1,nbine
       write(6,6000)ebinval(iwin),iwin,integint(iwin),
     @               expthmin1(iwin),expthmax1(iwin),
     @               ithmin1(iwin),ithmax1(iwin),
     @               expthmin2(iwin),expthmax2(iwin),
     @               ithmin2(iwin),ithmax2(iwin)
     enddo
   elseif(comp.ne.'n') then
     write(6,*)'Please enter [y] or [n]'
     goto 650
   endif
   write(6,*)'Would you also like a graphical comparison? (y/n)'
   read(5,*)graphq
   if(graphq.eq.'y') goto 7000
5900     continue

6000    format(f7.5,3x,i3,3x,i6,3x,i3,2x,i3,3x,i3,2x,i3,2x,
     @         i3,2x,i3,2x,i3,2x,i3)
8000     write(6,*)'Do you want to try a different cutoff value to'
   write(6,*)'compare with the data?'
   read(5,*) comp
   if(comp.eq.'y') goto 27

        write(6,*)'Enter 0 to end program here,'
   write(6,*)'  1 to integrate P(E,theta) over energy range: '
        read(*,*) iend
        if(iend.eq.0) goto 1000
c*************************************************************
c  allow user to select different energy window
```

```
c  THIS IS THE RE-ENTRY POINT FOR SUBSEQUENT WINDOWS
c
10 write(*,*)'Current Energy range is'
   write(*,*)ewinmin,' to ',ewinmax
   write(*,*)'or bin ranges ',nbine1,' to ',nbine2
   write(*,*)
   write(*,*)'Enter (0) to use a new energy range (1)'
   write(*,*)
   read(5,*)ichoice
   if(ichoice.eq.0) then
    write(*,*)'enter new values of min. and max. Energy'
    read(5,*)ewinmin,ewinmax
    write(6,*)'if you get non-sensical results, try a smaller energy'
    write(6,*)'range that is centered about your data'
    nbine1 = nint( nbine*(ewinmin - emin)/(emax - emin) + 1 )
    nbine2 = nint( nbine*(ewinmax - emin)/(emax - emin) )
    if(nbine2.gt.nbine) then
     write(6,*)'*************************************'
     write(6,*)'Screw up:  bin number ',nbine2,' was attempted'
     nbine2 = nbine
     write(6,*)'Max bin is now set to ',nbine2
     write(6,*)'*************************************'
    endif
    if (nbine1.gt.nbine2) then
     write(6,*)'That''s a mighty narrow energy window!!'
     nbine2 = nbine1
    endif
    write(*,*) 'multiplication of daf and data will proceed from'
    write(*,*) '(energy) bin ', nbine1, ' to bin ',nbine2
   endif
   write(6,*)'Enter output file name'
   read(5,*) outfile
c
c
c  Find the maximum ithmin1(iwin) and the minimum ithmax1(iwin)
c  and add together cpth(iwin,ith) for each iwin value ONLY
c  over this range.
c
   iabsthmin1 = 0
   iabsthmax1 = 90
   iabsthmin2 = 90
   iabsthmax2 = 180
   do iwin = nbine1,nbine2
    if (ithmin1(iwin).ge.iabsthmin1) then
     iabsthmin1 = ithmin1(iwin)
    endif
    if (ithmax1(iwin).le.iabsthmax1) then
     iabsthmax1 = ithmax1(iwin)
    endif
    if (ithmin2(iwin).ge.iabsthmin2) then
     iabsthmin2 = ithmin2(iwin)
    endif
    if (ithmax2(iwin).le.iabsthmax2) then
     iabsthmax2 = ithmax2(iwin)
    endif
   enddo
c
   write(*,*)'The usable (tdaf) range of THETA for integrating
P(theta,E)'
   write(*,*)'over Energy is:'
   write(*,*)iabsthmin1,' to ',iabsthmax1,' and ',
     @         iabsthmin2,' to ',iabsthmax2,' degrees '
c
c  now, add together the rows of the cpth(iwin,ith) matrix
```

```
c  to give the angular distribution for the useable window
c
   do ith = 1,nth
    pthfinal(ith) = 0.0
    varpthfinal(ith) = 0.0
   enddo
c
c Here comes the final result (in two parts, first and second half)
c
c        first half of theta distribution:
   do ith = iabsthmin1,iabsthmax1
    do iwin = nbine1,nbine2
     pthfinal(ith) = pthfinal(ith) + cpth(iwin,ith)
     varpthfinal(ith) = varpthfinal(ith) + (cptherr(iwin,ith))**2
    enddo  ! end of energy bin loop
c
c Find theta bins for which the variance is zero (i.e., no counts in
that
c  theta bin), and set the variance equal to the value it would have
if there
c  had been one count in the middle bin of the energy window
c This fix avoids having any variances = 0, which screw up the
betafits
c
    if (varpthfinal(ith).lt.1E-30) then
     if(pthfinal(ith).ne.0) stop 'screw-up with zero variances!'
     varpthfinal(ith) = 1/thdaf((nbine1+nbine2)/2,ith)**2
    end if
   enddo  ! end of theta bin loop

c        second half of theta distribution:
   do ith = iabsthmin2,iabsthmax2
    do iwin = nbine1,nbine2
     pthfinal(ith) = pthfinal(ith) + cpth(iwin,ith)
     varpthfinal(ith) = varpthfinal(ith) + (cptherr(iwin,ith))**2
    enddo  ! end of energy bin loop
    if (varpthfinal(ith).lt.1E-30) then
     if(pthfinal(ith).ne.0) stop 'screw-up with zero variances!'
     varpthfinal(ith) = 1/thdaf((nbine1+nbine2)/2,ith)**2
    end if
   enddo  ! end of theta bin loop
c
c  create raw data plots with error bars too.
c
   do iwin = nbine1,nbine2
    do ith = 1,nth
c
     pthraw(ith) = pthraw(ith) + ethetc(iwin,ith)
     varpthraw(ith) = varpthraw(ith) + (rptherr(iwin,ith))**2
c
    enddo
   enddo
c
c  now find the st.dev. in quadrature for each energy value
   do ith = iabsthmin1,iabsthmax1
    pthfinalerr(ith) = sqrt(varpthfinal(ith))
   enddo
c
   do ith = 1,nth
    pthrawerr(ith) = sqrt(varpthraw(ith))
   enddo
c
   do i = 1,nth
     write(73,*) th(i),pthraw(i),pthrawerr(i)
```

```
      enddo
c
      open (unit=74,file = outfile)
      do i = iabsthmin1,iabsthmax1
        write(74,7400) th(i),pthfinal(i),pthfinalerr(i)
      enddo
      do i = iabsthmin2,iabsthmax2
        write(74,7400) th(i),pthfinal(i),pthfinalerr(i)
      enddo
      close(unit=74)
7400     format(F6.1,F14.7,F14.7)
c
c  now, plot distribution using NCAR.
c
      if(iwkst.eq.3) goto 1234
           call opngks


c                 Plot (uncorrected) theta angular distribution
           call setusv('PB',2)
           call agseti('SET.',1)
           call agseti('FRAME.',1)
           call agsetc('LABEL/NAME.','L')
           call agseti('LINE/NUMBER.',100)
           call agsetc('LINE/TEXT.','P(theta)$')
           call agsetc('LABEL/NAME.','B')
           call agseti('LINE/NUMBER.',-100)
           call agsetc('LINE/TEXT.','CM Angle, theta (deg)$')
           call ezxy(th,pthraw,nth,'(uncorr) CM Angular Dist. (Theta)$')

c                 Plot DAF corrected theta angular distribution
           call setusv('PB',2)
           call agsetc('LABEL/NAME.','L')
           call agseti('LINE/NUMBER.',100)
           call agsetc('LINE/TEXT.','P(theta)$')
           call agsetc('LABEL/NAME.','B')
           call agseti('LINE/NUMBER.',-100)
           call agsetc('LINE/TEXT.','CM Angle, theta (deg)$')
           call ezxy(th,pthfinal,nth,'DAF corr. CM Angular Dist.
(Theta)$')
c
           call clsgks
c
           ierr = system(' mv gmeta tdist.img ')
           if(iwkst.eq.0) ierr = system(' ictrans -d sunview tdist.img
')
           if(iwkst.eq.1) ierr = system(' ictrans -d t4010 tdist.img')
           if(iwkst.eq.2) ierr = system(' ictrans -d X11 tdist.img')
c
1234      write(*,*)
      write(*,*)'Enter (0) to try a different energy range (1)'
      write(*,*)
      read(5,*)ich2
      if(ich2.eq.0) then
       goto 10
      else
       stop 'THANK YOU FOR USING MY CPU!!'
      endif
999       continue
7000      if(graphq.ne.'y') goto 1000
c  now, plot distribution using NCAR.
c

c  first find max of each following array for each energy
```

```
      do i = 1,nbine
       do j = 1,nth
        expmax(i) = 0
        dafmmax(j) = 0
       enddo
      enddo

      do inrg = 1,nbine
       do ith = 1,nth
        if(ethetc(inrg,ith).gt.expmax(inrg)) then
          expmax(inrg) = ethetc(inrg,ith)
        endif
        if(thdaf(inrg,ith).gt.dafmmax(inrg)) then
          dafmmax(inrg) = thdaf(inrg,ith)
        endif
       enddo
      enddo


      if(iwkst.eq.3) goto 1234
           call opngks

      do 2000 inrg = 1,nbine
c     inrg = 1
       do 1900 ith =1,nth
        expplot(ith) = ethetc(inrg,ith)
        dafplot(ith) = thdaf(inrg,ith)/dafmmax(inrg)*expmax(inrg)
1900       continue
        title = 'Raw data and DAF for energy bin '
        @   //char(inrg)//'$'


c                 Plot (uncorrected) theta angular distribution
      call agseti('FRAME.',2) ! two plots per frame
           call setusv('PB',2)
           call agsetc('LABEL/NAME.','L')
           call agseti('LINE/NUMBER.',100)
           call agsetc('LINE/TEXT.','P(thet)$')
           call agsetc('LABEL/NAME.','B')
           call agseti('LINE/NUMBER.',-100)
           call agsetc('LINE/TEXT.','CM Angle, thet (deg)$')
           call ezxy(th,expplot,nth,title)
           call points(th,dafplot,nth,-1,0)
      call FRAME

2000       continue
           call clsgks
c
           ierr = system(' mv gmeta tdis.img ')
           if(iwkst.eq.0) ierr = system(' ictrans -d sunview tdis.img ')
           if(iwkst.eq.1) ierr = system(' ictrans -d t4010 tdis.img')
           if(iwkst.eq.2) ierr = system(' ictrans -d X11 -bell
tdis.img')


      goto 8000

1000       stop
           end
```

Appendix C

MAVABFIT[a]: Energy and angular distribution analysis

I. Introduction

The final step in analysis of time- and position-sensitive data is the resolution of the 2-D $P_{true}(E_T,\theta)$ distribution into the two 1-D distributions $P(E_T)$, the translation energy distribution, and $\beta(E_T)$, the energy dependent anisotropy parameter. The code MAVABFIT accomplishes this resolution by a linear least-squares fit of the function

$$P_{true}(E_T,\theta) = P(E_T)\cdot\{1 + \beta(E_T)\cdot P_2(\cos\theta)\}$$

where $P(E_T)$ and $\beta(E_T)$ are the fitting parameters.

In order to obtain reasonable statistics on the function $\beta(E_T)$, is is necessary to assume that over a small enough window in E_T, the value of $\beta(E_T)$ is constant. In practice, the data is divided into translational energy windows of some fixed width, for example 0.4 eV, and E' is defined as a running parameter such that $E' \leq E_T \leq (E' + 0.4\text{ eV})$. β is determined from a linear least squares fit of the distribution $P_{true}(E_T,\theta)$ with both $P(E_T)$ and β as active fitting parameters. The quantity E' is then incremented in 30 meV steps beginning with $E'= 0$ and the fitting procedure is repeated until the entire energy range of the data is covered. In other words, $\beta(E_T)$ is constructed as a moving average over 0.4 eV windows in E_T. Using these values of $\beta(E_T)$, a second linear least squares fit in the single parameter $P(E_T)$ is determined for each individual energy bin, completing the separation of $P_{true}(E_T,\theta)$ into two one dimensional distributions, i.e., $P(E_T)$ and $\beta(E_T)$. An older version

[a] MAVABFIT stands for Moving AVerage Alpha-Beta FIT

of the code, ABFIT, is available if a moving average is not desired. The core of this program was written by Dr. David Leahy

II. Routines comprising the MAVABFIT program

MAVABFIT is compiled from the following routines and subroutines:

mavabfit.f: main calling routine

addsuffix.f: concatenates a prefix and a suffix string

interp.f: does linear interpolation between known function values.

The following routines are from Numerical Recipes[a]

covsrt.f: creates covariance matrix

fleg.f: calculates Legendre polynomials

gammln.f: gamma function routine

gammq.f: gamma function routine

gaussj.f: Gauss-Jordon matrix inversion

gcf.f: gamma function routine

gser.f: gamma function routine

lfit.f: linear least squares fit to a Legendre polynomial

MAVABFIT can be compiled on a UNIX platform by executing the `make` command:

```
make -f makemavabfit
```

[a] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes* (Cambridge University Press, Cambridge, 1992).

where the makefile is:

makemavabfit

```
FFLAGS= -C
mavabfit: mavabfit.o lfit.o gaussj.o covsrt.o fleg.o gammq.o gser.o
gcf.o gammln.o addsuffix.o interp.o
   ncargf77 mavabfit.o lfit.o gaussj.o covsrt.o fleg.o gammq.o gser.o
gcf.o gammln.o addsuffix.o interp.o -o mavabfit
```

III. Usage

A. Input

All user input to MAVABFIT is given interactively and is self-explanatory. MAVABFIT reads as input the `*.cpth` file produced by TDAFXDATA (cf. Appendix B). It is possible to fix the parameter β such that only the parameter $P(E_T)$ is adjusted in the fit. However, only the value of $\beta = 0$ may be chosen if β is fixed.

B. Output

MAVABFIT produces several output files which are explained below. Again, the * in the filenames corresponds to the prefix established by the TDAF execution.

***mav.angd**: the complete angular distribution information at each value of E_T at which $\beta(E_T)$ was determined. The angular distribution is presented consecutively for each value of E_T by three columns which consist of θ as the first column, the experimental values of $P(\theta)$ as the second column, and the values $\{1 + \beta(E_T) \cdot P_2(\cos\theta)\}$ with E_T = constant as the third column.

***mav.aoe**: three column file containing the translational energy distribution as E_T, $P(E_T)$, $\sigma\{P(E_T)\}$, where the final column gives the error bars (± 1 standard deviation) for $P(E_T)$.

***mav.boe**: three column file containing E_T, $\beta(E_T)$, $\sigma\{\beta(E_T)\}$.

***mav.saoe**: the same file as ***mav.aoe**, but without the third column.

***mavl.aoe**: three column file similar to ***mav.aoe**, except that the final least squares fitting to obtain $P(E_T)$.is done using the lower error bar values of $\beta(E_T)$ rather than the average values of $\beta(E_T)$. This gives an estimate of the error introduced into the $P(E_T)$.from the error in $\beta(E_T)$.

***mavu.aoe**: three column file corresponding to ***mavl.aoe** but for the upper error bars of $\beta(E_T)$.

IV. FORTRAN CODE

MAVABFIT.f

```
      program mavabfit
c This program reads in data from cpth.dat, and fits each energy
c bin.  Yow.          DJL 2/14/93
c Added code to integrate area under alphafit KER curve for branching
ratio
c concerns.           DRC 4/07/93
c Added code to give some simple statistics about KER curve.
c Included in this is:
c
c         Average P(Etrans)
c         Most Probable P(Etrans)
c         Highest KER observed
c
c This version first fits beta to windows whose energy widths depend
on
c the data, and on the user-entered parameter 'minevents'.
c It beta-fits all of the cpth data to these variable width windows,
c and then goes on to do an alphafit using interpolated beta curves.
c               DJL 5/25/93
c Full theta range: requires that .cpth input file is binned in
c one degree theta steps, with a total of 180 bins possible; the
.cpth
c file gives explicitly the number of dafxdata-filled bins for both
the
c first (0 < theta < 90) and second (90 < theta < 180) quadrants.
```

```
c               DJL 3/28/94

c Finally the moving average code is written.  This code takes abfit
c of 4/23/96 (with a few very minor bugs fixed) and adds the moving
c average capability to beta determinations.
c               DLO 4/25/96
      implicit none
      integer n2,n3,mmax
      parameter (n2 = 300)     ! max # of energy bins
      parameter (n3 = 180)     ! max # of angle bins
      parameter (mmax = 3)
      real*4 enrg(n2),emin,emax,erange,step,system,nrgdiff
      real*4 xra(n3),yra(n3),fit(n3)
      real*4 cpth(n2,n3),sig(n2,n3),cpthsum(n2,n3),varsum(n2,n3)
      real*4 sigfit(n3),covar(mmax,mmax),chisq,a(mmax)
      integer lista(mmax),ma,mfit,ncvm
      real*4 xx,xxsum,xysum,yysum
      real*4 alpha(n2),chisqr(n2)
      real*4 alphasig(n2),alphacent(n2)
      real*4 slope,intercept,pi,costheta
      real*4 nrg(n2),betasig(n2),beta(n2)
      real*4 betacent(n2),betaupper(n2),betalower(n2),betaorig(n2)
      real*4 betau(n2),betal(n2),alphau(n2),alphal(n2)
      real*4 alphausig(n2),alphalsig(n2),chisqru(n2),chisqrl(n2)
           real*4 alphaavg,alphaavgl,alphaavgu
           real*4 alphamp,alphamaxker,alphaint,alphaintl,alphaintu
      real*4 eventsum,nrgsum,ebinsum,event_threshold
      real*4 theta_val,cpth_val,sig_val,fixbetaval
c     integer lowedge(n2)
      integer ithmin(n2),ithmax(n2),nth(n2),minevents,events(n2)
      integer ithmin2(n2),ithmax2(n2),nth2(n2)
      integer ibad(n2),i,iwin,ith,index,nbetawin,nbine,ii
      integer inrgmin(n2),inrgmax(n2),ithminwin(n2),ithmaxwin(n2)
      integer ithminwin2(n2),ithmaxwin2(n2)
      integer nthwin(n2)
      integer writepass,nfile,ierr,iwkst
      character*20 betaroot,betaint,betafile,cpthroot,cpthfile,alphafile
      character*20 angdistfile,salphafile,angbinfile,tempname
      character*55 title
      character*3 char(174)
      character*1 fixbeta,steptype
      real*4 winwidtheV
      integer winwidth,iwinstart

c     ***********************************************
           character*25 titleorg
           open(unit=77,file="title.dat",status='old')
           read(77,3000) titleorg
3000       format(a25)
           write(*,*) titleorg
c***********************************************
c
      data char / ' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9',
     @   '10','11','12','13','14','15','16','17','18','19','20','21',
     @  '22','23','24','25','26','27','28','29','30','31','32','33',
     @  '34','35','36','37','38','39','40','41','42','43','44','45',
     @   '46','47','48','49','50','51','52','53','54','55','56','57',
     @   '58','59','60','61','62','63','64','65','66','67','68','69',
     @   '70','71','72','73','74','75','76','77','78','79','80','81',
     @   '82','83','84','85','86','87','88','89','90','91','92','93',
     @   '94','95','96','97','98','99','100','101','102','103','104',
     @   '105','106','107','108','109','110','111','112','113','114',
     @   '115','116','117','118','119','120','121','122','123','124',
     @   '125','126','127','128','129','130','131','132','133','134',
```

```
     @   '135','136','137','138','139','140','141','142','143','144',
     @   '145','146','147','148','149','150','151','152','153','154',
     @   '155','156','157','158','159','160','161','162','163','164',
     @   '165','166','167','168','169','170','171','172','173','174'
/
c

*******************Section 1:  Get inputs from user******************
   pi = 3.1415926
   write(6,*) 'Enter 2 for X11, 3 for batch'
   read(5,*) iwkst
   write(6,*) 'Event threshold for individual energy bin inclusion?'
   read(5,*) event_threshold

   write(6,*) 'Enter root name of .cpth file'
   read(5,*) cpthroot
   call addsuffix(cpthroot,'.cpth',cpthfile)
17 write(6,*) 'Fix beta at a given value? [y/n]'
   read(5,*) fixbeta
   if(fixbeta.eq.'y') then
    write(6,*) 'Enter value of beta to fix at'
    read(5,*) fixbetaval
     if(fixbetaval.ne.0.0) stop 'can only fix beta at zero!!!'
   elseif(fixbeta.ne.'n') then
    write(6,*) 'Please enter [y]es or [n]o, squidbreath'
    goto 17
   endif

*******************Section 2:  Read in 2-D P(E,theta) file********

   write(6,*) 'Reading in corrected cpth(E,theta) from ',cpthfile
   open(15,file = cpthfile, status = 'old')
   read(15,*) emin,emax,nbine  ! read in energy scale parameters
   erange = emax - emin
   step = erange/nbine
   write(6,*)
   write(6,*) 'Energy runs from',emin,' to',emax,' in',nbine,' bins,'
   write(6,*) 'amounting to ',1000*step,' meV/bin.'
   write(6,*)
c  write(6,*)'Enter the moving average spacing in bins [1 or higher]'
c  read(5,*)spacing
   write(6,*)'Currently the moving av. spacing is 1 bin'

   write(6,*)
   write(6,*) 'Determine betawindow width from # of [e]vents
     @or fixed energy [r]ange?'
101       read(5,*) steptype
   if(steptype.eq.'e') then
     write(6,*) 'Minimum number of events required for betafit step?'
     read(5,*) minevents
   elseif(steptype.eq.'r') then
102         write(6,*) 'Enter the width of betawindows in eV'
     read(5,*) winwidtheV
     winwidth = nint( winwidtheV / step )
     if(winwidth.eq.0) then
      write(6,*)'Choose a larger window width, furball'
      goto 102
     endif
c    numbetawin = nbine - winwidth + 1
c    do i = 1,numbetawin
c     lowedge(i) = i        !value in ebins
c    enddo
c   lowedge(numbetawin + 1) = emax !need this upper limit later
c   write(6,*)numbetwin,' betawindows created, each ',winwidth,
```

```
c     @         ' bins wide'
   else
     write(6,*)'Enter ''e'' or ''r'', you rubber rat'
     goto 101
   endif !if(steptype.eq.'e')

   do i=1,nbine
     read(15,*) ii,enrg(i),nth(i),nth2(i)
c    write(6,*) i,ii,enrg(i),(emin+(i-0.5)*step)
     nrgdiff = (1000*enrg(i)) - (1000*(emin+(i-0.5)*step))
     if ( (nrgdiff.gt.1e-2).or.(ii.ne.i) ) then
c      write(6,*) nint(1000*enrg(i)),nint(1000*(emin+(i-0.5)*step))
       write(6,4200) 'i = ',i,'  nrg =',1000*(emin+(i-0.5)*step)
       write(6,4200) 'ii = ',ii,' enrg =',1000*enrg(i)
4200            format(A5,i3,A8,F16.9)
              stop 'bad energy index'
     endif
     do index = 1,nth(i)+nth2(i)
       read(15,*) theta_val,cpth_val,sig_val
       ith = theta_val
       if (index.eq.1) then ! record theta limits for this energy bin
         ithmin(i) = ith
         ithmax(i) = ith + nth(i) - 1
       end if
       if ( index .eq. (nth(i)+1) ) then  ! 2nd quad theta limits
         ithmin2(i) = ith
         ithmax2(i) = ith + nth2(i) - 1
       end if
       cpth(i,ith) = cpth_val
       sig(i,ith) = sig_val
     end do ! index (theta bin) loop
c    write(6,*) nth(i)+nth2(i),' points read for energy =',enrg(i)
   end do      ! energy bin loop
   close(15)

*******************Section 3:  Define Betawindows
(thmin,max,Ewidth)**

c
c Step through energy index of cpth and establish a beta-fitting
c window, delimited in energy by inrgmin and inrgmax.
c Accumulate angular distributions in cpthsum array.
c Then fit the data to get beta.  Finally repeat by defining the next
c beta window.
c ---> i is energy bin index
c
c  i = 0              ! index of energy bins (increment by 1)
   iwinstart = 0      !index of starting energy bin for each window
   iwin = 0
100      iwin = iwin + 1          ! index of betawindows
   inrgmin(iwin) = iwinstart + 1 !lowest ebin in this betawindow
   i = iwinstart
   eventsum = 0
   nrgsum = 0
c
c ithminwin and ithmaxwin are the theta limits for each beta window
iwin.
c They represent the most stringent ithmin and ithmax values within
the
c energy interval of this beta window.
c
   ithminwin(iwin) = 0
   ithmaxwin(iwin) = 90
   ithminwin2(iwin) = 91
```

```
   ithmaxwin2(iwin) = 180

c Begin a betawindow

120      i = i + 1
c calculate number of events in energy bin i; if too small then skip
e bin
c ebinsum is the number of RAW events in at energybin i
c nrgsum/eventsum is <E> = SUM[E*P(E)]/SUM[P(E)]
   ebinsum = 0.0
   do ith = ithmin(i),ithmax(i) ! first quandrant in theta
     ebinsum = ebinsum + (cpth(i,ith)/sig(i,ith))**2
   end do
   do ith = ithmin2(i),ithmax2(i) ! 2nd quadrant in theta
     ebinsum = ebinsum + (cpth(i,ith)/sig(i,ith))**2
   end do

        if(ebinsum.ge.event_threshold) then
     if (ithmin(i).gt.ithminwin(iwin)) ithminwin(iwin)=ithmin(i)
     if (ithmax(i).lt.ithmaxwin(iwin)) ithmaxwin(iwin)=ithmax(i)
     if (ithmin2(i).gt.ithminwin2(iwin)) ithminwin2(iwin)=ithmin2(i)
     if (ithmax2(i).lt.ithmaxwin2(iwin)) ithmaxwin2(iwin)=ithmax2(i)
     eventsum = eventsum + ebinsum
     do ith = ithmin(i),ithmax(i)       ! sum over 1st quad in
theta...
       nrgsum = nrgsum + enrg(i)*(cpth(i,ith)/sig(i,ith))**2
       cpthsum(iwin,ith) = cpthsum(iwin,ith) + cpth(i,ith)
       varsum(iwin,ith) = varsum(iwin,ith) + sig(i,ith)**2
     end do
     do ith = ithmin2(i),ithmax2(i)     ! ...and 2nd quad in theta
       nrgsum = nrgsum + enrg(i)*(cpth(i,ith)/sig(i,ith))**2
       cpthsum(iwin,ith) = cpthsum(iwin,ith) + cpth(i,ith)
       varsum(iwin,ith) = varsum(iwin,ith) + sig(i,ith)**2
     end do
        endif

   if(steptype.eq.'e') then
     if ((eventsum.lt.minevents).and.(i.lt.nbine)) goto 120
   else  !using fixed width windows
     if ( i.ge.iwinstart + winwidth) then
       goto 150 !move on to next betawindow
     else
       goto 120 !not so fast--have to finish this betawindow
     endif
   endif
150      inrgmax(iwin) = i ! final assignments for window iwin
   if (ithminwin(iwin).gt.ithmaxwin(iwin)) then ! check for window
collapse
     ithminwin(iwin) = 45
     ithmaxwin(iwin) = 45
   end if
   if (ithminwin2(iwin).gt.ithmaxwin2(iwin)) then
     ithminwin2(iwin) = 135
     ithmaxwin2(iwin) = 135
   end if
   nthwin(iwin) = (ithmaxwin(iwin) - ithminwin(iwin) + 1)
     @          + (ithmaxwin2(iwin) - ithminwin2(iwin) + 1)
   events(iwin) = eventsum

   if(eventsum.lt.1) then
    write(6,*) events(iwin),' events between theta limits of ',
     @         ithminwin(iwin),' -',ithmaxwin(iwin),' and',
     @         ithminwin2(iwin),' -',ithmaxwin2(iwin)
    write(6,*) 'Therefore betafit will not be done.'
```

```
     iwin = iwin - 1  !don't count this empty window
     i=nbine    !this will cause exit
     goto 570
    endif

    nrg(iwin) = nrgsum/eventsum   !<E>
    write(6,*)'Beta window',iwin,':',
      @   enrg(inrgmin(iwin)),enrg(inrgmax(iwin)),'  Mean:',nrg(iwin)
    write(6,*) events(iwin),' events between theta limits of ',
      @         ithminwin(iwin),' -',ithmaxwin(iwin),' and',
      @         ithminwin2(iwin),' -',ithmaxwin2(iwin)

c  if (i.lt.nbine) goto 100      ! go start the next beta window
c instead, in mavabfit, we fit beta now for this window!

c  nbetawin = iwin
c  write(6,*) nbetawin,' beta windows'

*******************Section 4:  Find beta for each betawindow********

c
c ******************************************
c Master beta-window loop (iwin is still the betawindow index)
c   The beta fitting happens here
c
    if(iwin.eq.1) then
    write(6,*) 'Enter root name of output Beta files?'
    read(5,*) betaroot
    call addsuffix(betaroot,'.fit',betafile)
    open(11,file=betafile)  ! file for beta (and alpha) fit results
    endif

c  do iwin = 1, nbetawin
c Create 1-D arrays for lfit (Legendre polynomial coefficient fitting
routine)
      index = 0
      do ith = ithminwin(iwin),ithmaxwin(iwin)
        index = index + 1  ! index runs from 1 up
        xra(index) = cos((ith-0.5)*pi/180.) ! lfit wants cosines for x
array
        yra(index) = cpthsum(iwin,ith)
        sigfit(index) = sqrt(varsum(iwin,ith))
      end do
      do ith = ithminwin2(iwin),ithmaxwin2(iwin)
        index = index + 1  ! index carries over from 1st quad
        xra(index) = cos((ith-0.5)*pi/180.) ! lfit wants cosines for x
array
        yra(index) = cpthsum(iwin,ith)
        sigfit(index) = sqrt(varsum(iwin,ith))
      end do
c
      lista(1) = 1  ! basis fxn #1 = P0(x)
    if(fixbeta.eq.'y') then
      mfit = 1
      a(3) = fixbetaval
    else
      lista(2) = 3    ! basis fxn #2 = P2(x)
      mfit = 2  ! # of coefficients to vary (= # fxns for us here)
    endif
      ma = 3    ! order of the fit, i.e., # of basis fxns
      ncvm = 3  ! physical size of 2d covar array
c Go for it
      call lfit(xra,yra,sigfit,nthwin(iwin),
      @                     a,ma,lista,mfit,covar,ncvm,chisq)
```

```
c Gone done it
      alphacent(iwin) = a(1)
      betaorig(iwin) = a(3)/a(1)
c Calculate error of x = a(3)/a(1) = norm(d(a(3))/a(1) , -
a(3)*d(a(1))/a(1)**2)
      betasig(iwin) = sqrt( ( sqrt(covar(3,3)) /a(1) )**2 +
      @     ( a(3)*sqrt( covar(1,1) )/(a(1)*a(1)) )**2 )

c          -1 < beta < 2
       betacent(iwin) = min( 2, max(-1, betaorig(iwin) ) )
       betalower(iwin) = max(-1, betaorig(iwin)-betasig(iwin))
       betaupper(iwin) = min(2, betaorig(iwin)+betasig(iwin))

c Write out fit results to screen and to file
      writepass = 0
      nfile = 6
550         write(nfile,*) 'Fit results for window ',iwin
      if(a(1).lt.0)
      @   write(nfile,*)'WARNING:  BOGUS FIT RESULTS, alpha is
negative!
      @ !@#$%!@##$#@$%!#$%#^!@#%#$^!$#^!#$^^@^!#^#$^!@%!#$^#^@#^%'
      write(nfile,*) '   spanning the energy range from',
      @          enrg(inrgmin(iwin)),' to',enrg(inrgmax(iwin))
      write(nfile,*)'# events, 1st ebin, last ebin, <E>,
      @ theta limits'
      write(nfile,*) events(iwin),inrgmin(iwin),inrgmax(iwin)
      @  ,nrg(iwin),
      @   ithminwin(iwin),' -',ithmaxwin(iwin),' ;',
      @  ithminwin2(iwin),' -',ithmaxwin2(iwin)
      write(nfile,*) 'alpha =',a(1),'  +/-',sqrt(covar(1,1))
c  Note: the usual beta we are used to seeing is really divided
c  or normalized by the intensity at theta=54.7 degrees, i.e. a(1)
      write(nfile,*) 'beta        =',a(3)/abs( a(1) ),
      @          '  +/-',betasig(iwin)
      write(nfile,*) 'Chisq/(N-M) =',
      @          chisq/(nthwin(iwin)-mfit)
      write(nfile,*) 'Covariance matrix: '
      write(nfile,1000) covar(1,1),covar(1,3)
      write(nfile,1000) covar(3,1),covar(3,3)
      write(nfile,*)
      if (writepass.eq.0) then
        writepass = 1
        nfile = 11
        goto 550
      endif
c  end do       ! beta-window loop index (iwin)
c Now return to make the next betawindow
c Use a value other than 1 to increment iwinstart for coarser moving
av.
   iwinstart = iwinstart + 1

570      if(i.lt.nbine) goto 100 !else, we are done betafitting

   nbetawin = iwin
   write(6,*)
   write(6,*)nbetawin,' beta windows'
   write(6,*)

1000     format(3(E13.4))
c
590      call addsuffix(betaroot,'.boe',betafile)
   open(33,file=betafile)
   write(6,*)'Writing betas with errors to ',betafile
   if(fixbeta.eq.'y') then
```

```
      write(6,*) '************************************'
      write(6,*)'NOTE:  Beta is fixed at ',fixbetaval
      write(6,*) '************************************'
     endif

     call addsuffix(betaroot,'.angb',tempname)
     call addsuffix(tempname,'ins',angbinfile)
     open(88,file=angbinfile)
     do i = 1,nbetawin
       write(33,*) nrg(i),betaorig(i),betasig(i)
       write(88,*) nrg(i),nthwin(i)      !num of angle bins
     end do
     close(88)
     close(33)
c
*******************Section 5:  Interpolate beta to give a beta******
*******************    beta value for each energy bin***************

     call addsuffix(betaroot,'.int ',betaint)
     call addsuffix(betaint,'boe',betafile)
     write(6,*)'Writing interpolated betas to ',betafile
     open(35,file=betafile)
     call addsuffix(betaroot,'l.int',betaint)
     call addsuffix(betaint,'boe',betafile)
     write(6,*)'Writing interpolated lower betalimits to ',betafile
     open(34,file=betafile)
     call addsuffix(betaroot,'u.int',betaint)
     call addsuffix(betaint,'boe',betafile)
     write(6,*)'Writing interpolated lower betalimits to ',betafile
     open(36,file=betafile)
c
c Construct interpolated beta(E)'s from the betaorig(i) data points
c
c The subroutine interp returns the slope and intercept of the
beta(E)
c function for the linear interpolation between the adjacent points
in
c the nrg array
c
     do i = 1,nbine
       call interp(enrg(i),nrg,betacent,
       >              nbetawin,slope,intercept)
       beta(i) = enrg(i)*slope + intercept
            write(35,1500) enrg(i), beta(i)
       call interp(enrg(i),nrg,betalower,
       >              nbetawin,slope,intercept)
       betal(i) = enrg(i)*slope + intercept
            write(34,1500) enrg(i), betal(i)   !fixed typo 4/25/96 DLO
       call interp(enrg(i),nrg,betaupper,
       >              nbetawin,slope,intercept)
       betau(i) = enrg(i)*slope + intercept
            write(36,1500) enrg(i), betau(i)
     enddo
     close(35)
     close(34)
     close(36)
 1500   format(5x,f10.5,5x,f8.5)


******Section 6:  Do 3 alphafits for beta, beta+/- sigma********

c
c Find alpha(E) given the interpolated beta(E) values, and also
```

```
c keep track of integrated alpha(E)
c
   write(11,*)'Alpha fit results:'
   write(11,*)'energy    alpha      sigma     norm. chi squared'
        alphaavg = 0
        alphaavgl = 0
        alphaavgu = 0
   alphamp = 0
        alphamaxker = 0
   alphaint = 0
        alphaintl = 0
        alphaintu = 0
   do i=1,nbine
    if (nth(i)+nth2(i).le.2) then      ! test for daf-impossible
ebins
     ibad(i) = 1       ! warning flag for output loop
    else          ! go ahead and fit, even if there's no data
     yysum = 0 ! NOTE:  yysum only needs to be calculated once,
     xysum = 0 !        while xxsum and xysum depend on beta
     xxsum = 0
c Calculate sums for least squares formulae
     do ith = ithmin(i),ithmax(i)
       costheta = cos((ith-0.5)*pi/180.)
       xx = 1 + beta(i)*(1.5*costheta*costheta - 0.5)
       xysum = xysum + xx*cpth(i,ith)/(sig(i,ith)*sig(i,ith))
       xxsum = xxsum + xx*xx/(sig(i,ith)*sig(i,ith))
       yysum = yysum +
cpth(i,ith)*cpth(i,ith)/(sig(i,ith)*sig(i,ith))
     end do
     do ith = ithmin2(i),ithmax2(i)
       costheta = cos((ith-0.5)*pi/180.)
       xx = 1 + beta(i)*(1.5*costheta*costheta - 0.5)
       xysum = xysum + xx*cpth(i,ith)/(sig(i,ith)*sig(i,ith))
       xxsum = xxsum + xx*xx/(sig(i,ith)*sig(i,ith))
       yysum = yysum +
cpth(i,ith)*cpth(i,ith)/(sig(i,ith)*sig(i,ith))
     end do


c ************************************************************
          if(xxsum.eq.0.0) xxsum = 1e-10
c ************************************************************


c    write(6,*)i,nth(i)+nth2(i),xxsum,xysum,yysum
     alpha(i)= xysum/xxsum
          alphaavg = alphaavg + alpha(i)*enrg(i)
          if(alpha(i).gt.alphamp) then
           alphamp = enrg(i)
          endif
          if(alpha(i).gt.1e-8) then
           alphamaxker = enrg(i)
          endif
          alphaint = alphaint + alpha(i)
     alphasig(i) = sqrt( 1/xxsum )
c  reduced chisqr for the alphafit (1 parameter fit)
     chisqr(i) = (yysum - 2*alpha(i)*xysum
     @          + alpha(i)*alpha(i)*xxsum) / (nth(i)+nth2(i) - 1)
c write alphafit results to unit 11 ("betafile")
     write(11,2525) enrg(i),alpha(i),alphasig(i),chisqr(i)
     xysum = 0
     xxsum = 0
c Calculate sums again, for upper beta
     do ith = ithmin(i),ithmax(i)
```

```
        costheta = cos((ith-0.5)*pi/180.)
        xx = 1 + betau(i)*(1.5*costheta*costheta - 0.5)
        xysum = xysum + xx*cpth(i,ith)/(sig(i,ith)*sig(i,ith))
        xxsum = xxsum + xx*xx/(sig(i,ith)*sig(i,ith))
      end do
      do ith = ithmin2(i),ithmax2(i)
        costheta = cos((ith-0.5)*pi/180.)
        xx = 1 + betau(i)*(1.5*costheta*costheta - 0.5)
        xysum = xysum + xx*cpth(i,ith)/(sig(i,ith)*sig(i,ith))
        xxsum = xxsum + xx*xx/(sig(i,ith)*sig(i,ith))
      end do

c ************************************************************
          if(xxsum.eq.0.0) xxsum = 1e-10
c ************************************************************


      alphau(i)= xysum/xxsum

          alphaavgu = alphaavgu + alphau(i)*enrg(i)
          alphaintu = alphaintu + alphau(i)
      alphausig(i) = sqrt( 1/xxsum )

      chisqru(i) = (yysum - 2*alphau(i)*xysum
      @          + alphau(i)*alphau(i)*xxsum)/(nth(i)+nth2(i) - 1)
      xysum = 0
      xxsum = 0
c Calculate sums a last time, for lower beta
      do ith = ithmin(i),ithmax(i)
        costheta = cos((ith-0.5)*pi/180.)
        xx = 1 + betal(i)*(1.5*costheta*costheta - 0.5)
        xysum = xysum + xx*cpth(i,ith)/(sig(i,ith)*sig(i,ith))
        xxsum = xxsum + xx*xx/(sig(i,ith)*sig(i,ith))
      end do
      do ith = ithmin2(i),ithmax2(i)
        costheta = cos((ith-0.5)*pi/180.)
        xx = 1 + betal(i)*(1.5*costheta*costheta - 0.5)
        xysum = xysum + xx*cpth(i,ith)/(sig(i,ith)*sig(i,ith))
        xxsum = xxsum + xx*xx/(sig(i,ith)*sig(i,ith))
      end do

c ************************************************************
          if(xxsum.eq.0.0) xxsum = 1e-10
c ************************************************************


      alphal(i)= xysum/xxsum
          alphaavgl = alphaavgl + alphal(i)*enrg(i)
          alphaintl = alphaintl + alphal(i)
      alphalsig(i) = sqrt( 1/xxsum )
      chisqrl(i) = (yysum - 2*alphal(i)*xysum
      @          + alphal(i)*alphal(i)*xxsum)/(nth(i)+nth2(i) - 1)
     end if
    end do ! energy bin loop

*******************Section 7:  Statistics, files, plotting********

c
c Write out fit results
c
    call addsuffix(betaroot,'.aoe',alphafile)
    open(25,file=alphafile)
    call addsuffix(betaroot,'.saoe',salphafile)
    open(45,file=salphafile)      !simple aoe file in x,y format
```

```
      write(6,*)'Writing P(E) to ',alphafile
      call addsuffix(betaroot,'l.aoe',alphafile)
      open(24,file=alphafile)
      write(6,*)'Writing lower beta limit P(E) to ',alphafile
      call addsuffix(betaroot,'u.aoe',alphafile)
      open(26,file=alphafile)
      write(6,*)'Writing upper beta limit P(E) to ',alphafile
c           call addsuffix(betaroot,'.kcal',alphafile)
c           open(27,file=alphafile)
c           write(6,*)'Writing kcal P(E) to ',alphafile
      do i=1,nbine
       if (ibad(i).ne.1) then
        write(25,2526) enrg(i),alpha(i),alphasig(i)
        write(24,2526) enrg(i),alphal(i),alphalsig(i)
        write(26,2526) enrg(i),alphau(i),alphausig(i)
        write(45,*) enrg(i),alpha(i)
c             write(27,2526) enrg(i)*23.06,alpha(i),alphasig(i)
       end if
      end do
      close(25)
      close(24)
      close(26)
      close(45)
c           close(27)
2525        format(f8.4,2E13.5,F10.4)
2526        format(f8.4,2E13.5)
c
c Write out statistics for the alphafit KER curve
c
c           alphaavg = alphaavg / alphaint
c           alphaavgu = alphaavgu / alphaintu
c           alphaavgl = alphaavgl / alphaintl
c           call addsuffix(betaroot,'.stat',alphafile)
c           write(6,2530)alphaavg*23.06
c           write(6,2540)alphaavgu*23.06
c           write(6,2550)alphaavgl*23.06
c           write(6,2560)alphamp*23.06
c           write(6,2580)alphamaxker*23.06
      write(6,*)'*********************************************'
           write(6,2600)alphaint
           write(6,2650)alphaintu
           write(6,2700)alphaintl
      write(6,*)'*********************************************'
c           open(28,file=alphafile)
c           write(6,*)'Also writing statistics for the alphafit KER
curve ',
c     >                'to ',alphafile
c           write(28,2530)alphaavg*23.06
c           write(28,2540)alphaavgu*23.06
c           write(28,2550)alphaavgl*23.06
c           write(28,2560)alphamp*23.06
c           write(28,2580)alphamaxker*23.06
      write(11,*)'*********************************************'
           write(11,2600)alphaint
           write(11,2650)alphaintu
           write(11,2700)alphaintl
      write(11,*)'*********************************************'
      close(11)   ! finally close the fit results file
c           close(28)
c
c2530       format('Average KER for this channel using ',/,
c     >            'best Beta values [kcal/mol]: ',e11.4)
c2540       format('Average KER for this channel using ',/,
c     >            'upper Beta values [kcal/mol]: ',e11.4)
```

```
c2550    format('Average KER for this channel using ',/,
c     >          'lower Beta values [kcal/mol]: ',e11.4)
c2560    format('Most probable KER for this channel using best ',/,
c     >          'Beta values [kcal/mol]: ',e11.4)
c2580    format('Maximum observed KER for this channel using best
',/,
c     >          'Beta values [kcal/mol]: ',e11.4)
2600    format('Area under best alphafit KER curve is: ',f11.6)
2650    format('Area under low limit alphafit KER curve is: ',f11.6)
2700    format('Area under highlimit alphafit KER curve is: ',f11.6)
c
c Call NCAR to generate P(E) and Beta(E) plots
c
   write(6,*)'Plotting P(E) and Beta(E)'
   call opngks
        call setusv('PB',2)
        call agsetc('LABEL/NAME.','L')
        call agseti('LINE/NUMBER.',100)
        call agsetc('LINE/TEXT.','P(E)$')
        call agsetc('LABEL/NAME.','B')
        call agseti('LINE/NUMBER.',-100)
        call agsetc('LINE/TEXT.','Energy$')

c*********************************************************
        title = titleorg//'-DAF''d KER'//'$'
   call ezxy(enrg,alpha,nbine,title)
c*******************************************************
        call setusv('PB',2)
        call agsetc('LABEL/NAME.','L')
        call agseti('LINE/NUMBER.',100)
   call agsetc('LINE/TEXT.','Beta(E)$')
        call agsetc('LABEL/NAME.','B')
        call agseti('LINE/NUMBER.',-100)
        call agsetc('LINE/TEXT.','Energy$')

c ****************************************************
        title = titleorg//'-Beta vs Energy'//'$'
   call ezxy(enrg,beta,nbine,title)
****************************************************
        call setusv('PB',2)
        call agsetc('LABEL/NAME.','L')
        call agseti('LINE/NUMBER.',100)
   call agsetc('LINE/TEXT.','No. of Angular Bins Fit$')
        call agsetc('LABEL/NAME.','B')
        call agseti('LINE/NUMBER.',-100)
        call agsetc('LINE/TEXT.','Energy$')

c ****************************************************
        title = titleorg//'-Ang. Bins vs Energy'//'$'
   call ezxy(nrg,nthwin,iwin,title)
****************************************************



c
c Generate plots of beta window fits
c
   call addsuffix(betaroot,'.angdist',angdistfile)
   open(17,file=angdistfile)
   write(6,*)'Writing P(Theta) to ',angdistfile

   do iwin = 1, nbetawin
     do ith = 1,180
       xra(ith) = ith - 0.5
```

```
        costheta = cos(xra(ith)*pi/180.) ! cos(theta in radians)
        fit(ith) = alphacent(iwin)*
     @    ( 1 + betaorig(iwin) * ( 1.5*costheta*costheta - 0.5 ) )
        yra(ith) = 0  ! initialize yra (gotta zero all 180)
      end do
      do ith = ithminwin(iwin),ithmaxwin(iwin)
        yra(ith) = cpthsum(iwin,ith)
      end do
      do ith = ithminwin2(iwin),ithmaxwin2(iwin)
        yra(ith) = cpthsum(iwin,ith)
      end do
      do ith = 1,180
       write(17,*)xra(ith),yra(ith),fit(ith)
      enddo

c*****************************************************************
      title = titleorg//'; for '
      @   //char(inrgmin(iwin))//'-'//char(inrgmax(iwin))//'$'
c*****************************************************************
      call agseti('FRAME.',2)     ! two plots per frame
      call setusv('PB',2)
      call agsetc('LABEL/NAME.','L')
      call agseti('LINE/NUMBER.',100)
      call agsetc('LINE/TEXT.','P(Theta)$')
      call agsetc('LABEL/NAME.','B')
      call agseti('LINE/NUMBER.',-100)
      call agsetc('LINE/TEXT.','Theta$')
      call ezxy(xra,yra,180,title)
      call points(xra,fit,180,-1,0)
      call FRAME
    end do
    call clsgks
    ierr = system(' mv gmeta abfit.img')
    if (iwkst.eq.2) ierr = system(' ictrans -d X11 -bell abfit.img')

    close(17)
c
    end
```

addsuffix.f

```
    subroutine addsuffix(rootname,suffix,fullname)
c   this subroutine takes inputs rootname (e.g. oxygen) and
c   suffix (e.g. .cker) from calling program, and returns the
c   concatenated filename (e.g. oxygen.cker)
    implicit none
    character*5 suffix
    character*20 rootname,fullname
    integer loc1,length

    loc1 = index (rootname,' ') !find the first space in rootfile
    length = loc1 - 1 !length of filename rootword
    fullname = rootname(1:length) // suffix

    return

    end
```

interp.f

```
      subroutine interp(ebinval,ecenter,betapt,nbetapts,slope,intercept)

c     This subroutine findes the slope and interecept of Beta(e) for
c     each value of ebinval(i)

      parameter(n2 = 300)

      implicit none
      integer nbetapts,i
      real*4 ebinval,ecenter(n2),betapt(nbetapts)
      real*4 slope,intercept

      if(ebinval.le.ecenter(1)) then
       slope = 0.0
       intercept = betapt(1)
      elseif(ebinval.ge.ecenter(nbetapts)) then
       slope = 0.0
       intercept = betapt(nbetapts)
      else
c     if the ebinval was not outside the boundary of beta(E), find
c     out where it is.
c     first assign the two bounding neighbors of the point ebinval
c     i is the value of the ecenter point to the left of ebinval
      i = 0
100       i = i + 1
       if((ebinval.ge.ecenter(i)).and.(ebinval.le.ecenter(i+1))) then
        slope = (betapt(i+1) - betapt(i))/(ecenter(i+1)-ecenter(i))
c       write(6,*)'betapt(',i,') = ',betapt(i)
c       write(6,*)'slope = ',slope
c       write(6,*)'ecenter(',i,') = ',ecenter(i)

        intercept = betapt(i) - slope*ecenter(i)
        goto 200
       else
        goto 100
       endif
      endif

200       return

        end
```

covsrt.f:

```
        SUBROUTINE COVSRT(COVAR,NCVM,MA,LISTA,MFIT)
        DIMENSION COVAR(NCVM,NCVM),LISTA(MFIT)
        DO 12 J=1,MA-1
          DO 11 I=J+1,MA
            COVAR(I,J)=0.
11        CONTINUE
12      CONTINUE
        DO 14 I=1,MFIT-1
          DO 13 J=I+1,MFIT
            IF(LISTA(J).GT.LISTA(I)) THEN
              COVAR(LISTA(J),LISTA(I))=COVAR(I,J)
            ELSE
              COVAR(LISTA(I),LISTA(J))=COVAR(I,J)
            ENDIF
13        CONTINUE
14      CONTINUE
        SWAP=COVAR(1,1)
```

```
      DO 15 J=1,MA
        COVAR(1,J)=COVAR(J,J)
        COVAR(J,J)=0.
15    CONTINUE
      COVAR(LISTA(1),LISTA(1))=SWAP
      DO 16 J=2,MFIT
        COVAR(LISTA(J),LISTA(J))=COVAR(1,J)
16    CONTINUE
      DO 18 J=2,MA
        DO 17 I=1,J-1
          COVAR(I,J)=COVAR(J,I)
17      CONTINUE
18    CONTINUE
      RETURN
      END
```

fleg.f:

```
      SUBROUTINE FLEG(X,PL,NL)
      DIMENSION PL(NL)
      PL(1)=1.
      PL(2)=X
      IF(NL.GT.2) THEN
        TWOX=2.*X
        F2=X
        D=1.
        DO 11 J=3,NL
          F1=D
          F2=F2+TWOX
          D=D+1.
          PL(J)=(F2*PL(J-1)-F1*PL(J-2))/D
11      CONTINUE
      ENDIF
      RETURN
      END
```

gammln.f:

```
      FUNCTION GAMMLN(XX)
      REAL*8 COF(6),STP,HALF,ONE,FPF,X,TMP,SER
      DATA COF,STP/76.18009173D0,-86.50532033D0,24.01409822D0,
     *    -1.231739516D0,.120858003D-2,-.536382D-5,2.50662827465D0/
      DATA HALF,ONE,FPF/0.5D0,1.0D0,5.5D0/
      X=XX-ONE
      TMP=X+FPF
      TMP=(X+HALF)*LOG(TMP)-TMP
      SER=ONE
      DO 11 J=1,6
        X=X+ONE
        SER=SER+COF(J)/X
11    CONTINUE
      GAMMLN=TMP+LOG(STP*SER)
      RETURN
      END
```

gammq.f:

```
      FUNCTION GAMMQ(A,X)
      IF(X.LT.0..OR.A.LE.0.)PAUSE
      IF(X.LT.A+1.)THEN
        CALL GSER(GAMSER,A,X,GLN)
```

```
        GAMMQ=1.-GAMSER
      ELSE
        CALL GCF(GAMMQ,A,X,GLN)
      ENDIF
      RETURN
      END
```

gaussj.f:

```
      SUBROUTINE GAUSSJ(A,N,NP,B,M,MP)
      PARAMETER (NMAX=50)
      DIMENSION A(NP,NP),B(NP,MP),IPIV(NMAX),INDXR(NMAX),INDXC(NMAX)
      DO 11 J=1,N
        IPIV(J)=0
 11   CONTINUE
      DO 22 I=1,N
        BIG=0.
        DO 13 J=1,N
          IF(IPIV(J).NE.1)THEN
            DO 12 K=1,N
              IF (IPIV(K).EQ.0) THEN
                IF (ABS(A(J,K)).GE.BIG)THEN
                  BIG=ABS(A(J,K))
                  IROW=J
                  ICOL=K
                ENDIF
              ELSE IF (IPIV(K).GT.1) THEN
                PAUSE 'Singular matrix'
              ENDIF
 12         CONTINUE
          ENDIF
 13     CONTINUE
        IPIV(ICOL)=IPIV(ICOL)+1
        IF (IROW.NE.ICOL) THEN
          DO 14 L=1,N
            DUM=A(IROW,L)
            A(IROW,L)=A(ICOL,L)
            A(ICOL,L)=DUM
 14       CONTINUE
          DO 15 L=1,M
            DUM=B(IROW,L)
            B(IROW,L)=B(ICOL,L)
            B(ICOL,L)=DUM
 15       CONTINUE
        ENDIF
        INDXR(I)=IROW
        INDXC(I)=ICOL
        IF (A(ICOL,ICOL).EQ.0.) PAUSE 'Singular matrix.'
        PIVINV=1./A(ICOL,ICOL)
        A(ICOL,ICOL)=1.
        DO 16 L=1,N
          A(ICOL,L)=A(ICOL,L)*PIVINV
 16     CONTINUE
        DO 17 L=1,M
          B(ICOL,L)=B(ICOL,L)*PIVINV
 17     CONTINUE
        DO 21 LL=1,N
          IF(LL.NE.ICOL)THEN
            DUM=A(LL,ICOL)
            A(LL,ICOL)=0.
            DO 18 L=1,N
              A(LL,L)=A(LL,L)-A(ICOL,L)*DUM
 18         CONTINUE
```

```
            DO 19 L=1,M
              B(LL,L)=B(LL,L)-B(ICOL,L)*DUM
19          CONTINUE
          ENDIF
21      CONTINUE
22    CONTINUE
      DO 24 L=N,1,-1
        IF(INDXR(L).NE.INDXC(L))THEN
          DO 23 K=1,N
            DUM=A(K,INDXR(L))
            A(K,INDXR(L))=A(K,INDXC(L))
            A(K,INDXC(L))=DUM
23        CONTINUE
        ENDIF
24    CONTINUE
      RETURN
      END
```

gcf.f:

```
      SUBROUTINE GCF(GAMMCF,A,X,GLN)
      PARAMETER (ITMAX=100,EPS=3.E-7)
      GLN=GAMMLN(A)
      GOLD=0.
      A0=1.
      A1=X
      B0=0.
      B1=1.
      FAC=1.
      DO 11 N=1,ITMAX
        AN=FLOAT(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.NE.0.)THEN
          FAC=1./A1
          G=B1*FAC
          IF(ABS((G-GOLD)/G).LT.EPS)GO TO 1
          GOLD=G
        ENDIF
11    CONTINUE
      PAUSE 'A too large, ITMAX too small'
1     GAMMCF=EXP(-X+A*ALOG(X)-GLN)*G
      RETURN
      END
```

gser.f:

```
      SUBROUTINE GSER(GAMSER,A,X,GLN)
      PARAMETER (ITMAX=100,EPS=3.E-7)
      GLN=GAMMLN(A)
      IF(X.LE.0.)THEN
        IF(X.LT.0.)PAUSE
        GAMSER=0.
        RETURN
      ENDIF
```

```
      AP=A
      SUM=1./A
      DEL=SUM
      DO 11 N=1,ITMAX
        AP=AP+1.
        DEL=DEL*X/AP
        SUM=SUM+DEL
        IF(ABS(DEL).LT.ABS(SUM)*EPS)GO TO 1
11    CONTINUE
      PAUSE 'A too large, ITMAX too small'
1     GAMSER=SUM*EXP(-X+A*LOG(X)-GLN)
      RETURN
      END
```

lfit.f:

```
      SUBROUTINE LFIT(X,Y,SIG,NDATA,A,MA,LISTA,MFIT,COVAR,NCVM,CHISQ)
      PARAMETER (MMAX=3)
      DIMENSION X(NDATA),Y(NDATA),SIG(NDATA),A(MA),LISTA(MA),
     *    COVAR(NCVM,NCVM),BETA(MMAX),AFUNC(MMAX)
   real*4 qfit,nu,nu2
      KK=MFIT+1
c  write(6,*)'ma',ma,'mfit',mfit
c  write(6,*)'lista:',lista
      DO 12 J=1,MA
        IHIT=0
        DO 11 K=1,MFIT
c    write(6,*) k,lista(K)
          IF (LISTA(K).EQ.J) IHIT=IHIT+1
11      CONTINUE
        IF (IHIT.EQ.0) THEN
          LISTA(KK)=J
          KK=KK+1
        ELSE IF (IHIT.GT.1) THEN
          PAUSE 'Improper set in LISTA'
        ENDIF
12    CONTINUE
c  write(6,*)'final lista:',lista
      IF (KK.NE.(MA+1)) PAUSE 'Improper set in LISTA'
      DO 14 J=1,MFIT
        DO 13 K=1,MFIT
          COVAR(J,K)=0.
13      CONTINUE
        BETA(J)=0.
14    CONTINUE
      DO 18 I=1,NDATA
        CALL fleg(X(I),AFUNC,MA)
c  write(6,*)'successful call of funcs'
        YM=Y(I)
        IF(MFIT.LT.MA) THEN
c  get residuals for parameters which are held fixed
          DO 15 J=MFIT+1,MA
            YM=YM-A(LISTA(J))*AFUNC(LISTA(J))
15        CONTINUE
        ENDIF
        SIG2I=1./SIG(I)**2 !(errors)
        DO 17 J=1,MFIT      !(create covar matrix from params which
are fit)
          WT=AFUNC(LISTA(J))*SIG2I
          DO 16 K=1,J
            COVAR(J,K)=COVAR(J,K)+WT*AFUNC(LISTA(K))
16        CONTINUE
```

```
          BETA(J)=BETA(J)+YM*WT
17      CONTINUE
18    CONTINUE
      IF (MFIT.GT.1) THEN
        DO 21 J=2,MFIT
          DO 19 K=1,J-1
            COVAR(K,J)=COVAR(J,K)
19        CONTINUE
21      CONTINUE
      ENDIF
      CALL GAUSSJ(COVAR,MFIT,NCVM,BETA,1,1)
      DO 22 J=1,MFIT
        A(LISTA(J))=BETA(J)
22    CONTINUE
      CHISQ=0.
      DO 24 I=1,NDATA
        CALL fleg(X(I),AFUNC,MA)
        SUM=0.
        DO 23 J=1,MA
          SUM=SUM+A(J)*AFUNC(J)
23      CONTINUE
        CHISQ=CHISQ+((Y(I)-SUM)/SIG(I))**2
24    CONTINUE
   nu = ndata-mfit
   nu2 = nu / 2.0
   chisq2 = chisq / 2.0
   qfit = GAMMQ(nu2,chisq2)
   write(6,*)'Goodness of fit, Q = '
   write(6,1100) qfit
1100     format(f14.9)
      CALL COVSRT(COVAR,NCVM,MA,LISTA,MFIT)
      RETURN
      END
```

Appendix D

XYZLSQ: Creation of detector factors for the time- and position-sensitive detector.

I. Introduction

The XYZLSQ code generates factors which covert charge fraction and Δt data into particle position information based on a non-linear least-squares fit to dissociation data from O_2 as described in Chapter 2. The factors are then employed in the CMLWSA code given in Appendix A to routinely convert charge fraction and Δt data into product mass ratios, translational energy, and angular distributions. The code was written and/or compiled by Dr. David Leahy and is given in the present dissertation for the sake of completeness.

The conceptual basis for this code was presented in Chapter 1. This appendix presents practical information needed to run the program, a description of the subroutines involved, and the actual code itself.

II. Routines comprising the XYZLSQ program

XYZLSQ is compiled from the following routines and subroutines:

xyzlsq.f: Main routine which calls all other routines and performs the Levenberg-Marquardt non-linear least-squares optimization.

xyzin.f: Reads in the initial guesses for all fit parameters and many other constants which are adjusted only by the user.

genpin.f: Reads the fit parameters into the least-squares algorithm.

xyzcal.f: At the heart of the program, this routine does the conversion of charge fraction and Δt data into x, y, and z positions. On output it gives the length of the center-of-mass recoil vector for each coincident event.

jmdiag.f: Matrix manipulation routine.

matinv.f: Matrix inversion routine.

xyznrg.f: performs diagnostics upon completion of the fit

linfit.f: Linear least-squares fitting of data to a line.

XYZLSQ can be compiled on a UNIX platform by executing the `make` command:

```
make -f makexyz
```

where the makefile is:

makexyz

```
FFLAGS=-cg89 -dalign -O3
xyzlsq: xyzlsq.o xyzin.o genpin.o xyzcal.o jmdiag.o matinv.o xyznrg.o
linfit.o
   f77 -cg89 -dalign -O3 xyzlsq.o xyzin.o genpin.o xyzcal.o jmdiag.o
matinv.o xyznrg.o linfit.o -o xyzlsq
```

III. Usage

A. General information and input / output files

A minimum of two files are required on input. The aptly named `input` file contains initial values for all parameters, flags to denote whether a given parameter should

be adjusted or fixed during the fit, and various user-defined parameters which are never adjusted during the fit. The raw data file consists of the six charge fraction readings and the TAC reading for each coincident event. On output the file `param.dat` contains the values of the fit parameters from the last iteration. By selecting the appropriate flag in the `input` file, the initial guesses for the next run of XYZLSQ can be read directly from `param.dat`.

The following is a sample input file with each line annotated. The corresponding parameters discussed in Chapter 2 are given in parentheses after each annotation. The right handed coordinate system is defined with z normal to the plane of the detector, y pointing towards the top of the detector. The subscripts 1 and 2 refer to the upper and lower halves of the detector, respectively. R_{xy} is the projection of the recoil vector on the plane of the detector, while R_z is the projection along the radical beam axis.

input: input file for XYZLSQ code

`16`	number of parameters
`156.29852`	multiplicative wedge upper (a_{1x})
`v`	flag to vary (v) of fix (f) the preceding parameter
`-41.52153`	additive wedge upper (b_{1x})
`v`	fix/vary
`64.71810`	multiplicative strip upper (a_{1y})
`v`	fix/vary
`-6.58297`	additive strip upper (b_{1y})
`v`	fix/vary
`149.98367`	multiplicative wedge lower (a_{2x})
`v`	fix/vary
`-38.28748`	additive wedge lower (b_{2x})
`v`	fix/vary
`64.11207`	multiplicative strip lower (a_{2y})
`v`	fix/vary
`-5.96814`	additive strip lower (b_{2y})
`v`	fix/vary
`0.98`	cross talk wedge/zig-zag upper
`f`	fix/vary
`0.00140`	tilt of detector around x-axis (*tilt-x*)

v	fix/vary
0.98	cross talk wedge/zig-zag lower
f	fix/vary
0.01167	tilt of detector around y-axis (*tilt-y*)
v	fix/vary
1.00	slope of TAC conversion to Δt (*TAC-m*)
f	fix/vary
-0.04622	intercept of TAC conversion to Δt (*TAC-b*)
v	fix/vary
1.00	quadratic wedge factor (c_x)
f	fix/vary
1.40543	quadratic strip factor (c_y)
f	fix/vary
1	If 1, get parameters from `param.dat` file, If 0, get parameter values from `input` file.
0. 0.9	position of radical beam in detector coordinates: x_0, y_0
995	flight length L (mm)
0.14	R_{xy} error (mm)
0.07	R_z error (mm)
0.7	centroid radius error (mm)
	<u>spatial and temporal discrimination (mm)</u>
0.0 18.0	minimum and maximum acceptable radial recoil: R_{min}, R_{max}
1.0	maximum acceptable recoil error
1.5	maximum acceptable centroid radius
4. 18.	allowed range for |y| values, upper half
4. 18.	allowed range for |y| values, lower half
-20. 20.	allowed range for x values, upper half
-20. 20.	allowed range for x values, lower half
-60. 60.	allowed range for z values, i.e. $v_{beam} \bullet \Delta t$
50. 130.	allowed range for azimuthal angle φ
1	number of data files to fit
15000	number of data points to fit
500	skip first n data points in each file
raw_data.dat	name of raw data file
7	O_2 Schumann-Runge upper state vibrational quantum number
24.6	energy offset from the $O(^3P_2) + O(^3P_2)$ energetic limit
4.970	beam energy (kV)
15	number of non-linear least-squares fit iterations
.1	starting value for λ, a parameter of the fitting engine (default = 0.1)
0	output verbosity selector (0 is most verbose)
2	output files verbosity selector (0 is most verbose)

The output file `o2.dat` contains diagnostic information about the results of the fit, and is self-explanatory.

o2.dat: output file from XYZLSQ

```
tslope:          7.36594 ADC units/ns
tint:            3645.50 ADC units
centroid:      0.   0.90000000000000 mm
length:          995.000 mm
  12575 points read from raw_data.dat
O2 dissociation from vB =  7
Ediss of J=2,J=2:    1.53579; fit to mean:    1.51137
beam energy:       4.97000 keV
beam velocity:     173113. m/s
recoil (R):        34.7025 +/-   0.160000(xy)    8.00000E-02(z)
centroid radial uncertainty:   0.700000 mm
Raw data discrimination:       0.730101 %
Min detector radius:  0. mm  0. %
Max detector radius:    18.0000 mm   10.15792 %
Max recoil error:       1.00000 mm    16.7566 %
Max centroid radius:    1.50000 mm    8.45171 %
yu min:      4.00000 mm    1.19038E-02 %
yu max:      18.0000 mm    2.06730 %
yl min:      4.00000 mm    1.58718E-02 %
yl max:      18.0000 mm    2.38076E-02 %
xu min:     -20.0000 mm  0. %
xu max:      20.0000 mm  0. %
xl min:     -20.0000 mm  0. %
xl max:      20.0000 mm  0. %
z min:      -60.0000 mm    3.96794E-03 %
z max:       60.0000 mm   0.123006 %
phi min:     50.0000 degrees    17.1336 %
phi max:     130.000 degrees    19.8754 %
Total percentage disc'd:    50.1032 %
1

      Least Squares Pass:  0     CONVERGENCE HAS BEEN ACHIEVED
Bagged out on JMDIAG
Energy analysis:
Mean energy:     1511.42 meV
Std. dev. (wrt nrg mean):     16.9826 meV
Std. dev. (of the fit):       16.9841 meV
Peak:          1518 meV
FWHM:          21 meV
DeltaZ mean:     -0.781932 mm

   OUTPUT FROM FINAL LEAST SQUARES PASS

Param      Param Value     Uncertainty
  1*       157.30984       0.11978
  2*       -41.80790       0.03206
  3*        62.63658       0.04134
  4*        -6.19427       0.01215
  5*       150.29530       0.11462
  6*       -38.36240       0.02945
  7*        63.44763       0.04440
  8*        -6.04078       0.01172
  9          0.98000
 10*         0.00129       0.00001
 11          0.98000
 12*         0.01274       0.00003
 13          1.00000
 14*        -0.03871       0.00019
```

```
 15          1.00000
 16          1.46000
  Reduced Chi Squared:     1.12405
Variance of the Fit:     5.73180E-02
Standard Deviation of the Fit:     2.39412E-01
```

The output file `param.dat` contains the factors in a compact form which can be read back into XYZLSQ or used for data analysis with CMLWSA. The order of the parameters is identical to the order given in the `input` file.

param.dat: output file from XYZLSQ

```
16
  1    155.71308898926
  2   -41.409950256348
  3    57.684650000000
  4   -5.0801897048950
  5    149.54177856445
  6   -37.332431793213
  7    61.891620635986
  8   -5.5053655909881
  9   0.98000000000000
  10    1.1247047223151D-03
  11   0.98000000000000
  12    1.0954268276691D-02
  13    1.0038297176361
  14   -3.7268795073032D-02
  15    1.0000000000000
  16    1.6000000000000
```

In addition, the file `nrg.dat` contains the $P(E_T)$ distribution for visual inspection. Finally, the program NRGPLOT, compiled from the source code `nrgplot.f`, displays the data in a variety of contour plot formats which can be useful in interpreting systematic errors in specific regions of the detector face.

B. Systematic fitting of the parameters

The most important criterion for assessing whether the fit is satisfactory is the standard deviation of the fit. In addition, a truly converged set of parameters will converge on the next run of XYZLSQ in a single iteration. As with any non-linear fit, the

chances of reaching convergence depends (sometimes quite critically) on the initial guesses for the parameters. It is wise to begin a fit of a new data set with the converged parameters from a previous data set. Beyond these rather obvious points, there is a more-or-less systematic way of arriving at a converged set of factors suitable for use in general photodissociation applications.

(1) Begin from a previous data set parameter file by fixing all parameters except the four multiplicative and the four additive parameters, i.e. the a and b values in `input`. For the sake of speed in fitting, it is often desirable to begin with only 5000 data points. For final fitting the number of data points is increased, concurrent with an increase in required computing time. To begin, use x discrimination limits [-14,14] and y limits [8,18]. In general the φ discrimination limits are fixed at [50,130].

(2) As the fit reaches convergence with only these eight parameters, free the *tilt-x*, *tilt-y*, *TAC-m*, and *TAC-b* parameters in turn. It may not be necessary to leave these parameters free indefinitely, but they should be allowed to adjust at this point.

(3) Leave the quadratic strip factor fixed, but adjust its value by hand and refit the data. Typical values are between 1.4 and 1.6 for this parameter. It is possible to allow this parameter to be adjusted by the fit, but this often requires fixing one or more of the linear strip parameters.

(4) When converged to this point, use CMLWSA to view the image of the O_2 data set on the detector face. Normally the data does not go all the way to the beam block. If this is the case, the multiplicative strip factors, both linear and non-linear, must be fixed and increased "by hand." After converging the XYZLSQ fit, view the new image

using CMLWSA and iterate this procedure until a satisfactory image is obtained which nevertheless preserves most of the energy resolution gained by the fully optimized fit.

(5) Depending on the specific application, it may be wise to adjust the factors for data collection on a particular free radical, or even a particular combination of flight length, beam energy, etc. for a given radical. In general, the discrimination limits set in step (1) should also be adjusted to determine how much of the detector face may be used in the real experiments. The most grievously distorted parts of the spatial detector response occur for azimuthal angles φ = [0-30] and [150,180] and their compliments on the lower half of the detector. Without significant detector modifications it is doubtful whether these regions of the detector will ever be useful.

IV. FORTRAN code of XYZLSQ.

Routines are listed in the order given by the `makefile`, section III.

xyzlsq.f

```
      PROGRAM XYZLSQ
c Program reads in O2 data in wedge-strip-z charge data format
c and fits wedge-and-strip anode parameters to minimize the energy
spread
c and centroid size
c
c Subroutine xyzin.f is the lengthy input subroutine
c genpin.f reads in the initial guesses for the parameters
c xyzcal.f holds the algorithm for parameter-based conversion of
charge
c  data to x-y positions
c jmdiag.f, matinv.f are matrix manipulation programs used by the fit
c xyznrg.f performs diagnostics at the end of the fit
c
      implicit none
      COMMON /BLOCK1/ PARAM(20), NPARAM
      COMMON /BLOCK2/ ydata(50000),ydataerror(50000),NUMBER
      COMMON /BLOCK3/ HDIAG(20,20), EIGVEC(20,20), EIGVAL(20)
      COMMON /rawdata/ wedgeu(25000),wedgel(25000),stripu(25000),
        @ stripl(25000),zu(25000),zl(25000),z(25000),time(25000),x0,y0
      common /procdata/ xu1(25000),xl1(25000),yu1(25000),yl1(25000)

      real*4 z,time,zmean,zsum
      real*4 xu1,yu1,xl1,yl1
      integer wedgeu,stripu,zu,wedgel,stripl,zl,dummy,yesno
      REAL*8 PARAM, SIGMA(20), DELTA(20),x0,y0
```

```
      REAL*4 ydata,ydataerror,temp
      REAL*8 ARRAY(20,20), ALPHA(20,20), BETA(20), LAMBDA
      REAL*8 HDIAG, EIGVEC, EIGVAL
      REAL*4 CTRANS(50000), DTRANS(50000), DERIV(50000,16),
        @ PTEMP(20)
      REAL*8 TAU, PEMP, STEP, DEL, REFCOS, EPSILN, COSINE, NU
      REAL*8 DETERM, VARNCE, SIGMA0, CHIA, CHIB
      integer N,NUMBER,NPARAM,NREP,IPASS,LASTPS,NFIT,NUMFIT(20)
      integer I,ISUB,ILIMIT,K,K0,L,output,ierr,system

c     DATA PARAM        /   20*0.0 /
c     DATA ydata        / 1024*0.0 /
      DATA SIGMA        /   20*0.0 /
      DATA DELTA        /   20*0.0 /
      DATA ARRAY        /  400*0.0 /
      DATA ALPHA        /  400*0.0 /
      DATA HDIAG        /  400*0.0 /
      DATA EIGVEC       /  400*0.0 /
      DATA EIGVAL       /   20*0.0 /
      DATA BETA         /   20*0.0 /
c     DATA CTRANS       /  30000*0.0 /
c     DATA DTRANS       /  30000*0.0 /
c     DATA DERIV        / 300000*0.0 /
      DATA PTEMP        /   20*0.0 /
      DATA NUMFIT       /   20*0   /
      DATA VARNCE       /      0.0 /
      DATA SIGMA0       /      0.0 /

      PARAMETER         ( DEL    = 1.0D-6          )
      PARAMETER         ( REFCOS = 7.071067812D-1 )
      PARAMETER         ( EPSILN = 5.0D-7          )
      PARAMETER         ( TAU    = 1.0D-6          )
      PARAMETER         ( NU     = 10.0            )
      PARAMETER         ( ILIMIT = 12              )

      CALL GENPIN ( NFIT, NUMFIT, 1 )
      IF(Nfit.gt.16) stop 'Too many fit params for DERIV array'
c
c     write(6,*) 'Get parameters from param.dat? (1=yes)'
      read (5,*) yesno
      if (yesno.eq.1) then
        open(10,file='param.dat',status='old')
        read(10,*) dummy
        do i=1,nparam
          read(10,*) dummy,param(i)
        end do
        close(10)
        write(6,*)'Read in parameters from param.dat'
      end if

      CALL XYZLINDISC

      WRITE(6,*) 'Enter desired number of iterations:'
      READ (5,*) NREP
      WRITE(6,*) 'Enter starting value for lambda'
      READ (5,*) LAMBDA
c output flag:  0 gives final fit results with diagnostics
c        1 gives progress of fit step by step
c        2 gives sampling of residuals
c        3 gives matrices and xyz.dat
      write(6,*) 'Enter desired level of output (0 < 1 < 2 < 3)'
      read (5,*) output

      DO 175 N=1, NUMBER
```

```
175       SIGMA0 = SIGMA0 + 1.0 / YDATAerror(N)**2

   IPASS  = 0
   LASTPS = 0
   DETERM = 0.0
   LAMBDA = LAMBDA * NU

200       CHIA   = 0.0
   CALL GENCAL( CTRANS )
   DO 250 N=1, NUMBER
250       CHIA = CHIA + ((ydata(N) - CTRANS(N)) /ydataerror(N))**2
   CHIA = CHIA / (NUMBER - NFIT)

   WRITE(7,7300)

300       if (output.lt.1) goto 600
   WRITE(7,5300) IPASS
   WRITE(7,5400)
   DO 500 I=1, NPARAM
   DO 350 K=1, NFIT
   IF (NUMFIT(K) .EQ. I) GO TO 400
350       CONTINUE
   WRITE(7,5550) I, PARAM(I)
   GO TO 500
400       WRITE(7,5500) I, PARAM(I), SIGMA(K), DELTA(K)
500       CONTINUE
   WRITE(7,5700) DETERM, LAMBDA*MIN0(1,IPASS)
   WRITE(7,6300) COSINE, STEP
   WRITE(7,8000) CHIA
   IF ((NREP .NE. 0) .AND. (NFIT .NE. 0)) GO TO 600
   WRITE(7,6700) IPASS
   GO TO 4060

600       IF ((IPASS .LT. NREP) .OR. (NREP .LT. 0)) GO TO 700
   WRITE(7,6700) IPASS
   LASTPS = 1
   LAMBDA = 0.0
   GO TO 2600

700       IPASS  = IPASS + 1
   ISUB   = 0
   LAMBDA = LAMBDA / NU

C Here we calculate numbers (i.e. DERIV, BETA) for use in this
iteration,
C   including all subiterations
   DO 2200 K=1, NFIT
   K0   = NUMFIT(K)
   TEMP = PARAM(K0)
   PARAM(K0) = (1.0 + DEL)*TEMP
   IF (TEMP .EQ. 0.0) PARAM(K0) = DEL
   CALL GENCAL( DTRANS )
   DO 2100 N=1, NUMBER
   DERIV(N,K) = (DTRANS(N) - CTRANS(N)) / DEL
2100      IF (TEMP .NE. 0.0) DERIV(N,K) = DERIV(N,K) / TEMP
2200      PARAM(K0) = TEMP

   DO 2300 K=1, NFIT
   BETA(K) = 0.0
   DO 2300 N=1, NUMBER
2300      BETA(K) = BETA(K) + DERIV(N,K)*(ydata(N) - CTRANS(N)) /
     1                         ydataerror(N)**2
   DO 2400 K=1, NFIT
   DO 2400 I=1, NFIT
```

```
   ALPHA(I,K) = 0.0
   DO 2400 N=1, NUMBER
2400     ALPHA(I,K) = ALPHA(I,K) + DERIV(N,I)*DERIV(N,K) /
ydataerror(N)**2

   DO 2500 I=1, NFIT
2500     BETA(I) = BETA(I) / SQRT(ALPHA(I,I))

c Here is the repeat point for sub-iteration:
2600     write(6,*) IPASS, ISUB+1
   DO 2800 K=1, NFIT
   DO 2700 I=1, NFIT
2700     ARRAY(I,K) = ALPHA(I,K) / SQRT(ALPHA(I,I))/SQRT(ALPHA(K,K))
2800     ARRAY(K,K) = ARRAY(K,K) + LAMBDA

   IF (LASTPS .EQ. 0) GOTO 2875

C This code (up to 2875) is executed only when the fit is over
   DO 2850 I=1, NFIT
   DO 2850 K=1, NFIT
2850     HDIAG(K,I) = ARRAY(K,I)
c  CALL JMDIAG( NFIT )
   write(7,*)'Bagged out on JMDIAG'
   DO 2860 I=1, NFIT
   EIGVAL(I) = HDIAG(I,I)
   DO 2860 K=1, NFIT
2860     HDIAG(K,I) = ARRAY(K,I)

2875     CALL MATINV( ARRAY, NFIT, DETERM )
   IF (DETERM .EQ. 0.0) then
     write(7,*)'Array was singular!'
     stop
   end if

   IF (LASTPS .NE. 0) GO TO 4000

c This code is only executed for fits in progress
   DO 3000 I=1, NFIT
   DELTA(I) = 0.0
   DO 2900 K=1, NFIT
2900     DELTA(I) = DELTA(I) + ARRAY(I,K)*BETA(K)
   DELTA(I) = DELTA(I) / SQRT(ALPHA(I,I))
3000     PTEMP(I) = PARAM(NUMFIT(I))

   TEMP   = 0.0
   PEMP   = 0.0
   COSINE = 0.0
   DO 3010 I=1, NFIT
   COSINE = COSINE + DELTA(I)*BETA(I)
   TEMP   = TEMP + DELTA(I)**2
3010     PEMP   = PEMP + BETA(I)**2
   COSINE = COSINE / SQRT(TEMP) / SQRT(PEMP)

   STEP   = 1.0
3020     DO 3040 I=1, NFIT
c  write(6,*) I, delta(i)
   IF (ABS(STEP*DELTA(I)) .GE. EPSILN*ABS(TAU+PTEMP(I)))
     1                                GO TO 3120
3040     CONTINUE

   WRITE(7,6400) IPASS - 1
3060     LASTPS = 1
   LAMBDA = 0.0
3080     DO 3100 I=1, NFIT
```

```
3100      PARAM(NUMFIT(I)) = PTEMP(I)
   GO TO 2600

3120      DO 3140 I=1, NFIT
3140      PARAM(NUMFIT(I)) = PTEMP(I) + STEP*DELTA(I)
   CHIB = 0.0
   CALL GENCAL( CTRANS )
   DO 3160 N=1, NUMBER
3160      CHIB = CHIB + ((ydata(N) - CTRANS(N)) / ydataerror(N))**2
   CHIB = CHIB / (NUMBER - NFIT)

   IF (CHIB .LE. CHIA) GO TO 3200

   ISUB = ISUB + 1
   IF (ISUB .LE. ILIMIT) GO TO 3180
   WRITE(7,6750) IPASS - 1
   GO TO 3060

3180      IF (COSINE .LT. REFCOS) GO TO 3190
   STEP = 0.5*STEP
   GO TO 3020

3190      LAMBDA = LAMBDA*NU
   GO TO 3080

3200      DO 3300 I=1, NFIT
3300      SIGMA(I) = SQRT(CHIB*ARRAY(I,I) / ALPHA(I,I))
   CHIA = CHIB
   GO TO 300

4000      continue ! write(7,*)'Successfully inverted matrix'
   DO 4020 K=1, NFIT
   DO 4020 I=1, NFIT
4020      ARRAY(I,K) = CHIA*ARRAY(I,K) /
SQRT(ALPHA(I,I))/SQRT(ALPHA(K,K))

   DO 4040 K=1, NFIT
   SIGMA(K) = SQRT(ARRAY(K,K))
   DO 4040 I=K, NFIT
   IF (I .EQ. K) GO TO 4040
   ARRAY(I,K) = ARRAY(I,K) / SQRT(ARRAY(I,I))/SQRT(ARRAY(K,K))
4040      CONTINUE

4060      call xyznrg
   do N=1,number/2
     zsum = zsum + z(N)
   end do
   zmean = zsum / (number/2)
   write(7,*) 'DeltaZ mean:    ',zmean,' mm'
   VARNCE = FLOAT(NUMBER) * CHIA / SIGMA0
   WRITE(7,7600)
   WRITE(7,7700)
   DO 4120 I=1, NPARAM
   DO 4080 K=1, NFIT
   IF (NUMFIT(K) .EQ. I) GO TO 4100
4080      CONTINUE
   WRITE(7,7900) I, PARAM(I)
   GO TO 4120
4100      WRITE(7,7800) I, PARAM(I), SIGMA(K)
4120      CONTINUE
   WRITE(7,8000) CHIA
   WRITE(7,8600) VARNCE
   WRITE(7,8700) SQRT(VARNCE)
   if (output.LT.2) goto 4890    ! if zero, don't write out any data
```

```
4200      WRITE(7,5900)
   WRITE(7,6000)
   WRITE(7,6100)
   DO N=1, NUMBER, 100
     if (N.lt.NUMBER/2) then
       write(7,6150) N,wedgeu(N), stripu(N), zu(N),
     @ wedgel(N),stripl(N),zl(N)
       write(7,6175) xu1(N),yu1(N),xl1(N),yl1(N),z(N)
     end if
     TEMP = ydata(N) - CTRANS(N)
     WRITE(7,6200) ydata(N), CTRANS(N), TEMP, TEMP / ydataerror(N)
   end do

   if(output.LT.3) goto 4890      ! skip matrices and xyz output
c  open(17,file='xyz.dat')
c  DO N=1, NUMBER/2
c    write(17,6175) xu1(N),yu1(N),xl1(N),yl1(N),time(N)
c  end do
c  close(17)

   DO 4880 L=1, 3
   GO TO (4670, 4680, 4700), L
4670      WRITE(7,8500)
   GO TO 4750
4680      WRITE(7,8100)
   WRITE(7,8200)
   DO 4690 I=1, NFIT
4690      WRITE(7,7850) I, EIGVAL(I), SQRT(ALPHA(I,I))
   WRITE(7,8400)
   GO TO 4750
4700      WRITE(7,6500)
4750      DO 4870 K=1, NFIT, 4
   K0 = MIN0(4, NFIT-K+1)
   IF (K0 .EQ. 0) K0 = 4
   DO 4850 I=1, NFIT
   GO TO (4760, 4780, 4800), L
4760      WRITE(7,6250) (HDIAG(I,N), N=K, K-1+K0)
   GO TO 4850
4780      WRITE(7,6250) (EIGVEC(I,N), N=K, K-1+K0)
   GO TO 4850
4800      WRITE(7,6250) (ARRAY(I,N), N=K, K-1+K0)
4850      CONTINUE
4870      WRITE(7,6800)
4880      CONTINUE
4890      CLOSE(7)

   OPEN(8,file='param.dat')
   write(8,*) NPARAM
   DO I=1, NPARAM
     WRITE(8,*) I, PARAM(I)
   end do
   close(8)
   goto 4900   ! skip plotting for now


C PLOTTING JUNK
c         call opngks
c  call setusv('PB',2)
c  draw a boundary around the edge of the plotter frame
c     call bndary
c  suppress the frame advance
c  call agseti('FRAME.',2)
c  set limits for axes
c      call agseti('X/MAXIMUM.',20)
```

```
c       call agseti('X/MINIMUM.',-20)
c       call agseti('Y/MAXIMUM.',20)
c       call agseti('Y/MINIMUM.',-20)
c  suppress the drawing of curves by the EZ... routines
c       call agseti('SET.',-1)
c  draw the background using EZXY
c       call ezxy (xfov,yfov,nfov,' ')
c  put a plus sign at each of the x-y positions
c       call points (xfov,yfov,nfov,-1,0)
c  put a plus sign at the center of the image
c       call points (x0,y0,1,-2,0)
c        draw the blocking strip
c       call line(-20,-4,20,-4)
c       call line(-20,4,20,4)
c  draw the minimum and maximum reference circumference for
c        finding discrimination of data for detector center and
c        mass distribution purposes
c      call points (1.5,1.5,800,-1,0)
c       call points (20,20,800,-1,0)
c
c  call setusv('MS',1000)

c         DO N=1, NUMBER/2
c     call point (xu1(N),yu1(N))
c          call point (xl1(N),yl1(N))
c  end do

c  close gks
c       call clsgks
c        ierr = system( 'mv gmeta coincs.img' )
c         ierr = system( 'ictrans -d X11 coincs.img' )

C END PLOTTING JUNK

4900      CALL EXIT('Fit concluded')

5000      FORMAT(36HEnter desired number of iterations: )
5050      FORMAT(I4)
5100      FORMAT(33HEnter starting value for lambda: )
5200      FORMAT(D24.0)

5300      FORMAT(/6X, 19HLeast Squares Pass:, I3)
5400      FORMAT(/9HParameter, 3X, 15HParameter Value, 5X,
     1                 11HUncertainty, 5X, 9HIncrement/)
5500      FORMAT(1H , I2, 1H*, 3X, 3(F14.4))
5550      FORMAT(1H , I2, 4X, F14.4)
5700      FORMAT(12HDeterminant:, +1PE12.4, 6X,
     1                   7HLambda:, +1PE12.4)
5900      FORMAT(/8X,37HTABLE OF DEPENDENT VARIABLE RESIDUALS/)
6000      FORMAT(4HData, 6X, 8HMeasured, 4X, 10HCalculated,
     1                   2X, 8HResidual, 4X, 19HNormalized Residual)
6100      FORMAT(5HPoint, 5X, 8HVariable, 3(4X,
     1                   8HVariable)/)
6150      format(I5,6(4X,I5))
6175      format(5(F9.4))
6200      FORMAT(4(F13.5)/)
6250      FORMAT(4(2X, +1PE16.6))
6300      FORMAT(7HCosine:, F12.6, 8X, 5HStep:, F12.6)
6400      FORMAT(/6X, 19HLeast Squares Pass:, I3, 5X,
     1                   29HCONVERGENCE HAS BEEN ACHIEVED)
6500      FORMAT(//5X, 31HCOVARIANCE - CORRELATION MATRIX/)
6700      FORMAT(/6X, 19HLeast Squares Pass:, I3, 5X,
     1                   33HITERATION LIMIT HAS BEEN ACHIEVED)
6750      FORMAT(/6X, 19HLeast Squares Pass:, I3, 5X,
```

```
     1                    36HITERATION SUBLIMIT HAS BEEN ACHIEVED)
6800     FORMAT(1H )
6900     FORMAT(4HData,5X,25HI N D E P E N D E N T    ,
     1          15HV A R I A B L E,8X,21HD E P E N D E N T    ,
     2          15HV A R I A B L E)
7000     FORMAT(1H
,4X,5HPoint,5X,5HValue,5X,11HUncertainty,5X,5HValue,
     1          5X,11HUncertainty/)
7100     FORMAT(1H ,I3,(4X,+1PE16.8,4X,+1PE16.8,X))
7200     FORMAT(5X,35HDATA AND UNCERTAINTIES UTILIZED IN ,
     1          34HLEAST SQUARES REGRESSION PROCEDURE/)
7300     FORMAT(1H1)
7600     FORMAT(/3X,32HOUTPUT FROM FINAL LEAST SQUARES ,
     1                    4HPASS)
7700     FORMAT(/5HParam, 5X, 11HParam Value, 5X,
     1                    11HUncertainty)
7800     FORMAT(1H , I2, 1H*, 2X, 2(2X, F12.5))
7850     FORMAT(1H , I2, 3X, 2(2X, F12.5))
7900     FORMAT(1H , I2, 5X, F12.5)
8000     FORMAT(2X, 20HReduced Chi Squared:, F12.5)
8100     FORMAT(//5X, 32HEIGENVALUES OF SCALED CURVATURE
     1                    6HMATRIX/)
8200     FORMAT(/6HNumber, 10X, 10HEigenvalue, 13X,
     1                    12HScale Factor)
8300     FORMAT(1H , I3, 4X, 2(8X, +1PE16.8))
8400     FORMAT(//7X, 33HEIGENVECTORS OF SCALED CURVATURE
     1                    6HMATRIX//)
8500     FORMAT(//5X, 23HSCALED CURVATURE MATRIX//)
8600     FORMAT(20HVariance of the Fit:, +1PE16.5)
8700     FORMAT(30HStandard Deviation of the Fit:, +1PE16.5/)

   END
```

xyzin.f

```
   SUBROUTINE XYZLINDISC
c
c Reads in experimental parameters, and data, with discrimination
c Allows the reading in of multiple data sets with different beam
energies
c  for simultaneous fitting
c
c Requires:
c tcal.dat holds tdelay and some tdc calibration data points
c adc_cal.dat holds calibration for the adc converters
c phd.dat holds raw data upper and lower discrimination levels
c All other discrimination values read in from standard input
c O2 data filename(s) are read in from standard input
c
c Subroutine linfit.f is called to fit a line through the calibration
data
c
   implicit none
   COMMON /BLOCK1/ PARAM(20), NPARAM
   COMMON /BLOCK2/ ydata(50000),ydataerror(50000),NUMBER
   COMMON /experimental/ length,energyO(10),massO
   common /rawdata/ wedgeu(25000),wedgel(25000),stripu(25000),
     @ stripl(25000),zu(25000),zl(25000),z(25000),time(25000),x0,y0

   real*8 param,x0,y0
   real*4 Rmax,Rerrormax,Cmax,xu,yu,xl,yl,phi
   real*4 Rmin,qs,qw,fwu,fsu,fzu,fwl,fsl,fzl,tcu,tcl
```

```
      real*4 az,bz,zz,tiltx,tilty
      real*4 zmax,zmin,xumax,xumin,xlmax,xlmin,phimin,phimax
      real*4 yumax,yumin,ylmax,ylmin
      REAL*4 ydata,ydataerror
      REAL*4 length,beamnrg(10),energyO,massO,vbeam(10),nrgoffset
      REAL*4 R(10),Rxyerror,Rzerror,centersize
      real*4 tint,tslope,tdelay,z,time,numread
      REAL*4 awedgeu,bwedgeu,astripu,bstripu,ctwzu,ctszu
      REAL*4 awedgel,bwedgel,astripl,bstripl,ctwzl,ctszl
      real*4 tactrue(30),tacval(30),cw1,cs1,cz1,cw2,cs2,cz2
      real*4 vadc(30),adcw1(30),adcs1(30),adcz1(30)
      real*4 adcw2(30),adcs2(30),adcz2(30),adcscale
      real*4 sw1,ss1,sz1,sw2,ss2,sz2
      real*4 w1int,s1int,z1int,w2int,s2int,z2int
      real*4 TB(0:17),recoilnrg(0:17)

      integer wedgeu,wedgel,stripu,stripl,zu,zl
      integer NUMBER,N,discRmax,discCmax
      integer discyumax,discyumin,discylmax,discylmin
      integer discRmin,discRerrormax,discphimax,discphimin
      integer disczmin,disczmax,discxumin,discxumax,discxlmin,discxlmax
      integer I,nparam,NN,vib
      integer Nfiles,numpoints(10),discflag,discnum
      integer ntoss,dum1,dum2,dum3,dum4,dum5,dum6,dum7
      integer ntac,nadc
      integer iuw,ius,iuz,ilw,ils,ilz,itac,discraw
      integer idtl,idtu,idlsumu,idusumu,idlsuml,idusuml

      CHARACTER*40     FILENM(10)

      DATA   FILENM    /10* '                                        ' /
c
      massO = 15.9994 * 1.660565E-27            ! mass of O atom in kg
c
           TB(0) = 49356.81
           TB(1) = 50044.37
           TB(2) = 50709.53
           TB(3) = 51350.78
           TB(4) = 51968.26
           TB(5) = 52559.75
           TB(6) = 53121.44
           TB(7) = 53654.72
           TB(8) = 54154.92
           TB(9) = 54621.12
           TB(10) = 55049.72
           TB(11) = 55437.75
           TB(12) = 55782.95
           TB(13) = 56083.68
           TB(14) = 56338.43
           TB(15) = 56548.26
           TB(16) = 56716.64
           TB(17) = 56848.65
c
c TB(vB) is defined with respect to v = 0 of X state; 5.1166 is D0 of
c  X state in eV
c
      do i = 0,17
        recoilnrg(i) = TB(i) / 8065.479 - 5.1166
c       write(6,*) 'v =',i,' recoil energy = ',recoilnrg(i)
      end do
c
c
c *** bob stuff (tac and adc calibration)
c
```

```
c Read in time-to-amplitude converter calibration numbers (and
tdelay)
   open (12,file='tcal.dat',status='old')
   read (12,*) tdelay
   ntac = 0
50 ntac = ntac + 1
   read (12,*,end=60) tactrue(ntac),tacval(ntac)
   goto 50
60 ntac = ntac - 1
   close(12)
c
c  call linfit to get the linear fit to the TAC calibration curve
c
      call linfit(tactrue,tacval,ntac,tslope,tint)
   write(6,*)'tslope:',tslope,' tint:',tint
c
c       read in adc calibration curve data from fort.10
c
   open(10,file='adc_cal.dat',status='old')
        i = 0
        read(10 , *)cw1,cs1,cz1,cw2,cs2,cz2
70      i = i + 1
        read(10, *, end=80)vadc(i),adcw1(i),adcs1(i),adcz1(i),
     >                     adcw2(i),adcs2(i),adcz2(i)
   goto 70
80      nadc = i - 1
   close(10)
c
c       call linfit to get the linear fit to all the other ADC
channels
c       note, adcscale in units of adc unit/volt
c
   adcscale = 8064.0/10.0
   call linfit(adcw1,vadc,nadc,sw1,w1int)
c  write(6,*)'sw1:',sw1,' w1int:',w1int
   call linfit(adcs1,vadc,nadc,ss1,s1int)
c  write(6,*)'ss1:',ss1,' s1int:',s1int
   call linfit(adcz1,vadc,nadc,sz1,z1int)
c  write(6,*)'sz1:',sz1,' z1int:',z1int
   call linfit(adcw2,vadc,nadc,sw2,w2int)
c  write(6,*)'sw2:',sw2,' w2int:',w2int
   call linfit(adcs2,vadc,nadc,ss2,s2int)
c  write(6,*)'ss2:',ss2,' s2int:',s2int
   call linfit(adcz2,vadc,nadc,sz2,z2int)
c  write(6,*)'sz2:',sz2,' z2int:',z2int
c Read in raw data discrimination levels
c  format of input in unit 14:
c        upper wedge lower and upper level
c        upper strip lower and upper level
c        upper zig   lower and upper level
c        lower wedge lower and upper level
c        lower strip lower and upper level
c        lower zig   lower and upper level
c        tac         lower and upper level
   open (14,file='phd.dat',status='old')
   read(14,*) idusuml,idusumu,idlsuml,idlsumu,idtl,idtu
   close(14)
c
c *** end bob continetti-era code
c  write(6,*) 'x0, y0?'
   read(5,*) x0,y0
c  write(6,*)'Centroid coords:',x0,y0
c  write(6,*)'Flight path length following dissociation? (in mm)'
   read(5,*) length
```

```
   write(6,*)'flight length=',length,' mm'
c
c older recoil energy code, using ground vibrational level and
wavelength;
c    replaced by upper vibrational level method    DJL 8/93
c
c  O2 constants used below:  D00=5.1166 eV   omegae=1580.361 cm-1
c       omegae*xe = 12.0730 cm-1   Be = 1.44567 cm-1  alphae =
0.01579 cm-1
c
c  energyO(I) = (1/210.228)*1e7
c     @ + (4.5*1580.361 - 4.5*4.5*12.0730 + 2*1.44567 -
2*4.5*0.01579)
c     @ - (0.5*1580.361 - 0.5*0.5*12.0730 + 2*1.44567 -
2*0.5*0.01579)
c     @ - 5.1166*8065.479              ! energy in cm-1
c
c Convert to eV, and subtract off energy of atoms in (J=2,J=1)
c E(J=2) = 0; E(J=1) = 0.0196 eV; E(J=0) = 0.0281 eV
c
*************Note:  Commented out (6,*) stuff for ease of
viewing******
c  write(6,*)'Uncertainty in xy-recoil? (mm)'
   read(5,*) Rxyerror
c  write(6,*)'Uncertainty in z-recoil? (mm)'
   read(5,*) Rzerror
c  write(6,*)'Uncertainty in centroid radius (mm)?'
   read(5,*) centersize
c Read in discrimination parameters
c  write(6,*) 'Minimum and maximum radii on detector? (mm)'
   read(5,*) Rmin, Rmax
c  write(6,*) 'Maximum recoil error? (mm)'
   read(5,*) Rerrormax
c  write(6,*) 'Maximum centroid radius? (mm)'
   read(5,*) Cmax
c  write(6,*) 'Min. and max. upper y values? (mm)'
   read (5,*) yumin, yumax
c  write(6,*) 'Min. and max. abs(lower y) values? (mm)'
   read (5,*) ylmin,ylmax
c  write(6,*) 'Min. and max. upper x values? (mm)'
   read (5,*) xumin,xumax
c  write(6,*) 'Min. and max. lower x values? (mm)'
   read (5,*) xlmin,xlmax
c  write(6,*) 'Min. and max. (signed) z values? (mm)'
   read (5,*) zmin,zmax
c  write(6,*) 'Min. and max. phi value? (degrees)'
   read (5,*) phimin,phimax

c Parameter list:
c 1 awedgeu  2 bwedgeu  3 astripu  4 bstripu
c 5 awedgel  6 bwedgel  7 astripl  8 bstripl
c 9-12 cross talk terms
   awedgeu = PARAM(1)
   bwedgeu = PARAM(2)
   astripu = PARAM(3)
   bstripu = PARAM(4)
   awedgel = PARAM(5)
   bwedgel = PARAM(6)
   astripl = PARAM(7)
   bstripl = PARAM(8)
   ctwzu   = PARAM(9)
   ctwzl   = PARAM(11)
c tilt modification
c  ctszu   = PARAM(10)
```

```
c  ctszl    = PARAM(12)
   ctszu = 1.
   ctszl = 1.
   tiltx = param(10)
   tilty = param(12)
c end tilt modification
   az     = PARAM(13)
   bz     = PARAM(14)
   qw       = PARAM(15)
   qs       = PARAM(16)

c  write(6,*) 'How many oxygen data files?'
   read(5,*) Nfiles
   do I=1,Nfiles
c    WRITE(6,*) 'How many coincidence events from file',I,'?'
     read(5,*) numpoints(I)
c    WRITE(6,*) 'Skip how many events in file',I,'?'
     read(5,*) ntoss
c    WRITE(6,*)'Enter name of data file',I,':'
     READ (5,8000) FILENM(I)
c
c Get vibrational level, beam energy; calculate recoil energy and
distance
c
c  write(6,*) 'B state vibrational level for file',I,'? '
   read(5,*) vib
   write(6,*)'O2 dissociation from vB =',vib
   energyO(I) = recoilnrg(vib)   ! ground state (2,2) limit in eV
c  write(6,*)'Mean energy offset from (2,2) limit? (meV)'
   write(6,*)'(Enter 0 for (2,1) limit at 19.6 meV)'
   read(5,*) nrgoffset
   if (nrgoffset.eq.0) nrgoffset = 19.6
   energyO(I) = energyO(I) - nrgoffset/1000
   write(6,*)'Ediss of J=2,J=2:',recoilnrg(vib),
     @ '; fit to mean:',energyO(I)
   energyO(I) = energyO(I) * 1.6021917D-19     ! convert eV to Joules
c    write(6,*)'Beam energy for file',I,'? (in keV)'
     read(5,*) beamnrg(I)
     write(6,*)'beamenergy:',beamnrg(I),' keV'
     beamnrg(I) = beamnrg(I) * 1.602E-16    ! convert keV to Joules
     vbeam(I) = 1000 * sqrt(beamnrg(I) / massO)
     write(6,*)'beam velocity =',vbeam(I)/1000,' m/s'
     R(I) = 2 * length * sqrt(energyO(I)/beamnrg(I))
     write(6,*) 'relative particle recoil R =',R(I),' mm'
c Read in raw data
     OPEN (3, NAME = filenm(I), STATUS = 'OLD')
     do N = 1, ntoss
       READ(3,*) dum1,dum2,dum3,dum4,dum5,dum6,dum7
     end do
     N=1
c Begin discriminatory read-in loop
100        discflag = 0
     NN = number + N
c
     READ(3,*,END=350) iuw,ius,iuz,itac,ilw,ils,ilz
c
     if( nint((iuw+ius+iuz)/4.0) .lt. idusuml ) goto 200
     if( nint((ilw+ils+ilz)/4.0) .lt. idlsuml ) goto 200
     if( nint((iuw+ius+iuz)/4.0) .gt. idusumu ) goto 200
     if( nint((ilw+ils+ilz)/4.0) .gt. idlsumu ) goto 200
     if((itac.lt.idtl).or.(itac.gt.idtu)) go to 200
   goto 300
200      discflag = 1
   discraw = discraw + 1
```

```
c       correct ADC data using the appropriate calibration curves
c
300      wedgeu(NN) =   adcscale * ( ( cw1*sw1*float(iuw) ) + w1int )
        stripu(NN) =   adcscale * ( ( cs1*ss1*float(ius) ) + s1int )
        zu(NN) = 2e0 * adcscale * ( ( cz1*sz1*float(iuz) ) + z1int )
        wedgel(NN) =   adcscale * ( ( cw2*sw2*float(ilw) ) + w2int )
        stripl(NN) =   adcscale * ( ( cs2*ss2*float(ils) ) + s2int )
        zl(NN) = 2e0 * adcscale * ( ( cz2*sz2*float(ilz) ) + z2int )
c
      tcu = wedgeu(NN) + stripu(NN) + zu(NN)
      tcl = wedgel(NN) + stripl(NN) + zl(NN)
      fwu = wedgeu(NN)/tcu
      fsu = stripu(NN)/tcu
      fzu = zu(NN)/tcu
      fwl = wedgel(NN)/tcl
      fsl = stripl(NN)/tcl
      fzl = zl(NN)/tcl
c
      xu = -awedgeu*(fwu+(ctwzu-1)*fzu+(qw-1)*fwu*fwu)-bwedgeu
      yu = astripu*(fsu+(ctszu-1)*fzu+(qs-1)*fsu*fsu)+bstripu
      xl = awedgel*(fwl+(ctwzl-1)*fzl+(qw-1)*fwl*fwl)+bwedgel
      yl = -astripl*(fsl+(ctszl-1)*fzl+(qs-1)*fsl*fsl)-bstripl
      z(NN) = (((itac - tint)/tslope)-tdelay) * (1e-9) * vbeam(I)
      zz = az * z(NN) + bz + tiltx*(xu-xl) + tilty*(yu-yl)

      if ((xu-xl).eq.0) then
        phi=90
      else
        phi=(180/3.14159)*atan((yu-yl)/(xu-xl))
        if (phi.lt.0) phi=phi+180
      end if
      if (min((xu*xu + yu*yu),(xl*xl + yl*yl)).LT.(Rmin*Rmin)) then
        discRmin = discRmin + 1
        discflag = 1
      end if
      if (max((xu*xu + yu*yu),(xl*xl + yl*yl)).GT.(Rmax*Rmax)) then
        discRmax = discRmax + 1
        discflag = 1
      end if
      if (abs(R(I)-sqrt((xu-xl)*(xu-xl)+(yu-yl)*(yu-yl)+zz*zz))
      @          .GT. Rerrormax) then
        discRerrormax = discRerrormax + 1
        discflag = 1
      end if
      if (((xu+xl-2*x0)*(xu+xl-2*x0) + (yu+yl-2*y0)*(yu+yl-2*y0))
      @          .GT. 4*Cmax*Cmax ) then
        discCmax = discCmax + 1
        discflag = 1
      end if
      if (phi .LT. phimin) then
        discphimin = discphimin + 1
        discflag = 1
      end if
      if (phi .GT. phimax) then
        discphimax = discphimax + 1
        discflag = 1
      end if
      if (yu .LT. yumin) then
        discyumin = discyumin + 1
        discflag = 1
      end if
      if (yu .GT. yumax) then
        discyumax = discyumax + 1
        discflag = 1
```

```
      end if
      if (-yl .LT. ylmin) then
        discylmin = discylmin + 1
        discflag = 1
      end if
      if (-yl .GT. ylmax) then
        discylmax = discylmax + 1
        discflag = 1
      end if
      if (xu .LT. xumin) then
        discxumin = discxumin + 1
        discflag = 1
      end if
      if (xu .GT. xumax) then
        discxumax = discxumax + 1
        discflag = 1
      end if
      if (xl .LT. xlmin) then
        discxlmin = discxlmin + 1
        discflag = 1
      end if
      if (xl .GT. xlmax) then
        discxlmax = discxlmax + 1
        discflag = 1
      end if
      if (z(NN) .LT. zmin) then
        disczmin = disczmin + 1
        discflag = 1
      end if
      if (z(NN) .GT. zmax) then
        disczmax = disczmax + 1
        discflag = 1
      end if
      if (discflag.EQ.0) then
        ydata(NN) = R(I)
        ydataerror(NN) = ( sqrt( (xu-xl)*(xu-xl) + (yu-yl)*(yu-yl) )
     @ *Rxyerror + abs(zz)*Rzerror ) / R(I)
        time(NN) = (itac - tint)/tslope
        N = N + 1      !increment successful event index!
      else
        discnum = discnum + 1
      endif
      IF (N.LE.numpoints(I)) GOTO 100
C Finished with data file I
350         numpoints(I) = N - 1
      number = number + numpoints(I)
      write(6,*) numpoints(I),' points read from ',filenm(I)
      CLOSE (3)
    end do       ! end of data file loop (index I)

    do N = 1, NUMBER
      ydata(N + NUMBER)=0.0
      ydataerror(N + NUMBER) = centersize
    end do
    NUMBER=NUMBER*2          ! two "data points" per coincidence event

    numread = number/2 + discnum
    write(6,*)'Raw data discrimination:',100*discraw/numread,' %'
    WRITE(6,*)'Min detector radius:',Rmin,' mm',
      @          100*discRmin/numread,' %'
    WRITE(6,*)'Max detector radius:',Rmax,' mm',
      @          100*discRmax/numread,' %'
    WRITE(6,*)'Max recoil error:    ',Rerrormax,' mm',
      @          100*discRerrormax/numread,' %'
```

```
WRITE(6,*)'Max centroid radius:',Cmax,' mm',
  @          100*discCmax/numread,' %'
WRITE(6,*)'yu min:  ',yumin,' mm',
  @          100*discyumin/numread,' %'
WRITE(6,*)'yu max:  ',yumax,' mm',
  @          100*discyumax/numread,' %'
WRITE(6,*)'yl min:  ',ylmin,' mm',
  @          100*discylmin/numread,' %'
WRITE(6,*)'yl max:  ',ylmax,' mm',
  @          100*discylmax/numread,' %'
WRITE(6,*)'xu min:  ',xumin,' mm',
  @          100*discxumin/numread,' %'
WRITE(6,*)'xu max:  ',xumax,' mm',
  @          100*discxumax/numread,' %'
WRITE(6,*)'xl min:  ',xlmin,' mm',
  @          100*discxlmin/numread,' %'
WRITE(6,*)'xl max:  ',xlmax,' mm',
  @          100*discxlmax/numread,' %'
WRITE(6,*)'zmin:    ',zmin,' mm',
  @          100*disczmin/numread,' %'
WRITE(6,*)'zmax:    ',zmax,' mm',
  @          100*disczmax/numread,' %'
WRITE(6,*)'phimin:  ',phimin,' degrees',
  @          100*discphimin/numread,' %'
WRITE(6,*)'phimax:  ',phimax,' degrees',
  @          100*discphimax/numread,' %'
write(6,*)'Percentage thrown out:',100.*discnum/numread,' %'

OPEN(UNIT=7,FILE='o2.dat')
write(7,*)'tslope:      ',tslope,' ADC units/ns'
write(7,*)'tint:        ',tint,' ADC units'
write(7,*)'centroid:    ',x0,y0,' mm'
write(7,*)'length:      ',length,' mm'
do I=1,Nfiles
  write(7,*) numpoints(I),' points read from ',filenm(I)
  write(7,*)'O2 dissociation from vB =',vib
  write(7,*)'Ediss of J=2,J=2:',recoilnrg(vib),
  @ '; fit to mean:',energyO(I)/1.602E-19

  WRITE(7,*)'beam energy:   ',beamnrg(I)/1.602E-16,' keV'
  write(7,*)'beam velocity: ',vbeam(I)/1000,' m/s'
  write(7,*)'recoil (R):    ',R(I),' +/-',
  @          Rxyerror,'(xy)',Rzerror,'(z)'
end do
write(7,*)'centroid radial uncertainty:',centersize,' mm'
write(7,*)'Raw data discrimination:    ',100*discraw/numread,' %'
WRITE(7,*)'Min detector radius:',Rmin,' mm',
  @          100*discRmin/numread,' %'
WRITE(7,*)'Max detector radius:',Rmax,' mm',
  @          100*discRmax/numread,' %'
WRITE(7,*)'Max recoil error:   ',Rerrormax,' mm',
  @          100*discRerrormax/numread,' %'
WRITE(7,*)'Max centroid radius:',Cmax,' mm',
  @          100*discCmax/numread,' %'
WRITE(7,*)'yu min:  ',yumin,' mm',
  @          100*discyumin/numread,' %'
WRITE(7,*)'yu max:  ',yumax,' mm',
  @          100*discyumax/numread,' %'
WRITE(7,*)'yl min:  ',ylmin,' mm',
  @          100*discylmin/numread,' %'
WRITE(7,*)'yl max:  ',ylmax,' mm',
  @          100*discylmax/numread,' %'
WRITE(7,*)'xu min:  ',xumin,' mm',
  @          100*discxumin/numread,' %'
```

```
      WRITE(7,*)'xu max:  ',xumax,' mm',
        @          100*discxumax/numread,' %'
      WRITE(7,*)'xl min:  ',xlmin,' mm',
        @          100*discxlmin/numread,' %'
      WRITE(7,*)'xl max:  ',xlmax,' mm',
        @          100*discxlmax/numread,' %'
      WRITE(7,*)'z min:   ',zmin,' mm',
        @          100*disczmin/numread,' %'
      WRITE(7,*)'z max:   ',zmax,' mm',
        @          100*disczmax/numread,' %'
      WRITE(7,*)'phi min: ',phimin,' degrees',
        @          100*discphimin/numread,' %'
      WRITE(7,*)'phi max: ',phimax,' degrees',
        @          100*discphimax/numread,' %'
      write(7,*)'Total percentage disc''d:',100.*discnum/numread,' %'

 9999     RETURN

 8000     FORMAT(A)

      END
```

genpin.f

```
      SUBROUTINE GENPIN ( NFIT, NUMFIT, IFLAG )

      COMMON /BLOCK1/  PARAM(20), NPARAM

      real*8 PARAM

      integer        NUMFIT(20)
      CHARACTER*1      ANSWER

c     WRITE(6,5000)
      READ (5,5100) NPARAM

      NFIT = 0
c     WRITE(6,5200)
      DO 100 I=1, NPARAM
      NUMFIT(I) = 0
c     WRITE(6,5300) I
      READ (5,*) PARAM(I)
c     WRITE(6,*) PARAM(I)
      IF (IFLAG .EQ. 0) GO TO 100
c     WRITE(6,5400)
      READ (5,5600) ANSWER
      IF ((ANSWER .NE. 'V').and.(ANSWER .NE. 'v')) GO TO 100
      NFIT = NFIT + 1
      NUMFIT(NFIT)  = I
 100      CONTINUE

      RETURN

 5000     FORMAT(35HEnter number of parameters in fit: )
 5100     FORMAT(I4)
 5200     FORMAT(50HEnter initial values and flags for each
Parameter:)
 5300     FORMAT(4X,9HParameter,I3,5X,7HValue: )
 5400     FORMAT(22X,6HFlag: )
 5600     FORMAT(A1)

      END
```

xyzcal.f

```
      SUBROUTINE GENCAL ( CY )

      implicit none
      COMMON /BLOCK1/ PARAM(20), NPARAM
      COMMON /BLOCK2/ ydata(50000),ydataerror(50000),NUMBER
      COMMON /rawdata/ wedgeu(25000),wedgel(25000),stripu(25000),
        @ stripl(25000),zu(25000),zl(25000),z(25000),time(25000),x0,y0
      common /procdata/ xu1(25000),xl1(25000),yu1(25000),yl1(25000)
      common /experimental/ length,energy0(10),mass0

      real*8 PARAM
      real*4 CY(50000),ydata,ydataerror,z,time
      real*4 length,energy0,mass0
      REAL*8 xu,yu,xl,yl,x0,y0,xc,yc,eps,az,bz,zz
      real*4 xu1,xl1,yu1,yl1
      REAL*8 awedgeu,bwedgeu,astripu,bstripu,ctwzu,qw,qs
      REAL*8 awedgel,bwedgel,astripl,bstripl,ctwzl
      integer wedgeu,wedgel,stripu,stripl,zu,zl
      real*8 fwu,fsu,fzu,fwl,fsl,fzl,tcu,tcl
      real*8 tiltx,tilty
      INTEGER NUMBER,NPARAM,N


c Parameter list:
c 1 awedgeu  2 bwedgeu  3 astripu  4 bstripu
c 5 awedgel  6 bwedgel  7 astripl  8 bstripl
c 9-16 cross talk terms
      awedgeu = PARAM(1)
      bwedgeu = PARAM(2)
      astripu = PARAM(3)
      bstripu = PARAM(4)
      awedgel = PARAM(5)
      bwedgel = PARAM(6)
      astripl = PARAM(7)
      bstripl = PARAM(8)
      ctwzu   = PARAM(9)
      ctwzl   = PARAM(11)
c modification, so that I can include tilt parameters
      tiltx   = PARAM(10)
      tilty   = PARAM(12)
c end tilt modification
      az      = PARAM(13)
      bz      = PARAM(14)
      qw      = PARAM(15)
      qs      = PARAM(16)
c
      DO N=1, NUMBER/2
        tcu = wedgeu(N) + stripu(N) + zu(N)
        tcl = wedgel(N) + stripl(N) + zl(N)
        fwu = wedgeu(N)/tcu
        fsu = stripu(N)/tcu
        fzu = zu(N)/tcu
        fwl = wedgel(N)/tcl
        fsl = stripl(N)/tcl
        fzl = zl(N)/tcl
c
        xu = -awedgeu*(fwu+(ctwzu-1)*fzu+(qw-1)*fwu*fwu)-bwedgeu
        yu = astripu*(fsu+(qs-1)*fsu*fsu)+bstripu
        xl = awedgel*(fwl+(ctwzl-1)*fzl+(qw-1)*fwl*fwl)+bwedgel
        yl = -astripl*(fsl+(qs-1)*fsl*fsl)-bstripl
        zz = az * z(N) + bz + tiltx*(xu-xl) + tilty*(yu-yl)
c
```

```
c We assign these arrays for export to other routines, especially
xyznrg
c
      xu1(N) = xu
      yu1(N) = yu
      xl1(N) = xl
      yl1(N) = yl
c
      cy(N) = sqrt( (xu-xl)*(xu-xl) + (yu-yl)*(yu-yl) + zz*zz )
c
c  Centroid is corrected for time-dependence of each particle's
recoil
c         (first order, assumes z(N) << length)
      eps = z(N)/length
      xc = xu*(1+eps/2)/2 + xl*(1-eps/2)/2
      yc = yu*(1+eps/2)/2 + yl*(1-eps/2)/2
c
      cy(N + NUMBER/2) = sqrt( (xc-x0)*(xc-x0) + (yc-y0)*(yc-y0) )
    end do
    RETURN
    END
```

jmdiag.f

```
      SUBROUTINE JMDIAG ( NORDER )

      COMMON /BLOCK3/  HTOTAL(20,20), EIGVEC(20,20), EIGVAL(20)

      real*8 HTOTAL, EIGVEC, EIGVAL, THRES, ORDER
      real*8 FTHRES, C, D(82,4), S, T1, T2, W, DEGREE

      PARAMETER         ( DEGREE = 2.5D-17 )

      THRES  = 0.0D0
      ORDER  = FLOAT(NORDER)

      DO 200 I=1,NORDER
      DO 200 J=1,NORDER
      IF (I .EQ. J) GO TO 100
      EIGVEC(I,J) = 0.0D0
      THRES = THRES + HTOTAL(I,J)**2
      GO TO 200
100       EIGVEC(I,J) = 1.0D0
200       CONTINUE

      FTHRES = THRES*DEGREE/ORDER

300       THRES = THRES/ORDER
      IF (THRES .LE. FTHRES) RETURN

400       DO 500 I=1,NORDER
      DO 500 J=I,NORDER
      if (htotal(i,j).eq.0) stop('crash in jmdiag!')
      IF (ABS(HTOTAL(I,J)) .GE. THRES .AND. I .NE. J) GO TO 600
500       CONTINUE
      GO TO 300

600       T1 = -HTOTAL(I,J)
      T2 = 5.0D-1 * (HTOTAL(I,I) - HTOTAL(J,J))
      W  = T1 * SIGN(1.0D0,T2) / SQRT(T1*T1 + T2*T2)
      S  = W / SQRT(2.0D0 * (1.0D0 + SQRT(ABS(1.0D0 - W*W))))
      C  = SQRT(1.0D0 - S*S)
```

```
      DO 700 K=1, NORDER
      D(K,1) = HTOTAL(I,K)
      D(K,2) = HTOTAL(J,K)
      D(K,3) = EIGVEC(K,I)
700       D(K,4) = EIGVEC(K,J)

      DO 800 K=1,NORDER
      EIGVEC(K,I) = C*D(K,3) - S*D(K,4)
      EIGVEC(K,J) = S*D(K,3) + C*D(K,4)
      IF (K .EQ. I .OR. K .EQ. J) GO TO 800
      HTOTAL(I,K) = C*D(K,1) - S*D(K,2)
      HTOTAL(K,I) = HTOTAL(I,K)
      HTOTAL(J,K) = S*D(K,1) + C*D(K,2)
      HTOTAL(K,J) = HTOTAL(J,K)
800       CONTINUE

      HTOTAL(I,I) = C*C*D(I,1) - 2.0D0*C*S*D(J,1) + S*S*D(J,2)
      HTOTAL(J,J) = S*S*D(I,1) + 2.0D0*C*S*D(J,1) + C*C*D(J,2)
      HTOTAL(I,J) = (D(I,1) - D(J,2))*S*C + (C*C - S*S)*D(I,2)
      HTOTAL(J,I) = HTOTAL(I,J)

      GO TO 400

      END
```

matinv.f

```
C*********************************************************************
***
C
*
C       Subroutine: MATINV.FOR - v2.1
*
C
*
C       Date:       September 1982
*
C
*
C       Purpose:    To find the inverse of a symmetric matrix.
*
C
*
C       Parameters: MATRIX - on input,  the matrix to be inverted.
*
C                            on output, the inverted matrix.
*
C                   NORDER - the order of the matrix to be
*
C                            inverted.
*
C                   DETERM - the determinant of the input matrix
*
C
*
C       Comments:   See "Data Reduction and Error Analysis for
*
C                   the Physical Sciences" by P. R. Bevington.
*
C
*
```

```
C*********************************************************************
***
C Modified dimension sizes from 25 to 20    David Leahy, Sept 1992
C Also:  made function calls generic (rather than double prec.)
C   changed scalar definitions to generic (e.g., from 1.0D0 to 1.0)
C   changed variable type definitions (e.g., doub. prec. to REAL*8)

   SUBROUTINE MATINV ( MATRIX, NORDER, DETERM )

   REAL*8              MATRIX(20,20), DETERM, MAXVAL, TEMP
   INTEGER           INUM(20), JNUM(20)

   DETERM = 1.0

   DO 1000 K=1, NORDER
   MAXVAL = 0.0
   DO 100 I=K, NORDER
   DO 100 J=K, NORDER
   IF (ABS(MAXVAL) .GE. ABS(MATRIX(I,J))) GO TO 100
   MAXVAL  = MATRIX(I,J)
   INUM(K) = I
   JNUM(K) = J
100      CONTINUE

   IF (MAXVAL .NE. 0.0) GO TO 200
   DETERM = 0.0
   RETURN

200      N0 = INUM(K)
   N1 = JNUM(K)
   IF (N0 .EQ. K) GO TO 400
   DO 300 J=1, NORDER
   TEMP          = MATRIX(K,J)
   MATRIX(K,J)   = MATRIX(N0,J)
300      MATRIX(N0,J) = -TEMP
400      IF (N1 .EQ. K) GO TO 600
   DO 500 I=1, NORDER
   TEMP          = MATRIX(I,K)
   MATRIX(I,K)   = MATRIX(I,N1)
500      MATRIX(I,N1) = -TEMP

600      DO 700 I=1, NORDER
   IF (I .EQ. K) GO TO 700
   MATRIX(I,K) = -MATRIX(I,K) / MAXVAL
700      CONTINUE
   DO 850 I=1, NORDER
   IF (I .EQ. K) GO TO 850
   DO 800 J=1, NORDER
   IF (J .EQ. K) GO TO 800
   MATRIX(I,J) = MATRIX(I,J) + MATRIX(I,K)*MATRIX(K,J)
800      CONTINUE
850      CONTINUE
   DO 900 J=1, NORDER
   IF (J .EQ. K) GO TO 900
   MATRIX(K,J) = MATRIX(K,J) / MAXVAL
900      CONTINUE

   MATRIX(K,K) = 1.0 / MAXVAL
1000     DETERM = DETERM * MAXVAL

   DO 1400 L=1, NORDER
   K  = NORDER - L + 1
   N0 = INUM(K)
   N1 = JNUM(K)
```

```
      IF (N0 .LE. K) GO TO 1200
      DO 1100 I=1, NORDER
      TEMP          = MATRIX(I,K)
      MATRIX(I,K)   = -MATRIX(I,N0)
1100      MATRIX(I,N0) = TEMP
1200      IF (N1 .LE. K) GO TO 1400
      DO 1300 J=1, NORDER
      TEMP          = MATRIX(K,J)
      MATRIX(K,J)   = -MATRIX(N1,J)
1300      MATRIX(N1,J) = TEMP
1400      CONTINUE

      RETURN
      END
```

xyznrg.f

```
      SUBROUTINE xyznrg
      implicit none
c
      COMMON /BLOCK1/  PARAM(20), NPARAM
      COMMON /BLOCK2/  ydata(50000),ydataerror(50000),NUMBER
      COMMON /experimental/ length,energyO(10),massO
      COMMON /rawdata/ wedgeu(25000),wedgel(25000),stripu(25000),
        @ stripl(25000),zu(25000),zl(25000),z(25000),time(25000),x0,y0
      common /procdata/ xu1(25000),xl1(25000),yu1(25000),yl1(25000)
           common /smoothblock/ counts(-20:20,-20:20)
c
      integer nbins
      parameter (nbins=200) ! number of bins in nrg arrays (actually,
c           ! it's nbins + 1 bins, 'cuz the arrays go from 0:nbins)
c
      real*8 PARAM
      real*4 xu1,xl1,yu1,yl1
      integer wedgeu,stripu,zu,wedgel,stripl,zl,itheta
      real*4 calcY(50000),ydata,ydataerror,nrgstddev,nrgstddevfit
      REAL*4 nrg(25000),nrgsum,nrgmean,nrgsumfit
      real*4 xu2,xl2,yu2,yl2,nrg2
      real*4 deltaE,deltaR,deltax,deltay,bin,r,rprime
      real*4 z2d(-20:20,-20:20),nrg2d(-20:20,-20:20)
      real*4 recoilxy(-20:20,-20:20),deltaR2d(-20:20,-20:20)
      real*4 nrgz1d(-60:60),theta,Ptheta(-90:90)
      real*4 nzpos(-20:20,-20:20),nzneg(-20:20,-20:20)
      real*4 zposmean(-20:20,-20:20),znegmean(-20:20,-20:20)
      real*4 deltaz(-20:20,-20:20), deltazsmooth(-20:20,-20:20)
      real*4 length,energyO,massO,z,z2,time,x0,y0,energyO1,energyON
      integer nrgbin(0:4000),peakbin,leftbin,rightbin
      integer nrgbin3d,xbin,ybin,nrg3d(-20:20,-20:20,0:nbins)
      integer raw2d(-20:20,-20:20),counts
      integer nrgcorr(0:nbins),nrgcorrz(0:nbins)
      integer zbin,countsz1d(-60:60),nrgz2d(-60:60,0:nbins)
      INTEGER NUMBER,NPARAM,N,binsize,diagnostics,topbottom
      integer binsize,nrgmin,nrgmax
c
      binsize = 1  ! number of meV per bin in nrg arrays
      nrgmin = 1400   ! nrg of first bin in nrg arrays
      nrgmax = nrgmin + nbins*binsize   ! nrg of last bin in nrg arrays
c
c diagnostics = 0 writes nothing
c        = 1 writes nrg.dat
c        = 2 writes all the 1d, 2d and 3d diagnostic arrays
c        = 3 writes xyzcorr.dat (cmlwsa-ready input)
c
```

```
c  write(6,*) 'Write out 3-D diagnostics? (0=no, 1=nrg, 2=all)'
   read(5,*) diagnostics
c
c Calculate energy distribution given by data + fit parameters
   call gencal(calcy)
c
c Kluge:  energy01 is the energy0 of the first data set read in; this
c routine will correctly give the recoil energies nrg(N) only if the
beam
c energy is equal for all of the data sets!!  If the beam energies
are
c different, but the recoil energies are all the same, use the
commented
c out nrg(N) line
c
   energy01 = energy0(1) * 1000 / 1.60219e-19   ! convert to meV
   do N=1,number/2
     nrg(N) = energy01*calcy(n)*calcy(n)/(ydata(1)*ydata(1))
c    nrg(N) = energy01*calcy(n)*calcy(n)/(ydata(n)*ydata(n))
     nrgbin(nint(nrg(N)/binsize)) =
     @         nrgbin(nint(nrg(N)/binsize)) + 1      ! bin 'em up!
     nrgsum = nrgsum + nrg(N)
c also bin into 3-D nrg array
c points are binned twice; once by the top particle position, and
once
c   by the bottom particle position
     if ( (nrg(N).gt.nrgmin).and.(nrg(N).lt.nrgmax)
     @                .and.(diagnostics.gt.0) ) then
       topbottom = 0 ! 0 for top x,y bin; 1 for bottom x,y bin
700          if (topbottom.eq.0) then
         xbin = nint( xu1(N) )
         ybin = nint( yu1(N) )
         raw2d(xbin,ybin)=raw2d(xbin,ybin)
     @    + nint( (wedgeu(N)+stripu(N)+zu(N))/4.0 )
       else
         xbin = nint( xl1(N) )
         ybin = nint( yl1(N) )
         raw2d(xbin,ybin)=raw2d(xbin,ybin)
     @    + nint( (wedgel(N)+stripl(N)+zl(N))/4.0 )
       end if
       nrgbin3d = nint( (nrg(N) - nrgmin) / binsize )
       nrg3d(xbin,ybin,nrgbin3d) = nrg3d(xbin,ybin,nrgbin3d) + 1
       counts(xbin,ybin) = counts(xbin,ybin) + 1
       nrg2d(xbin,ybin) = nrg2d(xbin,ybin) + nrg(N)
       recoilxy(xbin,ybin) = recoilxy(xbin,ybin)
     @    + sqrt( (xu1(N)-xl1(N)) * (xu1(N)-xl1(N))
     @   + (yu1(N)-yl1(N)) * (yu1(N)-yl1(N)) )
       z2d(xbin,ybin) = z2d(xbin,ybin) + abs( z(N) )
       if (z(N).gt.0) then
         nzpos(xbin,ybin)=nzpos(xbin,ybin)+1
         zposmean(xbin,ybin)=zposmean(xbin,ybin)+z(N)
       else
         nzneg(xbin,ybin)=nzneg(xbin,ybin)+1
         znegmean(xbin,ybin)=znegmean(xbin,ybin)+z(N)
       end if
       if (topbottom.eq.1) goto 800
       topbottom = 1
       goto 700
c new arrays in z coordinate
800          zbin = nint( z(N) )
       nrgz1d(zbin) = nrgz1d(zbin) + nrg(N)
       countsz1d(zbin) = countsz1d(zbin) + 1
       nrgz2d(zbin,nrgbin3d) = nrgz2d(zbin,nrgbin3d) + 1
     end if
```

```
      end do
      nrgmean = nrgsum / (number/2)
c
c Write out binned results to nrg.dat
c
      if (diagnostics.lt.1) goto 999
      open(20,file='nrg.dat')
      do N=nrgmin/binsize,nrgmax/binsize
        write(20,*) N*binsize/1000., nrgbin(N)
      end do
      close(20)
      open(20,file='nrgfull.dat')
      do N=1,4000/binsize
        write(20,*) N*binsize/1000., nrgbin(N)
      end do
      close(20)
      write(6,*)'Wrote nrg.dat and nrgfull.dat'
c
c Write out nrg3d.dat array
c Normalize the nrg2d, raw2d, z2d, zposmean, znegmean, and recoilxy
arrays
c Calculate deltaz(xbin,ybin) array
c
      if (diagnostics.lt.2) goto 999
      open(21,file='nrg3d.dat')
      do xbin = -20 , 20
        do ybin = -20 , 20
          if (counts(xbin,ybin).gt.0) then
            nrg2d(xbin,ybin) = nrg2d(xbin,ybin)/counts(xbin,ybin)
            raw2d(xbin,ybin) =
nint(1.*raw2d(xbin,ybin)/counts(xbin,ybin))
            recoilxy(xbin,ybin) =
     @                recoilxy(xbin,ybin)/counts(xbin,ybin)
            z2d(xbin,ybin) = z2d(xbin,ybin)/counts(xbin,ybin)
            if ((nzpos(xbin,ybin).gt.0).and.(nzneg(xbin,ybin).gt.0))
then
            zposmean(xbin,ybin)=zposmean(xbin,ybin)/nzpos(xbin,ybin)
            znegmean(xbin,ybin)=znegmean(xbin,ybin)/nzneg(xbin,ybin)

      deltaz(xbin,ybin)=(zposmean(xbin,ybin)+znegmean(xbin,ybin))/2
            else
              deltaz(xbin,ybin)=0
            end if
            do N = 0,nbins
              if ( nrg3d(xbin,ybin,N).gt.0 )
     @       write(21,*) xbin,ybin,nrgmin+N*binsize,nrg3d(xbin,ybin,N)
            end do       ! N loop
          end if  ! counts>0 conditional
        end do  ! ybin loop
      end do  ! xbin loop
      close(21)
      call smooth(deltaz,deltazsmooth)
      call smooth(deltazsmooth,deltaz)
c
c Write out nrgz2d.dat array
c Normalize the nrgz1d array
c
      open(20,file='nrgz2d.dat')
      do zbin = -40 , 40
        if (countsz1d(zbin).gt.0) then
          nrgz1d(zbin) = nrgz1d(zbin)/countsz1d(zbin)
          do N = 0,nbins
            if ( nrgz2d(zbin,N).gt.0 )
     @            write(20,*) zbin,nrgmin+N*binsize,nrgz2d(zbin,N)
```

```
        end do  ! N loop
      end if  ! countsz1d>0 conditional
    end do  ! zbin loop
    close(20)
    write(6,*) 'Wrote nrg3d.dat and nrgz2d.dat'
c
c Find standard deviation of raw energy distribution
c
c Also find raw theta distribution P(theta)
c
c Also, the nrg2d array is used to calculate new x,y positions
c for each coincidence event.  The resulting array of
xu,yu,xl,yl,time
c events, cmlwsa-ready, will be written to xyzcorr.dat
c
c Also, use deltaz(xbin,ybin) correction to calculate nrgcorrz.dat
c
c nrgcorr.dat is derived from shifting each events energy by the
difference
c   between the overall mean energy and that bin's mean energy
c
    nrgsum = 0
    if (diagnostics.ge.3) open(30,file='xyzcorr.dat')
    do N = 1,NUMBER/2
      energyON = energyO1*ydata(N)*ydata(N) / (ydata(1)*ydata(1))
      nrgsum = nrgsum + (nrg(N) - nrgmean) * (nrg(N) - nrgmean)
      nrgsumfit = nrgsumfit + (nrg(N) - energyON) * (nrg(N) -
energyON)
      deltax = xu1(N) - xl1(N)
      deltay = yu1(N) - yl1(N)
      R = sqrt(deltax*deltax + deltay*deltay)
      itheta = nint( (180./3.1415926) * atan(R/z(N)) )
      Ptheta(itheta) = Ptheta(itheta) + 1
      deltaE = energyON - 0.5*( nrg2d( nint(xu1(N)),nint(yu1(N)) )
      @                 + nrg2d( nint(xl1(N)),nint(yl1(N)) ) )
      deltaR = sqrt( calcy(N)*calcy(N)*(1+deltaE/nrg(N))-z(N)*z(N)) -
R
      xu2 = xu1(N) + 0.5*deltaR*deltax/R
      xl2 = xl1(N) - 0.5*deltaR*deltax/R
      yu2 = yu1(N) + 0.5*deltaR*deltay/R
      yl2 = yl1(N) - 0.5*deltaR*deltay/R
      deltaR2d(xu1(N),yu1(N)) = deltaR2d(xu1(N),yu1(N)) + deltaR
      deltaR2d(xl1(N),yl1(N)) = deltaR2d(xl1(N),yl1(N)) + deltaR
      rprime = sqrt( (xu2-xl2)*(xu2-xl2) + (yu2-yl2)*(yu2-yl2) )
      if (diagnostics.ge.3) write(30,5000) xu2,yu2,xl2,yl2,time(N)
      nrg2 = energyON*(rprime*rprime+z(N)*z(N)) / (ydata(N)*ydata(N))
      if ( (nrg2.gt.nrgmin).and.(nrg2.lt.nrgmax) ) then
        bin = nint( (nrg2 - nrgmin)/binsize )
        nrgcorr(bin) = nrgcorr(bin) + 1
      end if
c Finally, use deltaz(xbin,ybin to calculate nrgcorrz array
      z2 = z(N) - deltaz( nint(xu1(N)), nint(yu1(N)) )
      nrg2 = energyON*(r*r+z2*z2) / (ydata(N)*ydata(N))
      if ( (nrg2.gt.nrgmin).and.(nrg2.lt.nrgmax) ) then
        bin = nint( (nrg2 - nrgmin)/binsize )
        nrgcorrz(bin) = nrgcorrz(bin) + 1
      end if
    enddo
    if (diagnostics.ge.3) then
      close(30)
      write(6,*) 'Wrote xyzcorr.dat'
    endif
    nrgstddev = sqrt(nrgsum / (number/2) )
    nrgstddevfit = sqrt( nrgsumfit / (number/2) )
```

```
c
c Find peak energy bin
c
   peakbin = 1
   do N=2,4000/binsize
     if (nrgbin(N).gt.nrgbin(peakbin)) peakbin = N
   end do
c Find half-height bins
   N = 0
   leftbin = 0
   rightbin = 0
100      N = N + 1
   if (nrgbin(N).gt.(nrgbin(peakbin)/2)) leftbin = N
   if (leftbin.eq.0) goto 100
   N = 4000/binsize
200      N = N - 1
   if (nrgbin(N).gt.(nrgbin(peakbin)/2)) rightbin = N + 1
   if (rightbin.eq.0) goto 200
300      write(7,*) 'Energy analysis:'
   write(7,*) 'Mean energy: ',nrgmean,' meV'
   write(7,*) 'Std. dev. (wrt nrg mean): ',nrgstddev,' meV'
   write(7,*) 'Std. dev. (of the fit):   ',nrgstddevfit,' meV'
   write(7,*) 'Peak:        ',peakbin*binsize,' meV'
   write(7,*) 'FWHM:        ',(rightbin-leftbin-1)*binsize,' meV'

c
c Normalize deltaR2d array
c
   do xbin = -20 , 20
     do ybin = -20 , 20
       if (counts(xbin,ybin).gt.0) then
         deltaR2d(xbin,ybin) = deltaR2d(xbin,ybin)/counts(xbin,ybin)
       end if  ! counts>0 conditional
     end do  ! ybin loop
   end do  ! xbin loop
c
   open(22,file='deltar2d.dat')
   open(23,file='raw2d.dat')
   open(24,file='z2d.dat')
   open(25,file='recoil.dat')
   open(26,file='deltaz.dat')
   do xbin = -20 , 20
     do ybin = -20 , 20
       if ( counts(xbin,ybin).gt.0 ) then
         write(23,*) xbin,ybin,raw2d(xbin,ybin)
         write(22,*) xbin,ybin,deltaR2d(xbin,ybin)
         write(24,*) xbin,ybin,z2d(xbin,ybin)
         write(25,*) xbin,ybin,recoilxy(xbin,ybin)
         write(26,*) xbin,ybin,deltaz(xbin,ybin)
       end if
     end do
   end do
   close(22)
   close(23)
   close(24)
   close(25)
   close(26)
   write(6,*)'Wrote raw2d.dat, z2d.dat, recoil.dat, deltar2d.dat'
c
c  Write out nrgcorr.dat, nrgcorrz.dat
c
   open(30,file='nrgcorr.dat')
   open(31,file='nrgcorrz.dat')
   do N = 1,nbins
```

```
      if (nrgcorr(N).gt.0) then
    write(30,*)
 (nrgmin+binsize*N)/1000.,nrgcorr(N),sqrt(1.*nrgcorr(N))
      else
    write(30,*) (nrgmin+binsize*N)/1000.,nrgcorr(N),1
      end if
      if (nrgcorrz(N).gt.0) then
    write(31,*)
 (nrgmin+binsize*N)/1000.,nrgcorrz(N),sqrt(1.*nrgcorrz(N))
      else
    write(31,*) (nrgmin+binsize*N)/1000.,nrgcorrz(N),1
      end if
    end do
    close(30)
    close(31)
c
c   Write out P(theta) array
c
    open(31,file='ptheta.dat')
    do itheta = 1,89
      theta = itheta*3.1415926/180.
      write(31,*) itheta, Ptheta(itheta)/sin(theta),
      @          sqrt(Ptheta(itheta))/sin(theta)
    end do
c note that the 90 degree point is split evenly between +z and -z;
these
c two must be summed to form a full one degree angle bin, equally
weighted
c with the other angle bins
    write(31,*) 90,Ptheta(90)+Ptheta(-90),sqrt(Ptheta(90)+Ptheta(-90))
    do itheta = -89,-1
      theta = itheta*3.1415926/180.
      write(31,*) 180+itheta, Ptheta(itheta)/sin(-theta),
      @          sqrt(Ptheta(itheta))/sin(-theta)
    end do
    close(31)
    write(6,*) 'Wrote nrgcorr.dat and ptheta.dat'
c
999       RETURN
5000      format(5(F9.4))
    end
c
c
c *** Subroutine smooth ***
c smooths 2D array
c smoothed array element = 4/9 orig value + 3/9 (av. of nearest
neighbors)
c                                + 2/9 (av. of diagonal neighbors)
c somewhat edited after lifting it from nrgplot.f
c
          subroutine smooth(ra,smoothra)
          implicit none
c
          integer nx,ny
c
          parameter ( nx = 41 )
          parameter ( ny = 41 )
c
          common /smoothblock/ counts(nx,ny)
c
          real*4 counts,ra(nx,ny),smoothra(nx,ny)
          integer xbin,ybin,near,diag,threshold
c
          threshold = 5
```

```
        do xbin = 2, nx-1
          do ybin = 2, ny-1
            if ( counts(xbin,ybin).ne.0 ) then
c
c The smooth correctly handles the edges of the data sets
c
              near = 0  ! number of nearest neighbors
              if (counts(xbin+1,ybin).ge.threshold) near = near + 1
              if (counts(xbin-1,ybin).ge.threshold) near = near + 1
              if (counts(xbin,ybin+1).ge.threshold) near = near + 1
              if (counts(xbin,ybin-1).ge.threshold) near = near + 1
              diag = 0  ! number of diagonal neighbors
              if (counts(xbin+1,ybin+1).ge.threshold) diag = diag + 1
              if (counts(xbin+1,ybin-1).ge.threshold) diag = diag + 1
              if (counts(xbin-1,ybin+1).ge.threshold) diag = diag + 1
              if (counts(xbin-1,ybin-1).ge.threshold) diag = diag + 1
              smoothra(xbin,ybin) =
     @  ( (4/9.) + (4-near)/12. + (4-diag)/18. )*ra(xbin,ybin)
     @  + ( ra(xbin+1,ybin) + ra(xbin,ybin+1)
     @  + ra(xbin-1,ybin) + ra(xbin,ybin-1) )/12.
     @  + ( ra(xbin+1,ybin+1) + ra(xbin-1,ybin+1)
     @  + ra(xbin+1,ybin-1) + ra(xbin-1,ybin-1) )/18.
            end if      ! counts conditional
          end do
        end do
c
        return
        end
```

linfit.f

```
*deck linfit
        Subroutine linfit(x,y,n,em,b)
   implicit none
c
c  this subroutine does a linear least squares fit to
c  a set of x,y data
c
        real*4 x(30),y(30)
   real*4 sigxy,sigx,sigy,sigxsq,em,b
   integer n,i
   sigx   = 0.0
        sigxy  = 0e0
        sigy   = 0e0
        sigxsq = 0e0
        do 10 i = 1,n
         sigxy  = sigxy + x(i)*y(i)
         sigx   = sigx  + x(i)
         sigy   = sigy  + y(i)
         sigxsq = sigxsq + x(i)*x(i)
10      continue
c
c  now, using the well-known least squares equations
c  calculate the slope and intercept.
c
       em = (float(n)*sigxy - sigx*sigy)/(float(n)*sigxsq -
     >      sigx*sigx)
       b  = ((sigy*sigxsq) - (sigxy*sigx))/(float(n)*sigxsq -
     >      sigx*sigx)
      return
      end
```