To the Graduate Council:

I am submitting herewith a dissertation written by Jong Yeon Lim entitled "Convoy Electron Emission Produced in Ion-Atom, Ion-Foil, and Grazing Angle Ion-Surface Collisions". I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Physics.

Stuart B. Elston, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Associated Vice Chancellor and
Dean of the Graduate School

# CONVOY ELECTRON EMISSION PRODUCED IN ION-ATOM, ION-FOIL, AND GRAZING ANGLE ION-SURFACE COLLISIONS

A Dissertation
Presented for the
Doctor of Philosophy
Degree

The University of Tennessee, Knoxville

Jong Yeon Lim

May 1996

# ABSTRACT

Doubly differential convoy electron emission peaks observed in experiments measuring the angular and energy distribution of electrons produced in ion-atom collisions and in grazing interactions of fast ions with atomically clean surfaces are presented by means of 2-dimensional non-linear fits to model functions. Experimental results for the multipole content of convoy "cusp" peaks for ion-atom collisions at relatively low energies are combined with previous results from the higher collision energy region to access the quality of various Born approximation approaches to calculation of the cross section. Inclusion of high-order multipoles for ion-solid foil collisions is made to demonstrate the significance of the transition from single- to multi-step collision production mechanisms. For grazing angle ion-surface collisions, results of efforts to develop reasonably simple, yet physically meaningful model functions that permit unbiased comparisons of the extracted convoy peak from the experimental electron distribution are also presented. Experimental results for 0.3 - 0.5 MeV/u carbon ions at grazing angles of 0.1 - 1.0 degree with a Si(100) crystal are compared to those of other collision systems. Acceleration as well as broadening of the convoy electron peak due to the dynamic image potential, or "wake" potential near the surface are observed in triply differential electron emission distributions.

# TABLE OF CONTENTS


I. INTRODUCTION ..... 1
I.1 Experimental Considerations ..... 4
I.2 Historical Remarks ..... 5

II. THEORETICAL ASPECTS ..... 12
II.1 Charge Transfer to the Continuum and the "Cusp" ..... 12
II.2 Binary Encounter Electrons ..... 17
II.3 Born Approximation ..... 21
II.4 Wake Potential ..... 23

III. EXPERIMENTAL APPARATUS AND TECHNIQUE ..... 29
III.1 Description of Experiment ..... 29
III.2 EN Tandem Accelerator ..... 30
III.3 Beam Steering and Collimation ..... 35
III.4 Preparation of UHV System ..... 37
III.5 Monitoring UHV Chamber Baking Cycles ..... 38
III.6 Magnetic Field Effects and Shielding ..... 39
III.7 Surface Characterization and Preparation ..... 44
III.8 Determination of Grazing Angle ..... 51
III.9 Spherical Sector Analyzer ..... 55
III.10 Position Sensitive Detector ..... 61
III.11 Charge State Analyzer and Electron Multiplier ..... 66

IV. DATA ACQUISITION AND ANALYSIS ..... 68
IV.1 Electronics ..... 68
IV.2 Extraction of Multipoles of Convoy Electron Emission in Ion-Gas and Ion-Foil Collisions ..... 74
IV.3 Measurement of Convoy Electron Emission in Ion-Foil and Ion-Surface Collisions ..... 78

V. RESULTS AND DISCUSSION ..... 84
V.1 Introduction ..... 84
V.2 Ion-Atom and Ion-Foil Collisions ..... 86
V.3 Grazing Angle Ion-Surface Collisions ..... 103
V.4 Comparison of All Collision Systems ..... 121

VI. CONCLUSION ..... 128

REFERENCES ..... 131

APPENDICES ..... 137
A. Determination of Multipole Moments ..... 138
B. 2-Dimensional Fitting Program ..... 145

VITA ..... 183

# LIST OF TABLES


Table Page

2.1 Examples of Important Collision Processes ..... 12

3.1 Experimental Parameters for Ion-Atom Collision Studies ..... 31

3.2 Experimental Parameters for Ion-Foil Collision Studies ..... 31

3.3 Experimental Parameters for Grazing Angle Ion-Surface Collision Studies .... 32

3.4 Properties of the Spherical Electrostatic Analyzer ..... 57

3.5 Spherical Sector Analyzer Calibration Parameters ..... 60

5.1 Multipole Moments for $O^{5+}$ + He (Ar) ..... 90

5.2 Multipole Moments for $O^{5+} + He \rightarrow O^{6+} + e^-$ ..... 91

5.3 Multipole Moments for $O^{5+} + Ar \rightarrow O^{6+} + e^-$ ..... 92

5.4 Comparison of $\beta_k$ Values for Convoy Electron Distribution (ELC) ..... 102

# LIST OF FIGURES


Figure Page

1.1 Normalized Spectra for Electron Capture and Loss to Continuum States in the Forward Direction for 2.5 MeV/u $O^{q+}$ and $Si^{q+}$ on Ar ..... 7

1.2 Doubly Differential Cross Section as a Function of $E_e$ and $\theta_e$ for Electron Loss and Capture to the Projectile Continuum States for 1.875 MeV/u $O^{5+}$ and 2.0 MeV/u $C^{6+}$ Incident on Ar ..... 8

2.1 Schematic Diagram Depicting the Motion of m and $M_P$ in the Lab Frame for the Double Scattering Process ..... 14

2.2 Electron Energy Spectra for 1.0 MeV/u $F^{9+}$ + $H_2$ and 0.5 MeV/u $C^{3+}$ + 20 $\mu g/cm^2$ Carbon Foil Observed at Zero Degrees ..... 19

2.3 Induced Surface Wake Potential Created by a Proton, Which Is Moving near an Al-vacuum Interface with Velocity $v = 4$ from Left to Right ..... 27

3.1 Schematic Diagram of the ORNL EN Tandem Van de Graaff Accelerator .... 33

3.2 Experimental Beam Line and the UHV Target Chamber for Ion-foil and Grazing Angle Ion-surface Collisions ..... 36

3.3 Temperature Fluctuation outside the UHV Chamber with Respect to the Elapsed Heating Time ..... 40

3.4 Residual Gas Partial Pressures inside the UHV chamber with Respect to the Elapsed Heating Time ..... 41

3.5 Electron Distribution in Angle and Energy under the Influence of External Magnetic Fields and the Effect of Helmholtz Coil Arrangement for 1.0 MeV/u $C^{3+}$ + 8 $\mu g/cm^2$ Carbon Foil ..... 43

3.6 Electron Distribution in Angle and Energy before and after shielding the Magnetic Field for Grazing Angle Ion-Surface Collisions ..... 45

3.7 Silicon(100) Surface Specification from the Information Given by the Donor .. 47

3.8 Convoy and Binary Peak Shifts in Emission Angle with respect to the Specular Beam before and after Heat Treatment of Si(100) ..... 50

3.9 Determination of Grazing Angle by 3rd Order Polynomial Fit ..... 52

3.10 Dependence of the Surface Tilting Angle on Surface and Faraday Cup Current ..... 54

3.11 Schematic Layout of Electrostatic Spherical Sector Analyzer ..... 56

3.12 Deflection Analyzer Voltage Calibration ..... 59

3.13 Position Sensitive Detector Electronics Layout ..... 62

3.14 Fitted and Geometric Angular Resolution Functions ..... 65

3.15 Illustration of the Charge State Analyzer Consisting of Two Parallel Plates . 67

4.1 Electronics Setup for Ion-Surface Coincidence Experiment ..... 69

4.2 TAC Spectrum for 0.5 MeV/u $C^{2+}$ + 20 μg/cm$^2$ Carbon Foil → $C^{4+}$ + $e^-$ + $e^-$ ... 72

4.3 Coincidence and Noncoincidence Multichannel Scanning Image for 6.15 MeV $O^{5+}$ + Ar → $O^{6+}$ + $e^-$ ..... 73

4.4 Display of High Resolution and Multichannel Scanning Images from the Spherical Sector Electrostatic Analyzer for 0.5 MeV/u $C^{2+}$ + 20 μg/cm$^2$ Carbon Foil ..... 75

4.5 Doubly Differential Distribution of Electrons emitted from 0.3 MeV/u $C^{2+}$ Incident on 20 μg/cm$^2$ Carbon Foil ..... 79

4.6 High Resolution Image and Corresponding Log Scale Contour Plots for 0.3 MeV/u $C^{1+}$ Incident on Si(100) at a grazing Angle of 0.1° with the Preset Cusp Energy of 246 eV ..... 81

5.1 Contour Plots of Electron Emission for ELC Produced in Ion-Atom Collisions . 88

5.2 Result of the Fitting Procedure for 1.41 MeV/u $O^{5+}$ + He → $O^{6+}$ + $e^-$ ..... 89

5.3 $\beta_{o1}$ and $\beta_{02}$ Coefficients for ELC of $O^{5+}$ on Ar (He), $He^+$ on H, and $He^+$ on He as a Function of Scaled Velocity ..... 94

5.4 Fourth-order Anisotropy Coefficient for ELC of $He^+$ on H and $O^{5+}$ on Ar (He) as a Function of Scaled Velocity ..... 96

5.5 Doubly Differential Electron Emission in Angle and Energy for 2.0 MeV/u $C^{5+}$ Incident on 1.8 μg/cm² Carbon Foil ..... 100

5.6 Result of Fitting with Multipole Expansion Function in Angle and Energy for 2.0 MeV/u $C^{5+}$ Incident on 1.8 μg/cm² Carbon Foil ..... 101

5.7 Fitting Result with a Gaussian Function for the Binary Encounter Electron Peak and a Multipole Expansion Function for the Cusp Peak for 0.5 MeV/u $C^{2+}$ Incident on 20 μg/cm² Carbon Foil ..... 104

5.8 Fitting Result without a Gaussian Function for the Binary Encounter Electron Peak and with a Multipole Expansion Function for the Cusp Peak for 0.5 MeV/u $C^{2+}$ Incident on 20 μg/cm² Carbon Foil ..... 105

5.9 Polar Angle Distribution of Cusp Electrons Emitted in Interaction of $C^{3+}$ Ions with a Kinetic Energy of 0.5 MeV/u on a Si(100) Surface ..... 107

5.10 Multichannel Scanning Image for 0.5 MeV/u $C^{3+}$ Incident on Si(100) at a grazing angle of 0.4° ..... 109

5.11 Result of Fitting with a Combination of Lorentzian and Gaussian Model Peak Functions ..... 111

5.12 Two applied Model Functions and the Result of the Fit in Contour Plot for 0.5 MeV/u $C^{3+}$ Incident on Si(100) at a Grazing Angle of 0.4 ° ..... 113

5.13 The Result of the Fit with a Generated and a Residue Plots for a Grazing Angle Ion-surface Collision System ..... 115

5.14 Extracted Convoy and Binary Electron Peaks Using the Fitted Parameters .. 116

5.15 Energy Distribution of Convoy Electrons before and after Removal of Binary Encounter Electrons ..... 118

5.16 Energy Shift, and Energy and Charge State Dependence of Left FWHM/2 for the Convoy Electron Peak without Removing the BEE Peak ..... 119

5.17 Angular Distribution of Electrons from Gas, Foil, and Crystal Targets ..... 122

5.18 Energy Distribution of Electrons from Foil and Crystal Targets ..... 123

5.19 Energy Distribution of Electrons with Varying Emission Angles for 0.5 MeV/u $C^{3+}$ + Si(100): 0.52° ..... 125

5.20 Energy Distribution of Electrons for 0.5 MeV/u $C^{3+}$ on Si(100) Crystal ..... 126

A.1 3 Dimensional Plots of Generated Multipole Distributions Displaying Combinations of All Possible Expansion Terms with the Dettman Cusp Term ($\beta_{00}$=1.0) ..... 139

A.2 Angular and Energy Resolution Functions for the Fitting Routine ..... 171

# I. INTRODUCTION

It is well known that the energies and emission angles of free electrons produced in atomic collisions are representative of the respective ionization mechanisms. The multiply differential cross section for electron ejection, differential in both energy and angle, is very rich in information about the structure and interaction mechanisms of the colliding systems. The fundamental atomic processes of electron capture and loss resulting from fast collisions of ions with gases, thin solid foils, and solid surfaces are investigated in this dissertation. For these processes, the electron energy and angular distribution near the forward direction exhibits a distinct, cusp shaped peak, which is centered at an electron velocity $v_e$ equal to the projectile velocity $V_P$. Electrons emerging from collisions with velocities near the projectile velocity are called convoy electrons (CE).[1] The cusp- shaped peaks resulting from ion-atom and ion-foil collisions involving fast ions are well described in terms of multipole moments of the ejected charge distribution, which permits direct comparison with theory. These CE mechanisms of ion-atom and ion-foil collision systems have been studied extensively over the last two decades, due to the availability of particle accelerators capable of producing highly charged ions with energies typically in the range of a few tens of thousands of electron volts per atomic mass unit (keV/u) to a few million electron volts per atomic mass unit (MeV/u). In the more complicated process of convoy electron emission in ion-surface interactions, however, we are far away from an analytical

electron emission in ion-surface interactions, however, we are far away from an analytical description of the peak.

When a charged particle is moving near a solid surface, it induces a dynamic image potential (DIP) due to polarization of surface electrons.[2,3] After traveling near a solid, a projectile emerges from the surface region accompanied by convoy electrons. As we shall see, CE may be accelerated by the dynamic image potential. The observed cusp peak in the electron spectrum is significantly broadened in energy as well as in angle in comparison to that observed in ion-atom and ion-foil interactions.

Scattering processes in grazing angle ion-surface collisions, which occur above or near the surface, have been a focus of recent experimental and theoretical studies. These investigations generally place a limit on the energy of the incident particles so that the normal component $E_\perp$ is less than ~100 eV, in order to reduce contributions from the bulk due to surface-penetrating trajectories.

In this work, measurements of the emission of convoy electrons from ion-gas, ion-foil, and grazing ion-surface collisions are presented. The observed cusp shapes in ion-gas and ion-foil experiments all seem to be characterized by a well-known multipole expansion formula. However, spectra in grazing angle ion-surface collisions are not well defined at the current state of the combined experimental and theoretical development. Since the cusp shape is known to be a sensitive measure of the collision dynamics, the deviation of the shape for grazing incidence collisions is expected to be indicative of specific ion-surface mechanisms which are absent in similar ion-atom collisions.

The main focus in this work is the comparison of the peak shape of convoy electron emission produced in grazing incidence ion-surface collisions with the related "cusp" emission produced in ion-atom and ion-foil collisions. First, we will briefly introduce previous investigations and discuss simple theoretical approaches to describing target-electron capture to the projectile continuum (ECC), projectile-electron loss to the projectile continuum (ELC), and a surface wake potential. Second, the experimental apparatus and the method of data reduction and analysis will be described. Third, results of an investigation of the ELC process at low and high velocities in ion-atom collisions will be compared with theoretically predicted values in terms of multipole moments of the CE distribution. Last, the characteristic differences between convoy electron emission peak shapes observed in previous measurements and in recent experiments measuring the angular and energy distribution of electrons emitted in grazing interactions of fast ions with atomically clean surfaces will briefly be studied.

In this study all discussion will be restricted to nonrelativistic charge transfer of a light particle between a heavy nucleus and another heavy nucleus which is incident with a high velocity. Since the relative speed, $v$, of the nuclei is large compared to the initial and final orbital speeds of the light particle, but small compared to the speed of light, $(\frac{v}{c})^2$ is negligible compared to unity, so that relativistic effects are not important. Moreover, the nuclear force plays no role in the charge transfer process, since the range of impact parameters is much larger than the range of the nuclear force (about $10^{-13}$ cm).

## I.1 Experimental Considerations

Progress in investigations of electron emission in ion-atom and ion-foil interaction processes has been accomplished experimentally as well as theoretically for several decades. Meanwhile, the improved availability of surface preparation and characterization techniques is responsible for a recent increase in experimental activity leading to an improved understanding of ion-surface interactions.

The main interest in our experiment originates from work reported a few years ago by the group of Kimura[4] and related work by the group of H. Winter in collaboration with J. Burgdörfer[5] and by Koyama *et al.*[6] Stemming from the pioneering work of those groups, which have clearly identified the convoy electron cusp accompanying ions specularly reflected from surfaces, we have initiated a series of experiments to measure electron emission produced in collisions of relatively slow ($\frac{v^2}{2} = \frac{E}{M_P} = 0.3 \sim 0.5$ MeV/u) and moderately heavy ions at grazing angles of incidence, less than ~1.0° with respect to the surface plane, with atomically clean and flat single-crystal surfaces. The silicon (100) surface was chosen because of its well-known crystal structure, easy preparation, and wide availability.

The spectrometers used in the previously mentioned early studies could only produce results integral in electron emission direction and differential solely in emission energy, or required physical rotation to resolve angular emission distributions. The measurements reported here use an emission angle-resolving, spherical sector electrostatic electron spectrometer equipped with a two dimensional position sensitive microchannel plate detector, and are fully differential in all three components of emission velocity without

requiring rotation of the apparatus. As a result, the details of emission anisotropies both in and normal to the scattering plane are directly accessible.

The experiments discussed in this dissertation are complex ones requiring complex apparatus and methods, and necessarily represent the cooperative effort of a group of workers who have built upon the earlier work of their predecessors. As such, it is impossible to present the new work which is the focus of this dissertation without mixing in a discussion of that other work. The components of this larger effort, which are unique and new to this dissertation include contributions to the apparatus discussed in chapter III (i.e., including magnetic shielding system for the electron spectrometer, the ultra-high vacuum chamber baking control system, and the ion detector system of the exit projectile charge state analyzer), to the analysis of results in the form of the multi-dimensional non-linear least-squares fitting routine which is described in chapter IV and presented in appendices, and to the interpretation of results as detailed in chapter V.

## I.2 Historical Remarks

Since experimental and theoretical evidence of the electron transfer to continuum mechanism was first reported in 1970,[7,8,9] considerable progress in developing both experimental and theoretical techniques has been made.

Following the pioneering efforts of Crooks, Rudd, and Macek in 1970, Dettman *et al.*[10] established a simple analytical form for the velocity spectrum of ejected electrons, collected into a cone centered about the projectile direction, using a first Born approximation. The

validity of their calculation is limited to projectile velocities much greater than the initial velocity of the electron to be transferred. Often this regime is termed the asymptotically high velocity region. Experimentally, an asymmetry of the ECC cusp had been noticed in the spectrum, corresponding to an enhanced probability for emission velocity lower than the projectile velocity in the laboratory frame.[11] In 1978 Shakeshaft and Spruch suggested that the inclusion of second order terms in a Born series expansion would explain this asymmetry.[12] Supporting experimental investigations[13,14,15] show that the inclusion of higher order Born terms is important in describing ECC.

While an ECC theory for the asymptotic velocity regime must include the second order Born approximation, ELC can be described within the first order Born approximation.[16,17] The first order Born approximation describes a symmetric peak, and high-order perturbative terms provide an asymmetry, which can enhance the probability for electron emission velocities less than the projectile velocity as compared with velocities higher than the projectile velocity. Figure 1.1 and 1.2 are comparisons of ECC and ELC cusps reproduced from M. Breinig *et al.*,[1] and electron distribution spectra, differential in angle and energy, measured in the course of the present study, respectively. A progression to a narrower and more symmetric cusp is seen as the incident projectile brings more electrons into the collision.

A vectorial dependence, $f(\vec{v}_e - \vec{V}_P)$, was developed by Day,[18] Briggs and Day,[19] and Burgdörfer *et al.*[17] to explain anisotropic terms in the emission cross section. In general the cross section can be written[16,20] as

$$\frac{d\sigma}{d\vec{v}} = \frac{f(v_e, V_P, \theta_e)}{|\vec{v}_e - \vec{V}_P|} = \frac{f(v_e, V_P, \theta_e)}{v_{e'}}, \qquad (1.1)$$

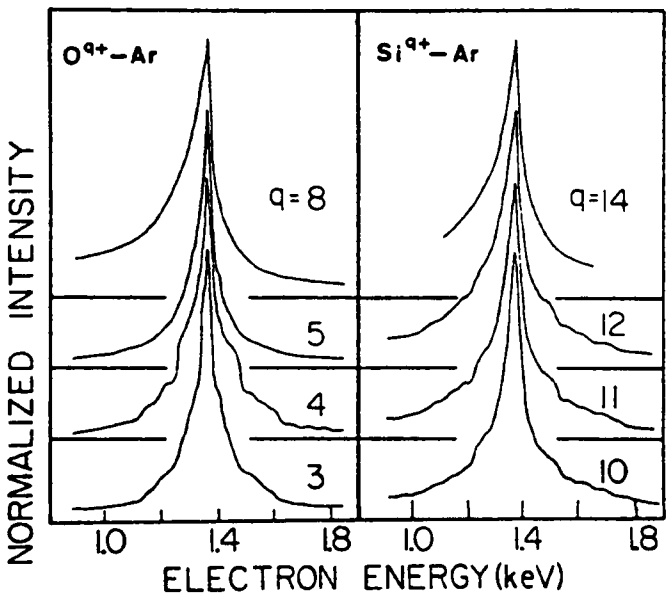
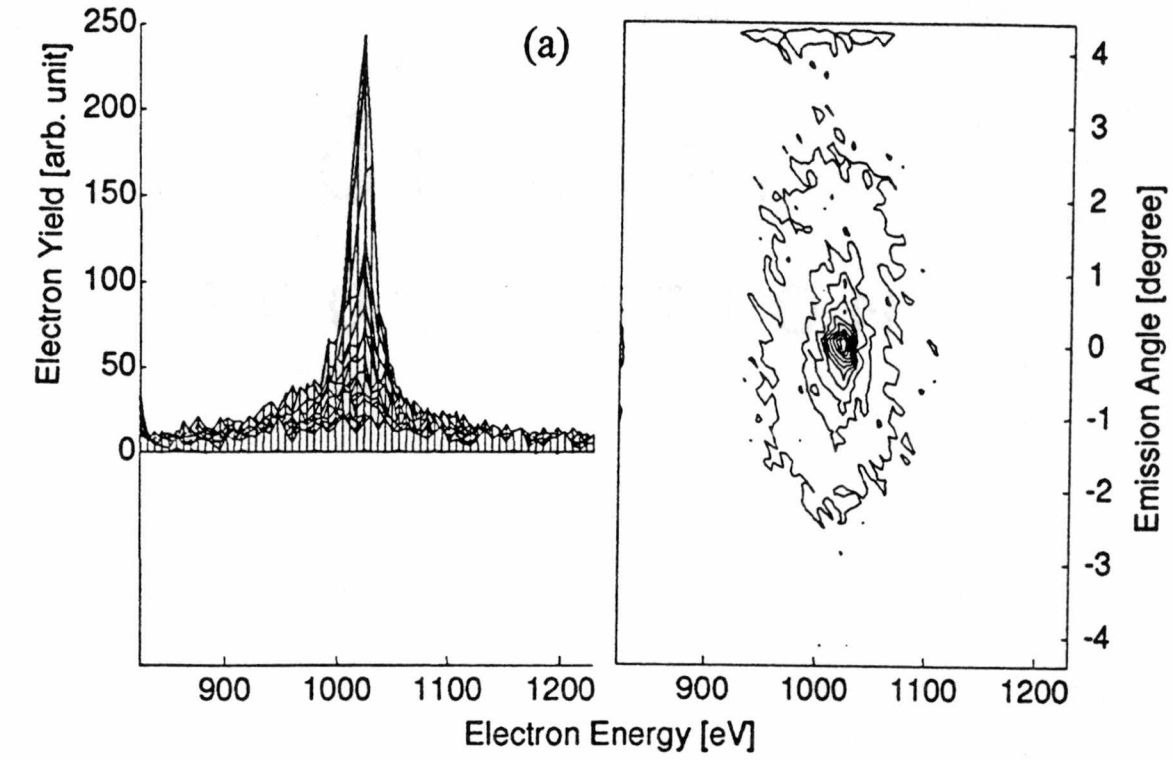
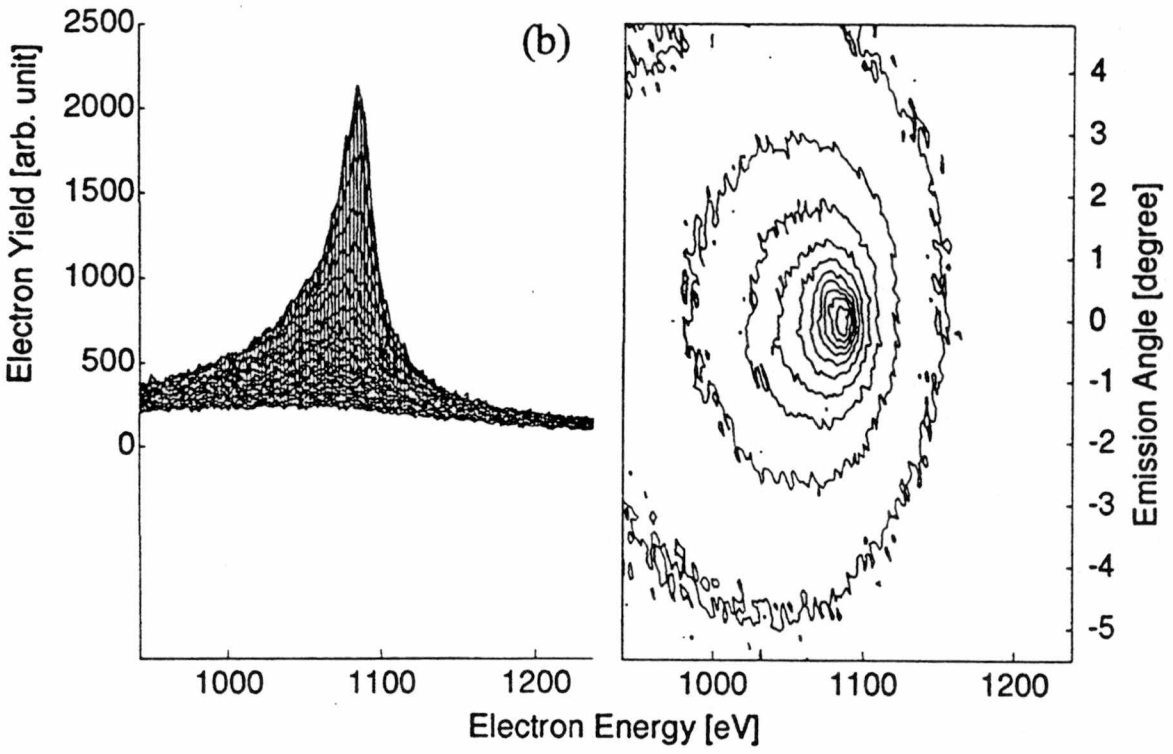



**Figure 1.1**: Normalized spectra for electron capture and loss to continuum states in the forward direction for 2.5 MeV/u $O^{q+}$ and $Si^{q+}$ on Ar. Zero-count base lines are shown for each spectrum. **Source**: M. Breinig, S. B. Elston, S. Huldt, L. Liljeby, C. R. Vane, S. D. Berry, G. A. Glass, M. Schauer, I. A. Sellin, G. D. Alton, S. Datz, S. Overbury, R. Laubert, M. Suter, Phys. Rev. A25, 3015 (1982).

**Figure 1.2**: Doubly differential cross section $\frac{d^2\sigma}{dE_e d\Omega_e}$ as a function of $E_e$ and $\theta_e$ for electron loss and capture to the projectile continuum states for (a) 1.875 MeV/u $O^{5+}$ and (b) 2.0 MeV/u $C^{6+}$ incident on Ar, respectively. The solid angle, $d\Omega_e$, denotes $\sin^2\theta_e d\theta_e d\phi_e$ in the laboratory frame. A strong skewing towards the low energy side, which is characteristic of ECC, is shown in the lower spectrum with 10 % intensity contours.


(a)
Electron Yield [arb. unit]
250
200
150
100
50
0
900
1000
1100
1200
Electron Energy [eV]
4
3
2
1
0
-1
-2
-3
-4
Emission Angle [degree]


(b)
Electron Yield [arb. unit]
2500
2000
1500
1000
500
0
1000
1100
1200
Electron Energy [eV]
4
3
2
1
0
-1
-2
-3
-4
-5
Emission Angle [degree]

where the primed variables are projectile frame variables, the unprimed are laboratory frame variables, and $\theta_e$ denotes the electron emission angle in the laboratory frame. Any asymmetry or anisotropy can be incorporated in the function $f(v_e, V_P, \theta_e)$.

In 1990 Jakubaßa-Amundsen[21] applied second order Born theory to the study of the ELC process in $He^+$ + He collisions at forward emission angles and found that the first Born approximation breaks down at collision velocities below twice the ground state electronic orbiting velocity, and the second order Born approximation becomes unreliable around $V_P \approx Z_P$. Also, experimentally-observed contamination of the electron loss peak by ECC was investigated by the inclusion of higher order Born terms in the Jakubaßa-Amundsen study. The negative skewness parameter, much weaker than in ECC, indicates that the peak is skewed towards the low energy side. It is anticipated that this formalism could be extended to multielectron projectiles and heavier targets[22,23] as were observed as part of this study.

In the last decade there has been increased interest in electron emission produced by grazing angle ion-surface collisions. Even though the problem is complex and involves a large number of electrons, considerable progress in both experimental and theoretical aspects has been made. As mentioned in the previous section, the broadening and acceleration of convoy electron peaks resulting from grazing angle ion-surface collisions have received particular attention, both experimentally and theoretically, from several groups.

In 1986 de Ferrariis and Baragiola reported an early observation of the energy spectrum of electrons produced by $H^+$ ions incident on an Al target.[24] A second early experiment

was performed by the Kyoto group of Hasegawa *et al.*[4] It is now generally accepted that the most probable velocity of the convoy electron peak, produced in grazing angle scattering of fast ions from solid surfaces, is faster than the projectile ion due to acceleration of convoy electrons by the surface wake potential, which is induced by the projectile ion near the surface. Meanwhile, Winter *et al.*[5] measured the energy spectrum of electrons emitted from a Si(111) surface with grazing incidence $H^+$, and observed significantly broadened, cusp-shaped peaks. They attributed this broadening to a deviation from a pure Coulombic final-state interaction produced by the dynamic image potential, or surface wake.

Unlike the situation for single ion-atom collisions involving few-electron systems, the current theory of electron emission in grazing ion-surface scattering appeals to quite simple models; a complete *ab initio* treatment of the complex dynamics of ion-surface collisions is beyond the capability of present theory.[25] A series of theoretical attempts by Iitaka *et al.*[26] and Kimura *et al.*[27] using classical trajectory Monte-Carlo (CTMC) simulations of the distribution of convoy electrons revealed a shifted convoy electron peak for the electron propagating in the fields of the projectile and its image. A more detailed approach to the analysis of convoy electron emission by Reinhold and Burgdörfer,[28] using ensembles of initial coordinates for both target and projectile electrons and employing the CTMC method, have demonstrated the presence of "rainbow scattering" of convoy electrons near the surface, and credit the rainbow scattering process as being responsible for the existence of a shifted and broadened peak.

# II. THEORETICAL ASPECTS

## II.1 Charge Transfer to the Continuum and the Cusp

By "charge transfer", we mean the transfer of an electron from one atomic system to another in the course of a collision. This charge exchange process is one of many possible electronic rearrangements (Table 2.1) that can occur during the collision of two atomic species and is one of the most extensively studied inelastic collision processes. At very high energies when the momentum transfer of the electron becomes important, this process becomes characteristic of a fast collision-induced reaction process.

Table 2.1

EXAMPLES OF IMPORTANT COLLISION PROCESSES[29]

| | |
|---|---|
| $A + B \rightarrow A + B$ | Elastic Collision |
| $A + B \rightarrow$ All Possible Results | Total Cross Section |
| $A + B \rightarrow A + B^*$ | Excitation |
| $A^* + B \rightarrow A + B$ | De-excitation |
| $A + B \rightarrow A + B^+ + e^-$ | Ionization |
| $A^+ + B \rightarrow A + B^+$ | Charge Transfer |
| $A + B^* \rightarrow A^* + B$ | Excitation Transfer |
| $A + BC \rightarrow AB + C$ | Rearrangement |
| $A + B + C \rightarrow AB + C$ | Reaction |

* implies the species is not in the lowest state.

$A^+$, $B^+$; bare nuclei for hydrogenic system.

For atomic collision processes it is very important to be specific in referring to "slow" or "fast" atom-atom collisions. A collision is "slow" or "fast" depending on whether the relative velocity of the colliding particle is much less than or much greater than the velocity of the active electrons bound in the colliding atoms. The "cusp" peak, which is normally observed in the angle and velocity distribution of electrons emitted in "fast" ion-atom collisions, is exhibited when the emergent electron velocity $\mathbf{v}_e$ matches that of the outgoing ion velocity $\mathbf{V}_P$ in both speed and direction.

In 1927, Thomas[30] gave the first classical treatment of the capture of a bound electron by a high velocity projectile when the initial orbital velocity of the electron is much less than the velocity of the projectile. He supposed that, for capture to take place, $m$ must acquire a velocity very close to the velocity of the incident nucleus $M_P$. In this process the bound electron is first scattered off the projectile at a laboratory angle of $\pi/3$, and then elastically off the target nucleus to redirect this velocity vector in the direction of the projectile without change of speed, as seen in Figure 2.1. The electron then emerges with a velocity very close to $V_P$. In a classical two body collision treatment, the projectile is scattered through the Thomas angle, $\theta = \sqrt{3} \cdot m/(2M_P)$, which for electron capture by a proton is about 1.6°. In the quantum mechanical treatment, the corresponding process is revealed by a peak in the differential projectile scattering cross section at this angle.

In the study of the light particle $m$, much of the description can be classical. Since the wavelength of $m$ between the two collisions will be of order $\hbar/mV_P$, which is very much smaller than the separation of the nuclei for almost all relevant values of the impact parameter $b$, the motion of $m$ between collisions is essentially classical. Thus each

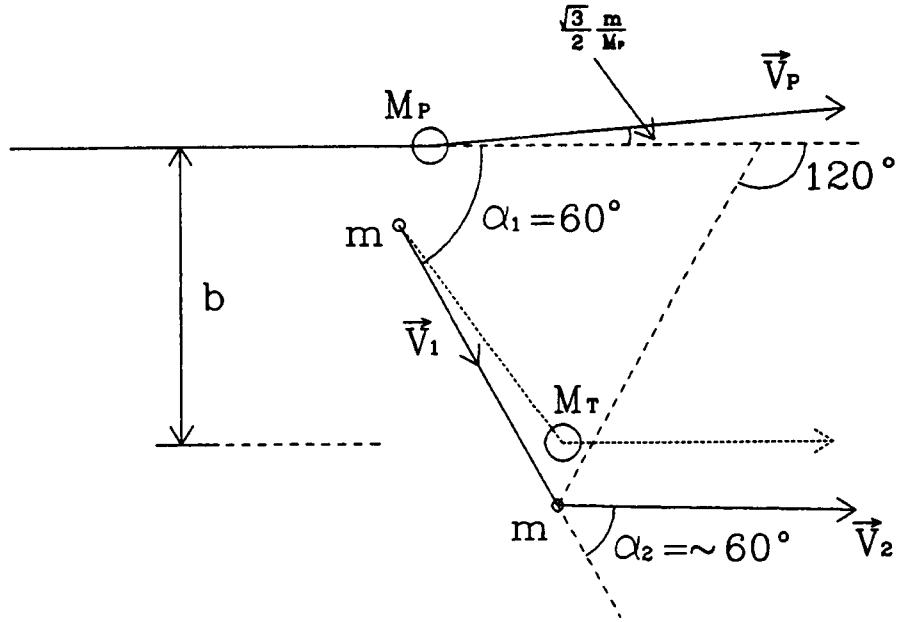



**Figure 2.1**: Schematic diagram depicting the motion of $m$ and $M_P$ in the lab frame for the Thomas double scattering process.

collision is truly a binary one and interference effects between the two collisions are negligible.

As explained in a classical treatment, we expect that if the relative velocity of the emergent electron and the incident nucleus is sufficiently small, the ionization of a target electron may proceed via a mechanism analogous to the double scattering mechanism for true bound-state capture. That is, "electron capture to the continuum (ECC)" happens if the emergent electron is moving relative to the projectile nucleus with a (positive) energy that is small compared to the ionization energy of the electron-projectile subsystem. This mechanism is a high momentum transfer process involving an electron in a bound state of the target atom being captured to a projectile continuum state.

If partially ionized projectiles (i.e. projectiles carrying initially bound electrons) undergo ion-atom collisions, the laboratory frame velocity distribution of the forward ejected electrons also displays a sharp cusp in both energy and angle centered near $\mathbf{v}_e = \mathbf{V}_P$. Whenever loosely bound projectile electrons are available, these electrons can be lost to low lying projectile-centered continuum states. This mechanism of electron loss to the continuum (ELC) is a low momentum transfer process involving an electron in a bound state of the projectile being excited into a continuum state of the projectile.

For an analytic form of the electron distribution in the forward direction, charge transfer to the continuum may be considered to be the dominant process in the production of the convoy peak. Thus, to develop the cross section, the continuum Coulomb wave function, describing the scattering of the electrons, could be introduced in the form of

$$\Psi_{\vec{k}}(\vec{r}) = (2\pi)^{-3/2} \cdot e^{\eta\pi/2} \cdot \Gamma(1+i\eta)^{i\vec{k}\cdot\vec{r}} \cdot {}_1F_1[-i\eta, 1, -i(kr + \vec{k}\cdot\vec{r})] \qquad (2.1)$$

$$\text{where } k = |\vec{k}|,\ \vec{k} = (m/\hbar) \cdot (\vec{v}_e - \vec{v}), \vec{v} = \mathbf{V}_P, \text{ and } \eta = Z_P/(a_0 k).$$

Dettman *et al.*[10] made the approximation $\eta >> 1$, or

$$e^{\eta\pi} \cdot |\Gamma(1+i\eta)|^2 = 2\pi\eta/(1 - e^{-2\pi\eta}) \approx 2\pi\eta. \qquad (2.2)$$

In essence, a cusp arises from the normalization factor $2\pi\eta$ in the squared modulus of the Coulomb wave function. The cusp in the differential cross section for charge transfer to the continuum was predicted using an approximation analogous to the Brinkman- Kramers (BK) approximation for true capture to a final bound state. Integrating over a collection cone of half angle $\theta_0$, they obtained the result

$$\frac{d\sigma_{BK}}{dv_e}(1s \rightarrow cont.) \sim \frac{2^{17}}{5} Z_P^3 Z_T^5 \left(\frac{e^2}{\hbar v}\right)^{10} \left(\frac{1}{v^2}\right) \{[(v_e - v)^2 + (v\theta_0)^2]^{1/2} - |v_e - v|\}(\pi a_0^2) \qquad (2.3)$$

From this equation, we note that the velocity dependence of the cross section is $\frac{1}{v^{12}}$, and therefore at very high velocities, the double scattering mechanism for charge transfer to the continuum, for which $\frac{d\sigma}{dv_e} \sim \frac{1}{v^{10}}$ as suggested by Shakeshaft and Spruch[12] to explain an asymmetry in the velocity spectrum, is important.

For the purpose of ECC and ELC cusp shape analysis, it is generally accepted that the previously mentioned form (Section I.2) of the cross section, $\frac{d\sigma}{d\vec{v}} \propto \frac{f(v_e, V_P, \theta_e)}{|v_e - V_P|}$, describes the emission of secondary electrons ejected in the forward direction with velocities near that of incident ion. Obviously the denominator, which is symmetric about $\mathbf{v}_e = \mathbf{V}_P$, is responsible for the gross, overall cusp shape and results from the Coulomb interaction between the outgoing projectile ion and the ejected electron. In order to characterize observed asymmetries, it has been found desirable to expand $f(v_e, V_P, \theta_e)$ in projectile frame multipoles[17,20,31] in the form of

$$\frac{d\sigma}{d\vec{v}_{e'}} = \frac{\sigma_0}{v_{e'}} \cdot \sum_k \beta_k(V_P) \cdot P_k(\cos\vartheta_{e'}). \tag{2.4}$$

where $v'_e = |\mathbf{v}_e - \mathbf{V}_P|$ in the projectile frame, $P_k(\cos\vartheta_{e'})$ are Legendre polynomials, and the expansion coefficients $\beta_k(V_P)$ are considered to be quantifiers that are ultimately derivable from theory (also experimentally evaluable). Details of this multipole expansion for parameterization of the cusp will be discussed in a later chapter.

## II.2 Binary Encounter Electrons

Binary encounter electrons (BEE)[32,33] are characterized by ionized target electrons, produced by Coulomb interaction with the incident (screened) nucleus, through direct and hard collisions with the projectile; i.e. loosely bound target electrons residing in outer shell collide elastically with the projectile nucleus and are ionized into the continuum, giving rise to a broad energy distribution. Since this process is analogous to elastic scattering of a free electron from a projectile ion, and the impact parameter between the target electron and the impinging ion is sufficiently small, the two body interaction between projectile and electron leading to BEE emission can be described as a Rutherford like scattering process, from which the cross section for the emission of BEE can be easily derived.[34]

Using energy and momentum conservation from classical two-body collision dynamics for heavy ion impact on a free electron, the energy of the recoiling electron or the BEE peak energy is given by

$$E_{BEE} = 4\frac{m}{M_P}E_P \cos^2\theta_{lab} \tag{2.5}$$

where $m$ and $M_P$ are the masses of the electron and the projectile, respectively, $E_P$ is the projectile energy, and $\theta_{lab}$ is the electron emission angle in lab frame. In comparison, the cusp electron energy is simply given by $\frac{m}{M_P}E_P$. For the forward electron emission measurement ($\theta_{lab} \approx 0°$), the expected BEE energy should be at 4 times the cusp electron energy, $4\frac{m}{M_P}E_P$.

When the quasi-free target electron or the binding energy of the electron in the initial target state is taken into account, the energy of the BEE is shifted to lower emission energies, i.e. smaller than $4\frac{m}{M_P}E_P$. Recently D. H. Lee *et al.*[35] developed an impulse-

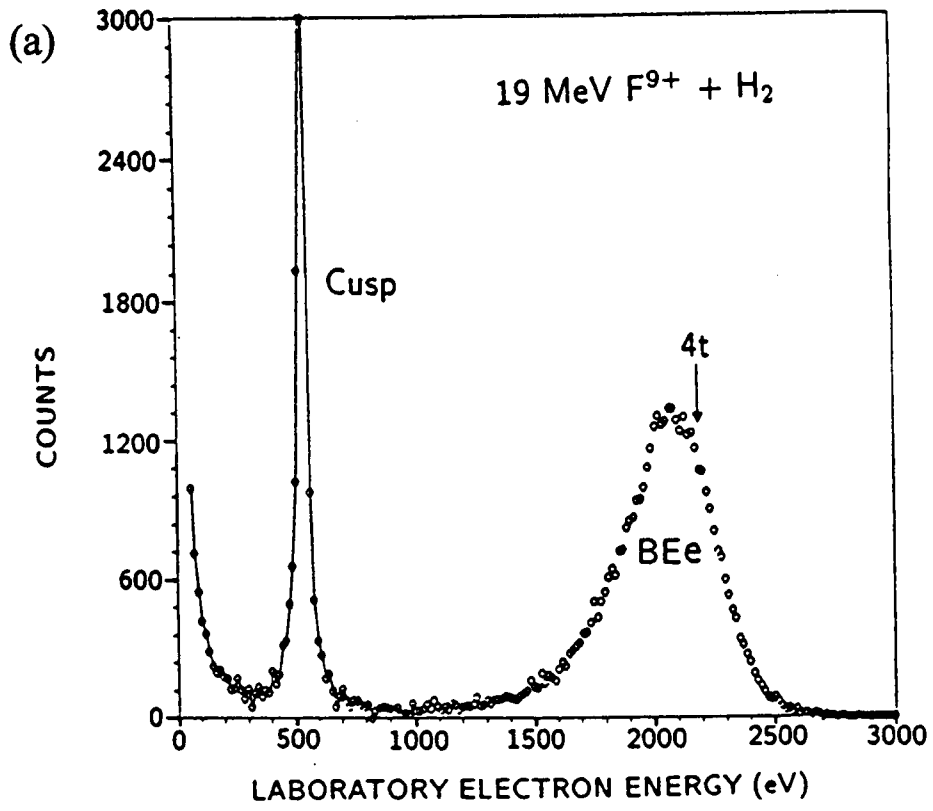
approximation (IA) model to describe this effect in which *quasi*-free target electrons undergo 180° Rutherford scattering in the projectile frame. The calculation of the BEE cross section is performed in the projectile frame within the impulse approximation treatment[36] as follows

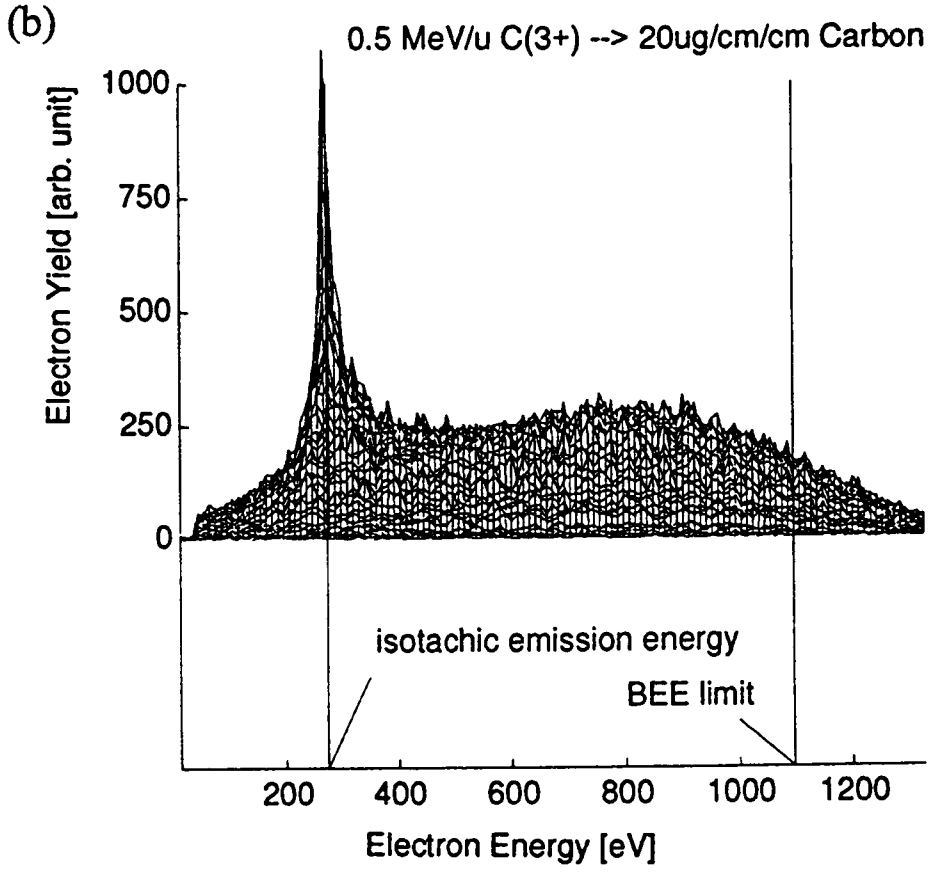
$$\left[\frac{d\sigma}{d\Omega}\right]_{BEE} = \sum_i \int \left[\frac{Z_P^2 e^4}{16E^2 \sin^4(\theta/2)}\right] |\psi_i(\mathbf{p}_i)|^2 d^3 p_i \qquad (2.6)$$

where $E$ is the electron energy in the projectile frame, $Z_P e$ is the projectile nuclear charge for the case of a bare projectile, and $\psi_i(\mathbf{p}_i)$ is the target-electron momentum wave function; the subscript $i$ refers to the $i$th target electron. The Rutherford scattering cross section of a free electron by a bare projectile ion is shown within the large parentheses on the right hand side. Integrating the 180° Rutherford scattering cross section over the incoming electron's initial momentum distribution around the target nucleus (Compton profile) and accounting for the binding energy of the target electron, they concluded that a description of the BEE within an IA treatment agrees well with measured BEE DDCS results for both the shifted position and the broadened peak shape.[35]

Figure 2.2 shows representative electron spectra for collisions of 1.0 MeV/u $F^{9+}$ incident on $H_2$ and 0.5 MeV/u $C^{3+}$ incident on a 20 $\mu g/cm^2$ carbon foil. In both cases the BEE peaks have very broad structures at the higher energy side of the spectra, and the shifted BEE positions to the lower energy side are clearly visible. For the case of 1.0 MeV/u $F^{9+}$ projectiles at relatively high collision energy, the BEE peak is well separated from the cusp.

**Figure 2.2**: Electron energy spectra for 1.0 MeV/u $F^{9+}$ + $H_2$ and 0.5 MeV/u $C^{3+}$ + 20 $\mu g/cm^2$ carbon foil observed at zero degrees. (a): The cusp, not shown at its full height, at energy t is about 5.5 times higher than the binary encounter peak. The maximum of the binary encounter peak appears at an energy slightly lower than 4t (arrow). **Source**: D. H. Lee, P. Richard, T. J. M. Zouros, J. M. Sanders, J. L. Shinpaugh, H. Hidmi, Phys. Rev. **A41**, 4816 (1990). (b): The recently measured doubly differential electron emission distribution for 0.5 MeV/u $C^{3+}$ incident on 20 $\mu g/cm^2$ carbon foil shows very different electron emission mechanism from ion-gas collision. The isotachic emission energy and binary encounter electron energy limit are shown at 274.3 and 1097 eV, respectively.


(a)
19 MeV F9+ + H2
Cusp
4t
BEe
COUNTS
LABORATORY ELECTRON ENERGY (eV)
0
600
1200
1800
2400
3000
0
500
1000
1500
2000
2500
3000


(b)
0.5 MeV/u C(3+) --> 20ug/cm/cm Carbon
Electron Yield [arb. unit]
1000
750
500
250
0
isotachic emission energy
BEE limit
200
400
600
800
1000
1200
Electron Energy [eV]

## II.3 Born Approximation

The number of particles $dN$, scattered per unit time into an element $d\Omega$ of a differential solid angle about the direction $(\theta,\phi)$, is determined by the probability of a collision between the incident particles and the target atoms. The connection between the number of target atoms and detected particles is given by the differential scattering cross section, $\frac{d\sigma}{d\Omega}$ or $I(\theta,\phi)$, of a target atom for scattering particle through an angle $(\theta,\phi)$ into a differential solid angle $d\Omega$.

It is well established that the asymptotic form for the outgoing scattered wave function at infinity $(r \to \infty)$ is given by

$$\Psi(\mathbf{r}) \sim e^{i\mathbf{k}\cdot\mathbf{r}} + \frac{1}{r}e^{i\mathbf{k}\cdot\mathbf{r}}f(\theta,\phi) \tag{2.7}$$

where $e^{i\mathbf{k}\cdot\mathbf{r}}$ is an incident plane wave and the function $f(\theta,\phi)$ is the scattering amplitude. In turn the differential cross section $I(\theta,\phi)$ is defined as

$$I(\theta,\phi) = |f(\theta,\phi)|^2 \tag{2.8}$$

and the total cross section is

$$\mathbf{Q} = \int I(\theta,\phi)d\Omega. \tag{2.9}$$

Introducing the free particle Green's function[37] to satisfy the outgoing wave boundary function, M.R. McDowell and J.P. Coleman derived the asymptotic equation at infinity[38]

$$\Psi^{+}(\mathbf{r}) \sim e^{i\mathbf{k}\cdot\mathbf{r}} - \frac{\mu}{2\pi}\frac{e^{ik\cdot r}}{r}\int e^{-i\mathbf{k}'\cdot\mathbf{r}'}V(\mathbf{r}')\Psi^{+}(\mathbf{r}')d\mathbf{r}'. \tag{2.10}$$

Therefore, when a structureless particle of mass $\mu$ undergoes the influence of a potential $V(\mathbf{r})$, satisfying the condition $\lim_{r\to\infty} r|V(\mathbf{r})| = 0$ (excludes Coulomb potential), and scatters from the scattering center, the expression for the scattering amplitude at infinity becomes

$$f(\theta,\phi) = -\frac{\mu}{2\pi}\int e^{-i\mathbf{k}'\cdot\mathbf{r}'}V(\mathbf{r}')\Psi^{+}(\mathbf{r}')d\mathbf{r}'. \tag{2.11}$$

A formal solution of the integral equation for $\Psi^+$ can be rewritten more concisely as

$$|\Psi^+\rangle = |\psi_0\rangle + G_0^+ V|\Psi^+\rangle. \tag{2.12}$$

The Green's function operators are defined as

$$G_0^+ \equiv \lim_{\varepsilon\to 0} \frac{1}{E-H_0+i\varepsilon}, \text{ and } G^+ \equiv \lim_{\varepsilon\to 0} \frac{1}{E-H+i\varepsilon} \tag{2.13}$$

where $H = H_0 + V$, $H_0$ is the free particle Hamiltonian, and $V$ is a perturbation. With the help of the inverse operator property, the operators $G^+$ and $G_0^+$ are related by the equations

$$G^+ = G_0^+ + G^+ V G_0^+ = G_0^+ + G_0^+ V G^+. \tag{2.14}$$

Finally the required formal solution of the unknown function $\Psi^+$ is written as

$$\Psi^+ = (1 + G^+ V)\psi_0 \tag{2.15}$$

where $\psi_0$ denotes the plane wave. Also the expression for the scattering amplitude can be given by

$$f(\theta,\phi) = -\frac{\mu}{2\pi}\langle\psi_0|V|\psi_0\rangle = -\frac{\mu}{2\pi}\langle\psi_0|T|\psi_0\rangle \tag{2.16}$$

where the transition operator $T$ is defined by the equation $T\psi_0 = V\Psi^+$, followed by

$$T = VG^+V = V + VG_0^+T. \tag{2.17}$$

In general, it is not possible to obtain exact solutions for these equations and therefore, some approximation scheme is required. Series iterations of equations for $G^+$, $\Psi^+$, and $T$ give

$$G^+ = G_0^+ + G_0^+ V G_0^+ + G_0^+ V G_0^+ V G_0^+ + \cdots\cdots \tag{2.18}$$

$$\Psi^+ = \psi_0 + G_0^+ V\psi_0 + G_0^+ V G_0^+ V\psi_0 + \cdots\cdots \tag{2.19}$$

$$T = V + VG_0^+V + VG_0^+VG_0^+V + \cdots\cdots, \tag{2.20}$$

which are commonly referred to as the Born series.

With the manipulation of the transition probability per unit time and the density of final states, the differential and total cross sections can be written

$$I(\theta,\phi) = \frac{\mu_i\mu_f}{4\pi^2}\left(\frac{k_f}{k_i}\right)|T_{if}|^2 \text{ and } \mathbf{Q} = \frac{\mu_i\mu_f}{4\pi^2}\left(\frac{k_f}{k_i}\right)\int|T_{if}|^2 d\Omega \qquad (2.21)$$

where $\mu_i$ and $\mu_f$ are the reduced masses before and after collision. The transition matrix takes the form $T_{if} = \langle\psi_f|V_f|\Psi_i^+\rangle = \langle\psi_f|V_f(1+G^+V_i)|\psi_i\rangle$ where $|\psi_i\rangle$ and $|\psi_f\rangle$ are the initial and final state wave function. The iterative solution to a Born series for $T_{if}$ will be

$$T_{if} = \langle\psi_f|V_f|\psi_i\rangle + \langle\psi_f|V_f G_0^+ V|\psi_i\rangle + \langle\psi_f|V_f G_0^+ V G_0^+ V|\psi_i\rangle + \cdots\cdots \qquad (2.22)$$

The *n*th Born approximation is obtained by retaining the first *n* terms of this expansion.

It has been addressed[38] that if the Born series converges for all energies $E$ greater than some energy $E_0$, i.e. $T_{if}$ tends to $\langle\psi_f|V_f|\psi_i\rangle$ as $E \to \infty$, then cross sections calculated in the first Born approximation are highly accurate for sufficiently large energies and furthermore one can expect to systematically improve on the first Born result at lower energies by taking more terms of the Born series into account.

## II.4 Wake Potential

A charged particle traveling through matter in the vicinity of a solid-vacuum surface induces an oscillatory electric field, called a "wake potential",[39,40] behind it. The emission of secondary electrons induced by the passage of a swift charged particle can be understood in terms of the excitation of individual electrons in the medium.

With the entry of a swift charged ion the medium responds to the presence of the added charge, causing transient readjustment of electrons and changes in the charge state of the

ion. Electrons may be stripped from the ion or captured from valence states or inner shells, resulting in polarization of the medium. Perturbation of electron orbits by the Coulomb field of the moving ions happens by dynamic screening of valence electrons, followed by continuous electron capture and loss into bound states on the ion. Swift ions emerging from a high-density medium also induce surface oscillatory modes, accompanied by electrons moving with velocities distributed near the velocity of the ion.[41,42] This oscillatory "bulk wake potential" as well as a wake electron density fluctuation inside the medium has been studied by many authors.[43,44]

In grazing angle ion-surface collision the "surface wake potential" or "dynamic image potential" by a swift particle moving "near" a surface plays an important role in the dynamics of convoy electron emission. Electrons can be trapped into wake-riding states by the potential minima associated with domains of electron density depletion with binding energies.[45,46] The notation, "near", means within the range of screening of the surface wake potential, that is the dynamic screening length

$$\lambda_D = v_F / w_s^2 \tag{2.23}$$

where $v_F$ denotes the Fermi velocity of the electron gas and $w_s$ represents the classical surface plasma frequency.

The boundary of the surface for the grazing incidence is assumed to lie around the region just above the top-most layer of the solid; the electronic surface or jellium edge, where the slope of electron density probability curve is very steep, is taken outside of the atomic surface by half of the atomic plane separation,[47] where the secondary electrons

originate. The information of spatial and time dependence of the surface wake potential is very important for electron transport and distribution by swift ions incident on solids.

Undispersiveness of many properties of the interface between two media like a solid-vacuum surface gives rise to consideration of the appropriate dielectric response function dependent on frequency. Recently Garcia de Abajo and Echenique[48] derived the wake potential as a function of space and time when the particle moves parallel with and perpendicular to the surface, using a dielectric analysis based upon the specular-reflection model and a Hamiltonian treatment. They first calculated the scalar potential by solving the Poisson equation, taking the dielectric function as

$$\epsilon(\mathbf{r},\varpi) = \epsilon(\varpi)\cdot\theta(-z) + \theta(z), \tag{2.24}$$

where $\theta(z)$ is the Heaviside step function, and applying suitable matching conditions. With the aid of Fourier-transformation two constants for homogeneous solutions in the bulk and in the vacuum were determined by the boundary conditions, which are the continuity of the potential and the electric displacement. The resulting surface wake potential, induced by a projectile ion with a charge $Z_1$, for an external trajectory ($z_0>0$) is symmetric with respect to the surface:

$$\phi_{\parallel}^{ind}(\mathbf{r},t) = \frac{Z_1}{2\pi}\int \frac{d\mathbf{Q}}{Q} e^{i\mathbf{Q}\cdot\check{\mathbf{R}}_e - Q(|z_0|+|z|)\frac{1-\epsilon(\varpi)}{1+\epsilon(\varpi)}}, \tag{2.25}$$

where $\mathbf{r}=(\mathbf{R},z), \mathbf{k}=(\mathbf{Q},k_z), \mathbf{v}=(\mathbf{V}_{\parallel},v_z)$, $\check{\mathbf{R}}=\mathbf{R}-\mathbf{V}_{\parallel}\cdot t$, and $\varpi=\mathbf{Q}\cdot\mathbf{V}_{\parallel}$; $\mathbf{R},\mathbf{Q}$, and $\mathbf{V}_{\parallel}$ represent components parallel with the surface. For an inner trajectory ($z_0<0$) the wake potential becomes

$$\phi_{\parallel}^{ind}(\mathbf{r},t) = \frac{Z_1}{2\pi}\int \frac{d\mathbf{Q}}{Q} e^{i\mathbf{Q}\cdot\check{\mathbf{R}}}\left[e^{-Q(|z_0|+|z|)\frac{1-\epsilon(\varpi)}{1+\epsilon(\varpi)}}\left\{\theta(z)-\frac{\theta(-z)}{\epsilon(\varpi)}\right\} + e^{-Q|z-z_0|}\left\{\frac{1}{\epsilon(\varpi)}-1\right\}\theta(-z)\right], \tag{2.26}$$

where the second set of curly braces represents the bulk wake potential. Figures 2.3 (a)-(e) show the surface wake potential created by a proton moving near an Al-vacuum interface in the plane containing both the normal to the surface and the particle trajectory for different positions of the ion. The perpendicular and parallel components of the force acting on electrons can be seen, and it is worth mentioning that the surface wake potential is also responsible for the deceleration of the projectile ion itself, which can be one of the origins of the surface stopping power for the ion.

Meanwhile, Figures 2.3 (f) and (g) describe the bulk wake potential, as calculated for a projectile moving with velocity $v = 2$ in a medium characterized by $\varpi_p = 0.919$, $\beta = 0.974$, and $\varpi_g = \gamma = 0$ with the use of the dielectric function[49]

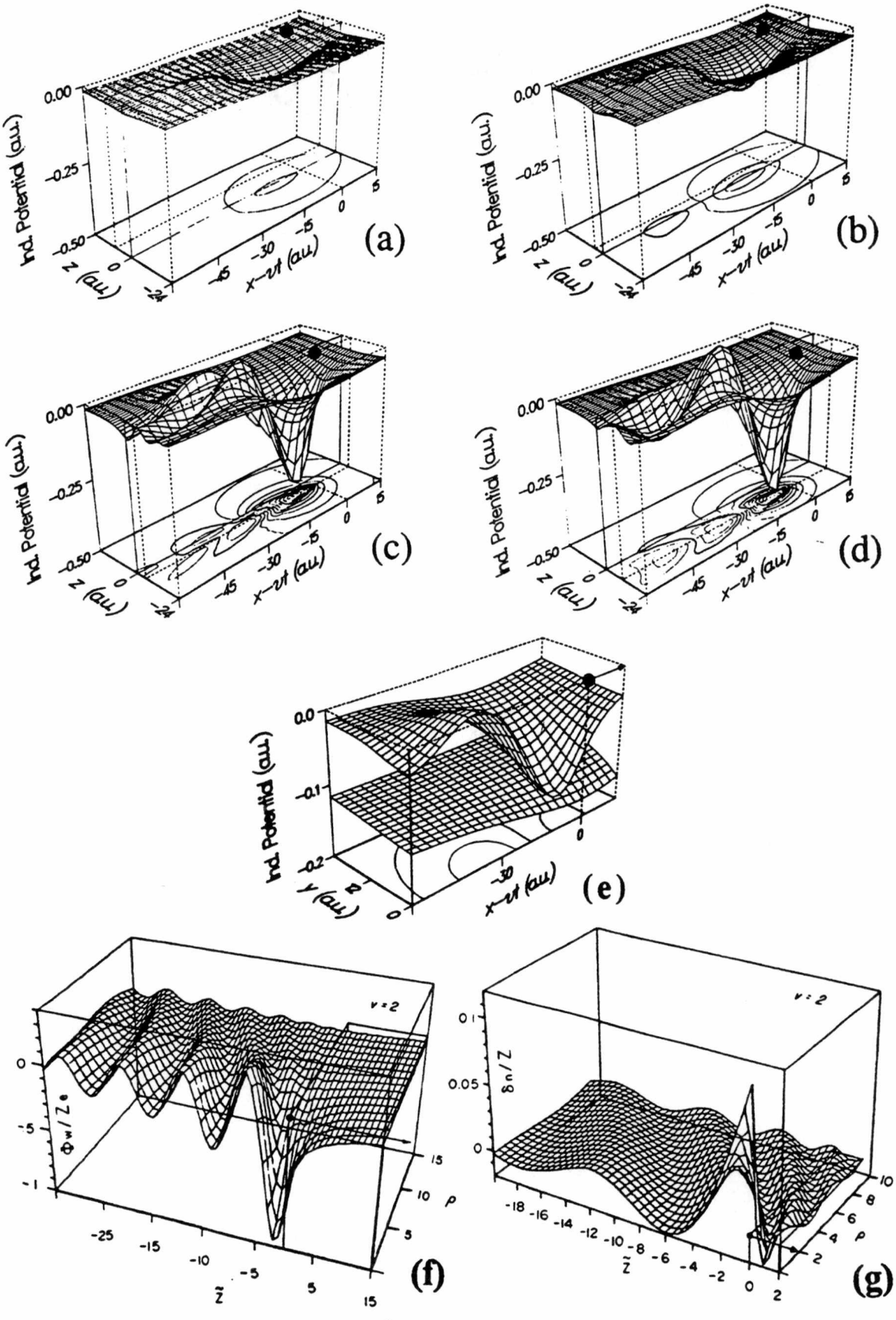
$$\in (k, \varpi) = 1 + \frac{\varpi_p^2}{\varpi_g^2 + \beta^2 k^2 + k^4/4m^2 - \varpi(\varpi + i\gamma)}. \qquad (2.27)$$

The surface plot (g), depicting electron density fluctuation induced by a moving ion under the same conditions, clearly shows the density enhancement at the origin.

**Figure 2.3**: Induced surface wake potential created by a proton, which is moving near an Al-vacuum interface ($r_s = 2$, $\gamma = 0.2$) with velocity $v = 4$ from left to right. The proton has been represented by a black circle. The wake potential is shown in the plane perpendicular to the surface and contains the trajectory for four different separations of the proton with respect to the surface: (a) $z_0 = 10$; (b) $z_0 = 5$; (c) $z_0 = -5$; (d) $z_0 = -10$. It is also plotted in a plane parallel with the surface for $z_0 = 5$ (e), $z = 0$ (upper sheet) and $z = 5$ (lower sheet; the potential has been shifted 0.1 a.u. downwards in this case). The distance between contour lines is 0.04 a.u. **Source**: F. J. Garcia de Abajo, P. M. Echenique, Phys. Rev. B**46**, 2663 (1992)

(f): Wake potential surface $\Phi_w(\rho, \check{z})/Z_e$, as calculated for a projectile moving with velocity $v$ in a medium described in text. This particle is located at the origin of the $(\check{z}, \rho, \phi)$ coordinate system as indicated by the dot and the arrow points in the direction of v.

(g): Surface depicting electron density fluctuations $\delta n/(\rho, \check{z})$ induced by a moving ion under the same condition of (f). The density enhancement at the origin is clearly visible. A series of wave fronts generated by the particle motion appear ahead of the particle. The crests of these waves extends to the side of and behind the particle, widening and blending smoothly to constitute disturbances. Note that the scales of $\check{z}$ and $\rho$ have been expanded as compared to those of (f) to exhibit details. **Source**: P. M. Echenique, R. H. Ritchie, W. Brandt, Phys. Rev. B**20**, 2567 (1979)


Ind. Potential (a.u.)
0.00
-0.25
-0.50
z (a.u.)
x−vt (a.u.)
(a)
(b)
(c)
(d)
Ind. Potential (a.u.)
0.0
-0.1
-0.2
y (a.u.)
(e)
v = 2
Φw / Z₀
z̃
ρ
(f)
v = 2
δn / Z
z̃
ρ
(g)

# III. EXPERIMENTAL APPARATUS AND TECHNIQUE

## III.1 Description of Experiment

The experimental apparatus used to collect the triply differential electron spectra for ion-gas, ion-foil, and ion-surface experiments was mounted on the 315 degree and 135 degree beamlines at the ORNL EN Tandem Van de Graaff accelerator, which is a maximum 7.0 MV machine adapted to study high-energy ion collisions with gases, solids, and electrons. We employ the ultra-high-vacuum (UHV) chamber with careful selection of *in-vacuuo* materials, a combination of cryogenic and turbo molecular pumping, and moderate baking with heating tapes. The ion beam provided by the accelerator was gas-stripped into a variety of positive charge states, energy selected, transported through a high vacuum beam line (pressure less than $10^{-7}$ Torr) to the differential pumping stage, and finally introduced into the UHV chamber, whose base pressure was in the range of low $10^{-10}$ Torr. Foil and crystal samples have been carefully selected and prepared. A rotatable electostatic analyzer and a position sensitive detector are located to collect electron spectra. The ion beam was slightly deflected or undeflected by the collision, and a 0.25" diameter exit hole was provided in the analyzer to allow the outgoing beam to be transported to a distant Faraday cup used for ion collection and normalization.

The other experimental setup for previously measured ion-gas spectra has been briefly introduced here for comparison only. The primary beams with energies of 7.74, 18.0, 28.4, and 37.8 au were $O^{5+}$ incident on Argon and Helium target gases . Tables 3.1, 3.2, and 3.3 detail the various collision systems studied in the course of this thesis.

## III.2 EN Tandem Accelerator

The EN Tandem Van de Graaff accelerator is a 7.0 MV machine used to study high-energy ion collisions with gases, solids, and electrons. Several beam lines can accommodate a wide range of space requirements. Negatively charged ion beams ranging from Hydrogen (Z=1) to Uranium (Z=92) (excluding the noble gases) are in principle available for injection into the accelerator from a sputter ion source, which is used primarily for producing ions from solid substances. By heating a cesium oven and sputtering cesium ions onto a source probe tip, the cesium ions lower the work function at the surface of the probe, so allowing source ions to become dislodged. Negative ions produced in the ion source are accelerated to approximately 100 keV, momentum analyzed by a 30° magnet, and injected into the accelerator. Its typical beam current is $1.0 \times 10^{-6}$ amperes. Figure 3.1 shows a schematic diagram of the accelerator.

Particles passing through the evacuated accelerating tube of the EN Tandem accelerator encounter two stages of acceleration. The terminal on its midway is held at a positive potential up to seven million volts. The negatively charged ions from the ion source are attracted to the positive potential at the terminal and are accelerated through

## Table 3.1

EXPERIMENTAL PARAMETERS FOR ION-ATOM COLLISION STUDIES

| | Ion - Gas | |
|---|---|---|
| Projectile Ion | $O^{5+}$ | $C^{4,5,6+}$ |
| Target | Helium, Argon | Argon |
| Ion Energy (MeV) | 6.15 : 14.28 : 22.50 : 30.00 | 24.00 |
| Cusp Energy (eV) | 210.9 : 489.6 : 771.4 : 1028.5 | 1097 |
| Cusp Energy (au) | 7.74 : 17.99 : 28.35 : 37.80 | 40.33 |
| Cusp Velocity (au) | 3.94 : 6.00 : 7.53 : 8.70 | 8.98 |
| Collision System | $O^{5+}$ + Ar (He) $\rightarrow$ $O^{6+}$ + $e^-$ | $C^{4,5,6+}$ + Argon |
| Remarks | NonCoincidence + Coincidence | NonCoincidence |

## Table 3.2

EXPERIMENTAL PARAMETERS FOR ION-FOIL COLLISION STUDIES

| | Ion - Foil | |
|---|---|---|
| Projectile Ion | $C^{4,5,6+}$ | $C^{1,2,3,4+}$ |
| Target | Carbon Foil (1.8 μg/cm$^2$) | Carbon Foil (20 μg/cm$^2$ |
| Ion Energy (MeV) | 24.00 | 2.40 : 3.60 : 6.00 |
| Cusp Energy (eV) | 1097 | 109.7 : 164.6 : 274.3 |
| Cusp Energy (au) | 40.33 | 4.03 : 6.05 : 10.08 |
| Cusp Velocity (au) | 8.98 | 2.84 : 3.48 : 4.49 |
| Collision System | $C^{4,5,6+}$ + 1.8μg/cm$^2$ C Foil | $C^{1,2,3,4+}$ + 20μg/cm$^2$ C Foil |
| Remarks | NonCoincidence | NonCoincidence |

Table 3.3

EXPERIMENTAL PARAMETERS FOR GRAZING ANGLE ION-SURFACE COLLISION STUDIES

| | Ion - Surface | |
|---|---|---|
| Projectile Ion | $C^{1,2,3,4+}$ | $C^{1,2,3,4+}$ |
| Target | Grazing Angle Si(100) | Grazing Angle Si(100) |
| Ion Energy (MeV) | 2.40 : 3.60 : 6.00 | 3.60 : 6.00 |
| Cusp Energy (eV) | 109.7 : 164.6 : 274.3 | 164.6 : 274.3 |
| Cusp Energy (au) | 4.03 : 6.05 : 10.08 | 6.05 : 10.08 |
| Cusp Velocity (au) | 2.84 : 3.48 : 4.49 | 3.48 : 4.49 |
| Collision System | $C^{1,2,3,4+}$ + Si(100) | $C^{1,2,3,4+}$ + Si(100) |
| Remarks | NonCoincidence | NonCoincidence + Coincidence |

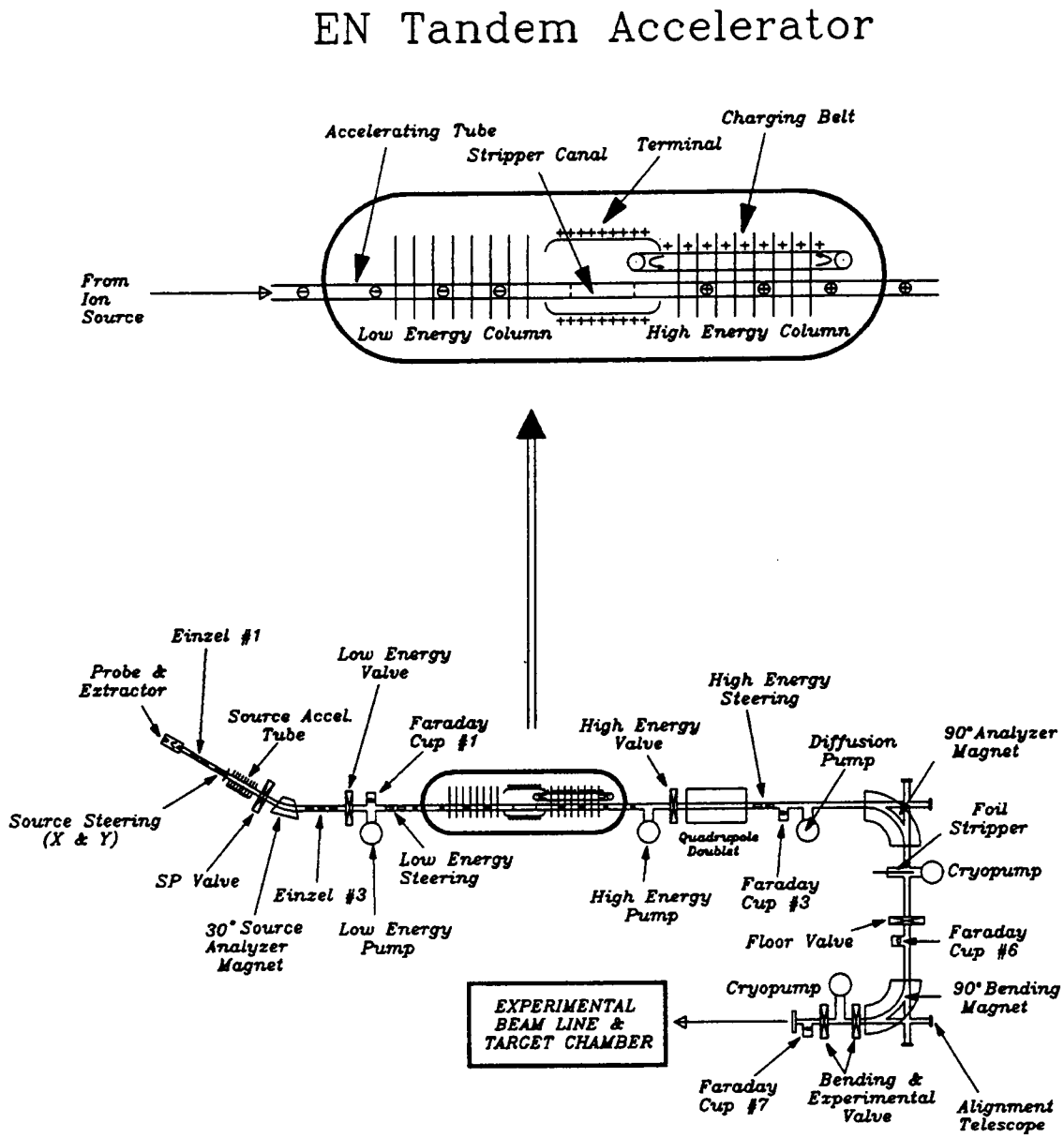



**Figure 3.1**: Schematic diagram of the ORNL EN Tandem Van de Graaff accelerator.

the first stage toward the terminal. At the terminal, the negatively charged ions pass through a stripping canal where electrons are stripped off the ions to leave them positively charged. These positively charged ions are eventually repelled by the positive potential at the terminal and are accelerated through the second stage away from the terminal. The positive potential at the terminal is maintained by dumping positive charge carried on a circulating, insulating belt onto the terminal. The potential at the terminal is varied by changing the amount of charge dumped onto the terminal from the belt. The EN Tandem accelerator itself is a metal vessel that is pressurized to 125 psi with insulating gases, sulfur hexafluoride ($SF_6$) and nitrogen ($N_2$), which insulate the high voltage at the terminal from surrounding objects.

The ion beam emerging from the Tandem accelerator has been accelerated to a velocity of up to six percent of the speed of light. After exiting the accelerator the ions are focused, using a quadrupole magnet, into a 90° analyzing magnet where only ions of a selected charge to momentum ratio are transmitted. After traveling through a short field free region, a second 90° switching magnet directs the ions into the beamline preceding our experimental chamber. These two 90° magnets with a thin carbon foil (5 or 10 $\mu g/cm^2$) stripper located between the two 90° magnets permit a range of highly stripped charge states to be produced for any ion at a given energy. The EN Tandem accelerator includes 14m long flight tube capable of performing very small angle scattering and detecting individual charge states.

## III.3 Beam Steering and Collimation

Due to the nature of grazing angle ion-surface collisions, i.e. with incidence angles of ion beams on target less than 1.0°, a well-defined and nearly parallel ion beam at the target was essential for good experimental results. Collimator #1 and #2 in Figure 3.2 were provided to collimate the incoming ion beams with two circular apertures of 0.75 mm (upstream) and 0.2 mm (downstream), separated by 150 cm, and a divergence angle less than 0.02°, which is enough to resolve grazing incidence angles, and to produce to a beam diameter smaller than 0.4 mm.

Any particles produced or scattered as a result of collisions with aperture edges were removed via a skimmer assembly consisting of a 0.2 mm tantalum aperture followed by a pair of plates to which a high voltage difference could be applied. This assembly was installed downstream between a differential pumping chamber and a 3 mm diameter differential pumping hole.

The collimation of the ion beam was also chosen to limit the spot size of the ion beam on the target surface located at the entrance focal point of the spectrometer. The collimated ion beam diameter defines an object size for the electron source which is energy-selected by the spectrometer, partially determines the energy resolution function, and reduces spurious secondary electrons produced in collisions between the ion beam and internal metal parts of the analyzer.

As illustrated in Figure 3.1, there are several beam steering and focusing devices in the EN Tandem accelerator. Skillful operation of those elements was required to maximize the transmitted beam currents in the Faraday cup, while maintaining consistent currents

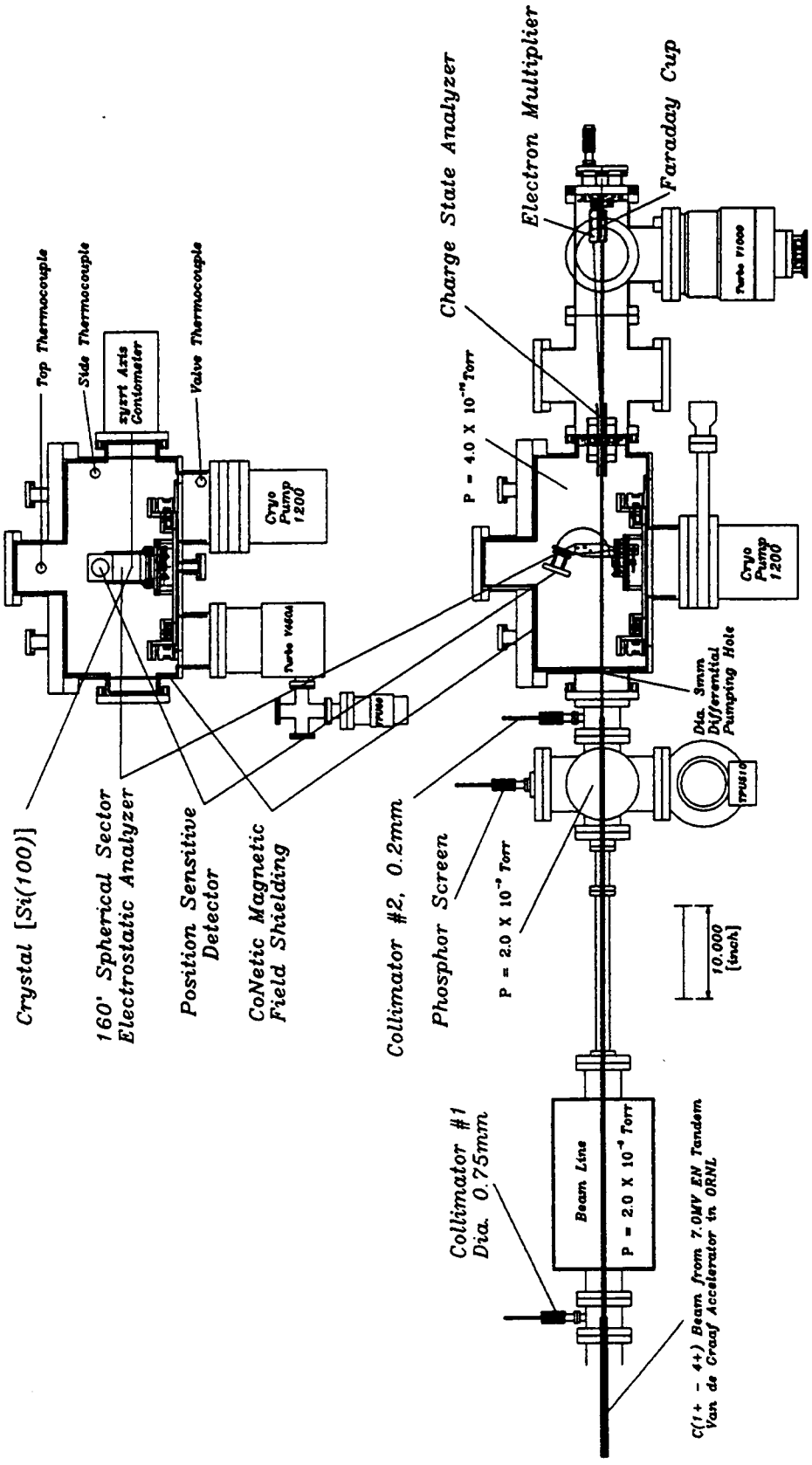



**Figure 3.2:** Experimental beam line and the UHV target chamber for ion-foil and grazing angle ion-surface collisions.

for convoy electron emission measurements and to reduce induced noise signals from the detector. The beam current was measured by transporting the outgoing ions into a series of collimators and an electrically isolated crystal target as well as a movable (perpendicular to the beam axis) Faraday cup, in which the large length-to-diameter ratio and a high voltage biased suppressor electrode was provided to prevent any secondary electrons produced in stopping the ions from escaping the cup.

## III.4 Preparation of UHV System

Figure 3.2 shows a schematic diagram of the main 24-inch diameter, stainless-steel scattering chamber, which is preceeded by a differential pumping stage and followed by a chamber housing for a scattered ion charge state analyzer and an assembly of an electron multiplier and a Faraday cup. This combination of 3 chambers presently includes all metal UHV-capable material, a combination of turbo molecular and cryogenic pumping-subsystems, and a spherical sector electrostatic electron spectrometer. This spectrometer is capable of angle-resolved measurements by means of pin-hole imaging of the source region on the face of a position sensitive microchannel plate detector.[50] This apparatus also has a layer of 0.01 inch thick CoNetic™ magnetic field shielding metal all over the inside of the UHV chamber. The existing system has achieved a base pressure ~$1x10^{-9}$ Torr inside the main chamber as well as the differential pumping stage without baking.

The main ion-surface collision apparatus is located inside the UHV main chamber, which is pumped by two turbo molecular pumps (450 l/s and 60 l/s) in series and by a

supplemental cryogenic pump (1000 l/s) as seen in Figure 3.2. This arrangement, with moderate baking, provides a residual gas pressure as low as $4x10^{-10}$ Torr.

A differential turbo molecular pumping stage isolates the main chamber and the beam line by a 3 mm differential pumping hole downstream and by a ~16" long, 2.75" diameter ion beam transport tube upstream. The 1000 l/s turbo molecular pump behind the main chamber maintains the pressure of the scattered ion analysis chamber in the low $10^{-9}$ Torr region.

## III.5 Monitoring UHV Chamber Baking Cycles

One important requirement for achieving an ultra-high vacuum in the experimental apparatus is the removal of hydrogen ($H_2$) and water ($H_2O$) from all surfaces inside the chamber. To meet this requirement, moderate and long term baking with two 750 W and two 600 W heating tapes is performed to reduce the prominent $H_2$ and $H_2O$ peaks, which on initial pumpdown produce about 1.0 and 10.0 x $10^{-10}$ amperes in the residual gas analyzer (RGA), respectively. Two 750 W heating tapes have been wrapped around main chamber and two 600 W tapes on the top of the main chamber.

A microcomputer-based heating control system, constructed as part of the present work, controls the entire heating process by holding the entire chamber at elevated temperatures without exceeding certain critical temperatures at carefully chosen thermocouple locations (Figure 3.2). The system consists of an IBM PC-compatible microcomputer with a

four-channel thermocouple ADC interface, a heater relay interface, a watchdog timer, and a control program written in C++.

The most important temperature limit imposed by the control system is ~75 °C above the cryogenic pump, so the lower heating tape is frequently switching on and off during a heating session. An approximate 200 °C limit is applied near the main copper gasket between the chamber and its lid so that leaks due to differential expansion of copper and stainless steel can be avoided.

A typical initial heating cycle is shown in Figures 3.3 and 3.4. Dramatic increases of $H_2$, $H_2O$, and $N_2$+CO partial pressures are noticeable. Typical residual gas pressure for terminating a chamber baking session is around $5x10^{-8}$ Torr, in which case the residual gas analyzer normally shows the ratio of $H_2$:$H_2O$:$N_2$+CO:$CO_2$ = 2.0:4.0:1.0:0.5 x $10^{-11}$ amperes. We have demonstrated a base pressure ~$4x10^{-11}$ Torr inside the main chamber, when isolated from the beamline.

## III.6 Magnetic Field Effects and Shielding

For electron emission measurements of ion-atom, ion-foil, and ion-surface collisions, even small residual magnetic fields can have a noticeable effect on the electron trajectories, especially for electrons less than a few hundred electron volts. The electron spectrometer was positioned so that the trajectories of the electrons being analyzed were in a vertical plane, as shown in Figure 3.2. The average magnetic field in the region surrounding the electron spectrometer was several hundred mG without shielding, based

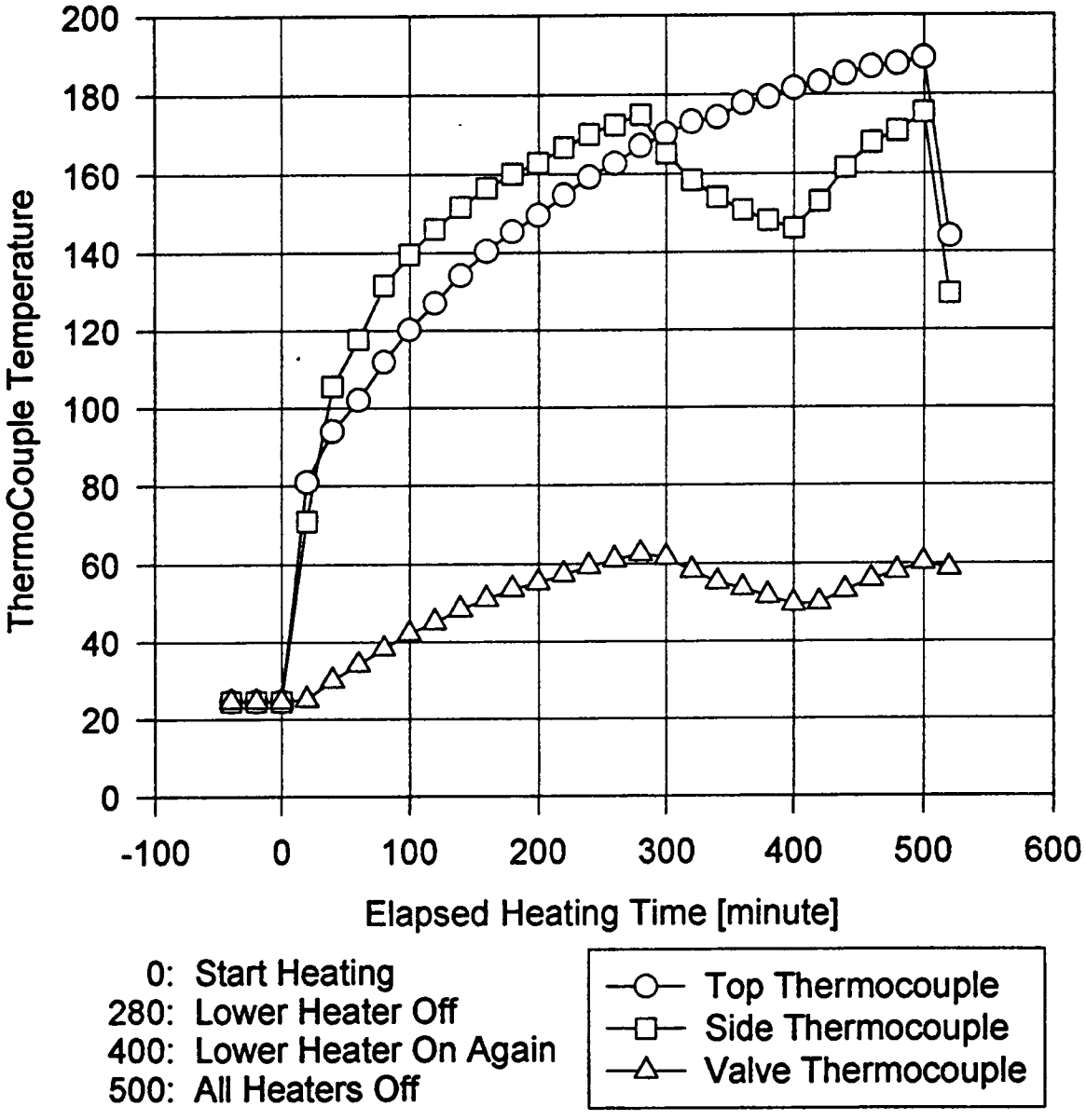



**Figure 3.3**: Temperature fluctuation outside the UHV chamber with respect to the elapsed heating time. The temperature limits for the valve and top thermocouples are 63 °C and 190 °C, respectively. At 280 min. only the lower heater was turned off to avoid excessive thermal load on the cryopump, and at 500 min. all heaters were turned off to prohibit the possible leak due to differential thermal expansions of copper and stainless steel at the seal between the main chamber and its lid.

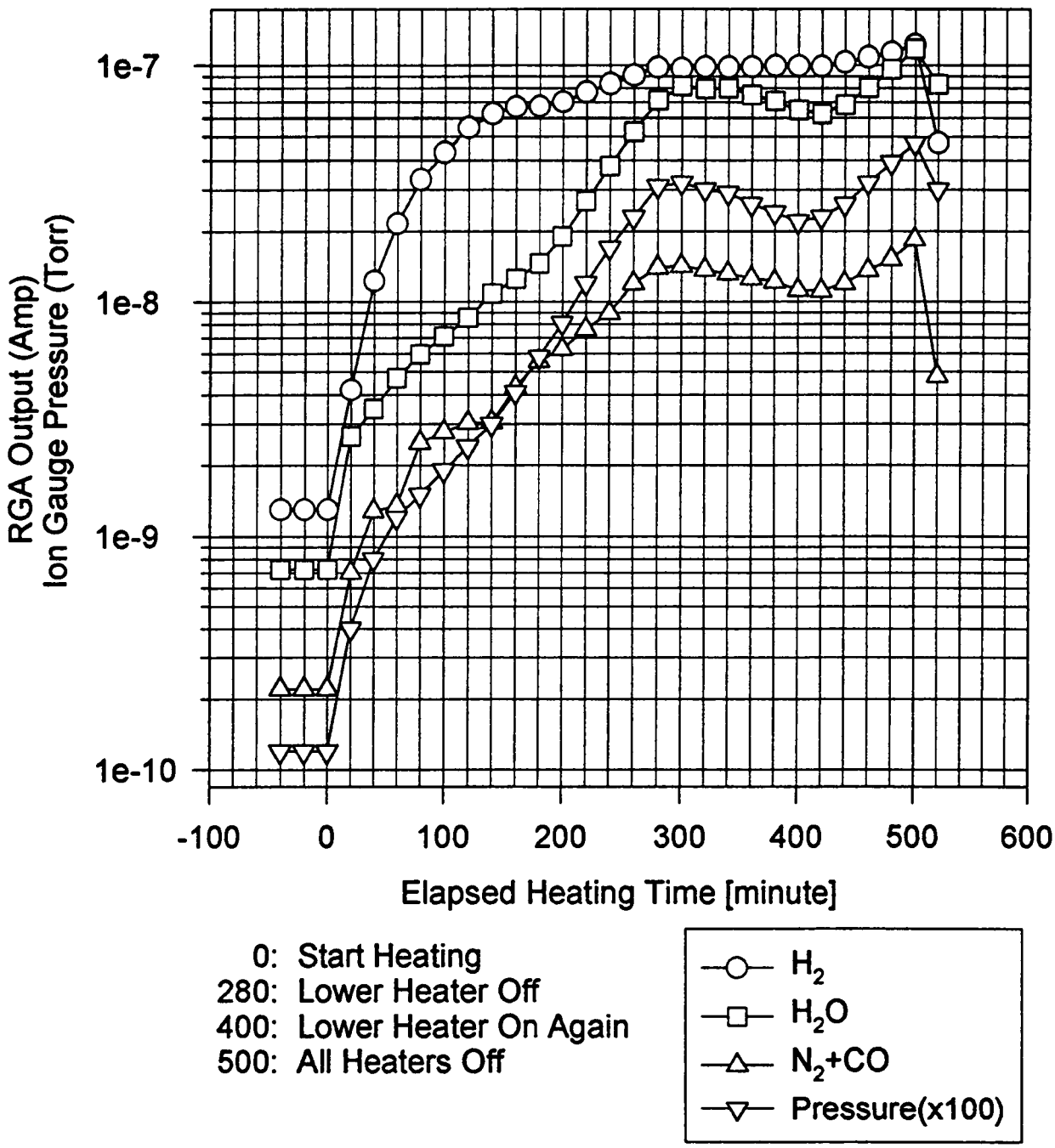



**Figure 3.4**: Residual gas partial pressures inside the UHV chamber with respect to the elapsed heating time. Hydrogen, nitrogen and carbon monoxide, and water components are rapidly rising, indicating the relative contribution of each to the total indicated ion gauge pressure inside the chamber.

on random measurements taken throughout a cubical volume sides ~20 cm. Since we have several magnetic field sources near the target chamber, including a large magnet on an adjacent beamline and DC or low frequency AC current sources as well as the earth's field, the magnitudes and orientations of the magnetic field are unpredictable and fluctuate wildly from position to position .

Typical effects of stray magnetic fields on spectrometer trajectories is illustrated in the angle and energy distributions shown in Figure 3.5. The energy-dependent angular deflection of electrons inside the spectrometer due to external magnetic fields is qualitatively expected.

There are two different ways by which the effects of magnetic fields in the region of the spectrometer can be minimized: (a) externally canceling the field with added, controlled sources or (b) enclosing the region in a high permeability metal shield. The first attempt at nullifying the external magnetic field was the former of these two methods. Three rectangular magnetic field coil pairs were mounted to separately null each component of the magnetic field in the vicinity of the main chamber. This arrangement was centered about the center of the UHV target chamber. By separately nulling each component of the field, and repeating the process until fields were established it was found that all components of the magnetic field at the center of the main chamber had been reduced to less than 20 mG. This method, however, reveals a problem that the effective volume for nullifying magnetic field is very small, and fields around center have been distorted so that the image had been washed out by the coil arrangements as seen in Figure 3.5.

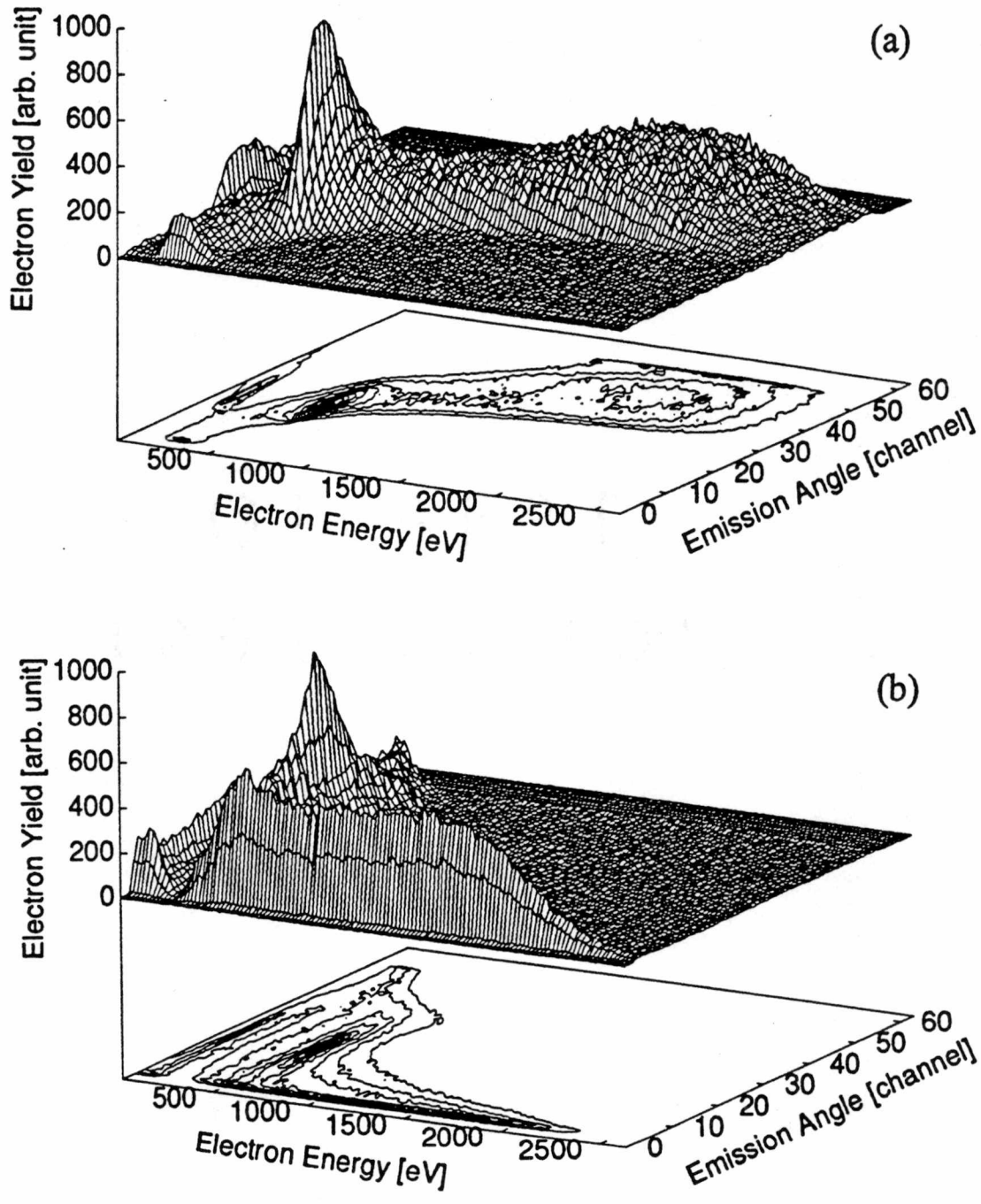



**Figure 3.5**: Electron distribution in angle and energy (a) under the influence of external magnetic fields and (b) the effect of the Helmholtz coil arrangement for 1.0 MeV/u $C^{3+}$ + 8 $\mu g/cm^2$ C foil.

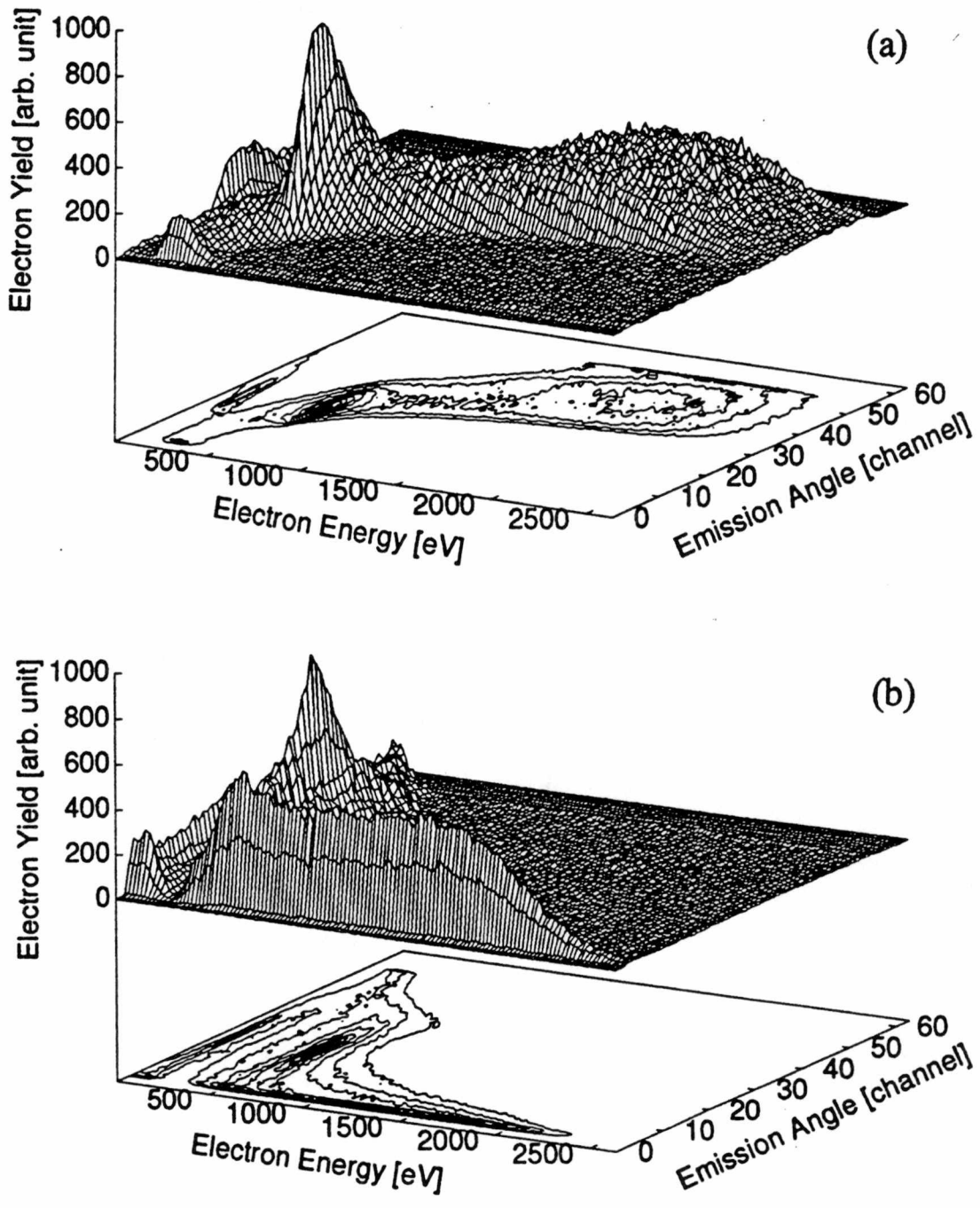


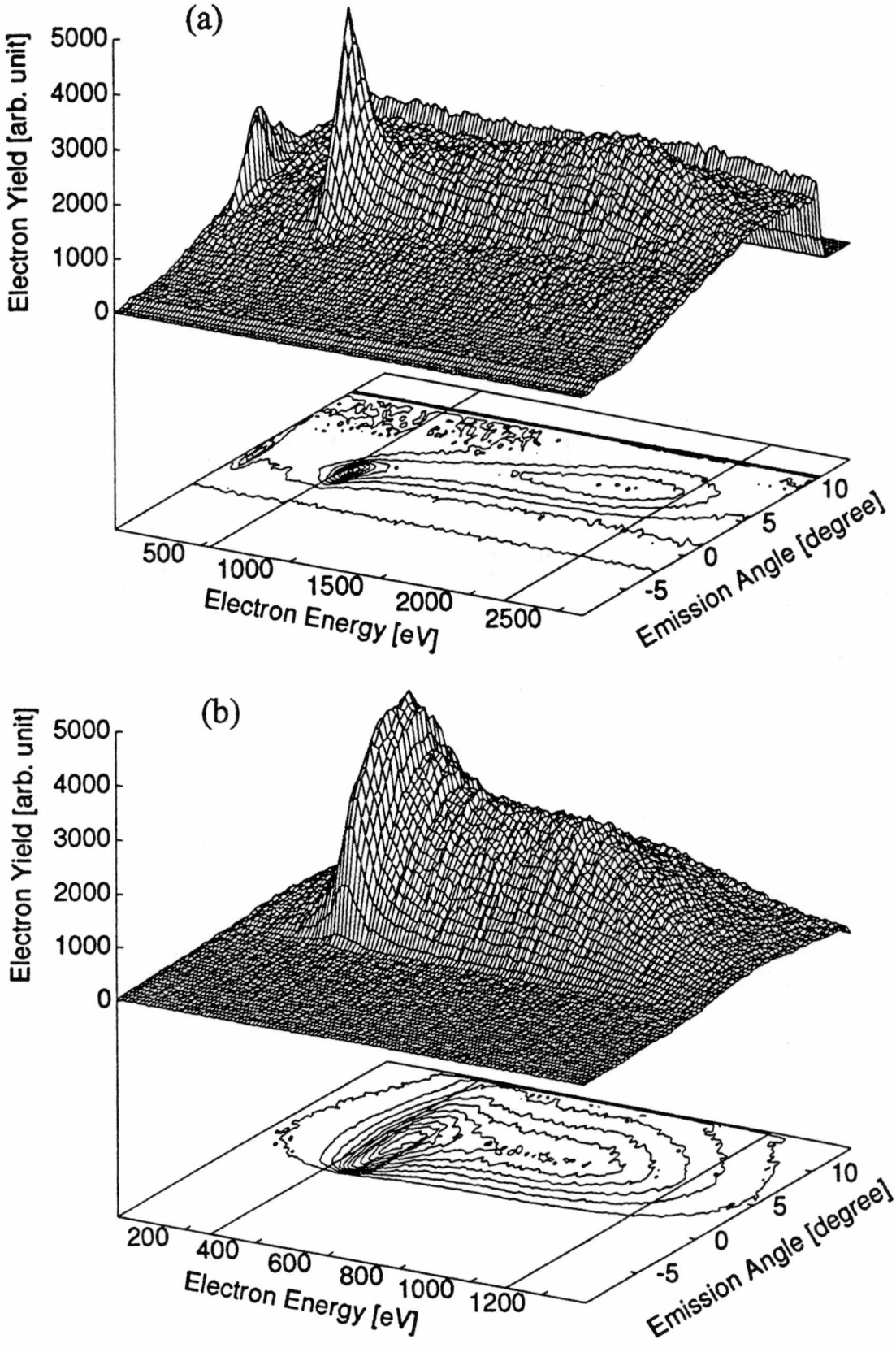

**Figure 3.5**: Electron distribution in angle and energy (a) under the influence of external magnetic fields and (b) the effect of the Helmholtz coil arrangement for 1.0 MeV/u $C^{3+}$ + 8 $\mu g/cm^2$ carbon foil.

A more successful attempt to reduce the effect of external, stray magnetic fields was achieved by enclosing the interior of the vacuum chamber with a high permeability metal shield constructed of commercially available 0.01 inch-thick Co-Netic™ magnetic shielding alloy. The shield concentrates magnetic flux lines that it intercepts and diverts them around the sensitive components. In theory a magnetic shield provides a low reluctance magnetic path for the interference field for DC and low frequency AC magnetic fields. The optimum shield is a closed, spherical configuration. In practice, effective shields can be cans, open-ended cylinders, five-sided boxes, U- or L-shaped brackets, and even flat plates. In our case we designed a one layer closed cylinder except for beam entrance, beam exit, and pumping holes. For manipulators, electrical feed-throughs, and goniometers, we enclosed the corresponding flanges with cylindrical shielding "cans" on the outside of the chamber. This method permitted a reduction of the magnetic field inside the UHV chamber to less than 5 mG, and restored all desired spectra without serious distortions in angle and energy. Figure 3.6 shows a complete spectrum after shielding against nearby magnetic field sources.

## III.7 Surface Characterization and Preparation

The specification of the orientation of the crystal targets employed in the ion-surface studies is shown in Figure 3.7, and the direction of incident ion beam to surface has been identified by information provided by the manufacturer. The parent single-crystal wafers of the silicon targets used in the ion-surface collision experiment were specially cut and

**Figure 3.6**: Electron distribution in angle and energy (a) before and (b) after shielding the magnetic field for grazing angle ion-surface collisions. (a): 1.0 MeV/u $C^{3+}$ incident on Si(100) at grazing angle of 0.5°. Two lines represent the isotachic emission energy and the BEE limit, respectively. (b): 0.5 MeV/u $C^{3+}$ incident on Si(100) at grazing angle of 0.2°. A very sharp cusp peak due to surface penetration for 1.0 MeV/u $C^{3+}$ beam is compared with that of 0.5 MeV/u $C^{3+}$.


(a)
Electron Yield [arb. unit]
5000
4000
3000
2000
1000
0
500
1000
1500
2000
2500
Electron Energy [eV]
10
5
0
-5
Emission Angle [degree]
(b)
Electron Yield [arb. unit]
5000
4000
3000
2000
1000
0
200
400
600
800
1000
1200
Electron Energy [eV]
10
5
0
-5
Emission Angle [degree]

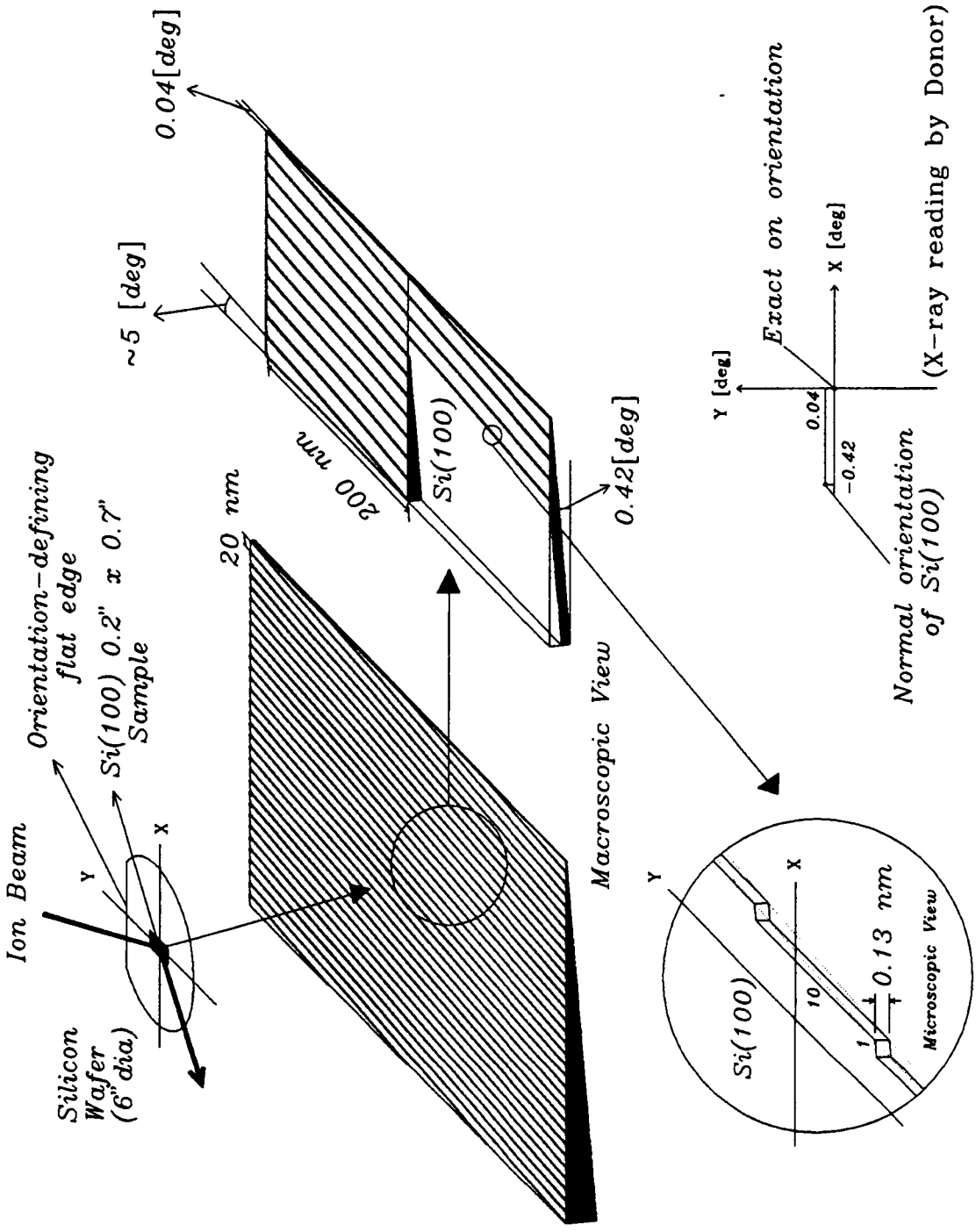



**Figure 3.7**: Silicon(100) surface specification from the information given by the donor.[51]

certified by X-ray diffraction measurements performed by the manufacturer and donor[51] to provide a crystal surface inclined at 0.4° ± 0.04° to the (100) plane, resulting in approximately regular terrace steps spaced by ~20 nm on average. Using additional orientation information supplied with the parent wafers, it was possible to mount the final target crystals so that the ion beams passed parallel to the average terrace edges. Assuming an orientation precision better than 5°, the distance between adjacent terrace edges encountered by the ions exceeded 200 nm on the average.

The crystal holder was a critical part of the ion-surface collision experiment since we were interested in reliable and reproducible grazing angles of incidence less than 1.0°, the capability to measure currents on the surface, and the capability to clean the surface by heat treatment. The silicon crystal was affixed by tantalum clips to two tantalum supports. The tantalum supports were mounted on an insulating ($Al_2O_3$) base and served as electrical contacts along the two edges of the crystal which were parallel to the incident beam direction, permitting connection of a low-voltage DC power source for direct Ohmic heating of the target, or of a picoammeter for measuring currents intercepted by the target when an ion beam was incident. All contact of target material with stainless steel was avoided to minimize contamination of the crystal surface with nickel, and possibly carbon, from stainless steel parts and tools.

The preparation of an atomically clean silicon surface free of impurities can be accomplished either by thermal treatment at a temperature greater than 1200 °C or bombardment with medium to low energy (2.0 - 0.2 keV) inert gas ions followed by annealing at ~700 °C[52]. The major contaminants of silicon surfaces are usually carbon and

oxygen, to monolayer coverage or greater, with possibly a much smaller quantity of heavy metals (Fe, Cu, Ni). Ion bombardment can remove all of these impurities, but inert gas ions are retained implanted in the surface and crystallographic damage induced by ion impact cannot be completely annealed. For these reasons we adopted a direct resistive heating treatment for the silicon surfaces employed in these measurements.

In order to remove the initial silicon-oxide layers and anneal the surface, the targets were heated by passing a current through the bulk crystal. Surface temperatures were judged by means of an optical pyrometer. The process typically consisted of an outgassing cycle at 400-500 °C for a few hours and then by a high-temperature "flash" for 1 to 2 minutes near 1200 °C while maintaining the pressure in the $10^{-9}$ Torr range or less, followed by a slowly controlled cooling (~2°/second) to the ambient temperature of surrounding hardware.[53] After that the crystal was found to be "useable" for about 20 hours before requiring further heat treatment, which means that during this time the coverage of the surface with residual gas atoms was small as judged by consequent changes in observed electron emission spectra. From data such as that in Figure 3.8, we could observe an angular shift of the convoy peak that increased after heating - by about 0.7° for convoy electron peaks produced with grazing angles less than 0.5°, while both angular shifts (before and after heating) converge to ~4.3° for grazing angles greater than 0.8°. It seems that the chemical purity of the crystal surface is no longer a concern when penetration of the incident ion is getting deeper. It is interesting to note that there is a maximum shift for convoy electrons around 0.5° after heating the crystal. This process was presumed to be good enough for the exploratory studies here to ensure that the

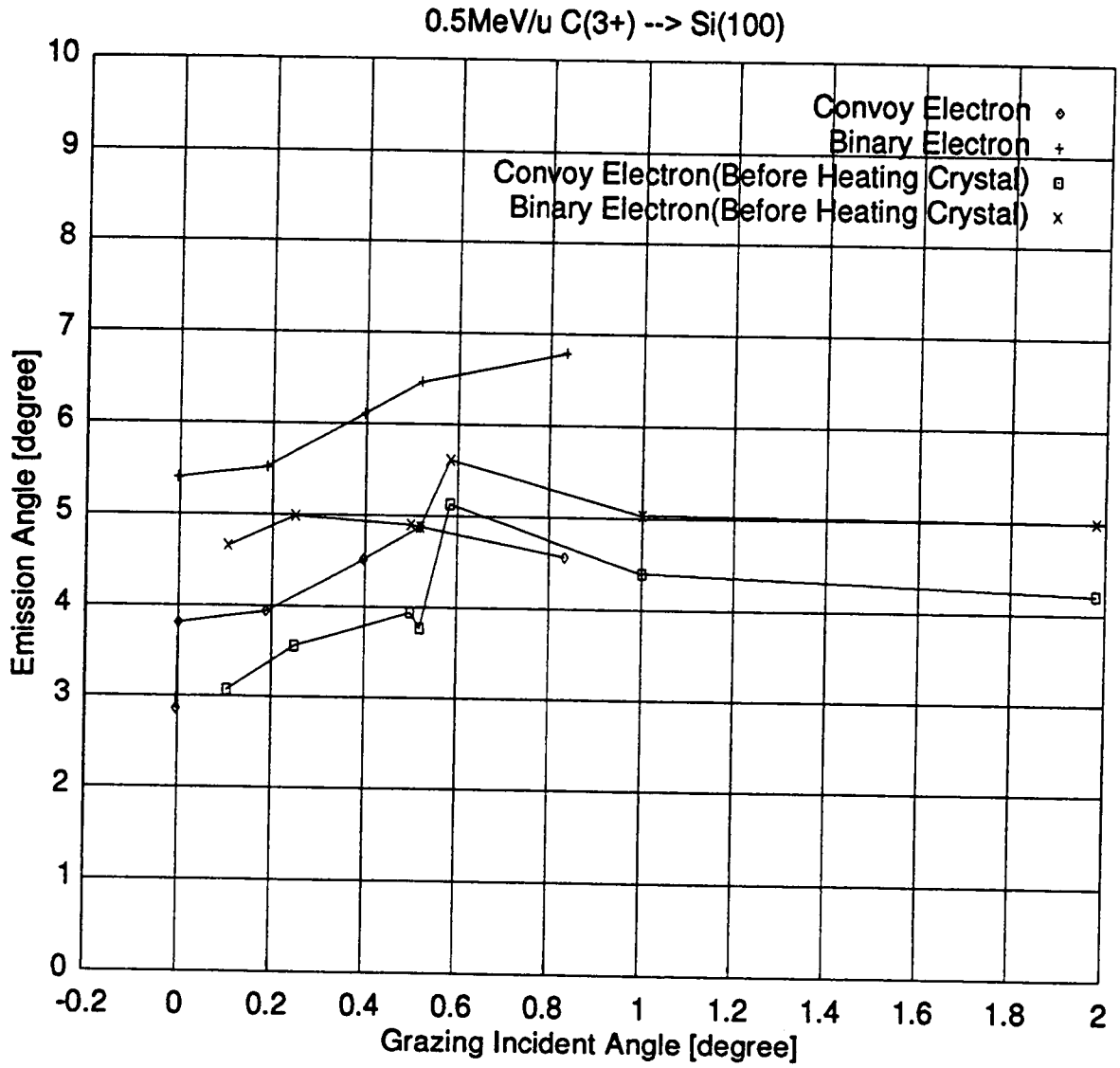



**Figure 3.8**: Convoy and binary peak shifts in emission angle with respect to the specular beam before and after heat treatment of Si(100). The error margins for convoy and BEE are about $\pm 0.3°$ and $\pm 1.0°$.

surface was still clean without measuring a low electron energy deflection (LEED) pattern or Auger electron spectra.

## III.8 Determination of Grazing Angle

The surface target base was mounted on a goniometer head which was in turn mounted on a manipulator with 5 degrees of freedom (x, y, z, r, and t), allowing alignment of the intersection of the ion beam, the center of the surface, and the entrance focus of the spectrometer at one point. In addition, the target mounting permitted adjustment of the ion beam angle of incidence and of perpendicularity between the incident beam and the axis of the incidence angle rotation. The grazing angle of incidence is defined as the compliment of the usual angle of incidence and is measured between the incident beam and the surface in the plane containing the incident beam and the surface normal. Because the incidence angle adjustment was achieved by a linear motion feed-through, the angle was periodically calibrated by measuring the deflection of a laser beam reflected from the surface of the crystal for different positions of the incidence angle motion feed-through. The laser beam deflections measured on the 6.0" clear window flange were linearized using trigonometric equation ($f(x) = D * \theta = D * \tan^{-1}(x/D)$ where $D$ = distance from the surface to the center of 6.0" window, $\theta$ = deflection angle, and $x$ = measurement), fitted by the third order polynomial equation ($g(z) = \theta(z)/2 = az^3 + bz^2 + cz + d$ where $g(z)$ = degree as a function of tilting number and $z$ = tilting number), and plotted in Figure 3.9 including the fit result and residual errors. From the result we determined that the error margin for the grazing

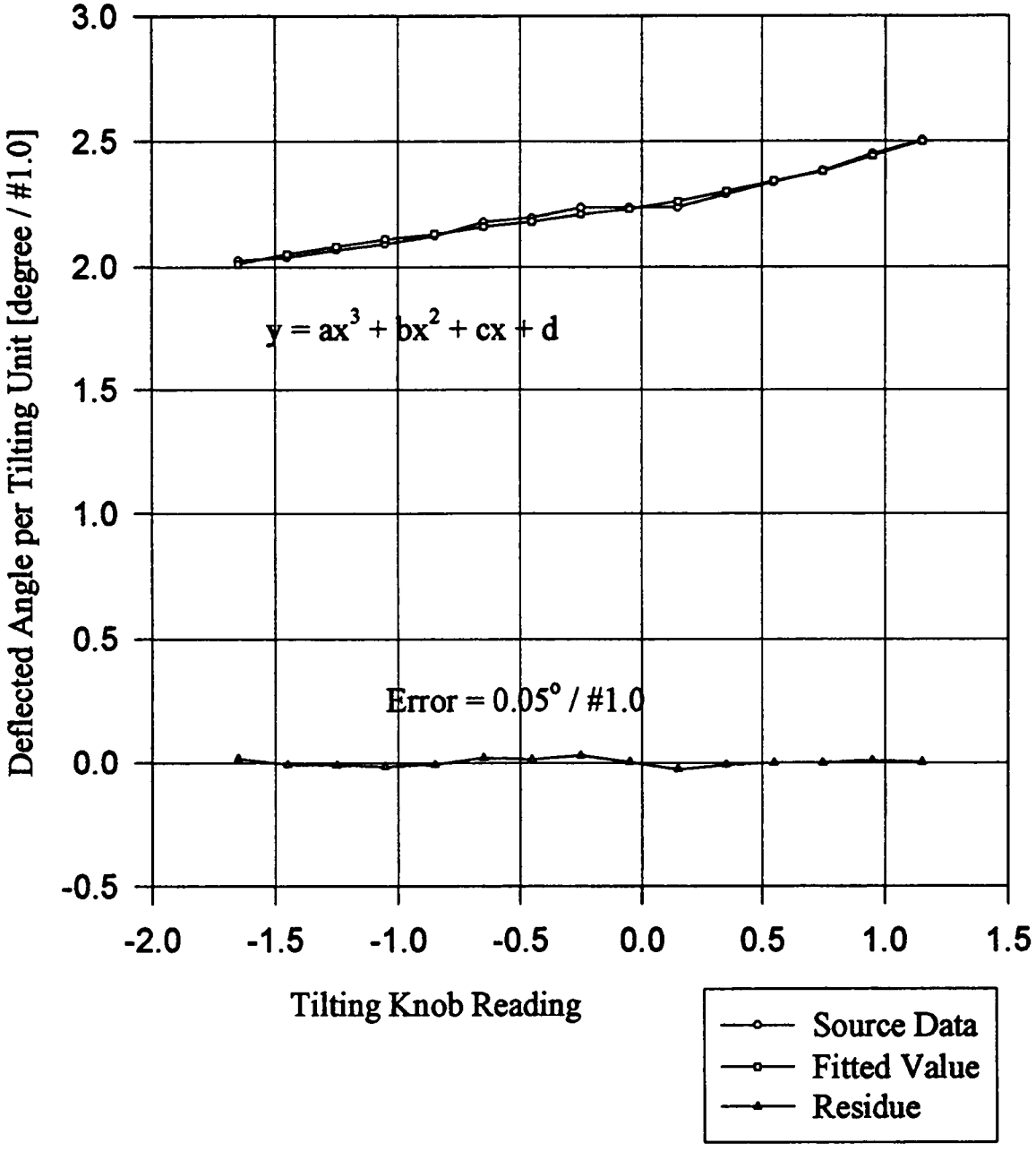



**Figure 3.9**: Determination of grazing angle by 3rd order polynomial fit. The error was decided by the residual plot and translated to at most 0.05°/#1.0 or ±0.01°.

angle is at most 0.05°/#1.0 (tilting knob reading), or ±0.01°.

The "zero" of this angle is determined by measuring the current on the surface produced by the incoming ion beam as a function of the position of the incidence angle feed-through; a sample plot of the dependence of the surface current and Faraday cup current on the grazing incidence angle is shown in Figure 3.10. Since the axis of the incidence angle motion lies on the front illuminated surface of the crystal, the crystal intersects a minimum beam cross section when its surface is parallel to the beam axis; if the crystal is tilted away from this zero direction, it intersects more of the ion beam (the crystal current rises) and the current in the Faraday cup decreases, leading to a minimum in the surface current and a maximum in the current measured in the Faraday cup. The current measured at the surface consists of three contributions: ions incident, ions leaving, and electrons ejected from the surface. For the positive angles of incidence (as defined here), the collisions involved are grazing and produce high electron yields. For negative angles the ions strike the leading edge of the crystal nearly normal to the local (edge) surface with penetrating collisions and consequently lower secondary electron yields. As a result the slope of the surface current versus grazing angle differs in magnitude as well as in sign for negative and positive angles.

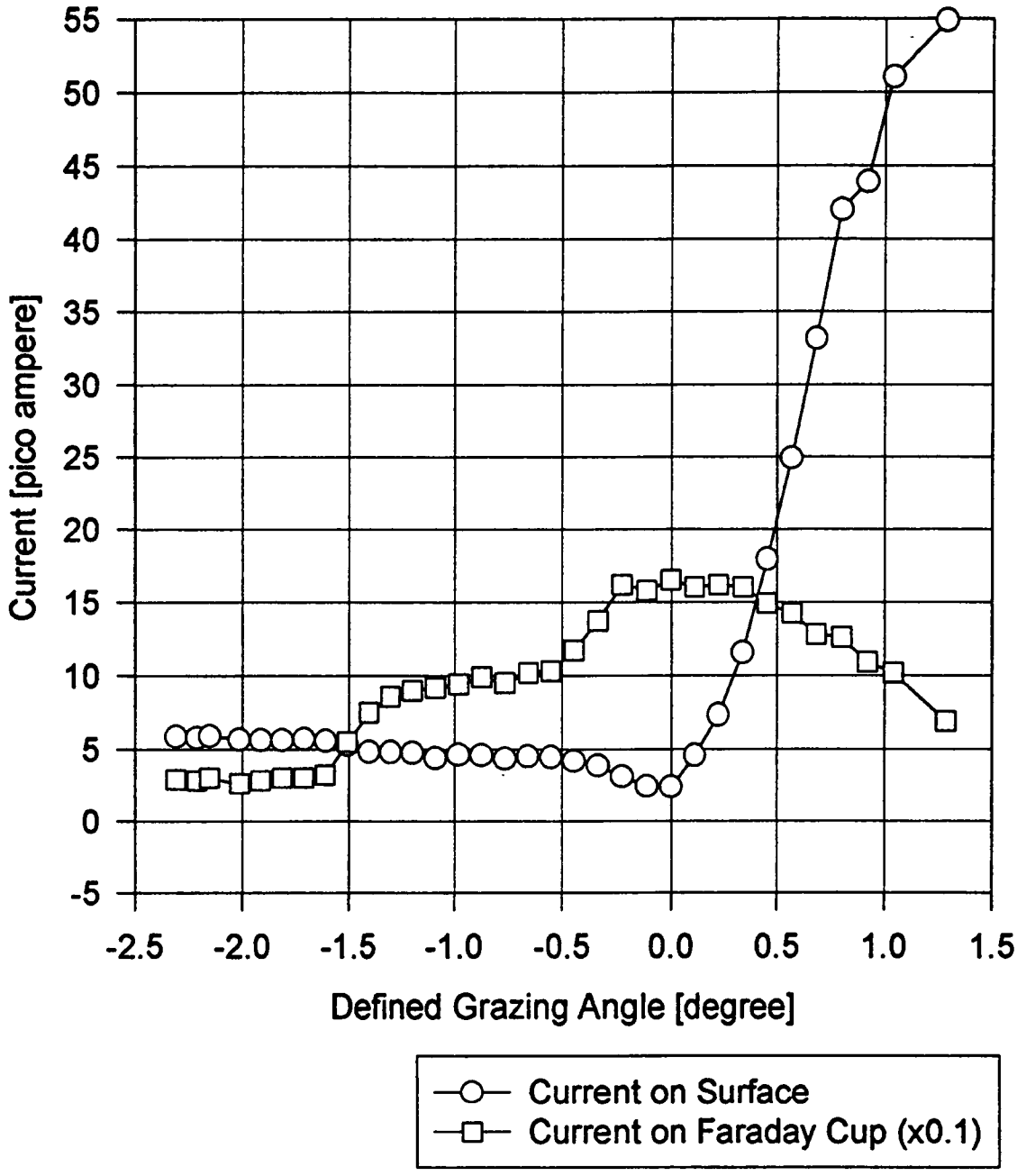



**Figure 3.10**: Dependence of the surface tilting angle on surface and Faraday cup current. Interception of the ion beam by the crystal produces a corresponding current decrease in the Faraday cup.

## III.9 Spherical Sector Analyzer

The electron spectrometer used to analyze the angle and energy of electron emission for ion-atom, ion-foil, ion-surface collisions was a 160° spherical sector electrostatic analyzer similar to that introduced by Purcell[54]. As shown in Figure 3.11, two electrically isolated concentric spherical copper sectors are held in place by two vertical stainless steel electrode mounting side plates. We placed entrance and exit apertures just outside the ground planes limiting the possible entrance positions into the analyzer and the exit position. It has been shown that the finite source and exit aperture sizes produce a finite resolution in energy and angle which may vary.[55]

The angular emission of the energy-selected electrons arriving at the exit aperture are approximately inverted from its entrance angle into the analyzer. This characteristic of the 160° electrostatic analyzer allows the entrance angle to be inferred from measurements of the position of the electron after it exits the analyzer. With a drift region of 61 mm length in a field free region between the 1.0 mm exit aperture and the microchannel plate entrance, the resistive anode-encoded position sensitive detector (PSD)[56] views an angle cone with a half-angle of about 12° for grazing angle ion-surface collision experiment.

Purcell's 180° analyzer was originally designed to provide stigmatic double-focusing to second order in the deflection plane or the angle of $\alpha$ in Figure 3.11, and allow all angles from the central ray to be focused to the image point. Seen from the Table 3.4 the basic differential equation and a second order solution contains one term with $\alpha^2$ but is independent of ß. So it is obvious that from the spherical symmetry this analyzer yields

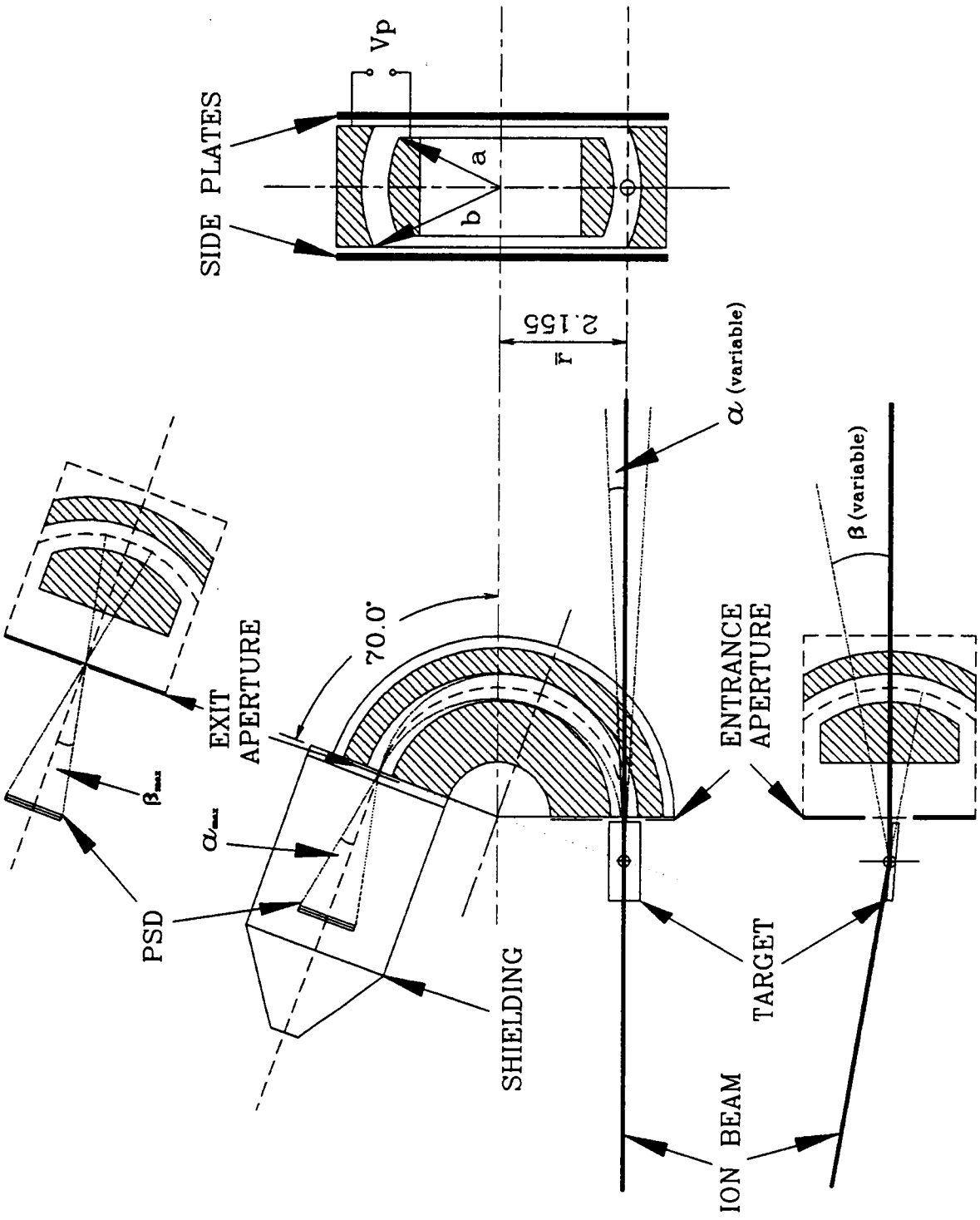



**Figure 3.11**: Schematic layout of electrostatic spherical sector analyzer.

perfect focusing in the plane of the angle of ß. However, under focusing condition, $\theta = \pi$, the last term in solution vanishes. It retains the only first power of $\alpha$ so that we have the first order focusing.

Table 3.4

PROPERTIES OF THE SPHERICAL ELECTROSTATIC ANALYZER[11]

| | |
|---|---|
| Electric Field | $\varepsilon_r = -V_p/r^2 \cdot a \cdot b/(b-a)$ |
| | $\varepsilon_\theta = \varepsilon_\phi = 0$ |
| Energy of Central Ray | $E_0 = q \cdot V_p/(b/a - a/b)$ |
| Analyzer Constant $k$ ($=qV_P/E_0$) | $b/a - a/b$ |
| Differential Equation | $\frac{d^2u}{d\theta^2} + u = \bar{r} \cdot E_0/r_0 \cdot E \cdot \cos^2\alpha$ where |
| | $u = r_0/r$ and $\bar{r} = (1/2)(a+b)$ |
| Solution of Differential Equation | $r_0/r = \frac{\bar{r} \cdot E_0 \cdot (1-\cos\theta)}{r_0 \cdot E \cdot \cos^2\alpha} + \cos\theta - \tan\alpha \cdot \sin\theta$ |
| Focusing Condition | $\theta = \pi$ |

The electrostatic analyzer used for this study has sectors subtending an angle of 160°. This 20° offset makes the external placement of surface and its holder possible. Also compared with the original 180° analyzer, this one preserves the focusing qualities to a good degree of the properties of the 180° analyzer.[57,57] Furthermore the radial electric field at the sector boundary, introducing severe aberration in the focal properties of the analyzer for the 180° design, can be minimized.

The angular acceptance of the analyzer is restricted to an approximate half-angle of 6° in the deflection plane as a result of interception of electron trajectories by the spherical sectors, leading to overall acceptance half-angles of about 6° in the deflection plane and

12° in the orthogonal direction. Two programmable Glassman MJ-series power supplies (+5 , -6 ) in the fixed ratio $V_o/V_i$ = -0.8041 (Figure 3.12) supply the deflection voltages for the outer ($V_o$) and inner ($V_i$) sectors so that the central trajectory is at ground potential. The analog output of a 16 bit digital-to-analog converter (DAC) controlling the analyzer deflection voltage supply was interfaced to an IBM PC-compatible for data acquisition. Periodic tests of the performance and energy calibration of the electrostatic analyzer were performed by observing the convoy electron emission produced by ions transmitted through a self-supporting 20 μg/cm$^2$ carbon foil positioned normal to the incident ion beam at the analyzer entrance focus.

The base width of a trapezoidal transmission function for the spherical spectrometer expresses the probability of transmission through the spectrometer of an electron of a given energy E and launched with an angle α with respect to the central ray in deflection plane

$$\frac{\Delta E}{E} = \frac{x_1+x_2}{2\bar{r}} + \alpha^2 \qquad (3.1)$$

where $x_1$ and $x_2$ are the displacements of the object and image points, respectively, and $\bar{r}$ is the radius of central ray, $\frac{1}{2}(a+b)$.[11] The magnification is unity, and the analyzer work equally well if the entrance and exit slits were interchanged since $x_1$ and $x_2$ are interchangeable. The transmission function can be approximated by a trapezoidal function of a top width (TW), and a full width at half maximum (FWHM) as $\Delta E$

$$\Delta E = E[\max(x_o, x_i)/2\bar{r} + \alpha^2] \qquad (3.2)$$

where $x_o$ and $x_i$ are the object and image sizes, respectively. In our case we do not have a defining aperture at the object point of the analyzer so that the size of the projectile beam

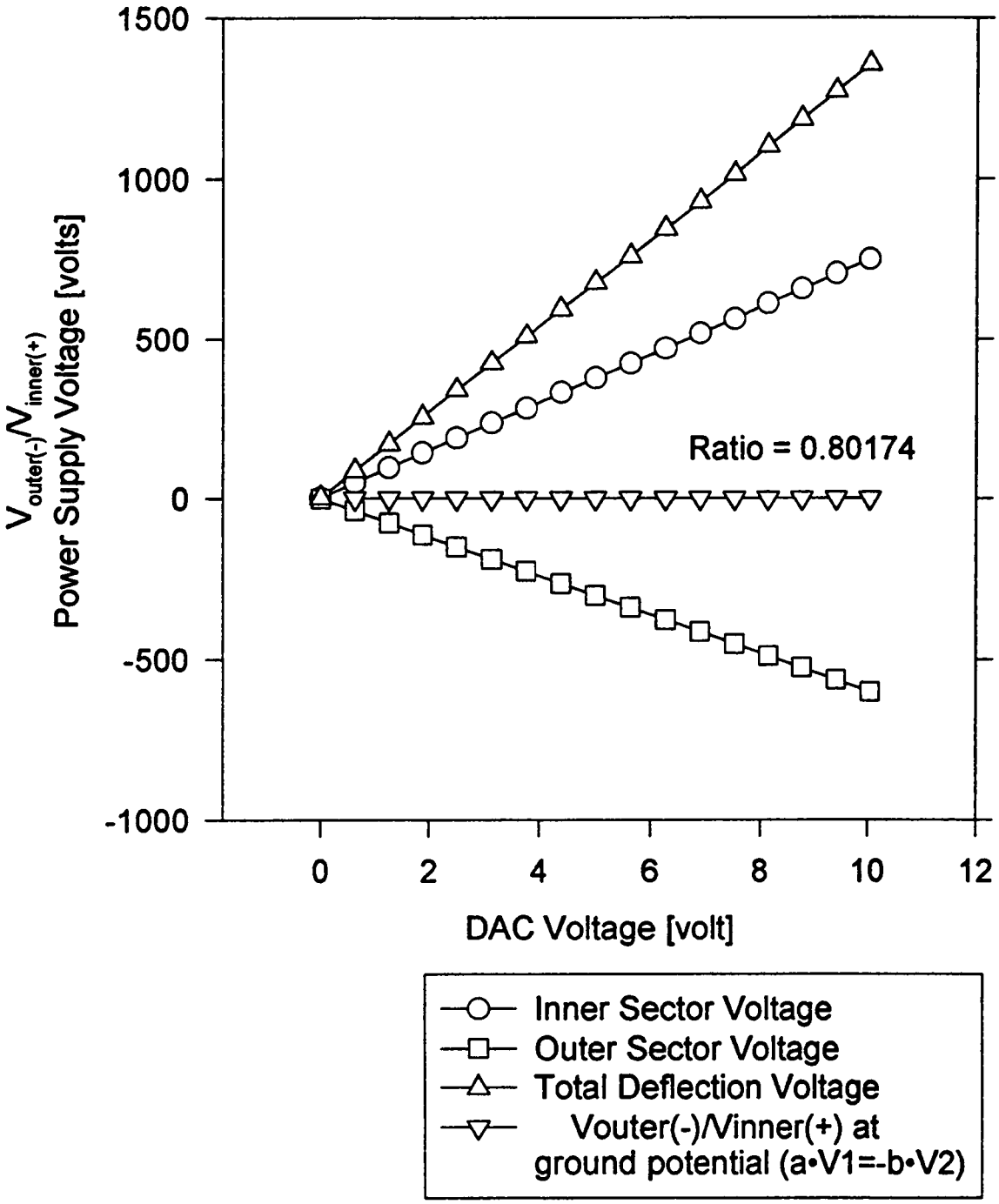



**Figure 3.12**: Deflection analyzer voltage calibration.

determines the range of energies $\Delta E$ allowed through the exit slit. α (alpha) is the emission angle as shown in Figure 3.11, and the aberration term can be eliminated since we can resolve this angle less than 0.5°. As a result we have the energy resolution functions defined by simply the sizes of the ion beam and exit aperture.

$$\frac{\Delta E}{E}|_{FWHM} = \frac{\max(x_o, x_i)}{2F} = \text{Constant} \tag{3.3}$$

$$\frac{|x_o - x_i|}{2F}|_{TW} = \text{Constant} \tag{3.4}$$

Physical constants for the analyzer are given in Table 3.5. 2-dimensionally fitted value for the transmission function seen in a late chapter is well agreed with the theoretically calculated value.

Table 3.5

SPHERICAL SECTOR ANALYZER CALIBRATION PARAMETERS

| | |
|---|---|
| Sector Angle | 160° |
| Radius of Inner Sector ≡ a | 4.880 cm |
| Radius of Outer Sector ≡ b | 6.069 cm |
| Angular Acceptance Angle | 6.0° |
| Theoretical Analyzer Constant ≡ $k$ | b/a - a/b = 0.440 V/eV |
| Measured Analyzer Constant ≡ $k$ | 0.454 ± 0.01 V/eV |
| Exit Aperture Size | 1.0 mm |
| Theoretical Energy Resolution ≡ FWHM | 0.91 % with Dia. 1.0 mm Beam |
| Fitted Energy Resolution ≡ FWHM | 1.02 ± 0.08 % (Table 5.2 and 5.3) |
| Calculated Ratio of $V_{outer}/V_{inner}$ | -0.8041 |
| Measured Ratio of $V_{outer}/V_{inner}$ | -0.80174 |

## III.10 Position Sensitive Detector

During drifting through a field free region from the exit focal point of the analyzer, electrons with initial angles near zero degree will remain close to a straight line extension of the central ray, and electrons with different angular divergences from the central axis will keep their track without being disturbed before arriving at the position sensitive detector (PSD). This position information of the corresponding energy selected electrons at the PSD is encoded and decoded by electronics.

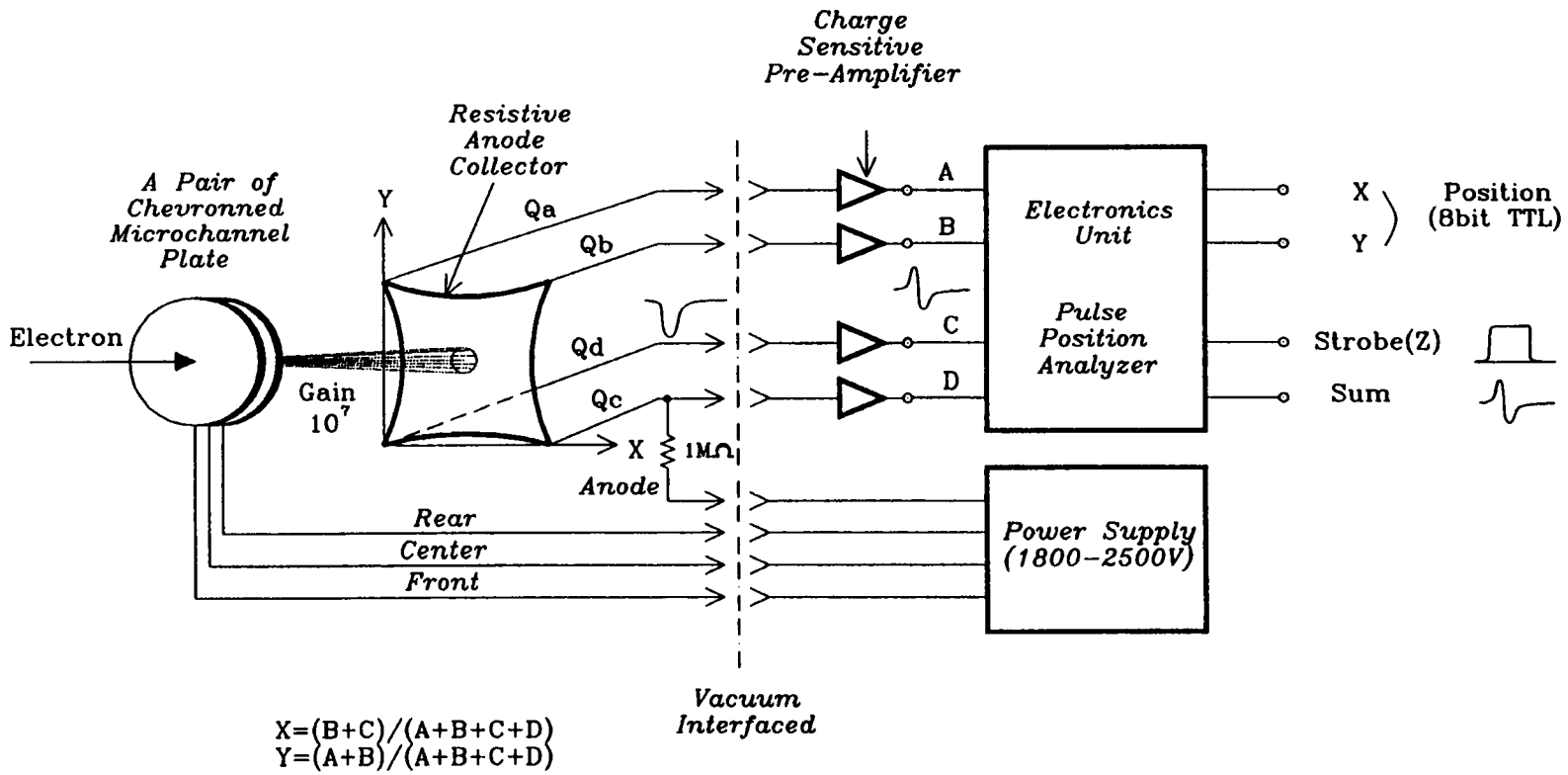
The PSD system includes a chevronned pair of microchannel plates (MCP), which is an array of $10^4$ - $10^7$ miniature electron multipliers oriented parallel to one another and has approximately $10^7$ gain, followed by a circular-arc-terminated resistive anode, charge sensitive pre-amplifiers, and position analysis electronics as illustrated in Figure 3.13. The MCPs are 25 mm in diameter and have 12 μm diameter channels biased at an angle of 8° to the input surface normal. About 1 kV per plate high voltages are kept during data acquisition, and the outgoing electrons are accelerated by 270 V to a resistive anode, which contains four independent corner electrodes where output pulses are detected and analyzed by standard charge sensitive pulse detection electronics. In turn the resulting charge pulses are analyzed by position analysis electronics followed by standard NIM electronics.

The position analyzer electronics unit, a Surface Science Laboratories[10] Ratiometric Position Decoder, generates a voltage output proportional to pulse position coordinates of the form,

$$X = \frac{B+C}{A+B+C+D}, \; Y = \frac{A+B}{A+B+C+D}, \tag{3.5}$$

**Figure 3.13**: Position sensitive detector electronics layout. A pair of chevronned micro-channel plate creates a charge cloud of an incident electron on the resistive anode via the electron multiplication process; the anode contains four corner electrodes where output pulses are detected. The position of the event is then analyzed by the position analyzer electronics.

# PSD Electronics Layout



which are two eight bit TTL digital quantities. After this pulse has been analyzed and its position signal is available at the X and Y outputs, a strobe pulse is sent.

Analog position pulses as well as digital pulses are also available for display on an oscilloscope to monitor the incoming signals during data acquisition. In order to prevent any low energy electrons from other sources, which are found to be negligible, we put a pair of 80-90 % transparent copper mesh grids in front of the detector surface.

For determination of the angular resolution, the exit aperture size and the distance between the exit aperture and the surface of the MCP were parameterized so that we have a pillbox shape angular resolution function: all electrons passing through the exit aperture of the analyzer with the same angle of emission will form an image on the detector equal to this size. Figure 3.14 shows geometrically-calculated and fitted angular resolutions as a function of electron energy for ion-atom collisions. It seems that there exists a slight dependence on the electron energy, but it remains in a question since analytical calculation shows no evidence of the energy dependence. In a realistic approach a trapezoidal-pillbox shape angular resolution function is considered to be applicable to the fitting process to explain the behavior in Figure 3.14. The 2-dimensional non-linear $\chi^2_\nu$ tests using a pillbox and a trapezoidal-pillbox shape angular resolution functions have been performed, but show less than 5% difference in $\chi^2_\nu$. Thus, results of multipole moments for ion-atom collisions in this dissertation are experimentally calculated with the pill-box shape angular resolution function. S. D. Berry *et al.* calculated the relative angular transmission function and approximated it for a pillbox shape.[58] More specific analysis with angular resolution functions will be dealt in the following chapter.

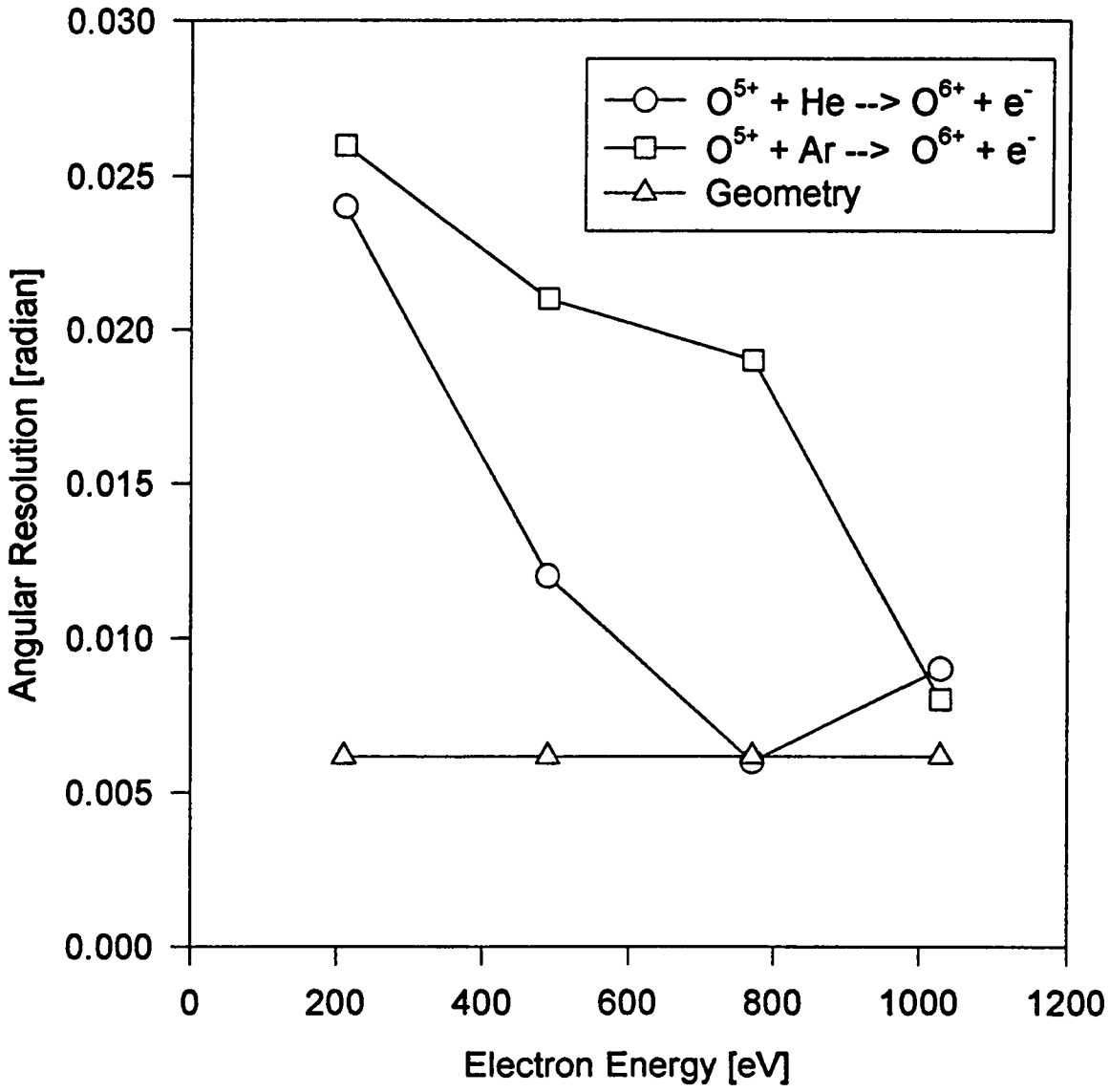



**Figure 3.14**: Fitted and geometric angular resolution functions.

## III.11 Charge State Analyzer and Electron Multiplier

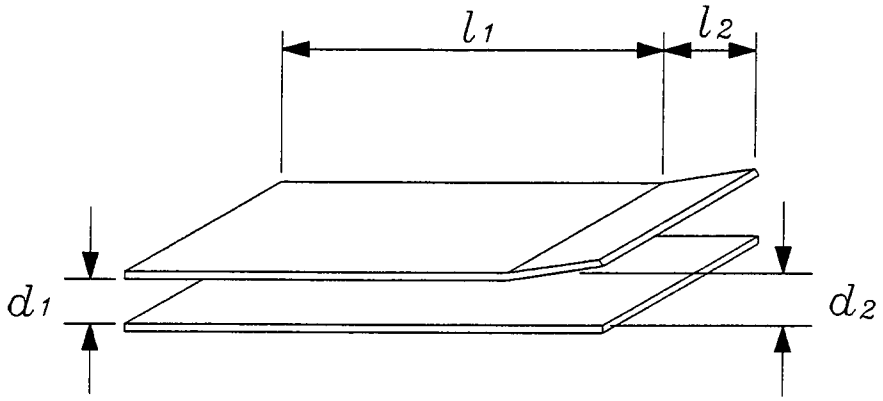
The ions reflected from the target surface as well as the foil around the scattering angle of the specular reflection were selected by an electrostatic charge state analyzer according to the charge state. The ions have been separated by a high voltage across two polished aluminum plates as seen in Figure 3.15, while entering the region at the point of one third of the total distance between the plates away from the straight plate. The total deflection at which ion detection took place was fixed during the experimental beam time so that only one parameter, a supplied high voltage, was given to select the interesting charge state, as approximately described by

$$s \approx \frac{10^{-3}qV}{4ME} \cdot \left(\frac{l_1^2}{d_1} + \frac{l_2^2}{d_2} + \frac{2l_1l_2}{d_1} + \frac{2l_2l_3}{d_2} + \frac{2l_1l_3}{d_1}\right) \text{ [cm]} \qquad (3.6)$$

where $V$ = voltage across plates in kV, $q$ = ion charge, $M$ = nuclear mass number, and $E$ = ion beam energy in MeV/u.

A movable ion detecting unit including a Faraday cup and a discrete dynode electron multiplier, which are separated by 0.9" to resolve the exit charge states, was mounted at the end of the experimental beam line as shown in Figure 3.2.

The discrete dynode electron multiplier, which is used to detect projectile ions of a particular exit charge state, consists of 21 dynodes electrically connected to each other by a resistor chain. The first dynode is in direct line-of-sight with the input aperture, enabling the detection of charged particles and ions by a series of multiplications of secondary electrons emitted from the previous dynode surface. With a high voltage supply of 2 kV, the discrete dynode electron multiplier has a gain of $10^7$ with maximum linear count rate of 5 MHz.




$l_1$ : length of straight section
$l_2$ : length of bent section
$l_3$ : distance between CSA and ion detector
$d_1$ : separation of parallel plates
$d_2$ : average separation of the bent plates

**Figure 3.15:** Illustration of the charge state analyzer consisting of two "parallel" plates.

# IV. DATA ACQUISITION AND ANALYSIS

## IV.1 Electronics

Electronics setup for data acquisition and storage of the information from the PSD, electron multiplier, and Faraday cup is shown in Figure 4.1. Recently developed coincidence experiment with the start signal from the sum pulse of the PSD and the stop signal from the electron multiplier enables us to measure convoy electron emission with particular charge states of outgoing ions in specific ranges of ion scattering angles for various collision systems. The IBM PC compatible computer controls the whole system with a 3-dimensional (angles in $x$ and $y$, and energy) histogramming software program, SIDA,[59] designed to acquire, display, and store the data. Dose/Scaler/DAC Interface and PSD Interface NIM modules have their own events defined as the completion of a preset dose of ion beam current detected by an electron multiplier and the detection of a strobe pulse following the position and sum signals from the pulse position analyzer, respectively. Ongoing "event signals" on two MetraByte model PI012 computer interface cards continuously interrupt the software program during data acquisition time.

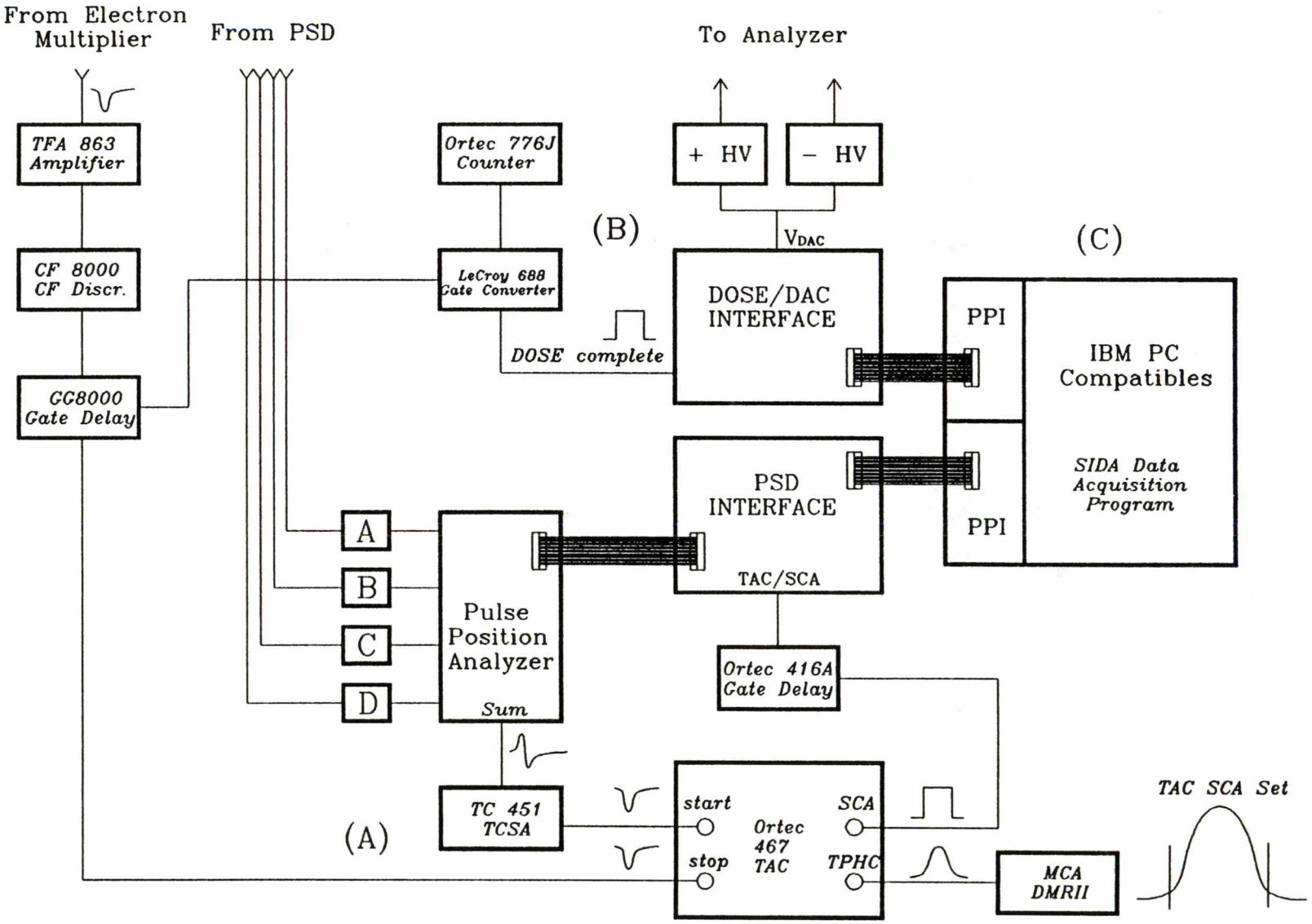
The determination of coincidences between PSD electrons and selected projectile ions is performed by Ortec Model 467 Time to Amplitude Converter/Single Channel Analyzer (TAC/SCA) NIM modules. The start pulse from the pulse position analyzer and the stop

**Figure 4.1**: Electronics setup for ion-surface coincidence experiment.

(A) A signal from the electron multiplier is amplified, discriminated, and sent to the stop input of TAC/SCA NIM module, meanwhile the anode signal from the PSD are also amplified and sent to the pulse position analyzer, which in turn sends a sum signal to the start input of TAC/SCA NIM module. These two signals provide output pulses whose amplitudes are proportional to the time difference between detection of the start and stop pulses and are displayed in TAC spectrum as shown in Figure 4.2. The carefully chosen time window represents the real coincidences which trigger positive digital SCA output to PSD interface and are recorded in coincidence spectrum as illustrated in Figure 4.3.

(B) The signal from the electron multiplier is also sent to the counter to determine the preset dose completion events. Upon receiving the dose completion signal by the dose event interface module, an interrupt signal is sent to the corresponding parallel port interface (PPI) card placed in the IBM compatible computer, which increments the input to a digital-to-analog (DAC) converter so that the programmable power supplies can provide the desired voltage to the spectrometer sectors.

(C) When an electron event from the PSD interface after the detection of a strobe pulse of the position analyzer is sent to PPI, the computer increments a register located at a memory address defined by the digital position and energy information of the event, and records the event in non-coincidence spectrum as also seen in Figure 4.3.


From Electron Multiplier
From PSD
To Analyzer
TFA 863 Amplifier
CF 8000 CF Discr.
GG8000 Gate Delay
Ortec 776J Counter
LeCroy 688 Gate Converter
(B)
+ HV
– HV
VDAC
DOSE/DAC INTERFACE
DOSE complete
(C)
PPI
IBM PC Compatibles
SIDA Data Acquisition Program
PPI
PSD INTERFACE
TAC/SCA
A
B
C
D
Pulse Position Analyzer
Sum
Ortec 416A Gate Delay
TC 451 TCSA
(A)
start
stop
Ortec 467 TAC
SCA
TPHC
MCA DMRII
TAC SCA Set

signal from the electron multiplier are inputs of this module, which in turn provides output pulses whose amplitudes are proportional to the time difference between detection of the start and stop pulses. These output pulses as seen in Figure 4.1 are the inputs to a Tennelec Nucleus DMR II Digital Multiplexer Router Unit, which display TAC spectrum obtained with PSD electrons providing the start signals and outgoing ions providing the stop signals, as illustrated in Figure 4.2. The peak corresponds to real coincidences between the PSD electrons and the selected charge states. The background counts for random coincidences. Carefully chosen time window around the peak represents the real coincidences of interest between the PSD electrons and corresponding ions, and these real coincidences within this time window trigger a positive digital SCA output to PSD interface NIM module.

The PSD position data is 8 bit $x$-address plus 8 bit $y$-address as generated by the position analyzer. In the case of PSD events, the combined 16 bit information and 12 bit digital energy information for the DAC input used for programming the spherical sector analyzer's voltages were characterized as the indices of a three- dimensional histogram. The PSD interface is wired to preset 8 bits of $y$-address to an input port, and 7 bits of $x$-address to the same input port chip except the least significant bit, which is the TAC/SCA input from Ortec Model 467 TAC/SCA NIM module. The PSD event from the data acquisition interrupt service routine interprets the least significant bits with TRUE or FALSE as coincidences and non-coincidences in the histogram spectrum in energy, respectively as shown in Figure 4.3.

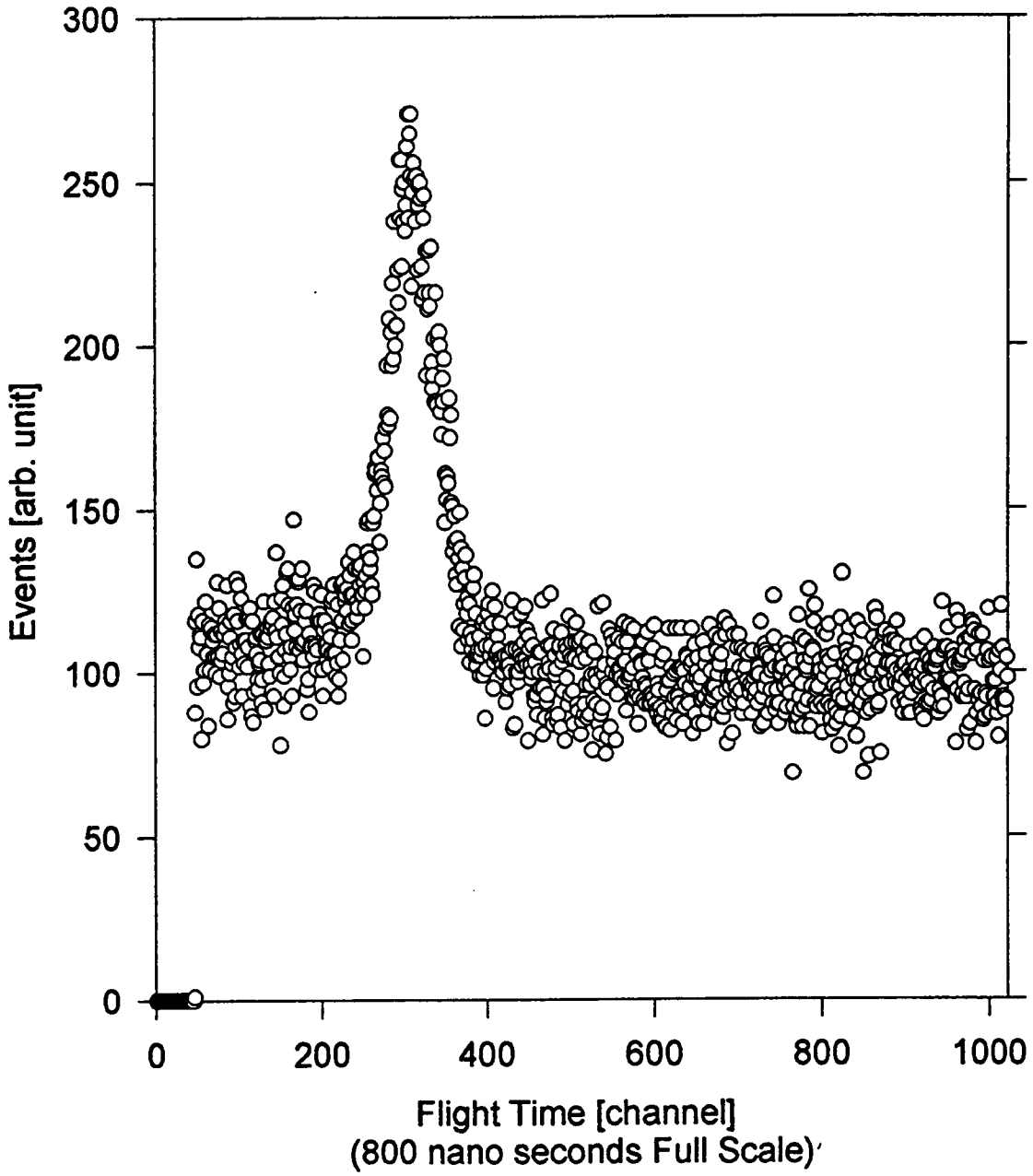



**Figure 4.2**: TAC spectrum for 0.5 MeV/u $C^{2+}$ + 20 μg/cm$^2$ carbon foil → $C^{4+}$ + $e^-$ +$e^-$.

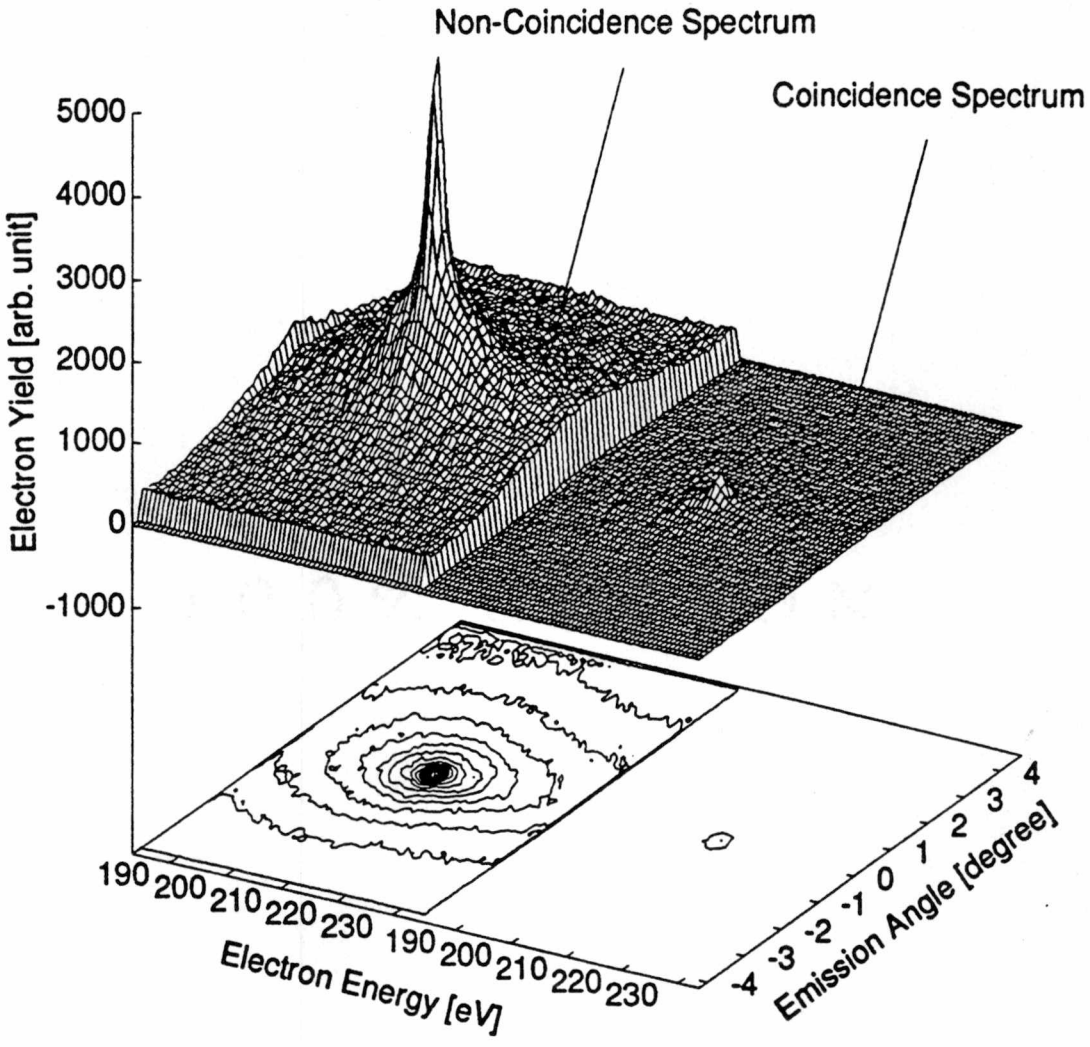



**Figure 4.3**: Coincidence and noncoincidence multichannel scanning image for 6.15 MeV $O^{5+} + Ar \rightarrow O^{6+} + e^-$

All combined information of $x$, $y$, and $E$ were defined in memory as an address where the number of events were stored. Also seen in Figure 4.4, the $x$-direction correspond to the direction of the plane perpendicular to the spherical sector analyzer's deflection plane so that the aberration of focusing can be minimized. The final precisions for $x$, $y$, and $E$ were reduced to 6 bits (64 channels) for $x$, 3 bits (8 channels) for $y$, and 7 bits (64 channels for coincidences and 64 channels for non-coincidences) for $E$ due to limited available memory.

For the dose completion events, the digital DAC output from the Dose/Scaler/DAC interface NIM module triggers the spherical sector analyzer's power supply to change the next desired deflection voltage, and data acquisition time at any particular energy channel is one second to have approximately two minutes required for a full 128 channel scan.

## IV.2 Extraction of Multipoles of Convoy Electron Emission in Ion-Gas and Ion-Foil Collisions

The doubly differential cross section of the cusp-shaped peak for electron ejection differential in both energy and angle has been qualitatively understood in terms of ELC and ECC, or collisional electron transfer to the continuum (ETC). The most responsible ion-atom process for convoy electron emission in ion-surface collision is ECC,[60] although projectile electron loss (ELC) may also contribute following capture to bound projectile states, as is known to occur in ion-solid collision.[61] Theoretical characterization of these

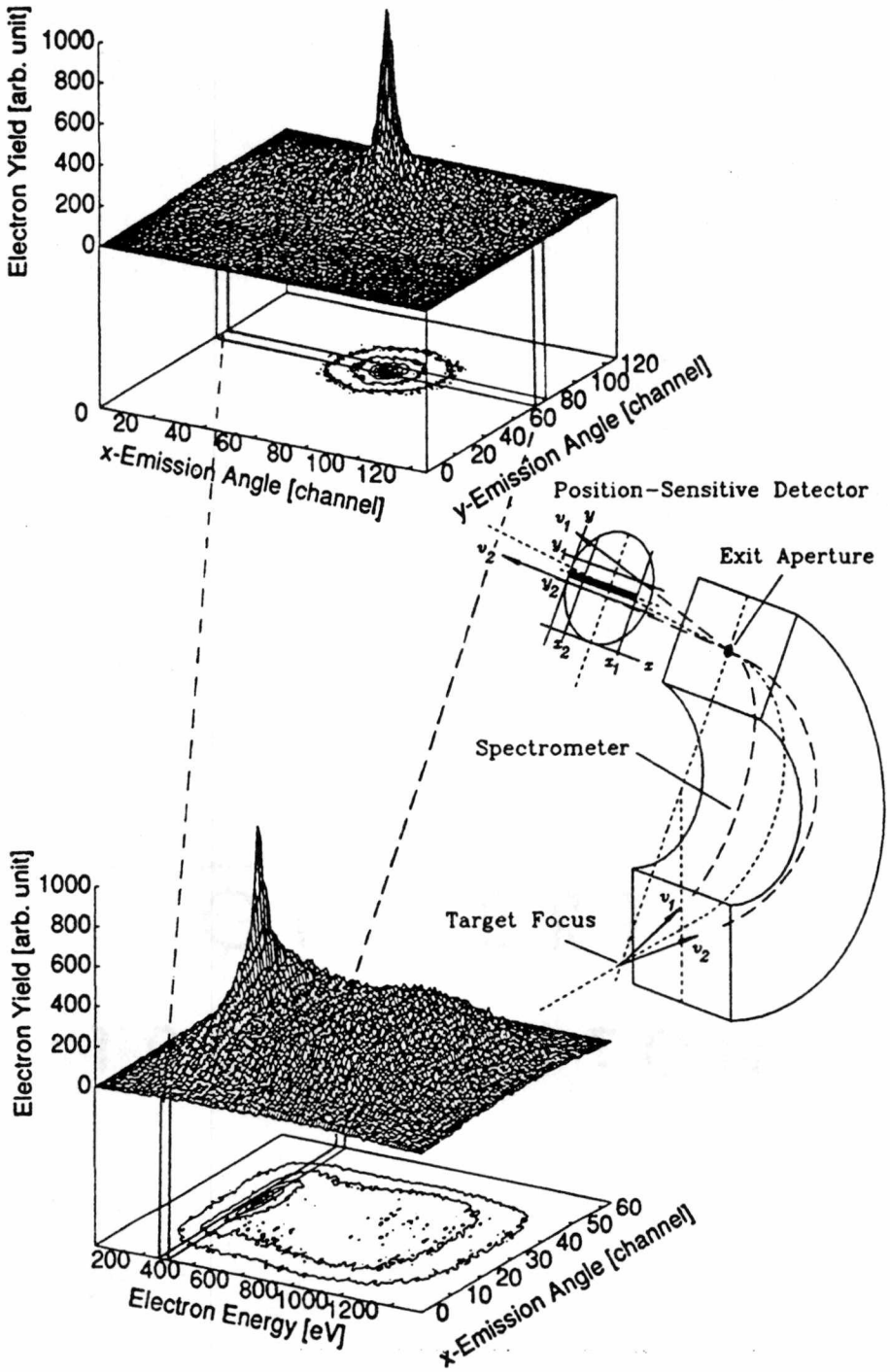



**Figure 4.4**: Display of high resolution and multi-channel scanning images from the spherical sector electrostatic analyzer for 0.5 MeV/u $C^{2+}$ + 20 μg/cm$^2$ carbon foil.

ETC processes has been established by the multipole expansion for ion-atom and ion-foil collisions.

The differential cross section of the form $\frac{1}{v_{e'}} \cdot (1 + A\cos\vartheta_{e'})$ describing ECC data with a p-wave contribution in the projectile frame had been found by Lucas *et al.*[62] followed by Briggs and Day[63], who had predicted a differential cross section of ELC with plane-wave Born approximation (PWBA) from 1s states of hydrogenic projectiles in the form of

$$\frac{d\sigma}{d\vec{v}_{e'}} = \frac{\sigma_0}{v_{e'}} \cdot (1 + \beta \cdot P_2(\cos\vartheta_{e'})) \tag{4.1}$$

where $\vec{v}_{e'} = \vec{v}_e - \vec{V}_P$ is the projectile frame electron velocity and $\vartheta_{e'}$ is the polar angle of projectile frame emission.

Both of these processes, however, have been successfully modeled by a cusp-shaped emission distribution in the lab frame which results from a projectile-frame model of the form

$$\frac{d\sigma}{d\vec{v}_{e'}} = \frac{\sigma_0}{v_{e'}} \cdot \sum_k \beta_k(V_P) \cdot P_k(\cos\vartheta_{e'}). \tag{4.2}$$

It was first introduced by Meckbach *et al.*[64] to quantitatively parameterize the observed ECC spectrum asymmetry.

While the above model applies for $v_{e'} << V_p$, it is more useful for the experimental analysis to expand $\beta_k$ in a power series in $v_{e'}$; $\beta_k = \sum_{j=0,1} v_{e'}^j \cdot \beta_{jk}$, $(\beta_k = \beta_{0k} + < \vec{v}_{e'} > \cdot \beta_{1k})$, resulting in

$$\frac{d\sigma}{d\vec{v}_{e'}} = \frac{\sigma_0}{v_{e'}} \cdot \sum_{jk} \beta_{jk} \cdot v_{e'}^j \cdot P_k(\cos\vartheta_{e'}) \tag{4.3}$$

To compare DDCS with the measured distributions $Q(E_e, \theta_e)$ of ejected electrons, the product of the spectrometer acceptance function[65] $S(E_e, E_e', \Omega_e, \Omega_e')$ and the cross section

$v_e \cdot \frac{d\sigma}{dv_e} = \frac{d^2\sigma}{dE_e d\Omega_e}$ for the electrostatic analyzer is integrated over the experimental acceptances in angle and energy. The transmission function $S(E_e, E'_e, \Omega_e, \Omega'_e)$ is approximated as being separable in energy and angle by earlier work;[21] $S(E_e, E'_e, \Omega_e, \Omega'_e) = R(E_e, E'_e) \cdot G(\Omega_e, \Omega'_e)$, resulting in the form of

$$Q(E_e, \theta_e) = \int_{\Delta\Omega_e} d\Omega'_e \cdot G(\Omega_e, \Omega'_e) \cdot \int_{\Delta E_e} dE'_e \cdot R(E_e, E'_e) \cdot \frac{d^2\sigma}{dE'_e d\Omega'_e}(E'_e, \Omega'_e) \qquad (4.4)$$

in the lab frame where $G(\Omega_e, \Omega'_e)$ and $R(E_e, E'_e)$ represent the angle and the energy resolution functions, respectively.

The energy resolution function $R(E_e)$, which is used to characterize the transmission of the spectrum for a fixed internal deflection field, is well approximated by a triangular or trapezoidal line shape specified by a FWHM proportional to the resolution $\frac{\Delta E}{E}$ and a TW proportional to the difference in entrance and exit apertures of the spectrometer as mentioned previously.

The rectangular approximation is used to determine the relative angular transmission function $G(\theta_e)$ for the laboratory azimuthal angle from the geometrical arrangement of limiting apertures. If all electrons are uniformly transported from the spectrometer through the exit aperture with a particular angle with respect to the central direction, they will form a uniform spot on the PSD, the same size of the exit aperture. With this assumption we can infer the angular resolution function of a "pillbox" shape, i.e. unity over an area of the approximate size of the exit aperture and zero elsewhere. Therefore, the angular uncertainty becomes $\delta\theta = \frac{a}{l}$ where $l$ is the drift distance between the exit aperture and the PSD, and $a$ is the diameter of the exit aperture.

However, the effect of the fringe fields and spectrometer angular aberrations due to interdependence of a polar angle and an azimuthal angle was expected so that the employment of a "trapezoidal pillbox" shape was suggested to correct this problem as mentioned in the section III.10. Obviously the calculation of multipole moments was proven to be very time dependent on the shape of the applied angular resolution function. Also the difference in reduced $\chi^2_\nu$ between results with "pillbox" and "trapezoidal pillbox" shape functions was negligible, at least, less than 5 %. Thus, the "pillbox" shape angular resolution function can be regarded as a good approximation. Detailed description of the fitting procedure will be seen in Appendices.

## IV.3 Measurement of Convoy Electron Emission in Ion-Foil and Ion-Surface Collisions

Figure 4.5 has an image produced by 0.3 MeV/u $C^{2+}$ ions with 20 $\mu g/cm^2$ carbon foil positioned normal to the incident ion beam at the analyzer entrance focus. The contours in the doubly differential electron emission were plotted in the rectangular position coordinates of the polar angle, $\theta_X$ ($x$-emission angle) and *Energy*. The contours around the cusp peak show the interference between binary encounter electrons and convoy electrons, which is in turn a consequence of the low collision energy in the present work. Therefore, the most important function of the foil results and the foil data is to serve as a "marker" or "calibration source" for the surface data. Having the foil present in the experimental chamber gives us a way in the surface experiment to know what a deeply penetrating

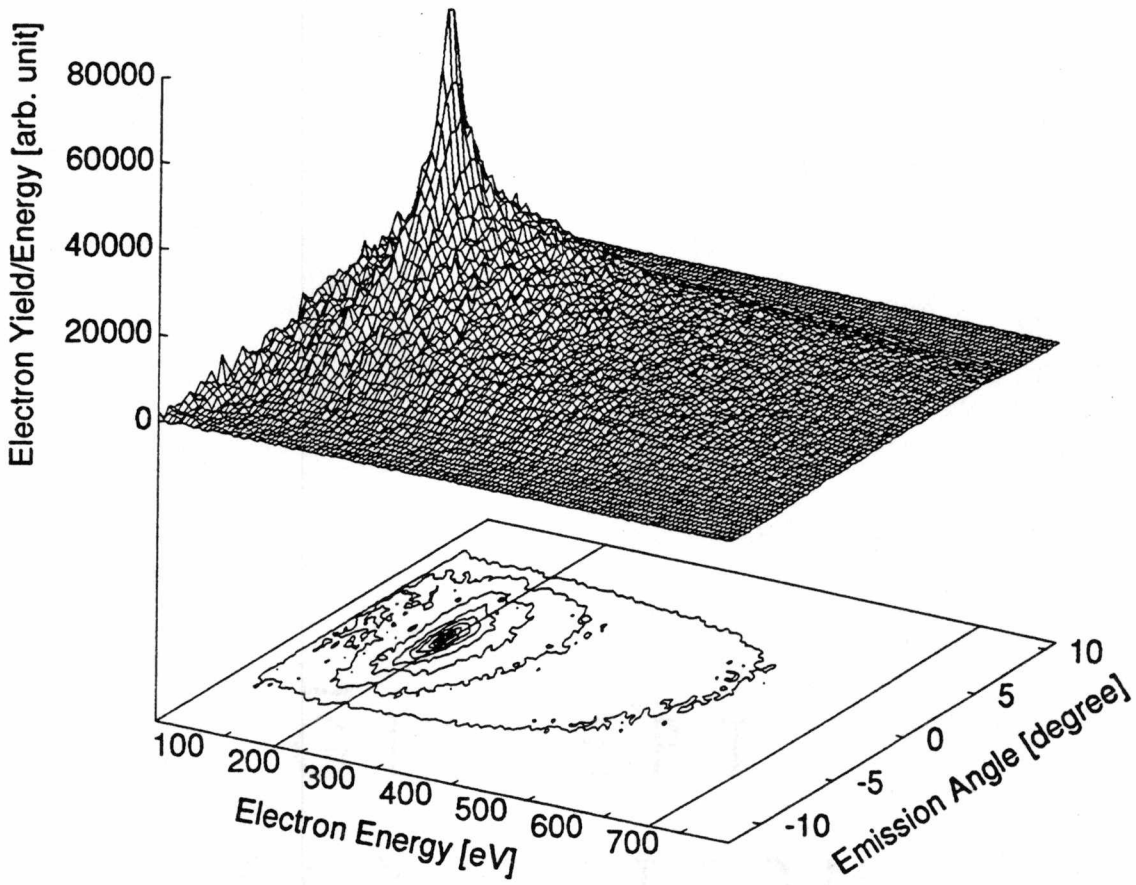


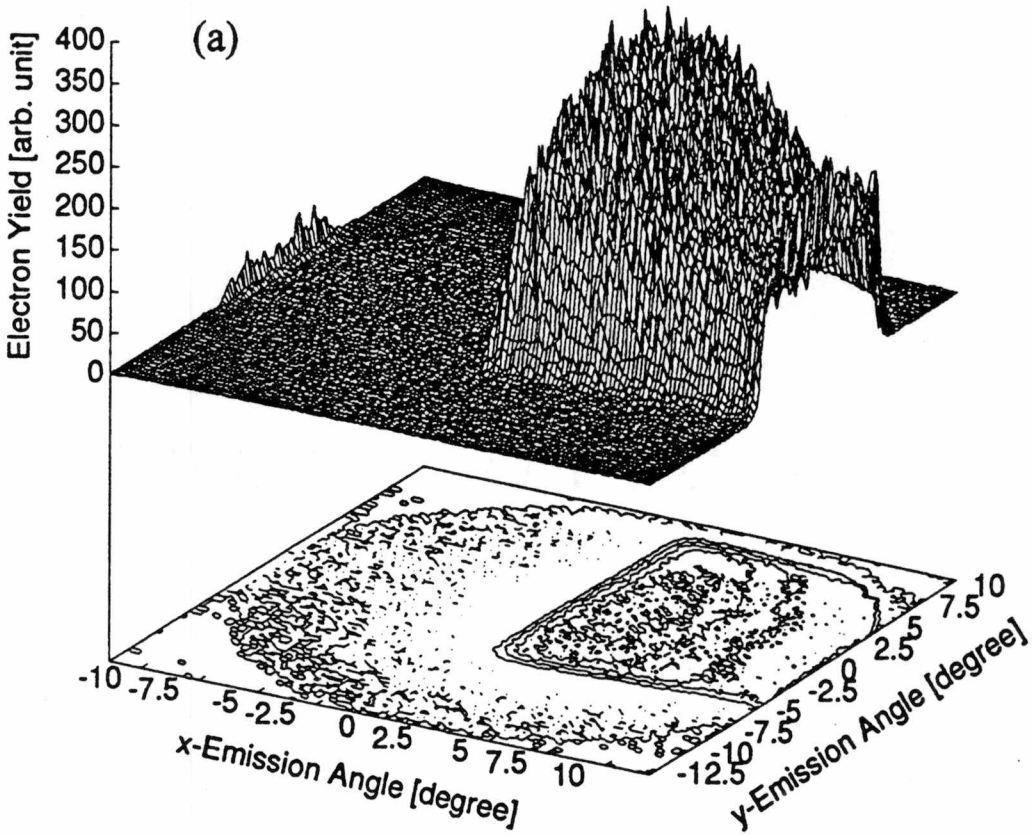

**Figure 4.5**: Doubly differential distribution of electrons emitted from 0.3 MeV/u $C^{2+}$ incident on 20 $\mu g/cm^2$ carbon foil.

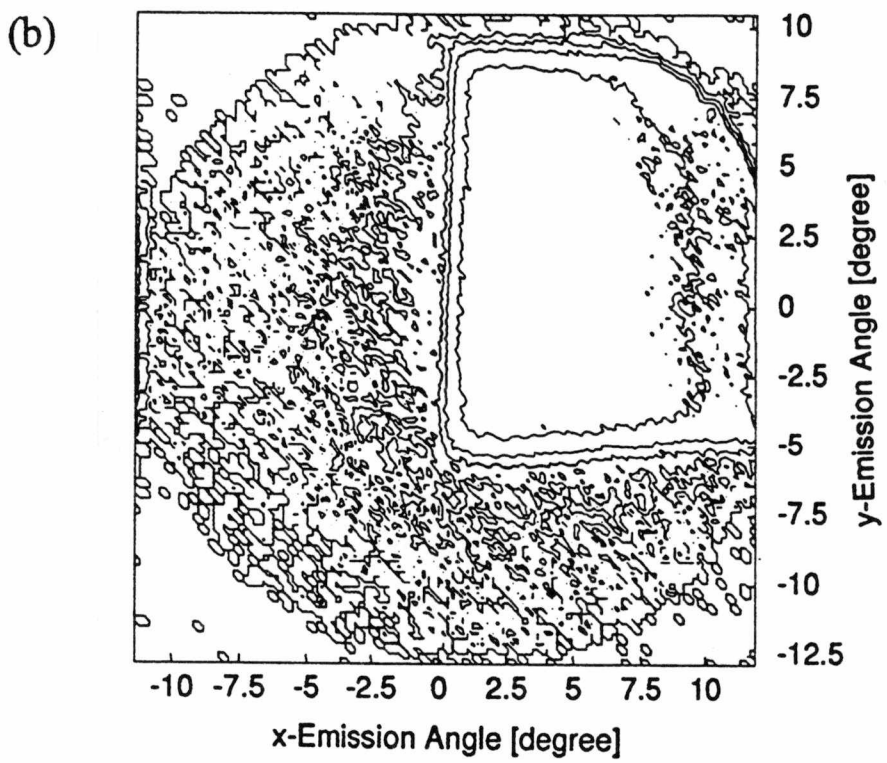
collision produces for emission. It also gives us a well-understood point in the energy scale, and allows us to verify that azimuthally isotropic emission looks isotropic (which among other things gives a measure of stray magnetic fields).

The convoy electron peak energy isotachic to carbon ions with a kinetic energy of 0.3 MeV/u was 164.6 eV. However, the experimentally observed peak energy was 157 eV as a consequence of two phenomena: (1) The mean attenuation length in a carbon foil of ~160 eV electrons is much less than the thickness of the foil target used. As a result, the observed emission is of convoy electrons produced in a ~1nm thick "last layer". (2) Electronic stopping power or energy loss produces a deceleration of the ions responsible for convoy electron production in this last layer of the foil by ~13.2 keV/u.[66] The observed peak energy is thus, within the analyzer resolution and precision of the accelerator energy (typically 1 part in $10^3$), in agreement with the expected value of 157.4 eV. Convoy peak energy measurements obtained with foil targets were used throughout the experiments as a double-check of the projectile ion energy.

Plots in Figure 4.6 display a typical 2-dimensional ($\theta_x$ and $\theta_y$) high resolution image in (a) and its contour plot of log scale intensity in (b) of convoy electron emission produced by 0.3 MeV/u $C^+$ ions incident on Si(100) surface with 0.1° grazing angle. This spectrum was obtained with the analyzer positioned at ~5° with respect to the incident ion direction and tuned to a pass energy of 246 eV, which is the maximum probable electron energy and determined by a preliminary scanning image spectrum. *Y*-emission (azimuthal) angle ranges in channel are carefully selected to minimize the aberration

**Figure 4.6**: High resolution image (a) and corresponding log scale contour (b) plots for 0.3 MeV/u $C^{1+}$ incident on Si(100) at a grazing angle of 0.1° with the preset cusp energy of 246 eV. Contours in the upper image are equally spaced intensities between zero and maximum count rate.


(a)
Electron Yield [arb. unit]
400
350
300
250
200
150
100
50
0
-10 -7.5 -5 -2.5 0 2.5 5 7.5 10
x-Emission Angle [degree]
-12.5 -10 -7.5 -5 -2.5 0 2.5 5 7.5 10
y-Emission Angle [degree]


(b)
10
7.5
5
2.5
0
-2.5
-5
-7.5
-10
-12.5
y-Emission Angle [degree]
-10 -7.5 -5 -2.5 0 2.5 5 7.5 10
x-Emission Angle [degree]

effect of the analyzer and to store the desired spectrum in a 3-dimensional histogram. For ion-surface collisions 64 channels were selected in *y*-emission angle, ~12°, to hold all electrons in 8 encoded emission channels in *y*-direction of the histogram. For this study the term 'polar angle' means the angle between specularly reflected ion and ejected electron, projected on the plane defined by the surface normal and the incoming ion trajectory; whereas 'azimuthal angle' is the angle between specularly reflected ion and ejected electron projected onto the surface.

Spectrum in the previous Figure 3.6 for 0.5 MeV/u $C^{3+}$ beam incident on Si(100) at a grazing angle of 0.2° shows very different characteristics from ion-atom or ion-foil collisions. One of the most striking characteristics is the significantly broadened peak of convoy electrons in angle and energy, and systematic analysis will be given in the next chapter.

# V. RESULTS AND DISCUSSION

## V.1 Introduction

Ejected electron distributions near the forward direction from collisions involving fast ions are often well described in terms of multipole moments of the ejected charge distribution, which permits direct comparison with theory for ion-atom and ion-foil collisions. An example of such a comparison will be presented in this chapter for the collision system of $O^{5+}$ + He, Ar targets. In the much more complicated process of convoy electron emission from ion-surface interactions we are far away from an analytical description of the convoy peak. The most advanced theoretical work to date seems to be that of Reinhold *et al.*,[60] which is a numerical simulation that implicates rainbow scattering as an important mechanism determining the shape of the convoy electron peak formed near surfaces. These calculations are very time-intensive and not directly scaleable to other collision systems so that the need arises to find simplified functions to describe a broader group of grazing angle ion-surface collision systems. Therefore, more empirical formulas were chosen, such as superpositions of Gaussian and Lorentzian functions, for the analysis of convoy electrons from surfaces. With these functions it was possible to describe dependences of physical parameters of convoy and binary electron distributions. Not all parameters of these functions have a physical meaning directly deductible as in the case of the parameters of multipole moments used to describe ion-atom collisions. However, some

parameters and especially their behavior with systematic changes of physical properties of the collision process can yield further insights into the process involved in the collision.

The focus of this study is also on the dynamics of inelastic atomic collisions with the surface, i.e. those which are associated with a change of the charge state and the excitation state of the atomic projectile. The energy range for the collision system is limited since the scattering processes that occur above the surface or near the surface is the main interesting phenomenon. By this reason we selected the normal component $E_{\perp}$ ($= E_0 \sin^2\theta$) less than or order of 100 eV of the incident particles to reduce contributions from the bulk due to surface penetrating trajectories. True grazing collisions, with a minimum of surface-layer penetration, are evident in the spectrum obtained from collisions of $C^{3+}$ ions with a silicon (100) surface shown in Figure 3.6, and are compared with the emission observed under conditions expected to result in substantial surface penetration. In the lower plot, which is the grazing incidence case, the incident projectile energy of 0.5 MeV/u with a surface-incidence angle of 0.2° results in a perpendicular kinetic energy of only ~70 eV, whereas in the upper plot, penetrating case of 1.0 MeV/u ions incident at 1.0°, the perpendicular energy is ~3.5 keV. It is seen that penetrating collisions produce a clearly defined cusp, similar to that observed with the same apparatus from ion solid-foil collisions, as well as a prominent binary encounter electron peak.

## V.2 Ion-Atom and Ion-Foil Collisions

The domination by strong "cusp" peak electrons, with relative vector velocities $|\mathbf{v}_{e'}| = |\mathbf{v}_e - \mathbf{v}_p| \ll 1$ au in the projectile frame, is a distinct signature in the forward electron spectrum for ion-atom collisions. Good *ab-initio* theories have been achieved for the emission of convoy electrons in gas-phase collision systems.

As mentioned in section IV.3, the multipole expansion of the doubly-differential cross section gives multipole moments which are characteristic of the ejected charge distribution for any particular collision system. The skewing of cusp peaks toward low emission energy, which is characteristic of ECC and is described by a strong negative dipole moment, $\beta_1$, has been explained by the inclusion of second-order terms in the Born approximation,[67] and is a consequence of the two-step Thomas capture mechanism. In contrast, the transversely polarized electron emission distribution which is representative of ELC processes is characterized by even order multipole moments such as quadrupole ($\beta_2$) and hexadecapole ($\beta_4$). This projectile ionization process (ELC) was described in the first-order Born approximation by strong quadrupole and hexadecapole moments of opposite signs, producing emission which is predominantly transverse in the projectile frame.

Burgdörfer *et al.*[18] calculated the maximum multipolarity, $k \leq 2n$, where $n$ is the principal quantum number of the initial electronic orbit, and predicted only even $k$ for ELC process in a Born approximation calculation when electrons of a given $n$ are lost. A projectile velocity range from 2 to 10 au for the collision of $He^+$ + H was chosen in their calculation of the 2nd and the 4th order anisotropic moments. In order to compare

experimentally obtained multipole moments with the theory, one existing result and one existing data set, which have the same collision systems but different projectile velocity ranges, were introduced here. For previously measured ELC ion-atom collision systems in coincidence, $O^{5+}(n=2)+Ar(He) \rightarrow O^{6+}+e^{-}$, the predicted electron distribution in the first Born approximation has the form with $k \leq 2n$

$$\frac{d\sigma}{dv} = (\frac{\sigma}{v}) \cdot [1 + \beta_2 P_2(\cos\theta) + \beta_4 P_4(\cos\theta)]. \qquad (5.1)$$

Plots in Figure 5.1 are the doubly-differential intensities in angle and energy with 4 different projectile velocities, and plots in Figure 5.2 show the fitting results (appendix A for the fitting function) in 3-dimensional perspective for 1.41 MeV/u $O^{5+}$ + He. The reduced $\chi^2_v$ for this case is about 1.25, and the goodness of the fit is visible in the 10% intensity contour and the residue plots. The reasonable fitting range had to be chosen to fit, $\Delta v_{\parallel} \sim \Delta v_{\perp} \sim$ 0.5 au, in order to maintain a large contribution in reduced $\chi^2_v$ by the singularity convoluted with the resolution functions and to extract the corresponding angular information. Initially, the fitting process was tested with only the first order radial velocity component terms ($j$=0), but it was found necessary to add the first order velocity terms ($j$=1). Also, $k$ = 1, 3, 5 terms were added to monitor theoretically disallowed terms of the first Born approximation, and were found to be contributing substantially to reductions in $\chi^2_v$ for the smaller ion velocities considered in this study.

Table 5.1 shows the previously obtained multipole coefficient $\beta_{jk}$s for the ELC data with 10.1, 14.3, and 16.2 au projectile velocities,[68,69] whereas Tables 5.2 and 5.3 present the recently obtained multipole moments for the ELC data with 3.9, 6.0, 7.5, and 8.7 au projectile velocities. To compare the result in this velocity range from 3.9 to 16.2 au ($O^{5+}$)

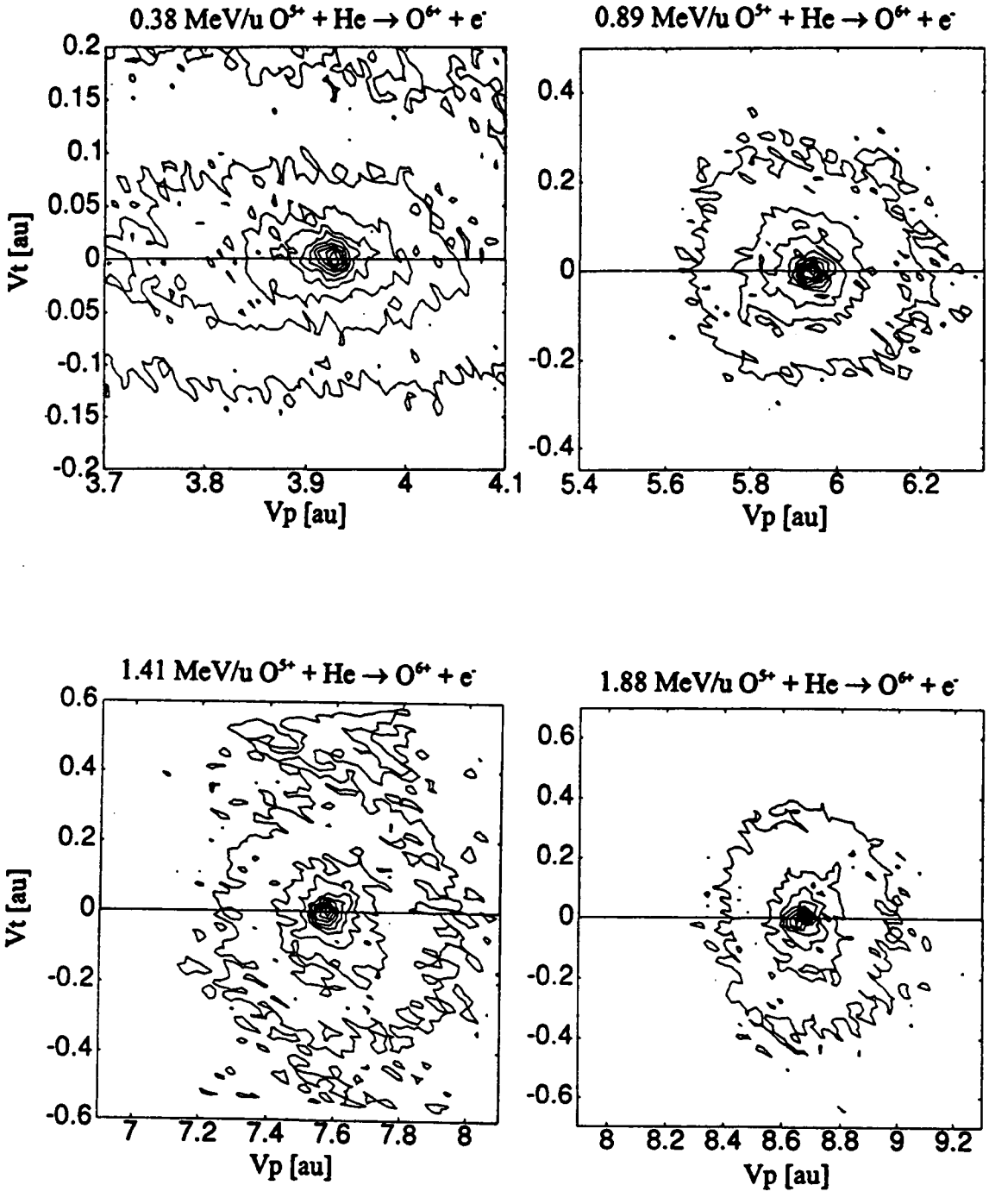



**Figure 5.1**: Contour plots for electron emission for ELC produced in ion-atom collisions.

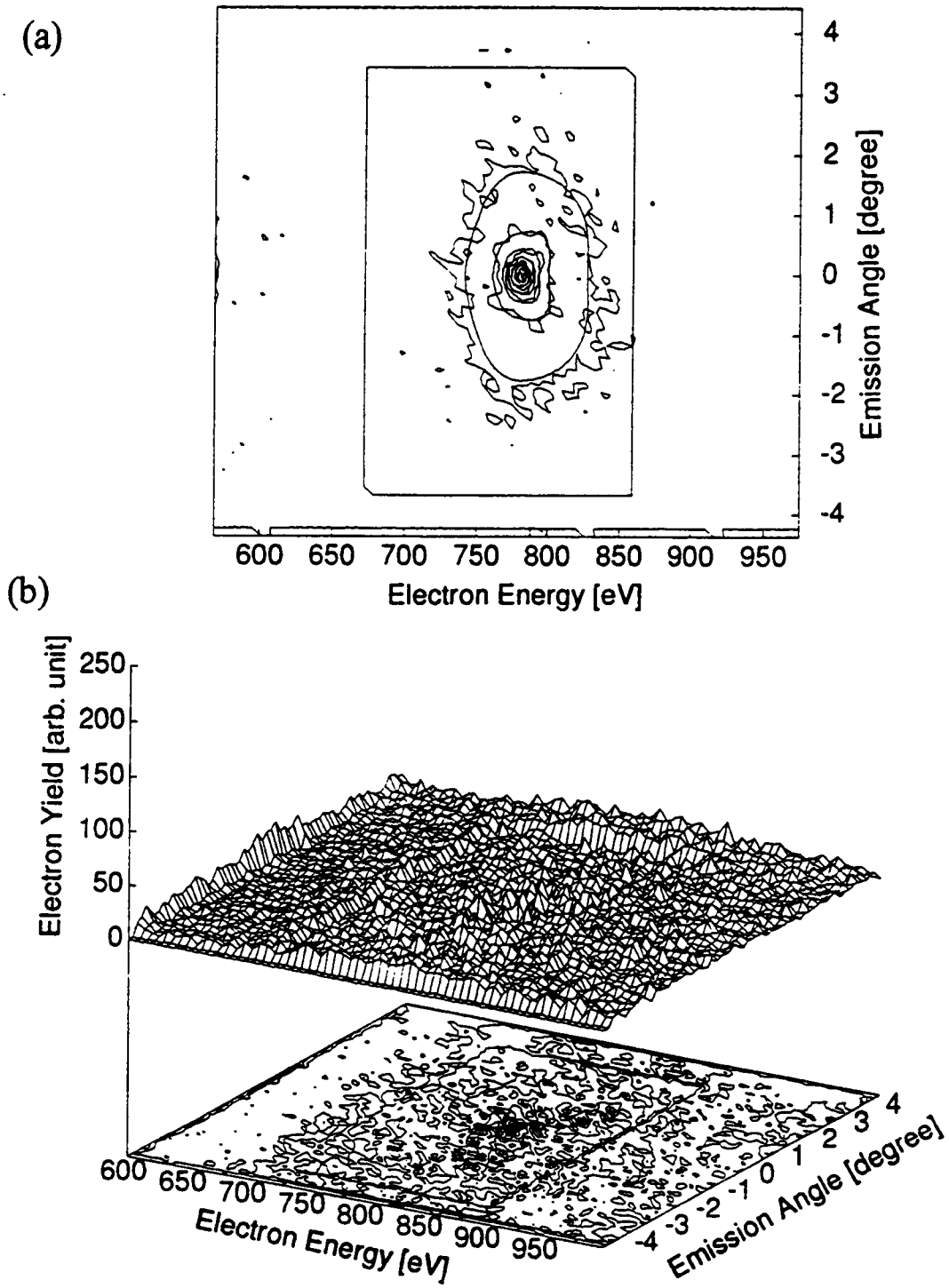



**Figure 5.2**: Result of the fitting procedure for 1.41 MeV/u $O^{5+}$ + He → $O^{6+}$ + $e^-$. The plot (a) shows the source and the fitted spectra in the same contour intensities, meanwhile the plot (b) illustrates the residue plot, which is the source data minus the fitted data.

Table 5.1

# Multipole Moments for $O^{5+}$ + He (Ar)

| Vcusp (au) | 10.1 | 14.3 | 16.2 |
|---|---|---|---|
| Ecusp (eV) | 1405.7 | 2811.3 | 3599.8 |
| TARGET | Argon : Helium | Argon : Helium | Argon : Helium |
| $\chi^2_\nu$ | 1.55 : 1.85 | 1.64 : 1.83 | 1.74 : 2.43 |
| BETA01 | -0.08 : 0.03 | -0.08 : 0.18 | 0.02 : 0.28 |
| BETA02 | -0.26 : -0.23 | -0.62 : -0.43 | -0.68 : -0.23 |
| BETA03 | -0.04 : 0.00 | -0.01 : -0.14 | -0.05 : -0.16 |
| BETA04 | 0.05 : -0.04 | 0.12 : 0.06 | 0.16 : -0.07 |
| BETA10 | 0.80 : 0.56 | -0.21 : 0.26 | -0.1 : 1.6 |
| BETA11 | -0.69 : -0.09 | -0.30 : -0.47 | -0.44 : -0.87 |
| BETA12 | -0.41 : -0.45 | 0.31 : -0.51 | -0.28 : -0.52 |
| BETA13 | 0.31 : 0.00 | 0.18 : 0.34 | 0.28 : 0.26 |
| BETA14 | -0.31 : 0.22 | 0.06 : 0.24 | -0.23 : 0.01 |
| Top Width (%) | 0.32 : 0.00 | 0.03 : 0.0 | 0.44 : 0.11 |
| FWHM (%) | 1.56 : 0.99 | 0.97 : 0.96 | 0.92 : 0.62 |
| Angular Res.(rad) | 0.007 : 0.0061 | 0.0052 : 0.0061 | 0.0049 : 0.0056 |
| TilT (%) | 1.6 : 2.4 | 0.82 : 2.3 | 1.3 : 2.3 |
| DOF | 973 : 958 | 2726 : 958 | 2718 : 1646 |

* Adapted from S.D. Berry[69]

Table 5.2

# Multipole Moments for $O^{5+} + He \rightarrow O^{6+} + e^-$

| Vcusp (au) | 3.9 | 6.0 | 7.5 | 8.7 |
|---|---|---|---|---|
| Ecusp (eV) | 210.85 | 489.58 | 771.39 | 1028.52 |
| $\chi^2_\nu$ | 1.634 | 1.349 | 1.247 | 1.505 |
| BETA01 | -0.028 (0.001) | -0.195 (0.003) | 0.109 (0.001) | 0.045 (0.001) |
| BETA02 | 0.308(0.001) | 0.105 (0.003) | 0.093 (0.001) | -0.164 (0.001) |
| BETA03 | -0.005 (0.001) | -0.105 (0.004) | -0.155 (0.001) | -0.134 (0.002) |
| BETA04 | -0.055 (0.002) | -0.175 (0.004) | -0.222 (0.001) | -0.157 (0.002) |
| BETA05 | 0.122 (0.002) | 0.135 (0.005) | -0.119 (0.001) | 0.064 (0.002) |
| BETA11 | -1.171 (0.004) | -0.041 (0.009) | 0.082 (0.001) | 0.002 (0.003) |
| BETA12 | 2.568 (0.005) | 0.313 (0.012) | -0.715 (0.001) | 0.123 (0.004) |
| BETA13 | -0.549 (0.006) | 0.436 (0.014) | 0.679(0.001) | 0.433 (0.004) |
| BETA14 | 1.465 (0.006) | 0.699 (0.016) | 0.826 (0.001) | 0.452 (0.004) |
| BETA15 | -0.950 (0.007) | -0.614 (0.018) | 0.237 (0.001) | -0.215 (0.005) |
| Top Width (%) | 0.190 (0.002) | 0.036 (0.002) | 0.224 (0.001) | 0.001 (0.001) |
| FWHM (%) | 1.083 (0.001) | 1.000 (0.001) | 1.048 (0.001) | 0.979 (0.001) |
| Angle Res.(rad) | 0.024(0.0001) | 0.012 (0.0001) | 0.006 (0.0001) | 0.009 (0.0001) |
| TilT (%) | 2.066 (0.003) | 0.548 (0.012) | -0.068 (0.001) | -0.110 (0.005) |
| Angle Step(rad) | 0.002438 | 0.002438 | 0.002438 | 0.002438 |
| DOF | 2782 | 1582 | 1382 | 1582 |

Table 5.3

# Multipole Moments for $O^{5+} + Ar \rightarrow O^{6+} + e^-$

| Vcusp (au) | 3.9 | 6.0 | 7.5 | 8.7 |
|---|---|---|---|---|
| Ecusp (eV) | 210.85 | 489.58 | 771.39 | 1028.52 |
| $\chi^2_\nu$ | 1.787 | 1.436 | 1.246 | 1.530 |
| BETA01 | -0.228 (0.001) | -0.355 (0.002) | -0.158 (0.003) | -0.191 (0.005) |
| BETA02 | 0.381(0.001) | 0.007 (0.003) | 0.041 (0.004) | -0.333 (0.007) |
| BETA03 | -0.059 (0.001) | -0.008 (0.003) | 0.002 (0.004) | -0.088 (0.008) |
| BETA04 | -0.039 (0.001) | -0.149 (0.004) | -0.145 (0.005) | -0.083 (0.009) |
| BETA05 | 0.149 (0.002) | 0.065 (0.005) | -0.030 (0.006) | -0.018 (0.011) |
| BETA11 | -0.874 (0.003) | -0.255 (0.008) | -0.310 (0.007) | -0.021 (0.015) |
| BETA12 | 1.221 (0.004) | 0.136 (0.011) | -1.093 (0.010) | 0.340 (0.021) |
| BETA13 | 0.155 (0.005) | 0.292 (0.012) | 0.371(0.012) | 0.533 (0.023) |
| BETA14 | 1.190 (0.005) | 0.800 (0.014) | 0.694 (0.013) | 0.567 (0.025) |
| BETA15 | -1.129 (0.006) | -0.412 (0.017) | -0.009 (0.015) | -0.114 (0.030) |
| Top Width (%) | 0.199 (0.001) | 0.078 (0.002) | 0.046 (0.006) | 0.097 (0.002) |
| FWHM (%) | 1.092 (0.001) | 1.000 (0.002) | 0.950 (0.004) | 0.978 (0.002) |
| Angle Res.(rad) | 0.026 (0.0001) | 0.021 (0.0001) | 0.019 (0.0001) | 0.008 (0.0001) |
| TilT (%) | 1.881 (0.002) | 0.352 (0.009) | 0.277 (0.014) | 0.426 (0.025) |
| Angle Step(rad) | 0.002438 | 0.002438 | 0.002438 | 0.002438 |
| DOF | 2782 | 1382 | 1382 | 1582 |

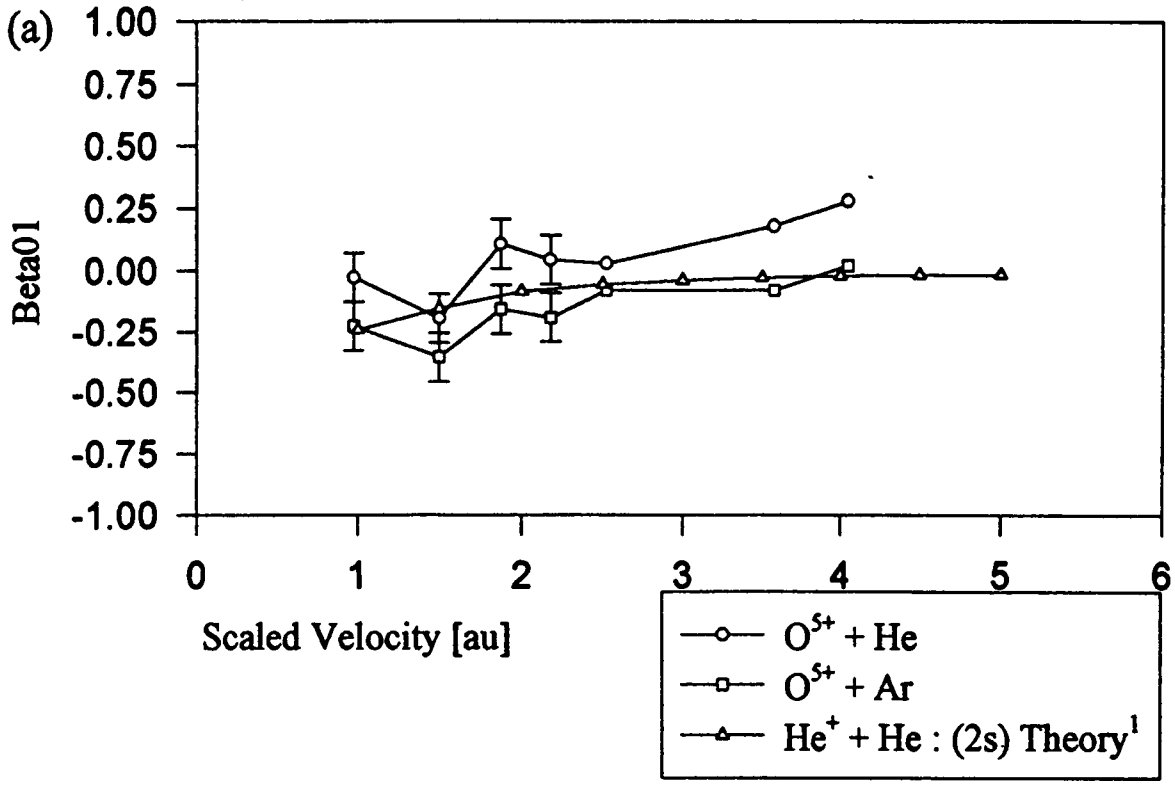
with that of theoretical calculation in the range from 2 to 10 au ($He^+$) the scaled velocity is introduced here.

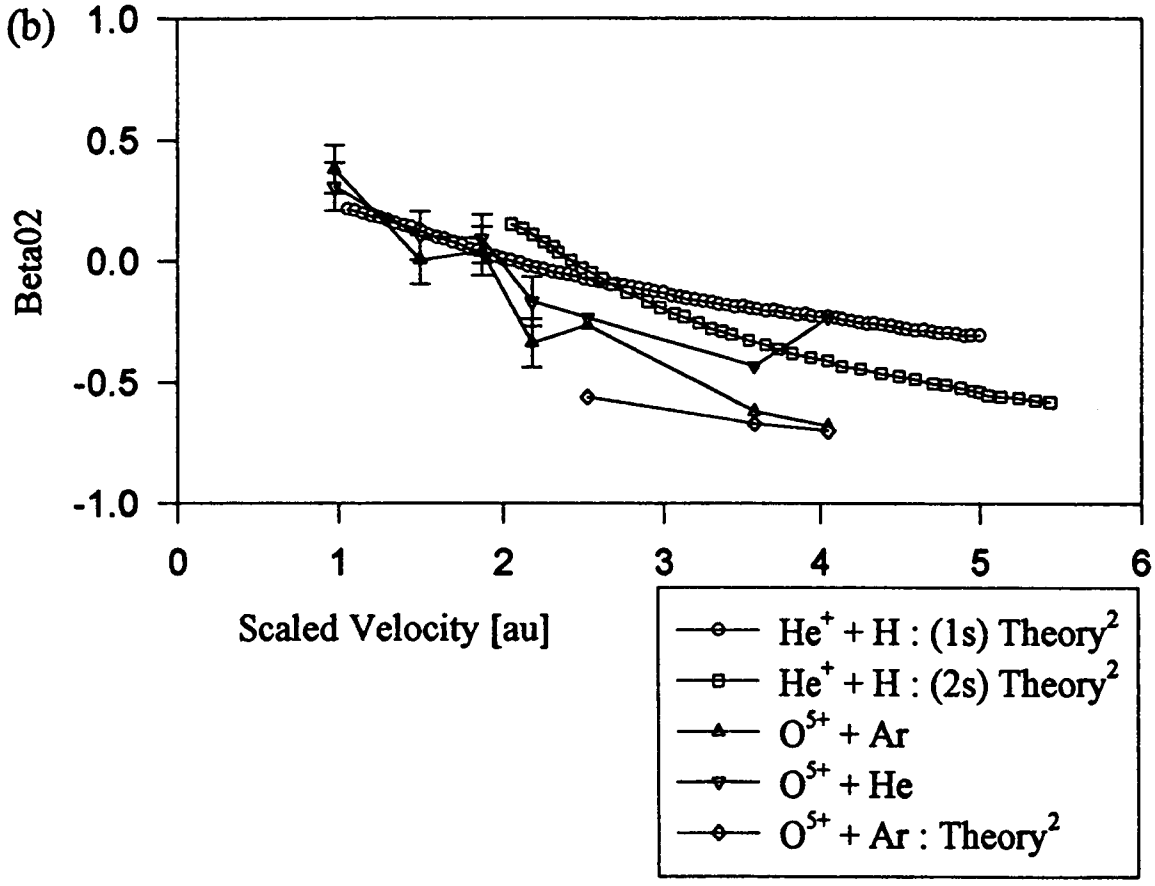
The characteristic velocity of an electron with principal quantum number $n$, $v_n$, is given by $v_n \sim v_0 Z/n$, where $v_0$ is the ground state Bohr velocity, so in au, $v_n \sim Z/n$. If ELC from the $n$ shell of a projectile ion $Z$ is considered, then the criterion for the first Born approximation is that $V_P >> v_n$, or when $V_P >> Z/n$. This happens for the scaled velocity $v_s = V_P \cdot (n/Z) >> 1$. The relative importance of higher-order Born terms will be determined by the size of $v_s$ relative to 1. Similarly, ELC data from different projectiles as in this study can be compared by considering them at the same scaled velocity, because what is important to the collision dynamics is the collision velocity relative to the active electron's characteristic velocity.

Therefore, from this perspective, the range of velocities for $O^{5+}$ which is analogous to the range 2 au to 10 au of $He^+$ ions is that which gives the same scaled velocities. ELC from $He^+$ means ELC from $n$=1 and He obviously has $Z$=2 so $v_s = (1/2) \cdot V_P$ (for ELC from $He^+$). ELC from $O^{5+}$ ($Z$=8), however, is dominated by loss of the outermost, loosely bound $n$=2 electron, thus $v_s = (2/8) \cdot V_P$ (for ELC from $O^{5+}$).

Experimental as well as theoretical curves in Figures 5.3 and 5.4 for $\beta_{01}$, $\beta_{02}$, and $\beta_{04}$ of a 2s electron (analogous to the $O^{5+}$ outer electron which dominates ELC) are plotted in terms of the scaled velocity mentioned above. The theoretical curve for $\beta_{02}$ of 2s electron in Figure 5.3 (b) changes sign at $v_s \sim$ 2.5. Qualitatively similar effect is also seen in experimental curves (Ar and He targets): $\beta_{02}$ changes sign in the same sense at around $v_s$= 2 au. The agreement is quite good considering that not only are the projectiles different,

**Figure 5.3**: $\beta_{01}$ (a) and $\beta_{02}$ (b) coefficients for ELC of $O^{5+}$ on Ar (He), $He^{+}$ on H ,and $He^{+}$ on He as a function of scaled velocity $v_s$. 1 au $\leq v_s \leq$ 5 au for $He^{+}$ (ELC) means 2 au $\leq V_P \leq$ 10 au. For $O^{5+}$, the analogous velocity range is 4 au $\leq V_P \leq$ 20 au. Experimental Data without error bars: adapted from S. D. Berry, *Multipole Characterization of the Electron Loss to the Continuum Cross Section at Asymptotically High Velocities*, Ph. D. Dissertation, University of Tennessee, 1985. Theory[1] : adapted from D. H. Jakubaßa-Amundsen, J. Phys. B: At. Mol. Opt. Phys. 23 (1990) 3335. Theory[2] : adapted from J. Burdörfer, M. Breinig, S. B. Elston, I. A. Sellin, Phys. Rev. A**28**, 3277 (1983).


(a)
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
-1.00
Beta01
0
1
2
3
4
5
6
Scaled Velocity [au]
O5+ + He
O5+ + Ar
He+ + He : (2s) Theory1


(b)
1.0
0.5
0.0
-0.5
-1.0
Beta02
0
1
2
3
4
5
6
Scaled Velocity [au]
He+ + H : (1s) Theory2
He+ + H : (2s) Theory2
O5+ + Ar
O5+ + He
O5+ + Ar : Theory2

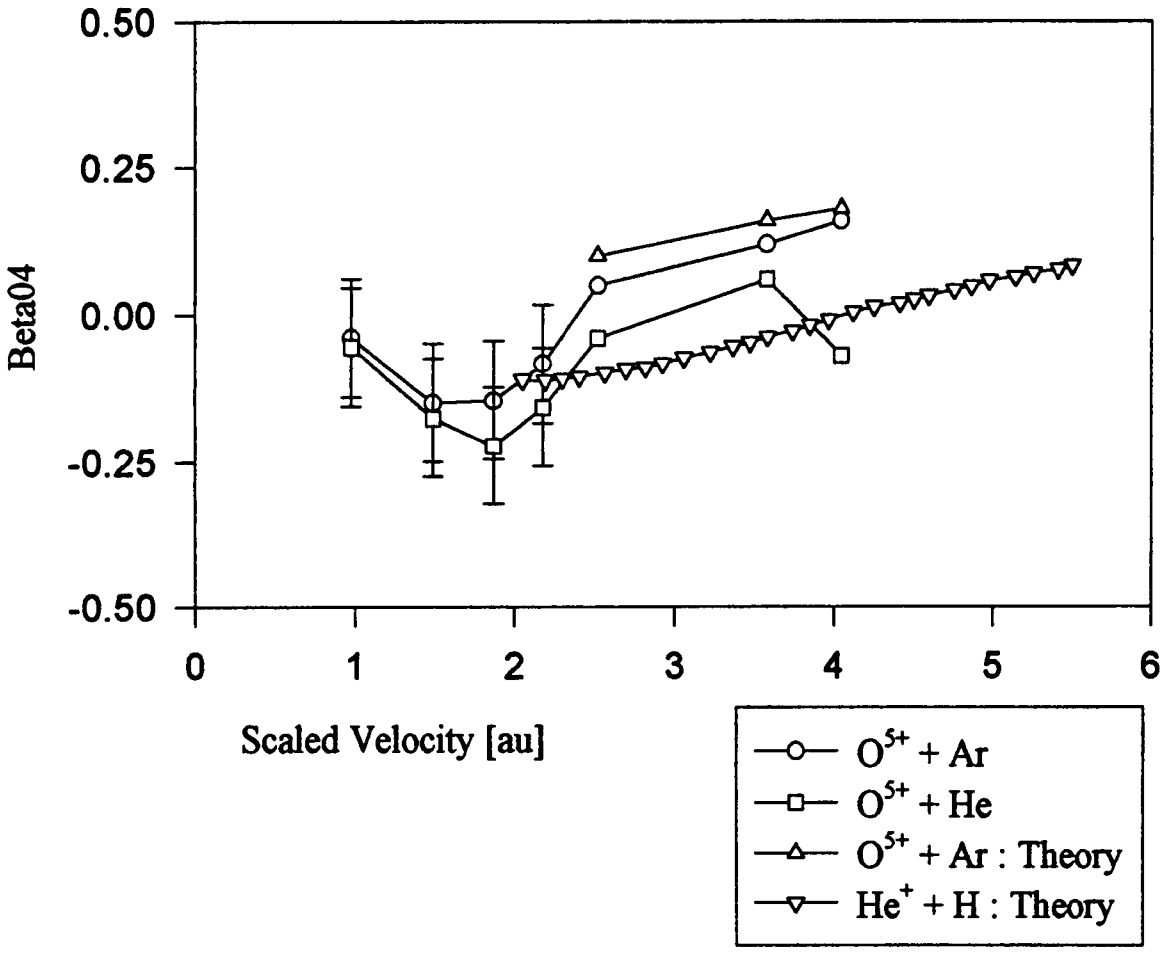



**Figure 5.4**: Fourth-order anisotropy coefficient for ELC of $He^+$ on H and $O^{5+}$ on Ar (He) as a function of scaled velocity $v_s$. Experimental Data without error bars: adapted from S. D. Berry, *Multipole Characterization of the Electron Loss to the Continuum Cross Section at Asymptotically High Velocities*, Ph. D. Dissertation, University of Tennessee, 1985. Theory : adapted from J. Burdörfer, M. Breinig, S. B. Elston, I. A. Sellin, Phys. Rev. **A28**, 3277 (1983).

the targets are too. There is good qualitative agreement of the detailed behavior of $\beta_{04}$, too, when scaled velocities are compared with allowed error bars. The theoretical curve shows the sign of $\beta_{04}$ is changing as projectile velocity sweeps down from high velocity; $\beta_{04} > 0$ at high $v_s > 4$ au and $\beta_{04} < 0$ at low $v_s$. For the experimental curve, this change in sign happens at $v_s \sim 3$ au for He target and at $v_s \sim 2.5$ au for Ar target. There appears to be a systematic target dependence: the zero crossings are consistent at higher $v_s$ for the lighter (He) target. Experimental minima in the vicinity of $v_s \sim 1.8$ au are the most striking features in Figure 5.4, but its interpretation is still in question even though there exists a little slope change in the theoretical curve at $v_s \sim 2$ au.

Very good agreement between theoretical and experimental results is also found for the case of argon target, i.e. $O^{5+}$ on Ar, at the highest three energies (2.5, 3.6, and 4.0 au), where the first order Born approximation is expected to be more reliable. In case of helium target the need for better theoretical treatment like electron-electron interaction has been suggested for the lack of good agreement in the high projectile velocity region.[69]

Recently Jakubaßa-Amundsen calculated singly as well as doubly inelastic ELC cross sections up to second order Born approximation for $He^+ + He$ collisions, and found that, for the total cross section, the first Born approximation breaks down at velocities below twice the ground state electronic orbiting velocities, and the second Born approximation becomes unreliable around $v \approx Z_P$.[22] From the Figure 5.3, we can see that the strong first Born approximation evidence, $\beta_1=0$, which is the skewness parameter for the ECC collision system, starts to deviate slightly around $v_s \sim 2$ au ($V_P \sim 8$ au for $O^{5+}$) and considerably around $v_s \sim 1.5$ au ($V_P \sim 6$ au) in our ELC result with the argon target. Its negatively

polarized monopole in the lower ion velocities indicates that the forward emission cusp peak is skewed towards the low-energy side. However, it seems to be much weaker than in the case of the ECC. These experimental results of multipoles for the validity of the Born approximation obviously satisfies the theoretically calculated boundaries in the projectile velocities.

The significance in this analysis arises from the $\beta_{02}$ and $\beta_{04}$ sign changes, which occur around $v_s \sim 2$ to 2.5 au for $\beta_{02}$ and around $v_s \sim 3$ to 4 au for $\beta_{04}$, both in experimental and in theoretical curves. Switch in signs means that the emission switches from predominantly transverse to the beam axis for $\beta_{02} < 0$ and $\beta_{04} > 0$ at high $V_P$ to the predominantly axial (longitudinal) for $\beta_{02} > 0$ and $\beta_{04} < 0$ at low $V_P$ as seen in Figure 5.1 and Figure A.1 in Appendix A. This is exactly what this sequence of ELC distributions does as the projectile velocity drops to $v_s \sim 1$ au ($V_P \sim 4$ au for $O^{5+}$)

The theoretical model mentioned above for $He^+ + He$ and $He^+ + H$ was successfully applied to experimental results for $O^{5+} + Ar$ (He) by scaling the collision velocity to the ELC electron's shell velocity. The experimental evidences of the theoretically given break-down region in the first and second order Born approximation for the ELC ion-atom single collision system make this experimental effort valid.

In contrast with asymmetrical skewing towards lower lab frame energies observed for ion-gas collisions, a series of theoretical works[17,70,71] and convoy electron measurements[68,72] pertaining of ion-foil collisions shows that the most dominant mechanism for convoy electron production within the bulk of the target is electron loss to the continuum of the

projectile. Yamazaki *et al.*[73], followed by Hülskötter *et al.*,[74] also concluded that the ELC dominates in convoy electron production mechanism for 2 MeV/u C projectiles incident on carbon foils, even when bare carbon ions are incident on the thinnest carbon foils. In order to verify this behavior in terms of multipole expansion parameters, similar collision systems are employed with various foil thicknesses and energies.

Table 3.2 shows the experimental characteristics for recent ion-foil collisions. Two sets of experimental setup were adapted to get forward emitted electron distributions with two different carbon foil targets, 1.8 and 20 μg/cm$^2$ in thickness. For the case of 1.8 μg/cm$^2$ C with 2.0 MeV/u $C^{5+}$ ion beam, we employed an electrostatic charge state analyzer downstream of the electron analyzer to measure charge state fractions of the emergent ions. The average charge state of the emergent ion with charge state fraction of 6% of $C^{4+}$, 44% of $C^{5+}$, and 50% of $C^{6+}$ is $\bar{q}_e = 5.44$, which is compared with 5.6[73] for the equilibrium thickness targets.

The spectrum in Figure 5.5 for 2.0 MeV/u $C^{5+}$ was fitted with multipoles up to k=10 to compare the high order multipole contributions in $\chi_\nu^2$ with the previous results for the ELC data.[68] The fitted values of $\beta_k$ are given in Table 5.4 for the various impact velocities. The result for 8.98 $V_P$ au (2.0 MeV/u) $C^{5+}$ + 1.8 μg/cm$^2$ C indicates that its dominant figure of the convoy electron distribution is the ELC as mentioned before. The corresponding goodness of the fit is seen in the residue plot of Figure 5.6. Even though a relatively low projectile energy with a thin target ($\bar{q}_e = 5.44$) was used, the enrichment in higher-order multipoles with decreasing projectile energy is still countable in this velocity range.

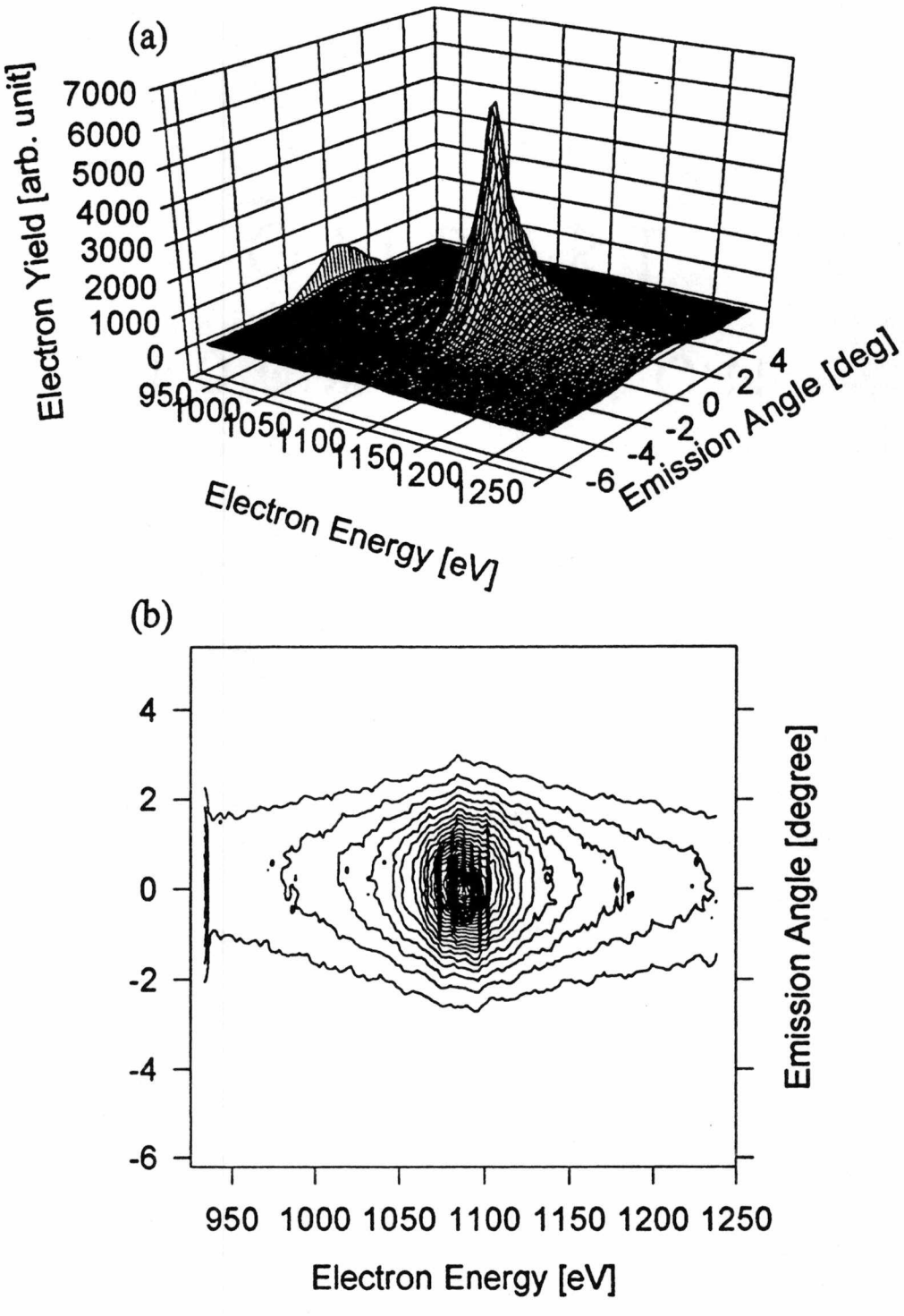



**Figure 5.5**: Doubly differential electron emission in angle and energy for 2.0 MeV/u $C^{5+}$ incident on 1.8 $\mu g/cm^2$ carbon foil.

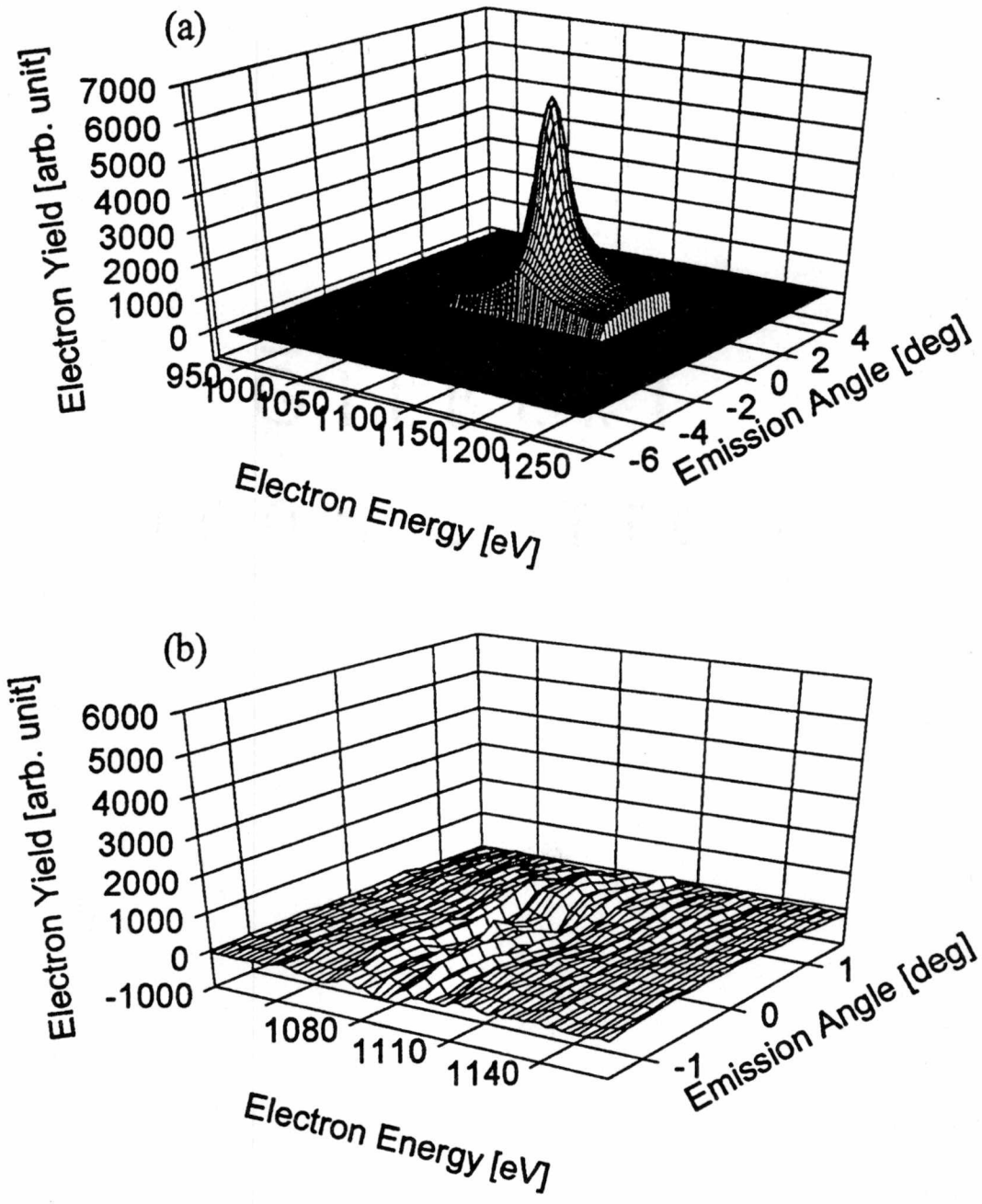



**Figure 5.6**: Result of fitting with multipole expansion function in angle and energy for 2.0 MeV/u $C^{5+}$ incident on 1.8 $\mu g/cm^2$ carbon foil. The reproduced plot (a) from the fitted parameters describes the convoy electron distribution. The plot (b) illustrates the goodness of fit in terms of residue spectrum which is the source minus the fitted data.

Table 5.4

COMPARISON OF $\beta_k$ VALUES FOR CONVOY ELECTRON DISTRIBUTION (ELC)

| $V_P$ (au) | Ion | Target ($\mu g/cm^2$) | $ß_2$ | $ß_4$ | $ß_6$ | $ß_8$ | $ß_{10}$ |
|---|---|---|---|---|---|---|---|
| 16.2* | $O^{5+}$ | 15 | -0.78 | 0.25 | -0.32 | 0.19 | -0.16 |
| 14.3* | $O^{5+}$ | 15 | -0.82 | 0.29 | -0.26 | 0.07 | -0.03 |
| 10.1* | $O^{5+}$ | 15 | -0.48 | 0.11 | -0.18 | 0.09 | -0.05 |
| 8.98 | $C^{5+}$ | 1.8 | -0.82 | 0.34 | -0.11 | 0.05 | -0.04 |
| *4.49* | *$C^{2+}$* | *20* | *0.12* | *-0.19* | *2.12* | *1.29* | *1.63* |
| **4.49** | **$C^{2+}$** | **20** | **-0.41** | **0.93** | **3.34** | **-1.01** | **1.13** |

* denotes previously published results.[68]

Nonetheless, the dominant terms are quadrupole and hexadecapole moments indicating strong evidence of the ELC process.

Meanwhile, relatively low energies of 0.2, 0.3, and 0.5 MeV/u $C^{1,2,3,4+}$ ion beams were selected to combine the grazing angle ion-surface measurements, which require very low perpendicular kinetic energy in the range of 100 eV or less, with the equilibrium thickness target of 20 $\mu g/cm^2$ carbon foil experiment at the same beam time. In the case of 4.49 $V_P$ au (0.5 MeV/u) $C^{2+}$ incident ion on 20 $\mu g/cm^2$ carbon foil, we encounter a significant characteristic difference, which is the existence of a very broad binary peak in the higher energy region as seen in the previous Figure 4.4. In other words, the energy gap between "convoy" and binary encounter peaks is relatively small so that the employment of multipole expansion is no longer valid unless we remove the binary encounter electrons. The result in *Italic* fonts in Table 5.4 represents the fitting result only with the multipole expansion, and was very unreliable even though the reduced $\chi^2_\nu$ was around 5.5 since the binary encounter peak is overriding the cusp peak. Without serious theoretical calculations

for the model function of the binary encounter peak, an alternative way to extract the multipole moments for the cusp has been considered using Gaussian or Lorentzian functions for the binary peak. For this collision system asymmetrical Gaussian has been introduced inside the original model function so that the multipole expansion and Gaussian functions can incorporate each other to search for the correct parameters characterizing the cusp peak. The **bold** fonts in Table 5.4 indicate its remarking result with the reduced $\chi^2_\nu$ of about 1.67, which is compared with the fitted result without a Gaussian function. The difference between the original and the fitted spectra with and without taking a Gaussian function into account can be easily seen by comparing Figure 5.7 with Figure 5.8. At current stage we are using a Gaussian function. In order to extract the exact multipole moments a considerable amount of work on theoretical model functions is required. This was one of preliminary attempts to separate two peaks in case of relatively low energy ion beam and not well-known model functions, and it has been proven to be very effective.

## V.3 Grazing Angle Ion-Surface Collisions

Electron emission in grazing incidence ion-surface interactions have been studied in an experimental investigation of electron emission processes which determine the shapes of convoy and BEE peaks as well as the charge and excitation of projectiles specularly reflected at grazing angles of incidence from atomically clean and flat surfaces. The measurements were extended to the high energy emission region to study the relative

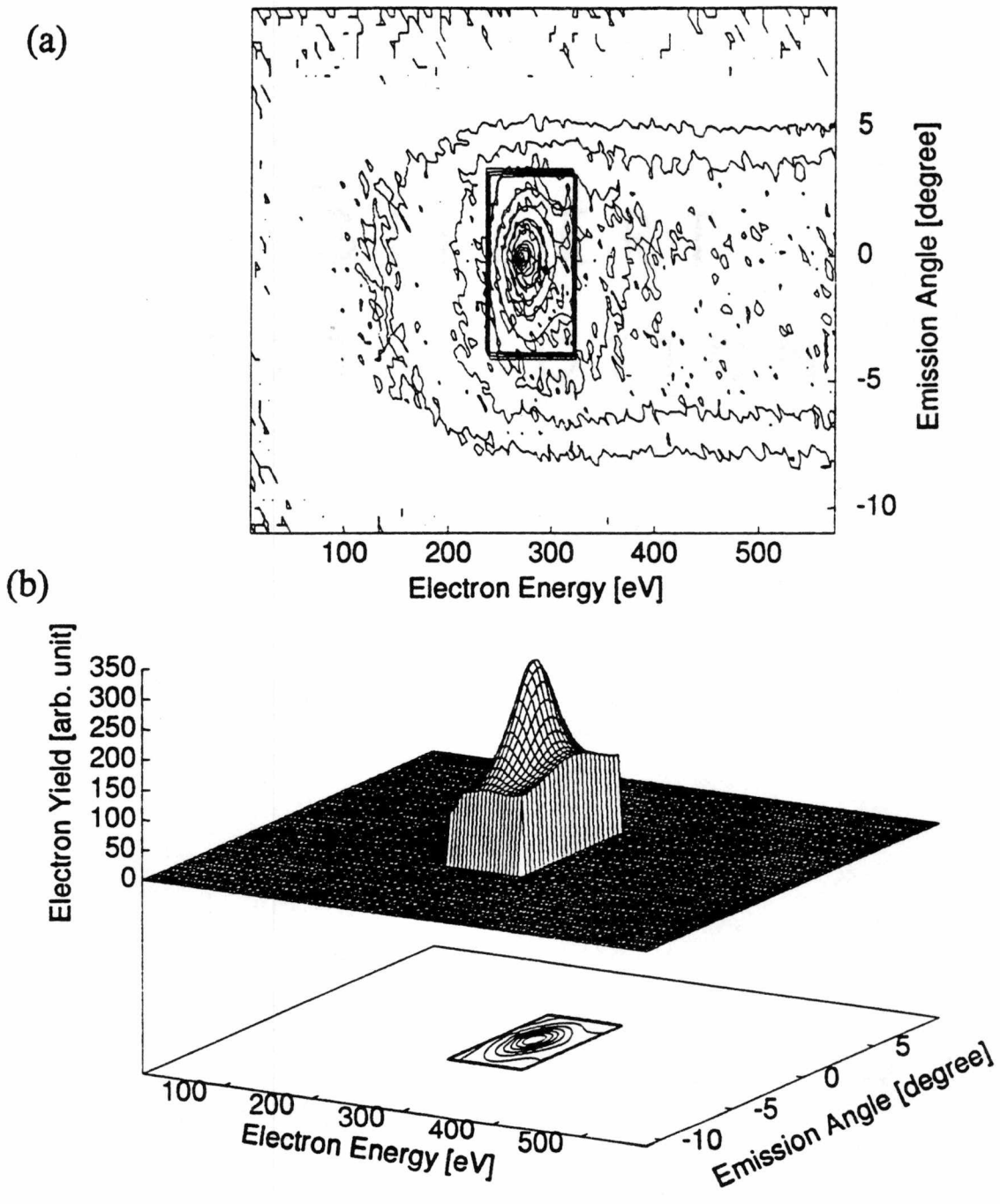



**Figure 5.7**: Fitting result with a Gaussian function for the binary encounter electron peak and a multipole expansion function for the cusp peak for 0.5 MeV/u $C^{2+}$ incident on 20 $\mu g/cm^2$ carbon foil.

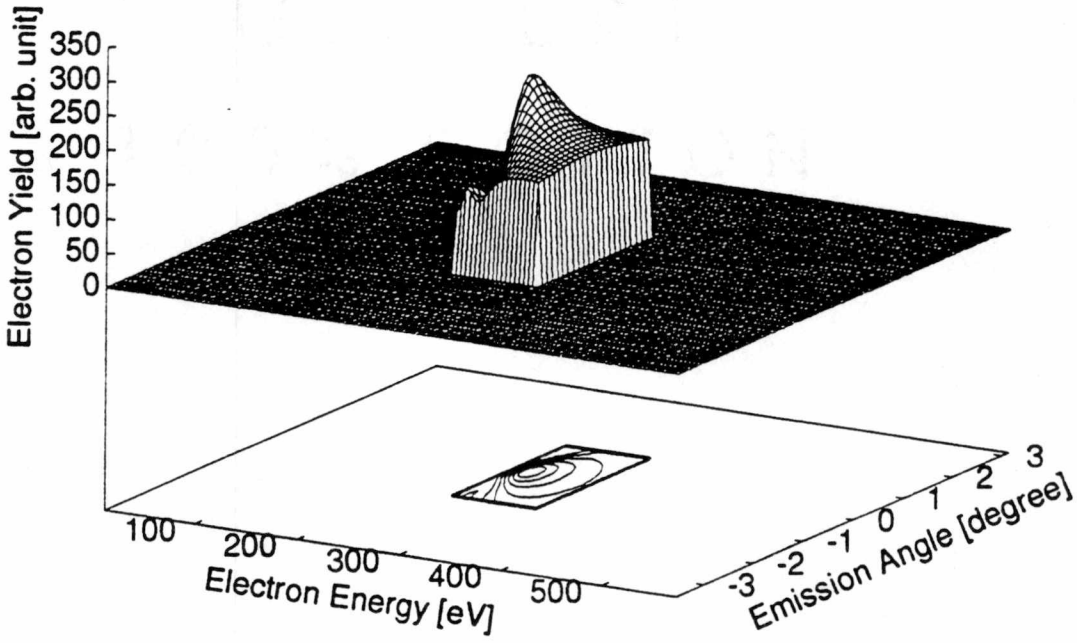



**Figure 5.8**: Fitting result without a Gaussian function for the binary encounter electron peak and with a multipole expansion function for the cusp peak for 0.5 MeV/u $C^{2+}$ incident on 20 $\mu g/cm^2$ carbon foil.

importance of the binary encounter and convoy emission as fundamental collision processes.

There exist several distinguishing features for grazing angle ion-surface collisions reported by several experimental groups. However, all results reported to this time have only measured singly differential or doubly differential electron distributions by means of separate angle observations, limiting the ideal experimental conditions. The employment of the PSD and the spherical sector electrostatic analyzer enables us to measure triply differential electron distributions, which is not directly available in other groups. The ability to observe electron emission as a function of azimuthal and polar emission angles with respect to the surface reveals very rich information.

As already seen in Figure 4.6, the angular distribution is much broader in both azimuthal and polar angles than in the previously mentioned case of ion-atom or ion-foil interactions, at least within our limited angle of view in the azimuthal direction. The polar distribution is asymmetric with a long tail towards higher angles away from the surface. To demonstrate this behavior, the polar angle convoy peak distribution is shown in Figure 5.9 for different incident angles of 6 MeV $C^{3+}$ ions upon a Si (100) surface. Meanwhile the polar angle shifts of the cusp peak as shown in Figure 3.8 are actually accelerated emission angles with respect to the reflected specular beam. The actual polar angle shifts in Figure 3.8 for 0.5 MeV/u $C^{3+}$ are about 4 to 5°. It is interesting to note that the magnitude of the shift increases over the grazing angle range from 0.0 to 0.6°, and eventually converges to a shift angle of ~4.2°. This can be explained in terms of cleanness of the surface; that is,

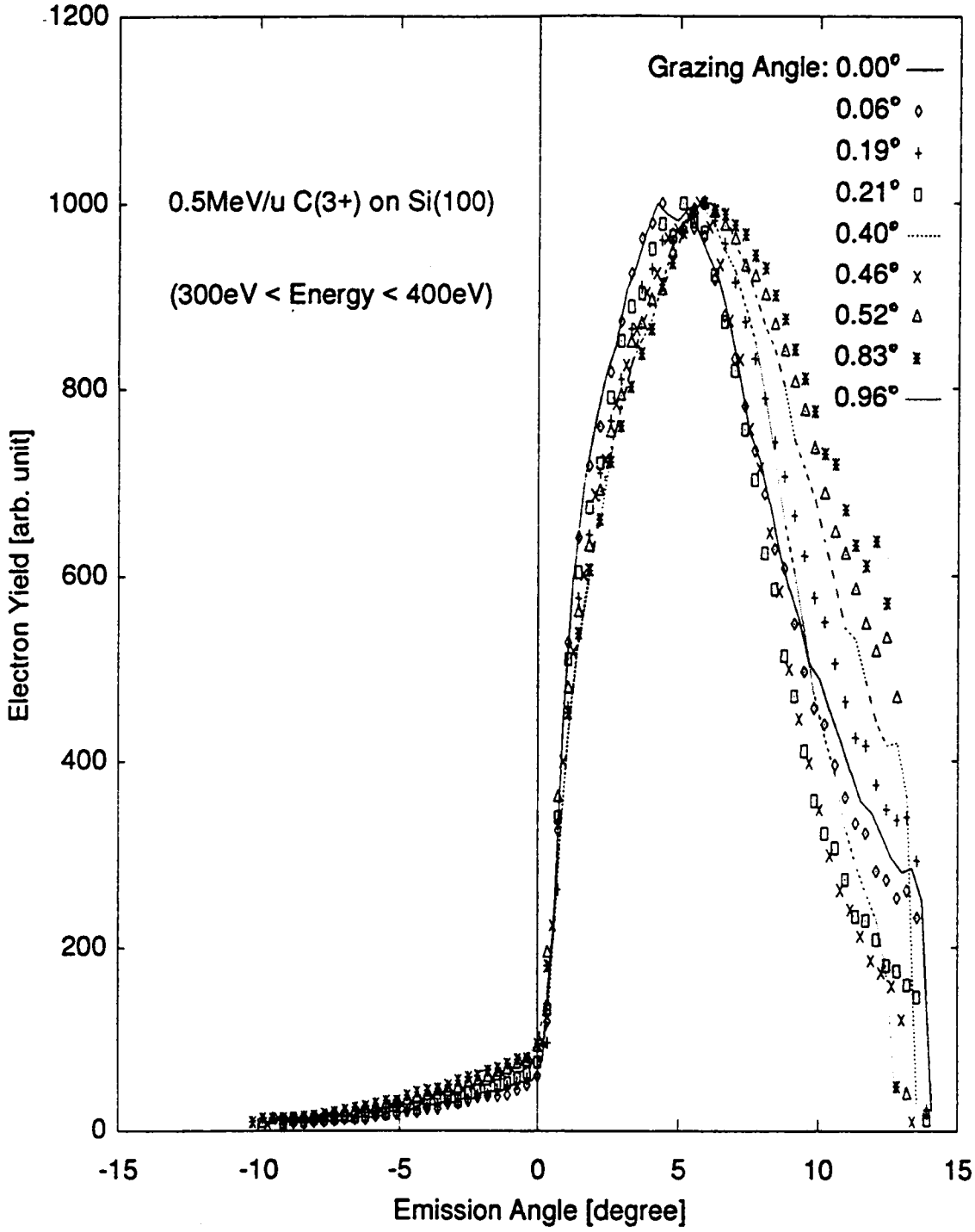



**Figure 5.9**: Polar angle distribution of cusp electrons emitted in interaction of $C^{3+}$ ions with a kinetic energy of 0.5 MeV/u with a Si(100) surface.

above a certain incidence angle any possible contamination of the target no longer plays an important role in the acceleration of the convoy electrons. In the case of grazing angles of near zero, the difference between the effects of DIP is significant; the angular shift is rapidly decreasing from 3.8° to 2.8°. The perpendicular kinetic energies for 0.2°, 0.6°, and 1.0° are 73 eV, 660 eV, and 1.8 keV, respectively. For near zero angles there is a sharp cut-off in all cases, because for grazing angles the electrons would have to pass either slightly above the surface or through the crystal. The image charge of the ion has a strong effect on the outgoing electrons which get repelled away from the surface and therefore produce a broad polar angle distribution. Thus, the growth of the positive polar angle shoulder is a sign for the growing influence of reactions involving ions specularly reflected in front of the surface.

From another view point we also have convoy and binary encounter peaks which are broader both in energy and polar angle for grazing angle ion-surface collisions than for ion-atom or ion-foil collisions. To produce the emission distribution shown in Figure 5.10, $C^{3+}$ ions with a kinetic energy of 0.5 MeV/u were directed onto the Si (100) surface at an angle of 0.4°. Figure 5.10 shows two distinct electron populations: one at ~300 eV and one in the range of (500-800) eV. Those peaks can be described as convoy electrons, produced in electron capture and loss to the continuum, and as binary encounter electrons, produced in collisions of the projectile with target electrons, respectively.

U. Thumm[75] argued that the cusp electron peak is only related to the final state interaction and therefore its shape remains unchanged. This was a very critical point for our data. From the spectrum itself we could not distinguish the cusp and the binary peaks,

at some point we needed two peaks separated. The most effective way was to use relatively simple, peaked model functions without resorting to complex theoretical calculations. At current state of the art, there are only modeling or simulation processes available for such complex dynamics as ion-surface collisions; an analytical description of the convoy electron peak is not available. A stage of modeling implies that a theoretical description involves only a few processes selected as presumably the most important ones among a large number of collision processes. One of the modeling processes currently presented was the rainbow scattering process, but as this includes a complex computer simulation with a very specific ion energy and target (0.3 MeV/u $Li^+$ ions interacting with a SnTe(001) surface), it is difficult to extrapolate or scale the result to another collision system. Thus, the modeling of the convoy and the binary peaks in this work is done solely for the purpose of the data analysis.

The first step for modeling the convoy peak in the present work is to extract or "strip" out the contributions that can be ascribed to binary encounter emission. A combination of Lorentzian and Gaussian functions for convoy and binary encounter peaks was adapted to fit to the ion-surface emission distribution data and tested several tens of times with reasonable initial parameters. 0.3 and 0.5 MeV/u $C^{2+,3+}$ ions with grazing incidence angles less than 1.0° were selected to fit for the convoy and binary peaks; a total of 16 possible combinations was tested under exactly the same fitting conditions. Figure 5.11 displays the reduced $\chi^2_\nu$ of the resulting best-fits to each of 17 different emission spectra considered, as a function of each of the 16 combinations of model functions. The model function combinations are denoted by *n*:ABCD, where *n* is an arbitrary numeric identifier and each

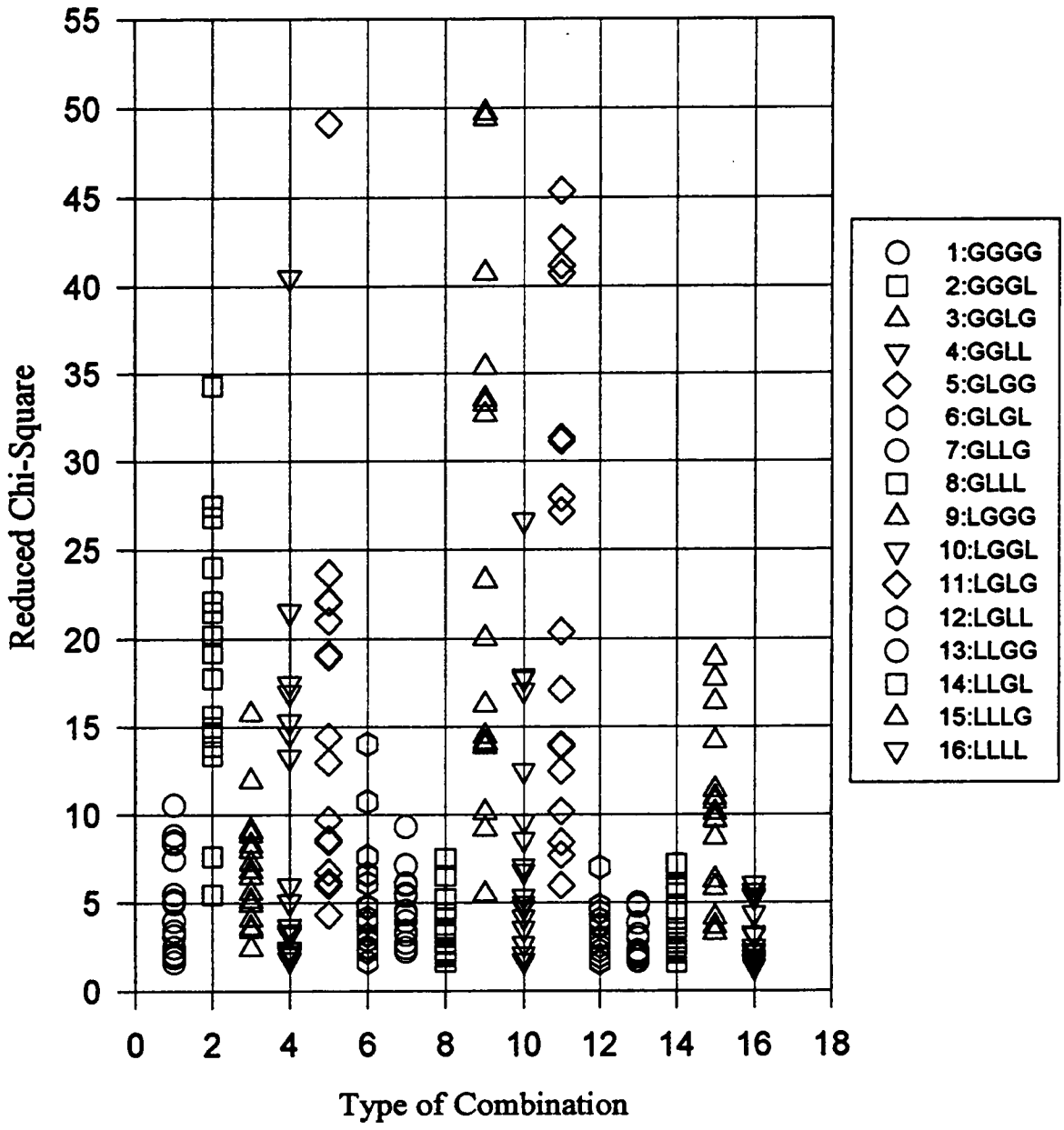



**Figure 5.11**: Result of fitting with a combination of Lorentzian and Gaussian model peak functions. A total of 17 spectra for 0.3 and 0.5 MeV/u $C^{2+,3+}$ ion beams were selected to fit; normally the lower 8 results in each column indicate those of 0.3 MeV/u $C^{2+,3+}$ spectra.

of A, B, C, and D can take on the value L (for a Lorentzian function) or G (Gaussian). A refers to the convoy peak energy distribution model function, B to the convoy angular model function, C to the binary peak energy function, and D refers to the binary peak angular distribution model function.

For the case of 0.3 MeV/u projectiles, the most effective combination was 16:LLLL, which means combination type 16: Lorentzian in angle and Lorentzian in energy for the convoy peak, and Lorentzian in angle and Lorentzian in energy for the binary encounter peak; with 0.5 MeV/u ions the best combination was 12:LGLL. However, the most suitable combination for both cases was 13:LLGG, in which the average $\chi^2_\nu$ was 2.9.

So far the best combined result for the general case was obtained using Lorentzians for convoy electrons and Gaussians for binary encounter electrons in both angle and energy. Figure 5.12 shows the applied functions, Lorentzian function (x,y) and Gaussian function (x,y), on the spectrum in Figure 5.10, which has two corresponding peaks, convoy and binary encounter peaks, respectively. The general model function F(x,y) was selected by combining two equations. Asymmetry parameters as well as polynomials up to 1st order were also selected to compensate for an unusual tilt or a background offset. The 2nd order polynomial term has been removed after several trials since its reduced $\chi^2_\nu$ contribution has been proven negligible.

$$F(x,y) = \text{Lorentzian}(x,y) + \text{Gaussian}\ (x,y) \tag{5.2}$$

$$\text{Lorentzian}(x,y) = \frac{a}{[1+(\frac{2X}{\Gamma_x})^2]\cdot[1+(\frac{2Y}{\Gamma_y})^2]} \cdot (1 + bX + cY) \tag{5.3}$$

where $X = x - x_{center}$, $Y = y - y_{center}$, and

$$\text{Gaussian}(x,y) = d \cdot \exp[-eU^2 - fV^2 - gUV] \cdot (1 + hU + iV) \tag{5.4}$$

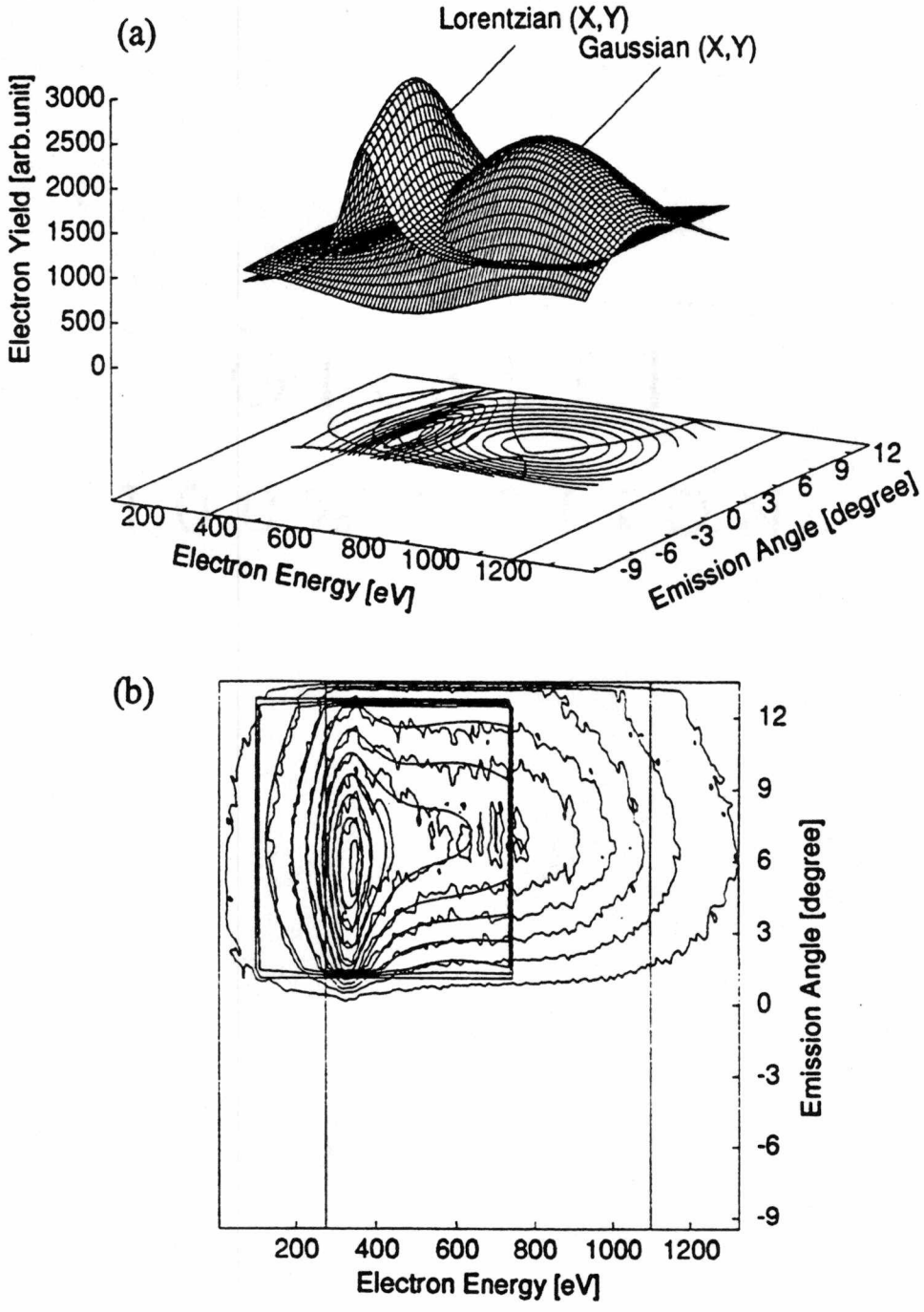



**Figure 5.12**: Two applied model functions (a) and the result of the fit in contour plot (b) for 0.5 MeV/u $C^{3+}$ incident on Si(100) at a grazing angle of 0.4°. The same intensity contour lines are drawn in the source and fitted spectra.

where $U = x - x'_{center}$, $V = y - y'_{center}$.

The fitting ranges in angle and energy were selected to preserve the interesting feature as much as possible; here the interesting feature is the cusp electron peak. The Gaussian and Lorentzian model functions proved to be very effective without theoretical aspects, and the resulting reduced $\chi^2_\nu$s for 2-dimensional non-linear fitting program were less than 5 as shown in Figure 5.11 for the case of 13:LLGG. This was a somewhat amazing result since these relatively low reduced $\chi^2_\nu$s seem to be very difficult to achieve for such a highly complex dynamic collision system even though we are only dealing with electrons in a jellium edge of the surface. The goodness of the fit with 10% intensity level contour plot clearly witnesses that the fitting result in Figure 5.12 is reasonable even though only half of binary encounter peak was actually fit.

This astonishing result suggests that proper model functions for binary and cusp peaks can produce the exactly interesting feature by a subtraction. In Figure 5.13 the residual plot has been created and also shows the goodness of the fit since the contour plot of the residue shows no significant structure in the removed part. These fitted data were separately applied to the raw data spectrum to extract the corresponding cusp and binary encounter electron emissions shown in Figure 5.14. Very smooth and continuous structure at least for the convoy peak without losing any substantial, underlying parts implies that the separated cusp and binary encounter peaks are relatively reliable for data analysis.

From this process it has been decided that the maximum intensity and the FWHM fractions of the cusp peak before and after removal of corresponding binary peak for 0.5 MeV/u $C^{3+}$ incident on Si(100) at a grazing angle of 0.4° were about 1.6 and 1.5,

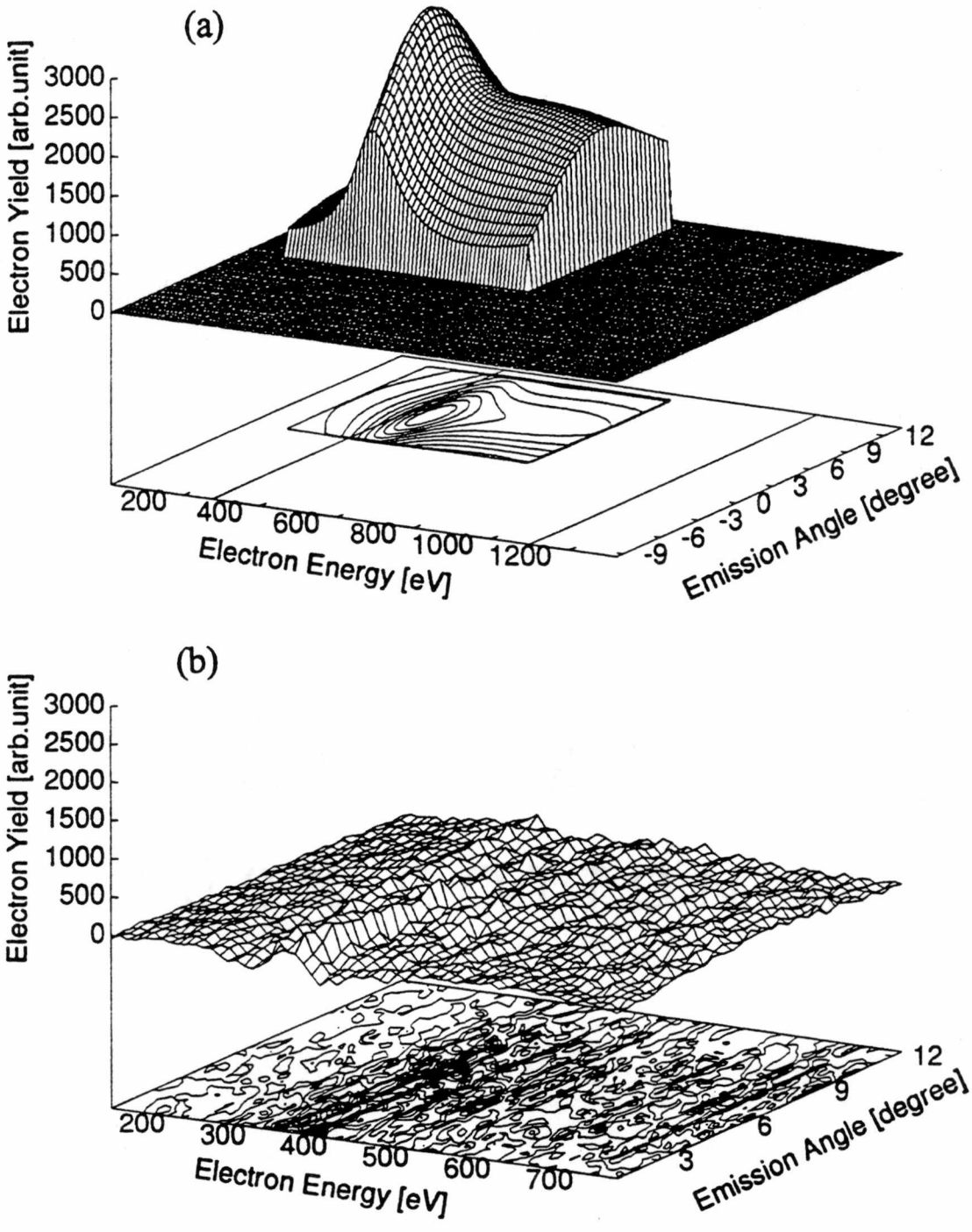



**Figure 5.13**: The result of the fit with a generated (a) and a residue (b) plots for a grazing angle ion-surface collision system.

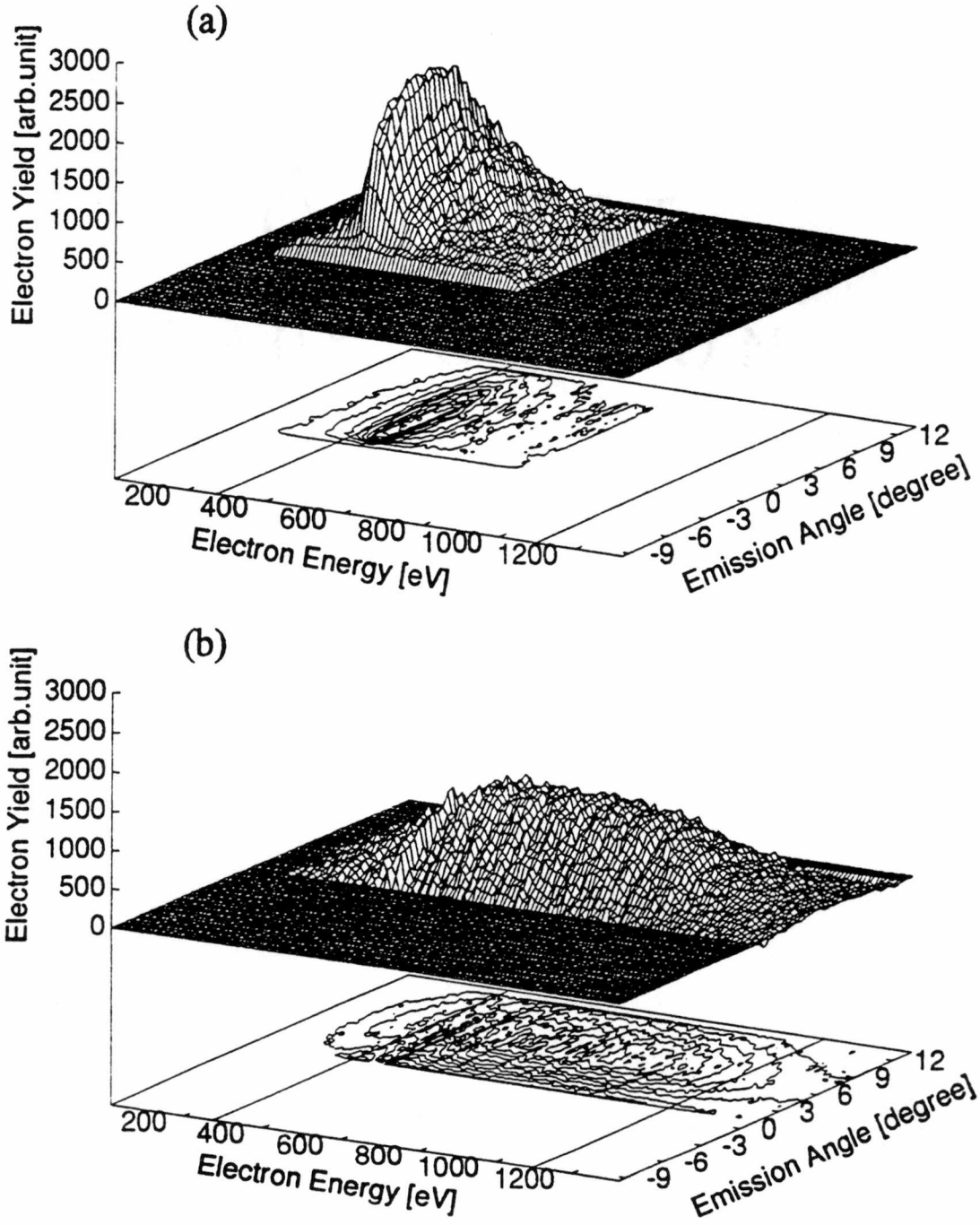



**Figure 5.14**: Extracted convoy (a) and binary electron (b) peaks using the fitted parameters.

respectively, and the position of the peak after removal of binary encounter electrons was ~323eV in energy channel and ~6° in angle channel. The energy shift and FWHM in angle and energy of the cusp peak were extracted from fitting and used to analyze the characteristics. It is evident that the extracted electron distribution of the convoy peak in energy is very symmetric, and this symmetry is not discernible in the original electron distribution spectrum in Figure 5.15.

As seen in Figure 5.16, the energy shift of the convoy electron peaks as well as energy and charge state dependences of left-FWHM/2 were carefully inspected and plotted from the original spectra and fitted data with error margins of 5 and 10 eV, respectively. The kinetic energies of 0.3 and 0.5 MeV/u $C^{2+,3+}$ were selected in order to study these behaviors of the convoy electron peaks with incident beam angles of 0.0° to 1.25°. For a kinetic energy of 0.3 MeV/u the peak of the convoy electrons was ~235 eV, independent of the incident charge state, which amounts to a shift of ~70 eV from the expected isotachic value of 164.6 eV. However, for the case of 0.5 MeV/u there is a dependence of the convoy peak shift on the incident ($C^{2+}$ vs $C^{3+}$) charge state for all measured grazing angles. $C^{2+}$ ions produce an almost constant energy shift of 55 eV, while $C^{3+}$ ions produce varying energy shifts from ~60 eV to 80 eV. This charge state dependence of the convoy peak is expected from the wake potential, which depends dynamically on the charge state of the ion. The maximum energy shift for 0.5 MeV/u $C^{3+}$ at a grazing angle of 0.5° was greater than the result of the polar angular shift of the convoy electron peak as seen in Figure 3.8; the ratio of the transverse and longitudinal acceleration for these cases was approximately 1.6. The perpendicular kinetic energies of 0.5 MeV/u carbon ion beam at incident angles of 0.25°

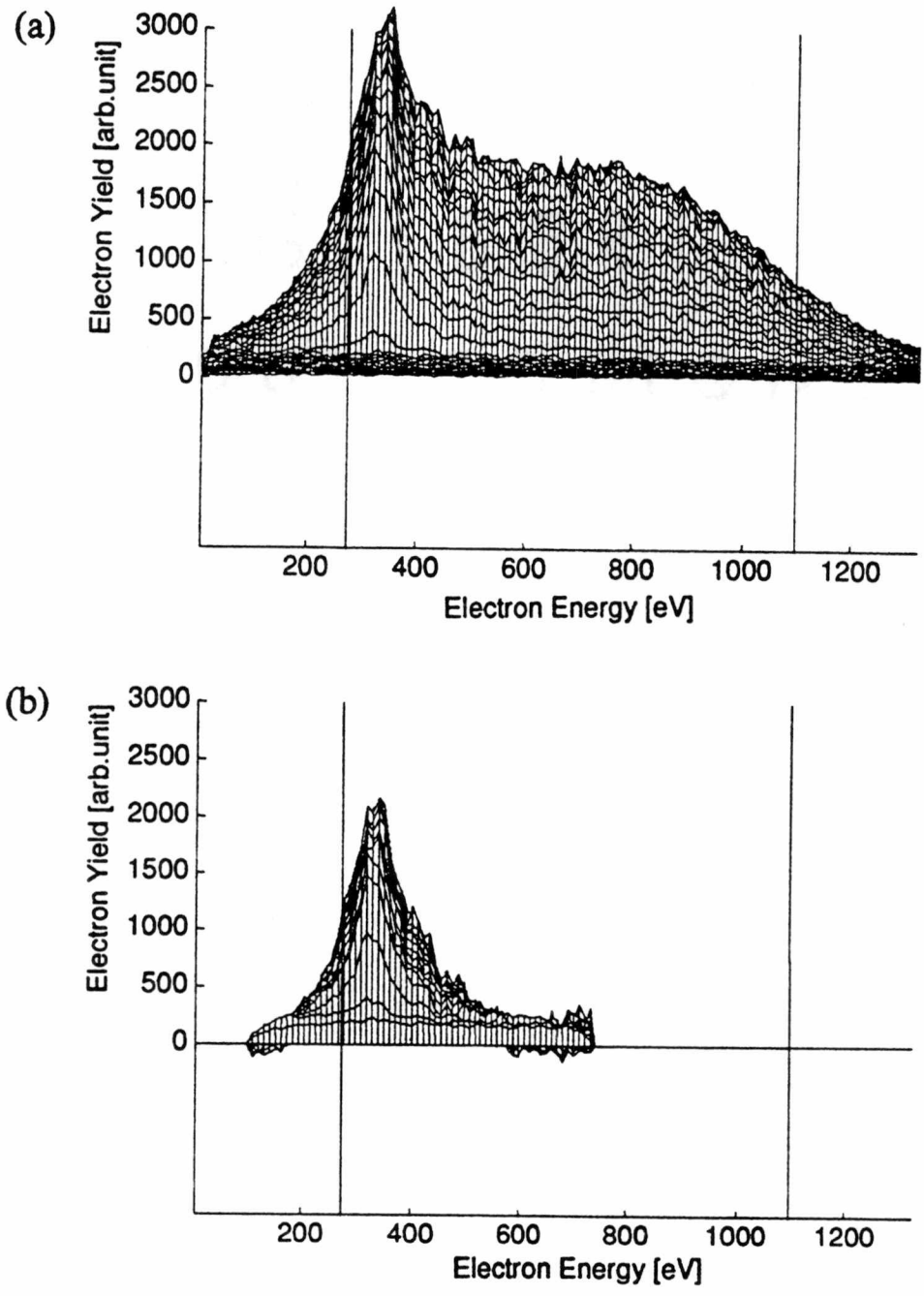



**Figure 5.15**: Energy distribution of convoy electrons before (a) and after (b) removal of binary encounter electrons. Fractions of maximum intensities and left FWHM/2s of both cases are ~1.5 and ~1.6, respectively.

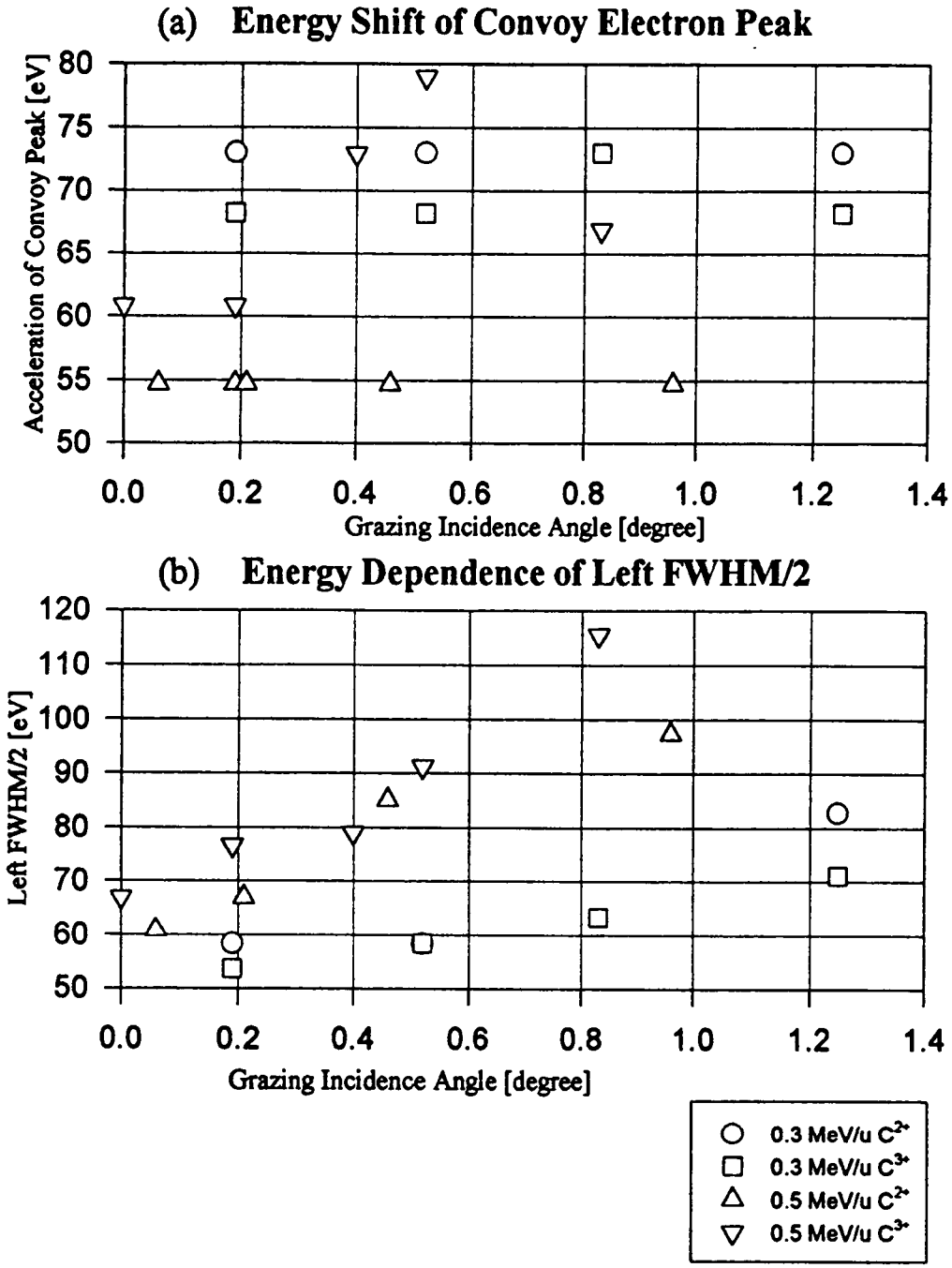



**Figure 5.16**: Energy shift (a), and energy and charge state dependence of left FWHM/2 (b) for the convoy electron peak without removing the BEE peak.

and 0.5° are 114 and 457 eV, respectively; similarly for 0.3 MeV/u carbon projectile at incident angles of 0.3° and 0.5° are 99 and 274 eV, respectively.

A possible explanation for different behaviors of the convoy peak at 0.5 and 0.3 MeV/u is that at 0.3 MeV/u the ions are reaching very early an equilibrium charge state, which means that the initial charge state has practically no influence on the exit charge state of the ion. Thus, the image charge state seen by the electrons does not change, and therefore the post-collision acceleration of the electrons is independent of the incident ion charge state.[76] In the case of ions at 0.5 MeV/u, the equilibrium charge state is not as much reached as in the aforementioned case as the collision process happens faster.

The lower plot of Figure 5.16 shows that there is no dependence of the left-FWHM/2 of the convoy peak on charge state, but a slight dependence on grazing angles including a stronger dependence for 0.5 MeV/u $C^{3+}$ ions. It seems that left-FWHM/2 is proportional to the grazing angle of the incident beam, and 0.5 MeV/u carbon beam has a slightly wider width than that of 0.3 MeV/u.

When observed spectra were compared with the fitted data of the convoy peak separated from the corresponding BEE peak, the effect of "peak pulling" due to the broad BEE peak was expected. For the case of 0.3 MeV/u $C^{2+,3+}$ the average peak shift of the separated convoy peak was about 45 eV, which is ~25 eV different from the result in Figure 5.16, and there is no charge state dependence. For 0.5 MeV/u $C^{2+,3+}$, average shifts are ~46 ($C^{2+}$) and ~51 eV ($C^{3+}$), and there exists a similar charge state dependence in Figure 5.16; $C^{3+}$ ion beam also has a maximum peak shift at a grazing angle of 0.5°. The difference of the energy shift of the convoy peak between with and without the BEE peak

for 0.3 MeV/u $C^{2+,3+}$ is much bigger than the case of 0.5 MeV/u $C^{2+,3+}$. It is evident from these results that the very close binary peak is actually pulling the nearby convoy electron peak towards higher energy side.

This experimental approach to separate the convoy and the binary encounter electron peaks from observed spectra may raise a question since the binary encounter electrons in the forward direction can also be considered convoy electrons. However, such an experimental trial for systematic analysis of a specific feature from complex dynamics can be a critical factor to understand the corresponding system. When exact analytical solutions are available, this method will be very appropriate to analyze a certain feature from complex systems.

## V.4 Comparison for All Collision Systems

All three collision systems, i.e. ion-atom, ion-bulk-solid, and ion-surface, have been treated so far individually only. It is of interest now to compare the results from this broad range of physical processes. From studies described in the previous sections, it is evident to the unaided eye that the ion-surface results have a markedly different character which must be a consequence of additional physics that should be eventually incorporated in the modeling process.

The characteristics of three different collision systems are seen in Figures 5.17 and 5.18, which combines convoy emission results from the three generic collision systems, with the emission peaks normalized to aid in comparison of shapes. The single collision (ion-atom)

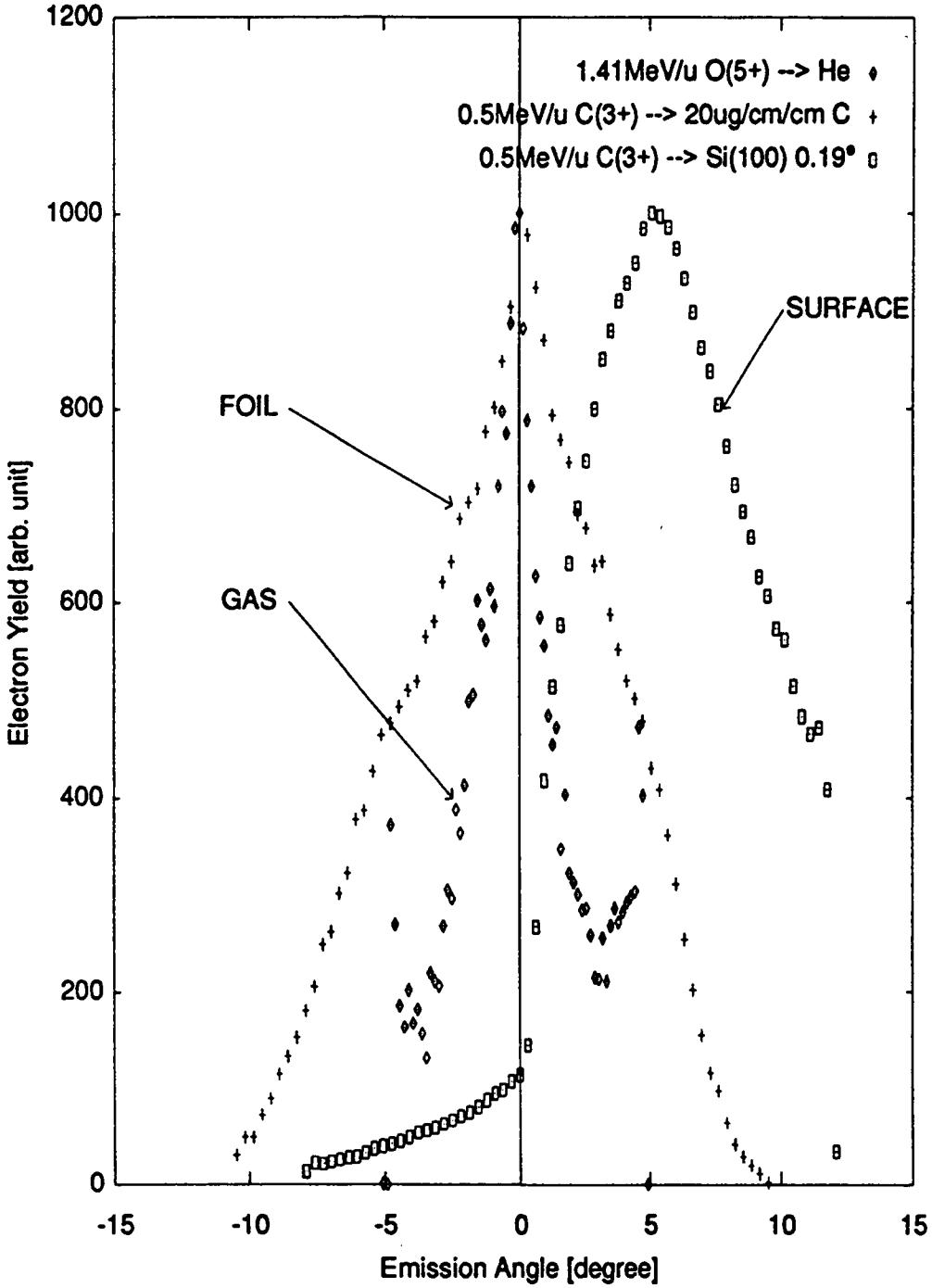



**Figure 5.17**: Angular distribution of electrons from gas, foil, and crystal targets.

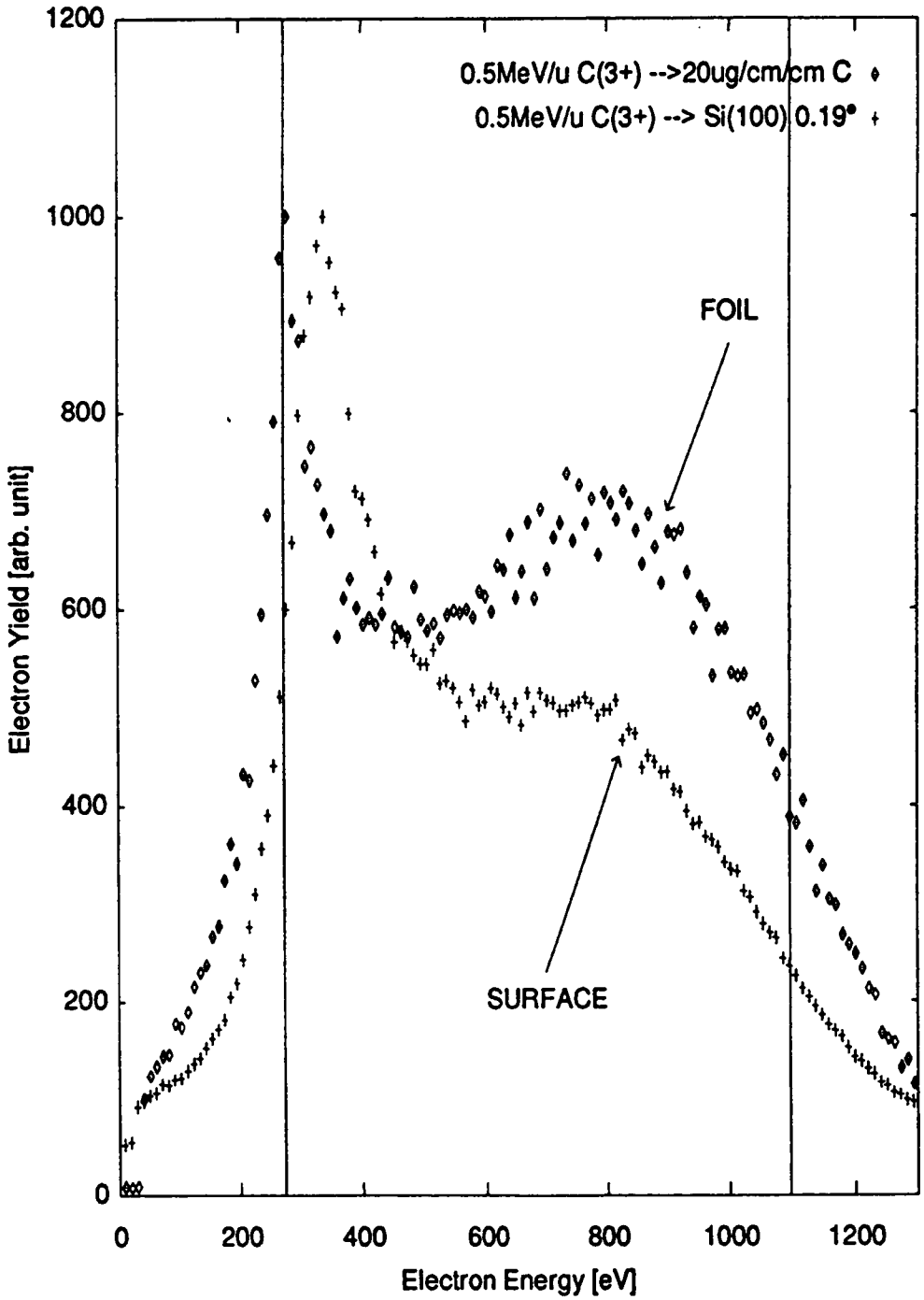



**Figure 5.18**: Energy distribution of electrons from foil and crystal targets.

production mechanism produced a different convoy peak shape from multi-collision production mechanisms. For ion-gas collisions the sharply defined convoy electron distribution was concentrated in angles of ~3°, whereas the multi-collision systems produced relatively wide convoy electron emissions in angles of ~10°.

When compared with results of ion-atom and ion-foil collisions, broadening of the cusp in angle and energy for grazing angle ion-surface collisions is especially prominent in Figures 5.17 and 5.18. The angular broadening of the convoy peak is more significant at the top of peaks than the energy broadening of the cusp. Meanwhile convoy electron distributions for ion-foil and ion-atom collisions show a cusp-like peak at the velocity equal to that of the projectile. As being reported by de Ferrariis and Baragiola[24] or the first time, followed by H. Winter *et al.* and Hasegawa *et al.*,[4,5] all the same experimental results were observed even though the target crystal and the projectile energies are all different. So far the broadening in angle and energy is the most remarkable feature of convoy electrons at grazing angle incidence of ions on a solid surface. H. Winter *et al.* explained this broadening by the DIP leading to an alteration of the final state Coulomb interaction between the electron and the projectile near the surface.

The broad peaks in the high energy half of the spectra of Figure 5.18 are due to the binary encounter electrons populated at the energies less than 4 times cusp electron energy since the binding energy of the electron in initial state is considered.

As seen in Figures 5.19 and 5.20 the most probable peak energy of the convoy electrons at grazing angle scattering was higher than the energy of electrons having the same velocity of projectile. The dependence of the convoy electron acceleration on the electron

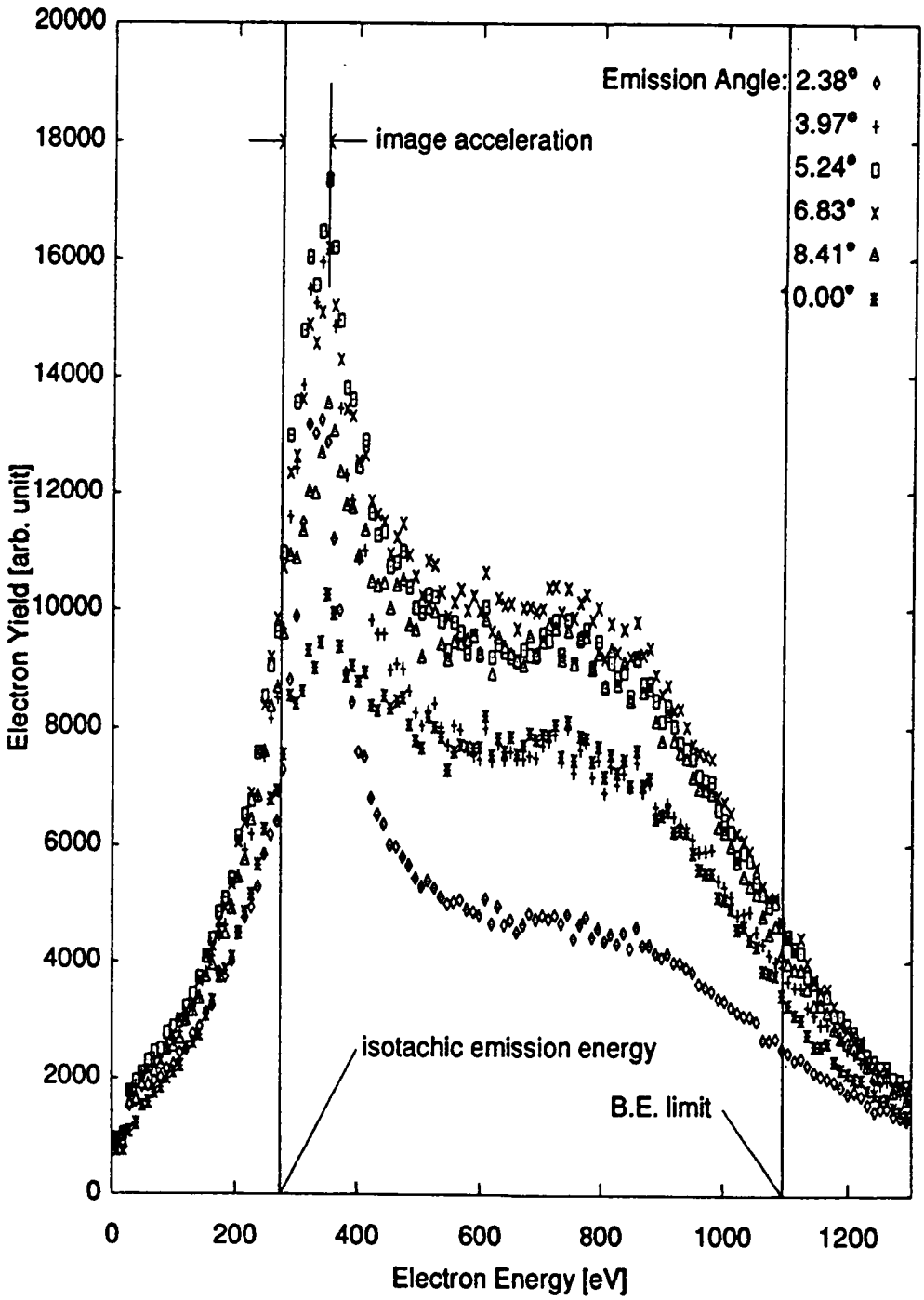



**Figure 5.19**: Energy distribution of electrons with varying emission angles for 0.5 MeV/u $C^{3+}$ + Si(100): 0.52°

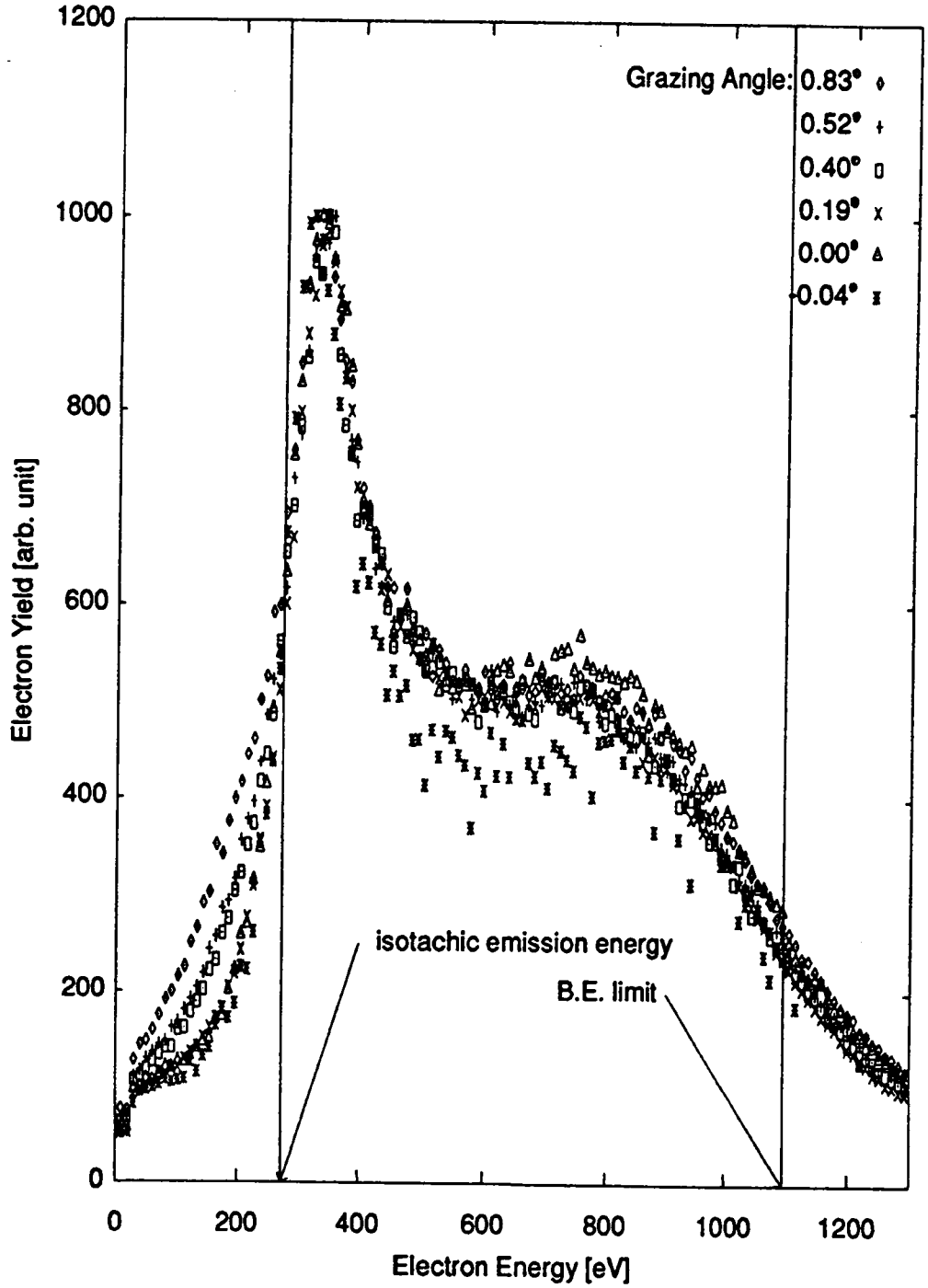



**Figure 5.20**: Energy distribution of electrons for 0.5 MeV/u $C^{3+}$ on Si(100) crystal

emission angle was investigated by splitting a spectrum (0.5 MeV/u $C^{3+}$ + Si(100) with 0.52°) into several energy slices in Figure 5.19, with a single position of the analyzer at ~5 degree with respect to the incoming ion trajectory. The maximum shift and probability of the convoy electrons were found at an emission angle of ~5°.

This acceleration has been investigated and explained in terms of the surface wake potential induced by the projectile ion near the surface. The interaction time between the ion and the surface or also between convoy electrons and the dynamic induced potential (DIP) is long enough to distinguish the acceleration of the convoy electrons significantly from other collision systems. The acceleration of convoy electrons by the image charge in the foil transmission experiment is also considered to be important,[77] but such acceleration may be small enough to ignore since at high velocity the time of interaction between the image charge and the convoy electron is very short, while at grazing angle incidence convoy electrons have enough time to interact with the charge induced by the ion.

The FWHM of the 0.5 MeV/u $C^{3+}$ incident on Si (100) with 0.52° as seen in Figure 5.18 increases with emission angle, and the maximum is found around 5.24°. This result is consistent with the result reported by M. Hasegawa *et al.*[4] for 0.98 MeV $He^{+}$ ions incident on SnTe(001).

The acceleration by the DIP is seen in Figure 5.19, the isotachic emission energy is 274 eV marked by a vertical line, and the shift to the high energy side is approximately 75 eV. Additionally the convoy electrons are also accelerated away from the surface, as seen in Figure 5.17, where the maximum of the polar angle distribution of the convoy peak is about 5° above the direction of the specularly reflected ion away from the surface.

# VI. CONCLUSION

A new experimental setup was presented to measure convoy electrons produced in collisions of relatively slow (0.3 and 0.5 MeV/u) carbon ions at grazing incidence with atomically clean and flat silicon(100) surfaces, comparing with similar ion-gas and ion-foil collisions. By utilizing a 160° spherical electrostatic analyzer with a position sensitive detector, it was possible to furnish more details in multiply differential electron emission than in singly differential cross section. Thus, large uncertainties in the angular distribution, which result in traditional measurements from a variety of fluctuating conditions between separate observation angle measurements, are eliminated.

As reported previously[26,27] and predicted by the Monte Carlo simulation performed by Reinhold[28] (and which implicates a rainbow scattering mechanism as responsible for forming the convoy peak), the grazing incidence convoy electron peak is dramatically broadened in comparison with those produced in ion-atom and ion-foil collisions, and in addition is accelerated by about 70 eV relative to the isotachic emission energy in the case of 0.3 MeV/u $C^{3+}$ projectiles grazing silicon (100) surfaces. In the present unique multiply differential measurements, the peak broadening becomes especially apparent in emission angle during a single positioning of the electron spectrometer.

The energy and angular shifts observed for 0.5 MeV/u $C^{3+}$ ions incident on Si(100) varied with changing incidence angle. The maximum energy shift of 79 eV relative to the

isotachic emission energy and the maximum angular shift of 5° with respect to the specularly reflected beam were observed at a grazing angle of 0.5°. Moreover, a charge state dependence of the acceleration of the convoy electrons towards the higher energy side for 0.5 MeV/u carbon ion beams was identified. However, this acceleration was found to be independent of the ionic charge state for 0.3 MeV/u carbon ion beams. At low ion velocities, the ions reach an equilibrium charge state very early in the collision so that the wake amplitude is independent of the incoming charge state; this is apparently not the case for higher ion velocities.

Independent study of the convoy peak separated from binary encounter emission using non linear least-squares fitting with empirical model functions also indicates that the shift in angle and energy is the general trend for grazing angle ion-surface collisions. In the case of the maximum shift for 0.5 MeV/u $C^{3+}$ at a grazing angle of 0.5°, the ratios of energy shifts between $C^{2+}$ and $C^{3+}$ were obtained as $\frac{\Delta E_{3+}}{\Delta E_{2+}}|_{spectrum} \approx 1.4$, $\frac{\Delta E_{3+}}{\Delta E_{2+}}|_{fit} \approx 1.2$. These results simply imply that the convoy electron peak is pulled towards the high energy side due to an adjacent, broad binary encounter emission peak ("peak pulling").

For ion-atom collisions, an experiment to observe consequences of the breakdown of the Born approximation at low collision velocities was performed in the velocity range of 3.0 to 16.2 au for $O^{5+}$ ion beams, by employing the multipole moment expansion method to the doubly differential emission cross section. The behaviors of the multipole moments $\beta_2$ and $\beta_4$ as well as $\beta_1$ with respect to the ion velocity agree well with theoretically predicted ones. The prediction that the first Born approximation fails at velocities below twice the

ground state electronic orbiting velocities was confirmed. Inadequacy of the second Born approximation at velocities below $v \approx Z_P$ was also observed.

Finally, experimentally determined values of $\beta_k$ from a recent ion-foil experiment confirms the need for high-order multipole terms to describe multiple collision production mechanisms and the rapid electronic evolution of highly excited states inside the bulk. A recent theoretical calculation by G. Szabo *et al.*[71] is in agreement with these experimental observation.

While the connection between electron emission observations from a grazing angle ion-surface collision system and theory is still in an immature state, the latest measurement of the yield of convoy electrons per incoming ion for 0.5 MeV/u $C^{3+}$ + Si(100) at an angle of 0.1°, by integrating over the angular and energy region of the convoy peak, agrees with a theoretically obtained number, ~0.1 electrons per ion.[78] This outcome is a signature of an agreement between our experiment and the theory in permitting comparison with predictions of model calculations of grazing angle ion-surface interaction.

# REFERENCES

[1] M. Breinig, S. B. Elston, S. Huldt, L. Liljeby, C. R. Vane, S. D. Berry, G. A. Glass, M. Schauer, I. A. Sellin, G. D. Alton, S. Datz, S. Overbury, R. Laubert, M. Suter, Phys. Rev. A 25, 3015 (1982); K. O. Groeneveld, W. Meckbach, I. Sellin, and J. Burgdörfer, Comments At. Mol. Phys. 4, 187 (1984).

[2] P.M. Echenique, R.H. Ritchie, N. Barberan and J. Inkson, Phys. Rev. B23, 6486 (1981)

[3] Y.H. Ohtsuki, *Charged Beam Interactions with Solids* (Taylor & Francis, London, 1983)

[4] M. Hasegawa, K. Kimura, M. Mannami, J. Phys. Soc. Japan 57, 1834 (1988)

[5] H. Winter, P. Strohmeier, J. Burgdörfer, Phys. Rev. A39, 3895 (1989)

[6] A. Koyama, Y. Sasa, H. Ishikawa, A. Misu, K. Ishii, T. Iitaka, Y. H. Ohtsuki, M. Uda, Phys. Rev. Lett. 65, 3156 (1990)

[7] G. B. Crooks, M. G. Rudd, Phys. Rev. Lett. 25, 1599 (1970)

[8] K. G. Harrison, M. Lucas, Phys. Lett. 33A 149 (1970); and 35A, 402 (1970)

[9] J. Macek, Phys. Rev. A1, 235, (1970)

[10] K. Dettmann, K. G. Harrison, M. W. Lucas, J. Phys. B: At. Mol. Phys. 7,269 (1974)

[11] C. R. Vane, I. A. Sellin, M. Suter, G. Alton, S. B. Elston, P. M. Griffin, R. S. Thoe, Phys. Rev. Lett. 40, 1020 (1978)

[12] R. Shakeshaft, L. Spruch, Phys. Rev. Lett. 41, 1037 (1978); Rev. Mod. Phys. 51 369, (1979)

[13] M. Breinig, S. B. Elston, I. A. Sellin, L. Liljeby, R. Thoe, C. R. Vane, H. Gould, R. Marrus, R. Laubert, Phys. Rev. Lett. 45, 1689 (1980)

[14] M. W. Lucas, W. Steckelmacher, J. Macek, J. E. Potter, J. Phys. B13, 4833 (1980)

[15] J. Macek, J. E. Potter, M. M. Duncan, M. G. Menendez, M. W. Lucas, W. Steckelmacher, Phys. Rev. Lett. 46, 1571 (1981)

[16] F. Drepper, J. S. Briggs, J. Phys. B: At. Mol. Phys. 9, 2063 (1976)

[17] J. Burgdörfer, M. Breinig, S. B. Elston, I. A. Sellin, Phys. Rev. A28, 3277 (1983)

[18] M. H. Day, J. Phys. B13, L65 (1980)

[19] J. S. Briggs, M. H. Day, J. Phys. B13, 4797 (1980)

[20] W. Meckbach, I. B. Nemirovsky, C. R. Garibotti, Phys. Rev. A24 1793 (1981)

[21] D. H. Jakubaßa-Amundsen, J. Phys. B: At. Mol. Opt. Phys. 23 3335 (1990)

[22] K. F. Man, W. Steckelmacher, M. W. Lucas, J. Phys. B: At. Mol. Phys. 19 401 (1986)

[23] W. Oswald, R. Schramm, H-D. Betz, Phys. Rev. Lett. 62 1114 (1989)

[24] L. F. de Ferrariis, R. A. Baragiola, Phys. Rev. A33 4449 (1986)

[25] J. Burgdörfer, in *Fundamental Processes and Applications of Atoms and Ion*, Ed: C. D. Lin, World Scientific Publ., Singapour, (1993)

[26] T. Iitaka, Y. H. Ohtsuki, A. Koyama, H. Ishikawa, Phys. Rev. Lett. 65 3160 (1990)

[27] K. Kimura, M. Tsuji, M. Mannami, Phys. Rev. A46, 2618 (1992)

[28] C. O. Reinhold, J. Burgdörfer, K. Kimura, M. Mannami, Phys. Rev. Lett. 73,2508 (1994)

[29] S. Geltman, *Topics in Atomic Collision Theory*, Academic Press, (1969)

[30] L. H. Thomas, Proc. R. Soc. A114, 561 (1927)

[31] M. W. Lucas, W. Steckelmacher, J. Macek, J. E. Potter, J. Phys. B13, 4833 (1980)

[32] N. Stolterfoht, D. Schneider, D. Burch, H. Wieman, J. S. Risley, Phys. Rev. Lett. 33, 59 (1974)

[33] N. Stolterfoht, in *Structure and Collisions of Ions and Atoms*, edited by I. A. Sellin, (Springer-Verlag Press, 1978), p. 155-199

[34] C. Kelbch, R. Koch, S. Hagmann, K. Ullmann, H. Schmidt-Bocking, C. O. Reinhold, D. R. Schultz, R. E. Olson, G. Kraft, Z. Phys. D22, 713 (1992)

[35] D. H. Lee, P. Richard, T. J. M. Zouros, J. M. Sanders, J. L. Shinpaugh, H. Hidmi, Phys. Rev. A41, 4816 (1990)

[36] D. Brandt, Phys. Rev. A27, 1314 (1983)

[37] J. Mathew, R. L. Walker, *Mathematical Methods of Physics* (Benjamin, N.Y., 1964);

B. Friedman, Principles and Techniques of Applied Mathematics (Wiley,N.Y., 1956)

[38] M. R. C. McDowell, J. P. Coleman, *Introduction to the Theory of Ion-atom Collisions* (North-Holland, 1969)

[39] E. Fermi, Phys. Rev. 56, 1242 (1939), and 57, 485 (1940)

[40] J. Neufeld, R. H. Ritchie, Phys. Rev. 98, 1632 (1955)

[41] M. H. Day, Phys. Rev. Lett. 44, 752 (1980)

[42] W. Brandt, R. H. Ritchie, Phys. Lett. 62A, 374 (1977)

[43] V. N. Neelavathi, R. H. Ritchie, Phys. Rev. Lett. 33, 302 (1974)

[44] P. M. Echenique, F. Flores, R. H. Ritchie, in *Solid State Physics*, edited by H. Ehrenreich and D. Turnbull (Academic, San Diego, 1990), Vol. 43, p.230

[45] V. N. Neelavathi, R. H. Ritchie, Phys. Rev. Lett. 33, 670(E) (1974), and 34 560(E) (1975)

[46] R. H. Ritchie, W. Brandt, P. M. Echenique, Phys. Rev. B14, 4808 (1976)

[47] R. Kawai, N. Itoh, Y. H. Ohtsuki, Surf. Sci. 114, 137 (1982)

[48] D. Chan, P. Richmond, Surf. Sci. 39, 437 (1973)

[49] P. M. Echenique, R. H. Ritchie, W. Brandt, Phys. Rev. B20, 2567 (1979)

[50] S. B. Elston, Nucl. Inst. and Meth. B24/25, 214 (1987)

[51] Texas Instruments Silicon Products Department, Sherman, TX

[52] B.A. Joyce, Surface Science 35 (1973) 1-7

[53] B. S. Swartzentruber, Y. W. Mo, M. B. Webb, and M. G. Lagally, J.Vac. Sci. Technol. A, Vol. 7, No.4, Jul/Aug (1989)

[54] E. M. Purcell, Phys. Rev. 54 818 (1938)

[55] R. DeSerio, Rev. Sci. Instrum. 60, 381 (1989)

[56] Surface Science Laboratories, Inc., Model 239G Position Sensitive Detector, 4151 Middlefield Rd., Palo Alto, CA 94303

[57] K. D. Sevier, Low Energy Electron Spectrometry, (John Wiley, New York, 1972).

[58] S. D. Berry, G. A. Glass, I. A. Sellin, K. -O. Groeneveld, D. Hofman, L.H. Anderson, M. Breinig, S. B. Elston, P. Engar, and M. M. Schauer, Phys. Rev. A 31, 1392 (1985)

[59] S. B. Elston, unpublished

[60] C. O. Reinhold, J. Burgdörfer, K. Kimura, and M. Mannami, Phys. Rev. Lett, 73, 2508 (1994)

[61] J. P. Gibbons,S. B. Elston, K. Kimura, R. DeSerio, I. A. Sellin, J. Burgdorfer, J. P. Grandin, A. Cassimi, X. Husson, L. Liljeby, M. Druetta, Phys. Rev. Lett. 67, 481 (1991)

[62] M. Lucas, W. Steckelmacher, J. Macek, J. E. Potter, J. Phys. B13 (1980)

[63] J. S. Briggs, M. H. Day, J. Phys. B13 4797 (1980)

[64] W. Meckbach, I. B. Nemirovsky, and C. Garibotti, Phys. Rev. A24 1793 (1981)

[65] F. Eager, E. N. Sikafus, Rev. Sci. Instrum. 48 1269 (1977)

[66] L. C. Northcliffe, and R. F. Shilling, Nuclear Data Tables A7, 233(1970)

[67] R. Shakeshaft, L. Spruch, Phys. Rev. Lett, 41, 1037 (1978)

[68] S. D. Berry, S. B. Elston, I. A. Sellin, M. Breinig, R. DeSerio, C. E. Gonzalez-Lapera, L.Liljeby, J. Phys. B: At. Mol. Phys. 19 L149 (1986)

[69] S. D. Berry, *Multipole Characterization of the Electron Loss to the Continuum Cross Section at Asymptotically High Velocities*, Ph. D. Dissertation, University of Tennessee, 1985 (unpublished)

[70] J. S. Briggs, and F. Drepper, J. Phys. B 11, 4033 (1978)

[71] G. Szabo, J. Wang, J. Burgdörfer, Phys. Rev. A 48, R3414 (1993)

[72] J. P. Gibbons, *Production and Transport of Convoy Electrons,* Ph. D. Dissertation, University of Tennessee, 1991 (unpublished)

[73] Y. Yamazaki, H. -P. Hülskötter, M. Breinig, R. DeSerio, J. Burgdörfer, T. Underwood, J. Gibbons, I. A. Sellin, Phys. Rev. A 34, 4493 (1986)

[74] H. -P. Hülskötter, M. Breinig, J. Burgdörfer, R. DeSerio, S. B. Elston, C. E.

Gonzalez-Kepera, C. Kelbch, H. Schmidt-Böcking, I. A. Sellin, Y. Yamazaki, Phys. Rev. A 33, 2156 (1986)

[75] Uwe Thumm, J. Phys. B: At. Mol. Opt. Phys. 25, (1992) 421-435

[76] H. Lebius, R. Minniti, J. Y. Lim, S. B. Elston, (to be submitted)

[77] J. Burgdörfer, Nucl. Instrum. & Methods B24/25 139 (1987)

[78] R. Minniti, C. O. Reinhold, private communication

# APPENDICES

# Appendix A

## - Determination of Multipole Moments

The experimentally observed electron distribution from ion-atom and ion-foil collisions is compared with the theoretical prediction of the CE distribution by being fitted to the multipole expansion discussed in Chapter IV. If the DDCS is expressed in projectile frame variables and the observed distribution is in the laboratory frame, a transformation of the projectile frame DDCS to a lab frame DDCS in energy and angle is necessary.

From the form of $\frac{d^2\sigma}{dEd\Omega}|_{proj} = \sigma_0 \sum_{jk} \beta_{jk} \cdot v_{e'}^j \cdot P_k(\cos\theta_{e'})$, the DDCS in lab frame is simply derived by a classical treatment, resulting in

$$\frac{d^2\sigma}{dEd\Omega}|_{lab} = \frac{v_e}{v_{e'}} \sigma_0 \sum_{jk} \beta_{jk} \cdot v_{e'}^j \cdot P_k(\cos\theta_{e'}) \qquad (A.1)$$

where the primed are the projectile frame variables. The conclusive electron distribution is then described by

$$Q(E_e, \theta_e) = \int_{\Delta\Omega_e} d\Omega_e' \cdot G(\Omega_e, \Omega_e') \cdot \int_{\Delta E_e} dE_e' \cdot R(E_e, E_e') . [\frac{v_e}{v_{e'}} \sigma_0 \sum_{jk} \beta_{jk} \cdot v_{e'}^j \cdot P_k(\cos\theta_{e'})]. \qquad (A.2)$$

However, for the case of a larger azimuthal slice width than the angular resolution function, the above equation should be integrated over PSD's $y$-direction, i.e. $y$-emission angle. The actual experimental intensity becomes

$$Q^{exp}(E_e, \theta_e) = 2 \int_0^{w/2} Q(E_e, \theta_e(\sqrt{x^2+y^2})) dy. \qquad (A.3)$$

With the aid of previously described calculations in numerical integration over energy as well as analytical integration over angle[1], the 2-dimensional fitting routine could evaluate the triple integral of a typical 64 by 128 channel spectrum in several hours with a properly chosen initial parameter set. The complete fitting program is presented in Appendix B.

In this program the multipole expansion function is truncated to $0 \le k \le 5$ ($j = 0, 1$), $k = 6,\ 8,\ 10$ ($j = 0$), since the negligible contributions of the fitted $\beta_{jk}$ to $\chi_\nu^2$ and the relatively large errors in the fitted $\beta_{jk}$ beyond $k = 10$ are expected. However, the trend of $\beta_{06}$, $\beta_{08}$, and $\beta_{010}$ with increasing target thickness for ion-foil collisions suggests that further expansion terms may be needed for the excitation or the electron capture-loss to the projectile continuum state with the bulk material of the target. The first order velocity term, $j = 1$ ($k \le 5$), can be added to adequately fit the spectra although there exist theoretical predictions only for $j = 0$. As seen in Figure A.1 the contours with $j = 1$ term are not significantly different from the $\beta_{00}$ term except for the outer few contours. The total number of $\beta_{jk}$s for the fitting are determined only after many early attempts, including both the number of angular moments $k$ and radial velocity corrections $j$.

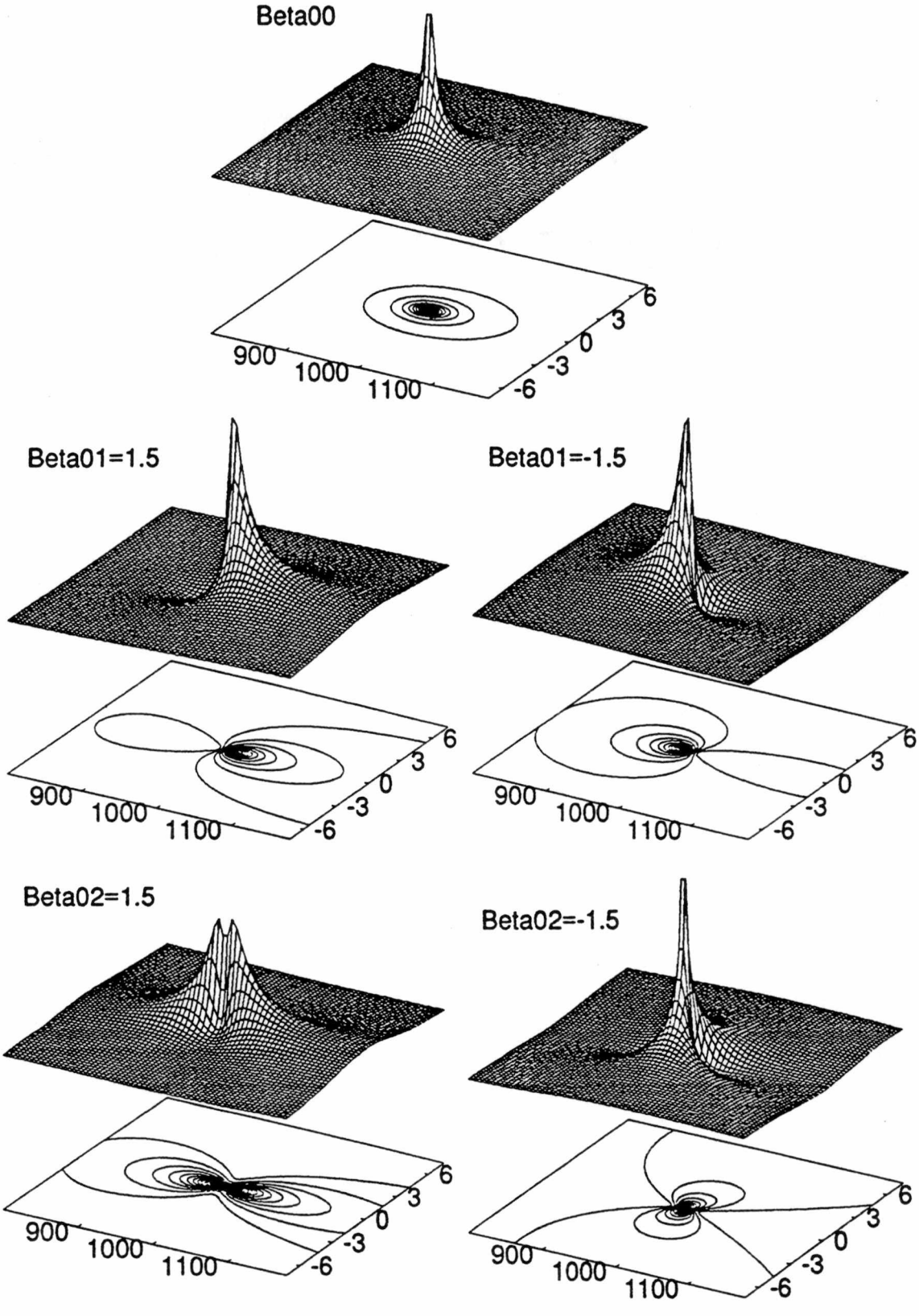
Except these 15 $\beta_{jk}$s, there are 16 additional parameters to facilitate the fitting program as shown in Appendix B. Total 31 parameters are allowed to have arbitrary subsets of parameters to control the floating ranges between the maximum and the minimum values. Also this subset can be chosen to hold some parameters fixed, while the remaining parameters are varied during the calculation. In this way various combinations of fitted parameters could be easily tried to test the significance of adding particular terms. This program automatically searches the minimum value of $\chi_\nu^2$ until the convergence value of $\chi_\nu^2$ is reached during the calculation.

Plots in Figure A.1 show a number of generated electron distributions with every possible multipole expansion term, $\beta_{jk}$ ($k = 0-5$ ($j = 0, 1$), $k = 6, 8, 10$ ($j = 0$)) $= \pm 1.5$, combined with the isotropic contribution $\beta_{00} \equiv 1.0$ (Dettmann cusp term). The contours represent multipoles of 10 % intensities of the maximum yield occurring at the given center ($E_e = 986$ eV, $\theta_e = 0.0°$) of each plot with arbitrary resolution values of $\frac{\Delta E}{E} = 0.011$ and $\delta\theta = 0.005$ in the convolution.

---

[1] J. P. Gibbons, *Production and Transport of Convoy Electrons, Ph.D. Dissertation*, the University of Tennessee, 1991 (unpublished)

**Figure A.1**: 3 dimensional plots of generated multipole distributions displaying combinations of all possible expansion terms with the Dettmann cusp term ($\beta_{00}$ =1.0). The magnitudes of the $\beta_{jk}$ values are ± 1.5. Scales in $x$ and $y$ axes indicate directions specifying energy [eV] and x-Emission angle [degree], respectively. The contours represent multipoles of 10 %, with the maximum yield occurring at the given center ($E_e$ = 986 eV, $\theta_e$ = 0.0°) of each plot, corresponding to $v_{e'} = 0$. (continued on next pages)


Beta00
900
1000
1100
6
3
0
-3
-6
Beta01=1.5
Beta01=-1.5
Beta02=1.5
Beta02=-1.5

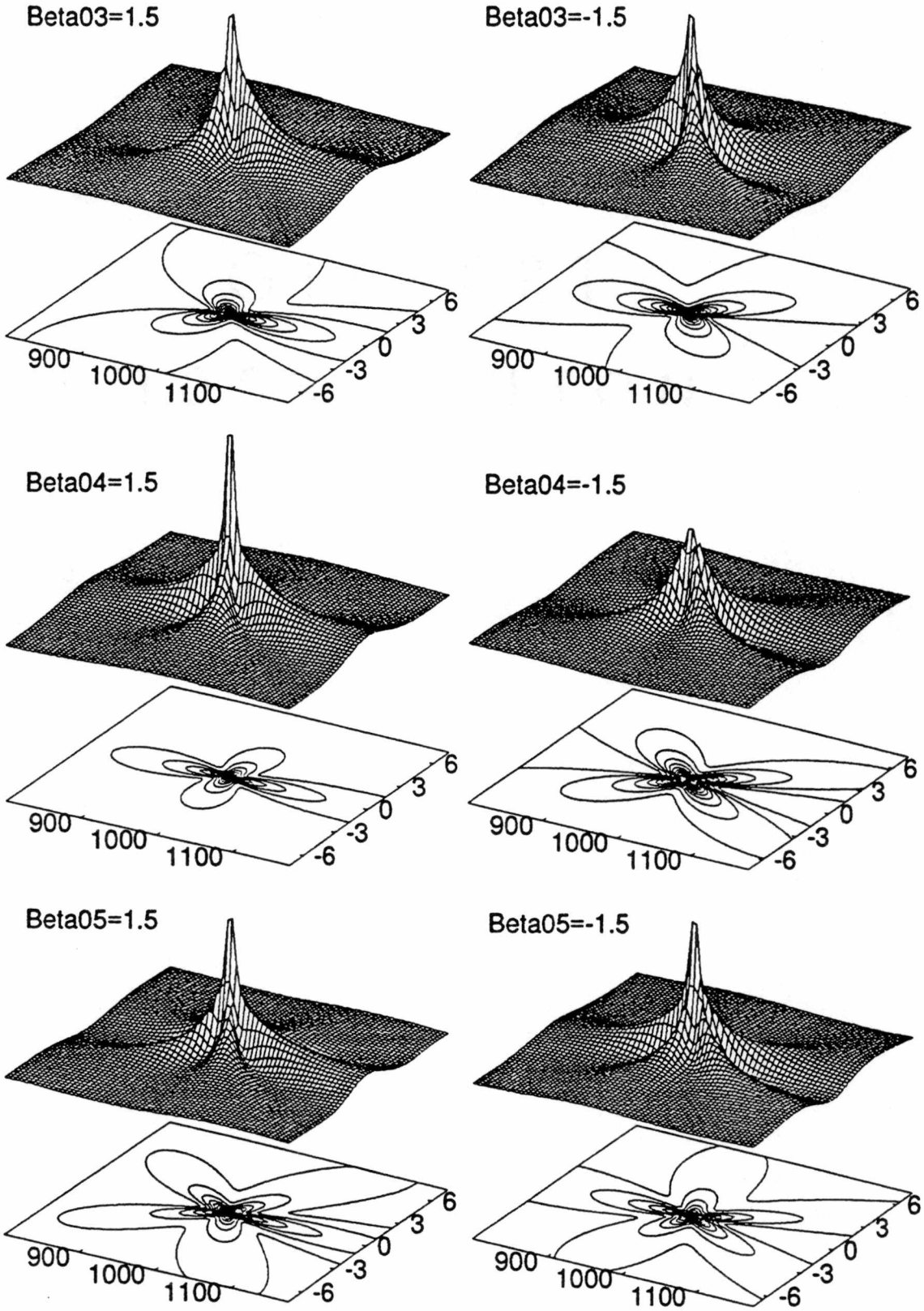



**Figure A.1: (continued)**

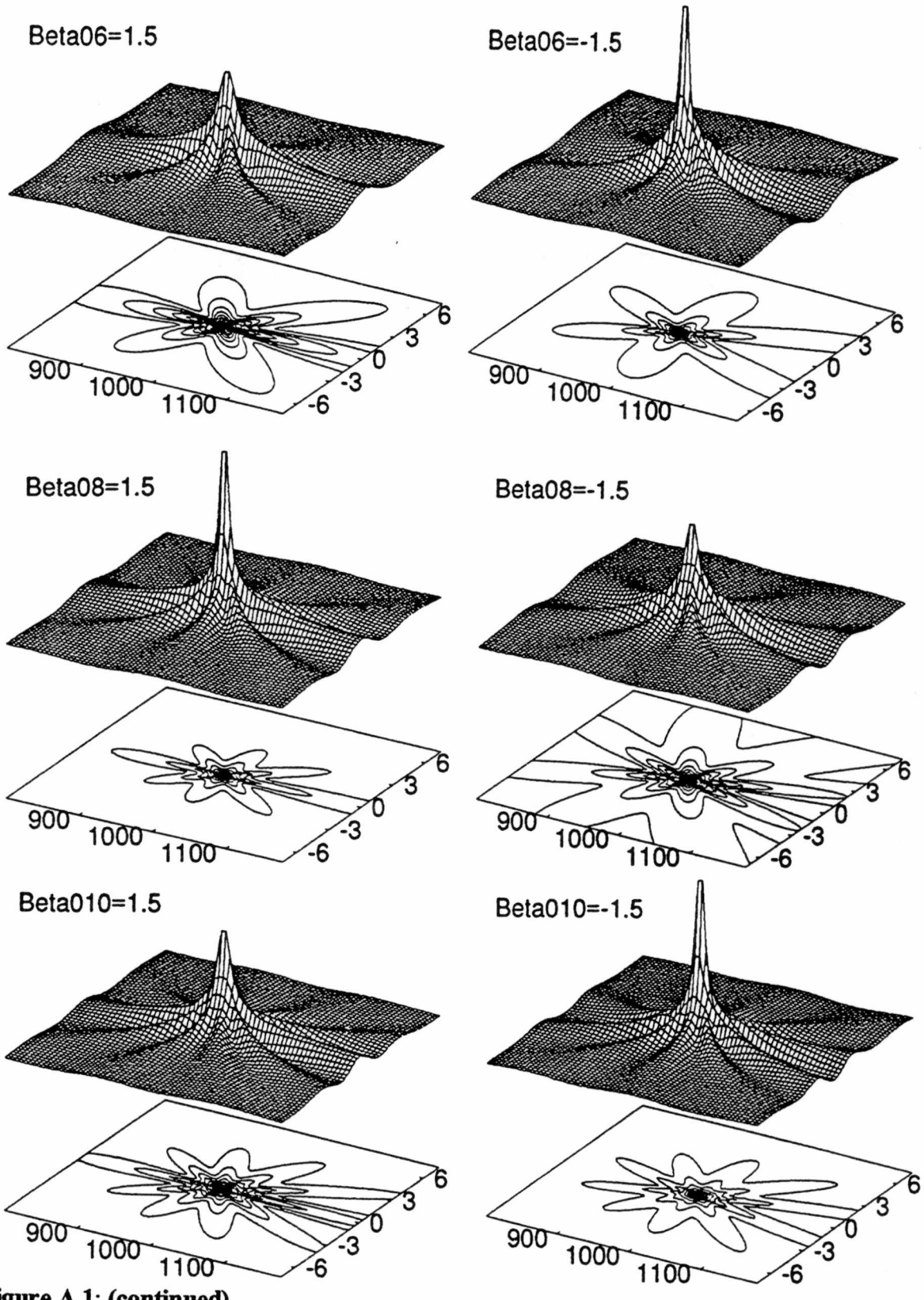



**Figure A.1: (continued)**

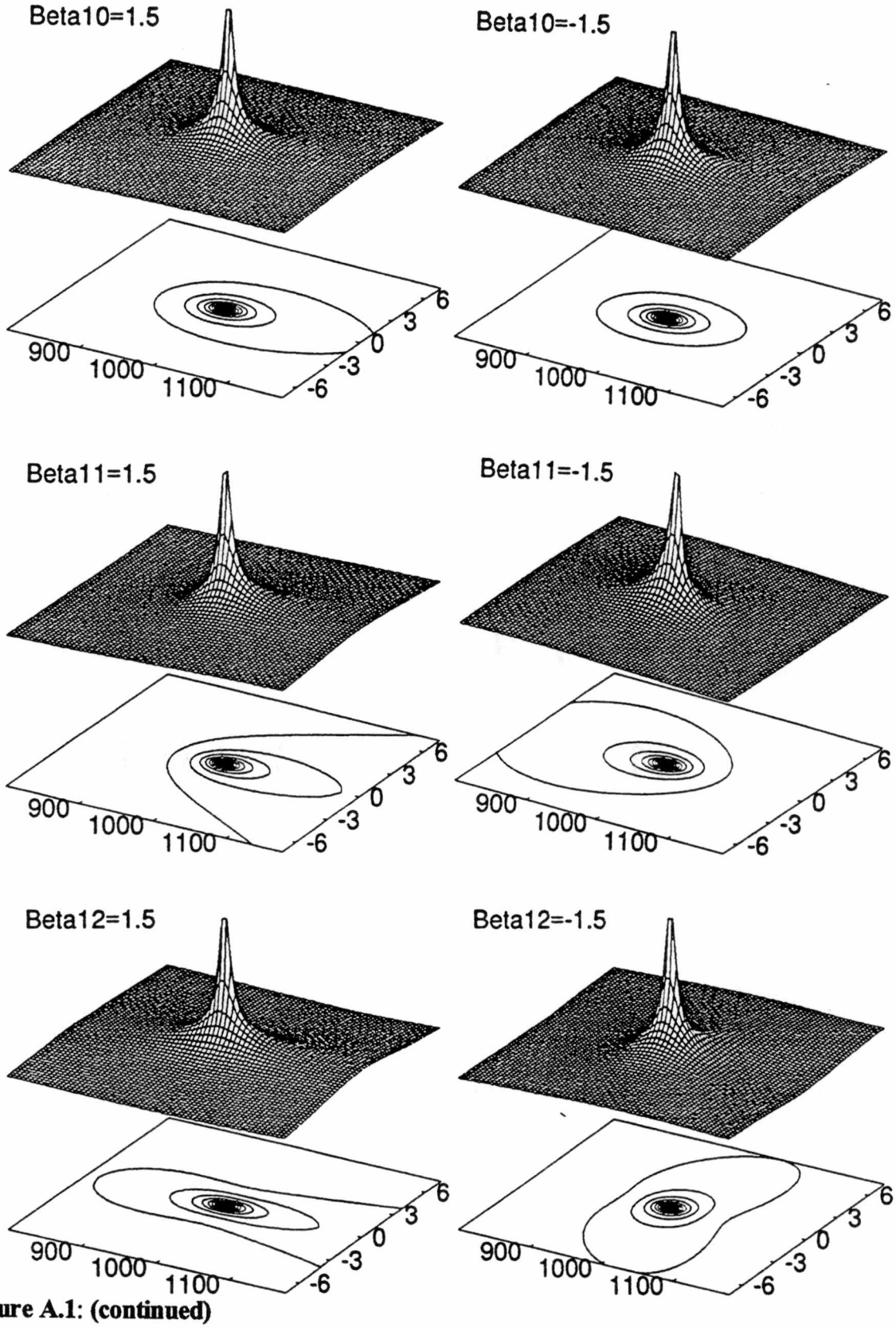



**Figure A.1: (continued)**

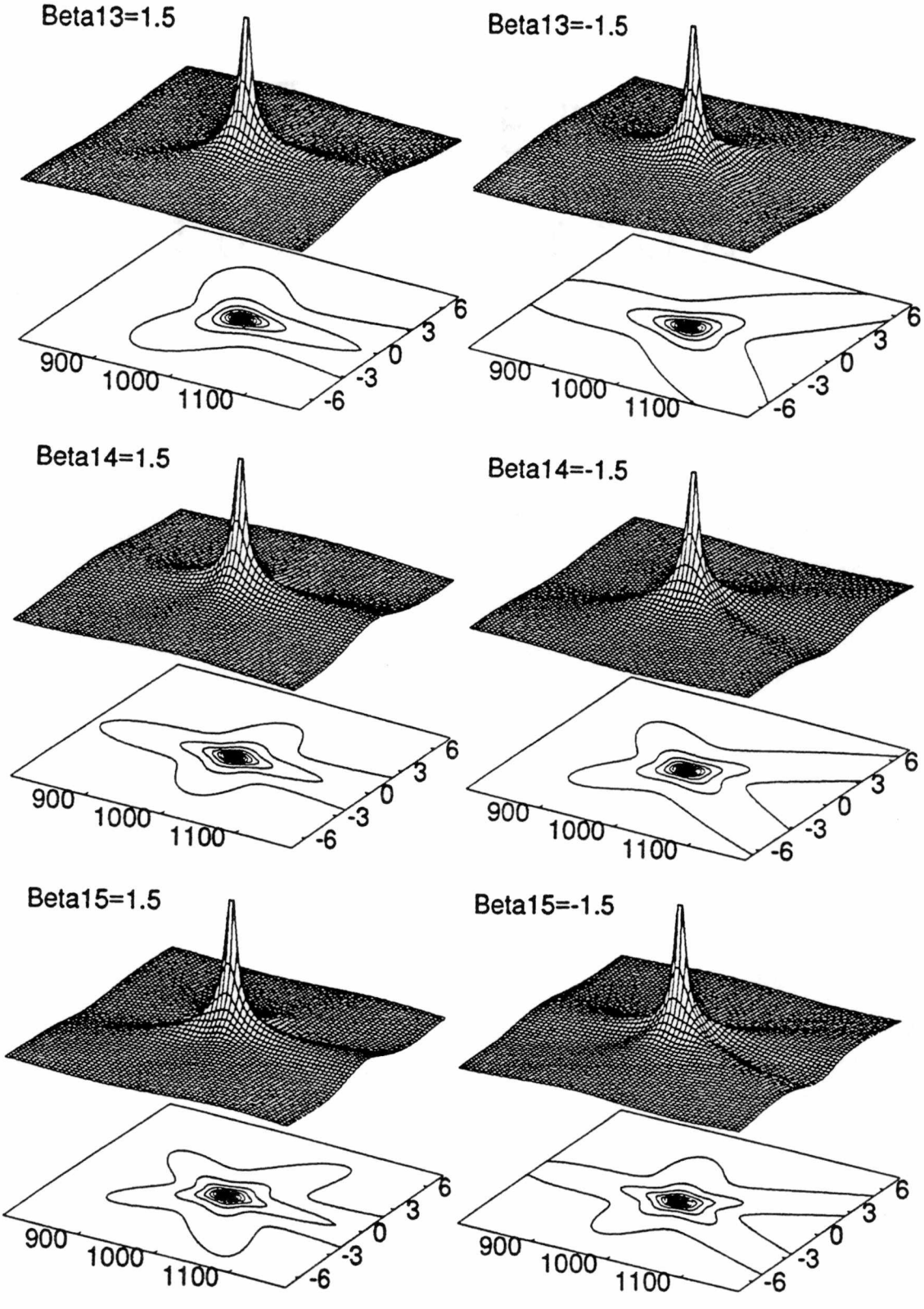



**Figure A.1: (continued)**

# Appendix B

## - 2 Dimensional Fitting Program

*N* experimental data points are fitted to two dimensional model which has *n* adjustable parameters *a, b, c...* The model predicts a functional relationship between the measured independent and dependent variables,

$$z(x,y) = z(x,y;a,b,c...) \tag{B.1}$$

where the dependence on the parameters is indicated explicitly on the right-hand side.

The maximum likelihood estimate of the model parameters is obtained by minimizing the quantity

$$\chi^2 = \Sigma[\{z(x,y) - z(x,y;a,b,c...)\}/\sigma]^2 \tag{B.2}$$

called the "chi-square." A "typical" value of $\chi^2$ for a "moderately" good fit is $\chi^2 = DOF$ where $DOF = N\text{-}M$, $M$ = # of parameters. More precise is the statement that the statistic $\chi^2$ becomes normally distributed with a mean $DOF$ and a standard deviation $\sqrt{2 \cdot DOF}$.

If we take the derivative of above equation with respect to the parameters, we obtain equations which must hold at the $\chi^2$ minimum,

$$0 = \Sigma[\{z(x,y) - z(x,y;a,b,c...)\}/\sigma] \cdot z'(x,y;a,b,c...). \tag{B.3}$$

Above equation is, in general, a set of $M$ nonlinear equations for the $M$ unknown parameters.

It is, however, not convenient to drive an analytical expression for calculating the parameters of a non-linear function *z(x,y)*. Instead, $\chi^2$ must be considered a continuous function of the $n$ parameters describing a hypersurface in $n$-dimensional space. In the **Gradient Search** method of least squares, all the parameters are incremented simultaneously, with the relative magnitudes adjusted so that the resultant direction of travel in parameter space along the gradient of $\chi^2$. In **Parabolic Extrapolation**, instead of searching the $\chi^2$ hypersurface to map the variation of $\chi^2$ with parameters, we can find an approximate analytical function which describes the $\chi^2$ hypersurface and to use this function to locate the minimum directly.

For the best result, the **Gradient-Expansion** algorithm is applied to the program.[2] This algorithm which combines the best features of the gradient search with the method of linearizing the fitting function can be obtained by increasing the diagonal terms of the curvature matrix *ALPHA* by a factor *LAMDA* which controls the interpolation of the algorithm between the two extremes. The value of the constant factor *LAMDA* should be chosen small enough to take advantage of the analytical solution, but large enough that $\chi^2$ decreases.

---

* Get the fitting data file using SidaXfer.exe, then set
  #define 'msfile (64x128)', 'hrfile (128x128)', 'csfile (64x64)' in FitT.h.
* Compile the whole fitting programs with IBM C++ for OS/2.
  command: "icc jyfit.cpp fitt.cpp fitparm.cpp model.cpp bevington.lib
* Set the initial parameter file carefully. (i.e. ASTEP and SLOPE are dependent on the data file size).
* Run JYFIT.
  command: jyfit 'data file' 'parameter file'
* Possible Questions during Running.
  . Change fitting data size from 64x128 to 64x64?
  – Answer should be 'yes', or 'no' according to the ASTEP and the SLOPE inside the parameter file.
  . Fit Cusp?
  – Answer should be 'yes', or 'no'. It will remove the cusp area from the fit. The size is dependent on the ARF and the ERF
  . Enter angular fitting range, 0 - 63.

---

[2] P. R. Bevington, *Data Reduction and Error Analysis for the Physical Sciences,* McGraw Hill Book Company (1969)

– Type the angular fitting range you want. Eq: "23 59"

. Enter energy fitting size, 0 - 63 (if reduced) or 0 - 127.

– Type the energy fitting range you want. Eq: "14 117"

. Save result?

– Answer should be 'yes', or 'no'. If 'yes', you can name your own result file. If 'no', default file, 'default.dat', is produced.

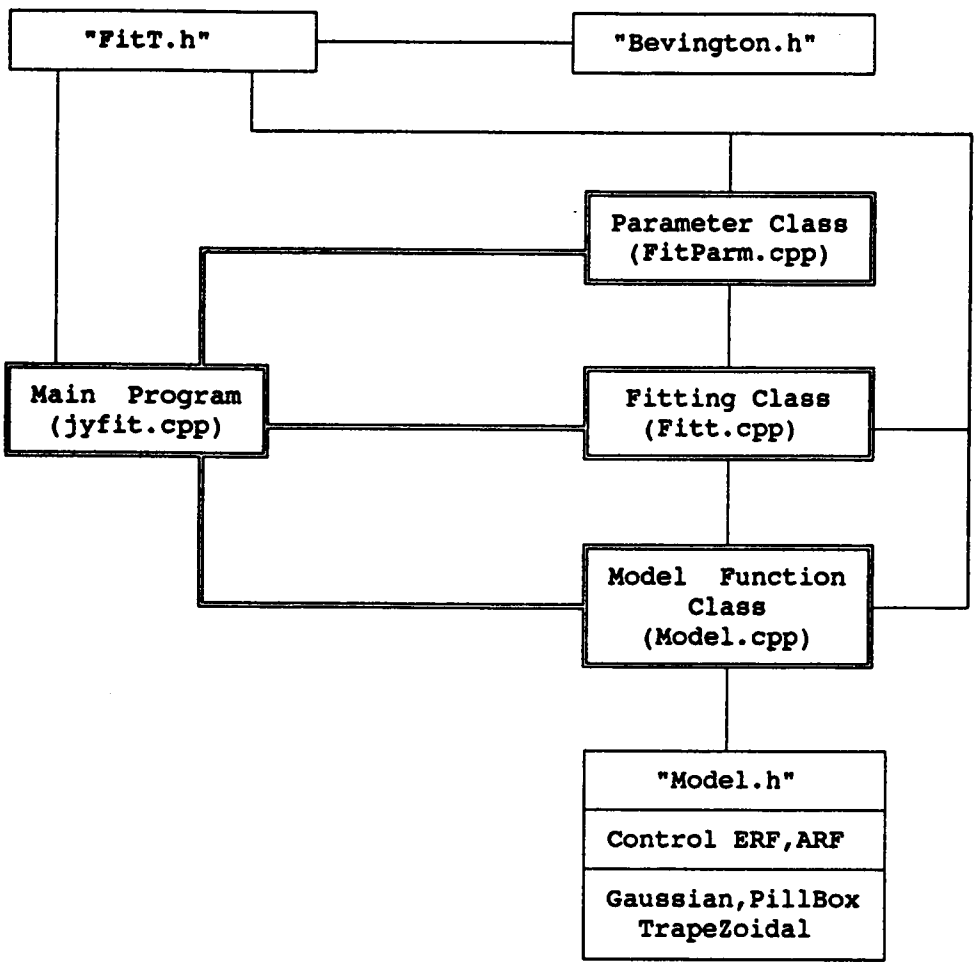
. Save parameter file?

– Answer should be 'yes', or 'no'. If 'yes', you can name your own parameter file.

. Another parameter?

– Answer should be 'yes', or 'no'. It is not tested completely yet. 'No' is recommended. If 'yes', you can change each component of a parameter file, and run again with that parameter set.

FLOW CHART for JYFIT



* Parameter File Format
. Control: -3 Not used in fitting, set to 0.0 inside program
-2 Fixed, retain its original value
-1 floating > Lower Limit
0 floating
1 floating < Upper Limit
2 floating > Lower Limit, and < Upper Limit
. Maximum matrix size is 50 by 50 not including control -2 or -3.
More than 50 by 50 matrix can be used by changing the size in Bevington.lib.
. PillBox ARF == using only ARFin
Trapezoidal ARF == using ARFin and ARFout
Trapezoidal ERF == using TW and FWHM
Gaussian ERF == using only FWHM

. parm[0] to parm[19] : Beta Parameter, or whatever.
parm[20] = TILT : % slope in x-angle
parm[21] = ACEN : Angle center
parm[22] = ECEN : Energy center
parm[23] = TW : % of ECUSP for Trapezoidal ERF
parm[24] = FWHM : % of ECUSP for Trapezoidal and Gaussian ERF
parm[25] = ARFin : radian for PillBox and Trapezoidal ARF
parm[26] = ARFout : radian for Trapezoidal ARF
parm[27] = ASTEP : Angular step in radian
parm[28] = ORIGIN : Vdac origin in Volt
parm[29] = SLOPE : Vdac slope in Volt
parm[30] = Aconst : Analyzer conversion constant in eV/V
parm[31] = Bconst : Analyzer conversion constant in eV
parm[32] = ECUSP : Theoretical cusp energy in eV
parm[33] = BKGDa : Tilting slope in energy channel
parm[34] = BKGDb : Back ground counts in energy channel
parm[35] = SLICE : # of channels in y-angle

| | | | | | |
|---|---|---|---|---|---|
| 1 | 36 | : Number of Parameters | | | |
| 2 | 100 | : Number of Maximum Iterations | | | |
| 3 | 1.000000e-05 | : Chi Square Convergence | | | |
| 4 | | | | | |
| 5 | Parameter | Lower Limit | Upper Limit | Delta Parm | Control |
| 6 | | | | | |
| 7 | 140.4937 | 1.000000 | 100000.0 | 1.000000 | -1 |
| 8 | 2.582546 | -10.00000 | 10.00000 | 0.01000000 | 2 |
| 9 | -0.2257644 | -1.000000 | 1.000000 | 0.01000000 | 2 |
| 10 | -1.293704 | -10.00000 | 10.00000 | 0.01000000 | 0 |
| 11 | -0.04572034 | -1.000000 | 2.000000 | 0.01000000 | 2 |
| 12 | 0.1276098 | -10.00000 | 10.00000 | 0.01000000 | 0 |
| 13 | 0.004229448 | -1.000000 | 1.000000 | 0.01000000 | 2 |
| 14 | 0.3840455 | -10.00000 | 10.00000 | 0.01000000 | 0 |
| 15 | -0.009087751 | -1.000000 | 2.333330 | 0.01000000 | 2 |
| 16 | 0.3462708 | -10.00000 | 10.00000 | 0.01000000 | 0 |
| 17 | -0.01808387 | -1.000000 | 1.000000 | 0.01000000 | 2 |
| 18 | -0.08237907 | -10.00000 | 10.00000 | 0.01000000 | 0 |
| 19 | 0.006190471 | -1.000000 | 2.410900 | 0.01000000 | 2 |
| 20 | 0.03035457 | -1.000000 | 2.449700 | 0.01000000 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| 21 | -0.01607316 | -1.000000 | 2.455200 | 0.01000000 | 2 |
| 22 | 0.0000000 | 1.000000 | 1.000000 | 1.000000 | -3 |
| 23 | 0.0000000 | 1.000000 | 1.000000 | 1.000000 | -3 |
| 24 | 0.0000000 | 1.000000 | 1.000000 | 1.000000 | -3 |
| 25 | 0.0000000 | 1.000000 | 1.000000 | 1.000000 | -3 |
| 26 | 0.0000000 | 1.000000 | 1.000000 | 1.000000 | -3 |
| 27 | 0.1854471 | -10.00000 | 10.00000 | 0.01000000 | 2 |
| 28 | 30.55053 | 10.00000 | 50.00000 | 0.01000000 | 2 |
| 29 | 28.52505 | 10.00000 | 50.00000 | 0.01000000 | 2 |
| 30 | 0.1060660 | 0.0000000 | 10.00000 | 0.01000000 | -2 |
| 31 | 0.9972889 | 0.0000000 | 10.00000 | 0.01000000 | -2 |
| 32 | 0.0050000 | 1.000000 | 1.000000 | 1.000000 | -2 |
| 33 | 0.0050000 | 1.000000 | 1.000000 | 1.000000 | -2 |
| 34 | 0.002438000 | 1.000000 | 1.000000 | 1.000000 | -2 |
| 35 | 0.486000 | 1.000000 | 1.000000 | 1.000000 | -2 |
| 36 | 0.004000 | 1.000000 | 1.000000 | 1.000000 | -2 |
| 37 | 803.7860 | 1.000000 | 1.000000 | 1.000000 | -2 |
| 38 | 0.000000 | 1.000000 | 1.000000 | 1.000000 | -2 |
| 39 | 489.5780 | 1.000000 | 1.000000 | 1.000000 | -2 |
| 40 | 0.00000 | -10.0000 | 10.00000 | 0.0100000 | -2 |
| 41 | -24.91752 | -100.0000 | 1000.000 | 0.1000000 | 2 |
| 42 | 8.000000 | 1.000000 | 64.00000 | 1.000000 | -2 |

* OUTPUT

. During the calculation: iter1.prm, iter2.prm, iter3.prm,....., and default.dat (renew continuously).

. Interim stages of the fitting are transparent. Running another program, 'ModelMain.exe', with iter?.prm
produce the corresponding GnuPlot data file, so that the GnuPlot can show the exact stage.

. After calculation: Result file (result.dat)
Parameter file
GnuPlot data file
.. 'data file'.SGNU; source file
.. 'data file'.FGNU; fitted file
.. 'data file'.RGNU; residue file

* Error Handling

. This fitting program can handle HR, MS, CS datafiles from the SIDA data acquisition program.

. The parameter file should be examined very carefully. First, check the total parameter number, limits, delta parm, and unusual white spaces. Then, check if the initial values are between boundaries. Finally, most important, regulate the angular step and the SLOPE of DAC according to 'size of fitting ranges'. Practically 128 channels can be changed to 64 channels inside the program, so ASTEP and SLOPE in the parm file should be set before changing the corresponding size in the fitting program.

. For putting 'control!=-2,-3' for FWHM..... There is a danger of encountering the 'floating point invalid operation' error. This is due to a smaller FWHM than a SLOPE (or energy step). After any iteration, a new FWHM is calculated. If a new and an old FWHM produce the same returning values, then the difference is "ZERO" inside 'FitT::FDeriv(). This causes the expected error inside 'FitT::Curfit(). It is recommended to turn FWHM contol=-2. If the FWHM is big enough, you can optimize FWHM by putting control=2,1,0, or -1.

. In rare case there may exist an error related to "full swap file". Just change the swappath in 'config.sys' file, and run it again.

. So far all errors inside the fitting program have been removed. However, there are possibilities to have unexpected errors related to the floating value over flow. In this case just reduce the fitting ranges. Once a proper initial parameter set is found, it is unlikely to have the floating point overflow so that its original fitting ranges can be recovered.

* Changing a Model function.

. The model class can be replaced by any arbitrary model functions. In this case, write the complete 'model.cpp' and its corresponding 'model.h'. Then modify the result file routine, FitT::result(), and the parameter file. It should not be difficult if the returning values and the passing values are carefully chosen . So just see the above flow chart or the original model function for a reference.

## /* JYFIT.cpp */

```
#include "fitT.h"
int main(int argc, char **argv)
{
  if (argc != 3) {ErrDisplay(1); exit(-1);}
  time_t timer; struct tm *tblock;

  // Prohibits the underflow exception of the coprocessor
  _control87(EM_UNDERFLOW,EM_UNDERFLOW);

  int ASize,ESize;
  SetSize(argv,ASize,ESize);                    // Set the size of data file

  parmM ParmR(argv);                            // Constructor for parameter
  ParmR.GetParm();                              // Read and get parameters

  Display(0);                                   // Check resolution functions
  Model model(ParmR.Control,ASize,ESize);       // Constructor for model function

  float **Mdata=new float*[ASize];
  for (int i=0;i<ASize;i++)  Mdata[i]=new float[ESize];
  GetData(argv,Mdata,ASize,ESize);

  Range R;                                      // Constructor for fitting range
  fitT fitting(ParmR.nrOfParm,Mdata,R,ParmR.parmBound, ParmR.Control,ASize,ESize);

  char *ans=new char[10],*ansP=new char[10],*parmF=new char[50];  *ans=*ansP='n';
  do {
    if ((*ans=='y')||(*ans=='Y'))  ParmR.ChangeParm();

    /*========= Down Hill Search ==========================================*/
    int iteration=0;
    float oldChiSqr,ChiSqr=1.0e30,fLamda=1.0e-3;
    do { oldChiSqr=ChiSqr;  iteration++;
      timer=time(NULL);  tblock=localtime(&timer);
      Display(1);  cout<<iteration<<"\t"<<asctime(tblock)<<endl;
      fitting.CurFit(model,ParmR.parm,ParmR.DeltaParm,fLamda,ChiSqr);
```

```
      ParmR.WriteParm(argv,parmF,iteration);  // Interim parm. file
      if ((fLamda>1.0e5)||(iteration>=ParmR.nrOfIter)) break;
    } while((oldChiSqr-ChiSqr)/oldChiSqr>ParmR.Convergence);
    timer=time(NULL); tblock=localtime(&timer);
    Display(2); cout<<iteration<<"\t"<<asctime(tblock)<<endl;
    /*========== End of Down Hill Search ==========*/

    fitting.Result(argv,ChiSqr,ParmR.parm,iteration,fLamda,model);
    Display(3); cin>>ansP;
    if ((*ansP=='y')||(*ansP=='Y')) ParmR.WriteParm(argv,parmF,0);
    Display(4); cin>>ans;
  } while ((*ans=='y')||(*ans=='Y'));

  GnuPlotData(argv,Mdata,ParmR.parm,R,ASize,ESize,model);
  delete[] Mdata;  delete[] ans;  delete[] ansP;  delete[] parmF;  delete[] tblock;
  return 0;
}


/*-------------------------------------------------------------------

   FUNCTION: ErrDisplay()
  ------------------------------------------------------------------*/
void ErrDisplay(int Q)
{
  switch(Q) {
    case 1: {cerr<<"Usage:\tJYFIT  'data file'  'parameter file'"<<endl;  break;}
    case 2: {cerr<<endl<<"Input data file doesn't exist."<<endl;  break;}
    case 3: {cerr<<"Input parameter file doesn't exist."<<endl;  break;}
    default: break;
  }
}


/*-------------------------------------------------------------------

   FUNCTION: Display()
  ------------------------------------------------------------------*/
void Display(int P)
{
  switch(P) {
    case 0: {cout<<endl<<"\tThis version is using ";
             #ifdef PillBoxARF
             cout<<"a PILLBOX shape angular, and "<<endl;
             #endif
             #ifdef TrapeZoidalARF
             cout<<"a TRAPEZOIDAL shape angular, and "<<endl;
             #endif
             #ifdef TrapezoidalERF
             cout<<"\ta TRAPEZOIDAL shape energy resolution functions."<<endl;
             #endif
             #ifdef GaussianERF
```

```
            cout<<"\ta GAUSSIAN shape energy resolution functions."<<endl;
            #endif
            break;}
    case 1: {cout<<endl<<endl<<"\t\t\tIteration No. ";   break;}
    case 2: {cout<<endl<<endl<<"\t\tIteration stopped at ";   break;}
    case 3: {cout<<endl<<"\t\tSave parameter file?"<<endl<<"\t\t";   break;}
    case 4: {cout<<endl<<"\t\tAnother parameter set?"<<endl<<"\t\t";   break;}
    default: break;
  }
}


/*=============================================================================

   FUNCTION: SetSize()
  ============================================================================*/
void SetSize(char *argv[],int &ASize,int &ESize)
{
  char *ans=new char[10]; *ans='n';
  ifstream CheckIn;
  CheckIn.open(argv[1],ios::in);
  if(!CheckIn) {ErrDisplay(2); exit(1);}
  CheckIn.close();

  #ifdef Data6464    // CS data file
    ASize=64; ESize=64;
  #endif

  #ifdef Data64128   // MS data file
    cout<<endl<<"Change fitting data size from 64x128 to 64x64?"<<endl;  cin>>ans;
    if ((*ans=='y')||(*ans=='Y')) {ASize=64; ESize=64;}
    else  {ASize=64; ESize=128;}
  #endif

  #ifdef Data128128  // HR data file
    cout<<endl<<"Change fitting data size from 128x128 to 64x64?"<<endl;  cin>>ans;
    if ((*ans=='y')||(*ans=='Y'))  {ASize=64; ESize=64;}
    else  {ASize=128; ESize=128;}
  #endif
}


/*=============================================================================
   FUNCTION: void GetData()
   OPERATION: This function reads data file.
  ============================================================================*/
void GetData(char **argv,float **Mdata,int &ASize,int &ESize)
{
  float **Rdata;
  char string[40];
  FILE *input;
```

```
  input=fopen(argv[1],"r");
  int AS,ES;

  #ifdef Data6464        // CS data file
    AS=64; ES=64;
  #endif
  #ifdef Data64128       // MS data file
    AS=64; ES=128;
  #endif
  #ifdef Data128128      // HR data file
    AS=128; ES=128;
  #endif

  Rdata=new float*[AS];
  for (int i=0; i<AS; i++) {
    Rdata[i]=new float[ES];
    for (int j=0; j<ES; j++) {fscanf(input,"%s",string); Rdata[i][j]=atof(string);}
  }

  #ifdef Data6464        // CS data file
    for (i=0;i<ASize;i++) for (int j=0;j<ESize;j++) Mdata[i][j]=Rdata[i][j];
  #endif
  #ifdef Data64128       // MS data file
    if ((ASize==64)&&(ESize==64))
      for (i=0;i<ASize;i++) for (int j=0;j<ESize;j++) Mdata[i][j]=Rdata[i][2*j]+Rdata[i][2*j+1];
    else
      for (i=0;i<ASize;i++) for (int j=0;j<ESize;j++) Mdata[i][j]=Rdata[i][j];
  #endif

  #ifdef Data128128      // HR data file
    if ((ASize==64)&&(ESize==64))
      for (i=0;i<ASize;i++)
        for (int j=0;j<ESize;j++)
          Mdata[i][j]=Rdata[2*i][2*j]+Rdata[2*i][2*j+1]+Rdata[2*i+1][2*j]+Rdata[2*i+1][2*j+1];
    e lse
      for (i=0;i<ASize;i++) for (int j=0;j<ESize;j++) Mdata[i][j]=Rdata[i][j];
  #endif

  delete[] Rdata;
  fclose(input);
}


/*=============================================================================
   FUNCTION:  Range::Range()
  ============================================================================*/
Range::Range()
{
  cout<<endl<<"Enter the angular fitting range, "
  #ifdef Data6464       //CS datafile
      <<"0 - 63."
  #endif
```

```
  #ifdef Data64128     //MS datafile
     <<"0 - 63."
  #endif
  #ifdef Data128128    //HR datafile
     <<"0 - 63(if reduced) or 0 - 127."
  #endif
     <<endl;
  cin>>Angle1>>Angle2;

  cout<<endl<<"Enter the energy fitting range, "
  #ifdef Data6464      //CS datafile
      <<"0 - 63."
  #endif
  #ifdef Data64128     //MS datafile
      <<"0 - 63(if reduced) or 0 - 127."
  #endif
  #ifdef Data128128    //HR datafile
     <<"0 - 63(if reduced) or 0 - 127."
  #endif
     <<endl;
  cin>>Energy1>>Energy2;
}
```

## /* fitT.cpp */

```
#include "fitT.h"
float (Model::*ModelF)(float *parm,float angleCh,float energyCh) = &Model::Access;
/*=============================================================================
   CONSTRUCTOR: fitT()
  ============================================================================*/
fitT::fitT(int nrParmC,float **Mdata,class Range R,float **parmBC,int *ControlC,int ASize,int ESize)
{
  set_new_handler(MemWarn);
  NrOfParm=nrParmC;  Asize=ASize; Esize=ESize;
  angle1=R.Angle1; angle2=R.Angle2;  energy1=R.Energy1;  energy2=R.Energy2;

  zval=new float*[angle2+1];  weight=new float*[angle2+1];   weight1=new float*[angle2+1];
  Wm=new float*[Asize];      // Take over Wm[i][j] from model function
  for (int i=0;i<Asize;i++)  Wm[i]=new float[Esize];
  for (i=0;i<=angle2;i++) {
    zval[i]=new float[energy2+1];   weight[i]=new float[energy2+1];   weight1[i]=new float[energy2+1];
    for (int j=0;j<=energy2;j++)  zval[i][j]=0;  weight[i][j]=1.0;
  }

  MIN=10000;  MAX=0;
  for (i=angle1;i<=angle2;i++)
    for (int j=energy1;j<=energy2;j++) {
      zval[i][j]=Mdata[i][j];
      if (zval[i][j]==0) weight[i][j]=1.0;
      else weight[i][j]=1.0/zval[i][j];
      if (zval[i][j]>MAX) {MAX=zval[i][j]; POS[0]=i; POS[1]=j;}
```

```
      if (zval[i][j]<MIN) MIN=zval[i][j];
    }

  parmB=new float*[NrOfParm];  Control=new int[NrOfParm];  SigmaParm=new float[NrOfParm];
  for (i=0;i<NrOfParm;i++) {
    parmB[i]=new float[2];  Control[i]=ControlC[i];  SigmaParm[i]=0.0;
    for (int j=0;j<2;j++)  parmB[i][j]=parmBC[i][j];
  }

  int AngleRange,EnergyRange,FLAG=0;
  AngleRange=angle2-angle1+1;  EnergyRange=energy2-energy1+1;
  iNrOfParm=0;   // Number of parameters for fitting routine; Control != -2,-3
  for (i=0;i<NrOfParm;i++)  if ((Control[i]!=-2)&&(Control[i]!=-3)) iNrOfParm++;
  DOF0=AngleRange*EnergyRange-iNrOfParm;    // Initial Degree of Freedom
  flag1=new int[NrOfParm];  flag2=new int[iNrOfParm];

  // Fitting Control
  for (i=0;i<NrOfParm;i++) {
    if ((Control[i]==-2)||(Control[i]==-3)) flag1[i]=-1;
    else {flag1[i]=FLAG;  flag2[FLAG]=i;  FLAG++;}
  }
}


/*-------------------------------------------------------------------------
   DESTRUCTOR: ~fitT()
-------------------------------------------------------------------------*/
fitT::~fitT()
{
  delete[] zval;    delete[] weight;    delete[] weight1;    delete[] Wm;    delete[] parmB;
  delete[] flag1;   delete[] flag2;     delete[] Control;    delete[] SigmaParm;
}


/*-------------------------------------------------------------------------
   FUNCTION:  void fitT::CurFit()
   OPERATION:  Make a least-square fit to a non-linear function with a linearization of the fitting function.
-------------------------------------------------------------------------*/
void fitT::CurFit(class Model model,float *parm,float *DeltaParm,float &fLamda,float &ChiSqr)
{
  int i,j,x,y,iflag=-1
  double *beta,**alpha,det;
  float ChiSq1,*parmA,*parmX,*deriv,**array,I, modelV;
  parmA=new float[iNrOfParm];  parmX=new float[NrOfParm];  beta=new double[iNrOfParm];
  deriv=new float[iNrOfParm];  alpha=new double*[iNrOfParm];  array=new float*[iNrOfParm];
  for (i=0; i<iNrOfParm; i++) {alpha[i]=new double[iNrOfParm];  array[i]=new float[iNrOfParm];}

  cout<<"1:  Evaluating alpha and beta matrices"<<endl;
  for (i=0;i<iNrOfParm;i++) {beta[i]=0.0;  for (j=0;j<=i;j++)  alpha[i][j]=0.0;}

  cout<<"0                    25                    50                    75
            100"<<endl;
```

```
cout<<"└──────────┴──────────┴──────────┴──────────┴──────────┘"<<endl;
// Pull out (model.*ModelF)(parm,x,y) by one loop to reduce calculating time by 1/NrOfParm
for (x=angle1;x<=angle2;x++) {
  I=(float)(x-angle1)*76.0/(float)(angle2-angle1);
  if ((int)I!=iflag) {for (int k=iflag;k<(int)I;k++) cputs("▒"); iflag=(int)I;}
  for (y=energy1;y<=energy2;y++) {
    FDeriv(model,parm,x,y,DeltaParm,deriv); modelV=(model.*ModelF)(parm,x,y);
    for (i=0;i<iNrOfParm;i++) {
      beta[i]+=(zval[x][y]-modelV)*deriv[i]*weight[x][y];
      for (j=0; j<=i; j++) alpha[i][j]+=deriv[i]*deriv[j]*weight[x][y];
    }
  }
}
for (i=0;i<iNrOfParm;i++) for (j=0;j<=i;j++) alpha[j][i]=alpha[i][j];

cout<<endl<<"2: Evaluating Chi Square at starting point"<<endl;
FindChiSq(model,parm,ChiSq1);
/**** Invert modified curvature matrix to find new parameters ****/
REPEAT:
for (i=0;i<iNrOfParm;i++) {
  for (j=0;j<iNrOfParm;j++) array[i][j]=alpha[i][j]/sqrt(alpha[i][i]*alpha[j][j]);
  array[i][i]=1.0+fLamda;
}

MatInv(array,iNrOfParm,det);      // Bevington.lib
for (i=0;i<iNrOfParm;i++) {
  parmA[i]=parm[flag2[i]];
  for (j=0; j<iNrOfParm; j++) parmA[i]+=beta[j]*array[i][j]/sqrt(alpha[i][i]*alpha[j][j]);
}
for (i=0;i<NrOfParm;i++) {parmX[i]=parm[i]; if (flag1[i]!=-1) parmX[i]=parmA[flag1[i]];}

cout<<"3: If Chi Square increased, increase fLamda and try again"<<endl;
FindChiSq(model,parmX,ChiSqr); cout<<"ChiSqr = "<<ChiSqr<<endl;
if (ChiSqr>ChiSq1) {
  fLamda*=10.0;
  if (fLamda>1.0e5) {cout<<"\tfLamda blew up beyond 1.0e5."<<endl; ChiSqr=ChiSq1; goto OUT;}
  goto REPEAT;
}
OUT:

cout<<"4: Evaluating parameters and uncertainties"<<endl;
for (i=0;i<iNrOfParm;i++) {
  if (fLamda<=1.0e5) parm[flag2[i]]=parmX[flag2[i]];
  else fLamda*=10.0;
  SigmaParm[flag2[i]]=sqrt(fabs(array[i][i]/alpha[i][i]));
}

fLamda/=10.0;
ofstream DF("default.dat",ios::out);
for (i=0;i<iNrOfParm;i++) {cout<<parm[flag2[i]]<<","; DF<<parm[flag2[i]]<<",";}
cout<<endl<<"\tfLamda = "<<fLamda<<", "<<"Chi Square = "<<ChiSqr<<endl;
```

```
  DF<<endl<<"\tfLamda = "<<fLamda<<", "<<"Chi Square = "<<ChiSqr<<endl;
  DF.close();

  delete[] parmA;  delete[] parmX;  delete[] beta;  delete[] deriv;  delete[] alpha;  delete[] array;
}



/*-------------------------------------------------------------------------------
   FUNCTION: void fitT::FindChiSq()
   OPERATION: Evaluate reduced Chi Square for fit to data.
  -------------------------------------------------------------------------------*/
void fitT::FindChiSq(class Model model,float *parm,float &ChiSqr)
{
  for (int i=0;i<NrOfParm;i++) {
    switch(Control[i]) {
      case 2: {       // Between maximum and minimum
              if (parm[i]>parmB[i][1]) parm[i]=parmB[i][1];
              if (parm[i]<parmB[i][0]) parm[i]=parmB[i][0];  break;}
      case 1: {       // fit < maximum
              if (parm[i]>parmB[i][1]) parm[i]=parmB[i][1];  break;}
//    case 0: {parm[i] is floating.}
      case -1:{       // fit > minimum
              if (parm[i]<parmB[i][0]) parm[i]=parmB[i][0];  break;}
//    case -2:{parm[i] has not been changed, so retains its original value.}
      case -3:{       // Not for fitting routine. Set to 0.0
              parm[i]=0.0;  break;}
      default: break;
    }
  }

  DOF=DOF0;
  for (i=0;i<=angle2;i++)  for (int j=0;j<=energy2;j++)  weight1[i][j]=1.0;
  model.Mweight(Wm);              // Take over Wm[][] from model function
  // Wm[][]=1 : normal, Wm[][]=0 : to remove a certain region from fitting
  NrOfEx=0;
  for (int x=angle1;x<=angle2;x++)
    for (int y=energy1;y<=energy2;y++) {
      weight1[x][y]=weight[x][y]*Wm[x][y];
      if (weight1[x][y]==0.0)  {DOF--; NrOfEx++; Exclude=1;}
    }

  ChiSqr=0.0;
  for (x=angle1;x<=angle2;x++)
    for (int y=energy1;y<=energy2;y++)
      ChiSqr+=pow(zval[x][y]-(model.*ModelF)(parm,x,y),2.0)*weight1[x][y];
  ChiSqr/=DOF;
}


/*-------------------------------------------------------------------------------
   FUNCTION: void fitT::FDeriv()
```

```
   OPERATION: Evaluate derivatives of function for least-square search for arbitrary function given by
              model function.
 ================================================================================*/
void fitT::FDeriv(class Model model,float *parm,float AngleCh,float EnergyCh,float *DeltaParm,float *deriv)
{
  float parmI,Delta,yFit;
  for (int i=0;i<iNrOfParm;i++) {
    parmI=parm[flag2[i]];  Delta=DeltaParm[flag2[i]];  parm[flag2[i]]=parmI+Delta;
    yFit=(model.*ModelF)(parm,AngleCh,EnergyCh);  parm[flag2[i]]=parmI-Delta;
    deriv[i]=(yFit-(model.*ModelF)(parm,AngleCh,EnergyCh))/2.0/Delta;  parm[flag2[i]]=parmI;
  }
}


/*================================================================================
   FUNCTION: void fitT::Result()
   OPERATION: Prints out reduced CHI2, parameters, and so on. Also gives information of the input
              file. The same details are written in "result.dat" file.
================================================================================*/
void fitT::Result(char *argv[],float &ChiSqr,float *parm,int &iteration,float &fLamda,class Model model)
{
  float pchi2=pchisq(DOF,ChiSqr);  /* Return the result of CHI2 test: Bevington.lib */
  // Set local time
  time_t timer;  struct tm *tblock;  timer=time(NULL);  tblock=localtime(&timer);
  float A1,A2,E1,E2;

  (model.*ModelF)(parm,0,0);  // Parameter initialization
  model.Conversion(A1,E1,angle1,energy1);  model.Conversion(A2,E2,angle2,energy2);
  cout.precision(7);
  cout<<endl<<endl<<"\t\t      Data file = "<<argv[1]<<endl;
  cout<<"\t\t      Parameter file = "<<argv[2]<<endl;
  cout<<"\t\t      Peak Maximum = "<<MAX<<endl;
  cout<<"\t\t      Fit Array Size :"<<endl;
  cout<<"\t\t        Angle :: "<<angle1<<" - "<<angle2<<endl;

  #ifdef PlotIn_au
    cout<<"\t\t              ("<<A1<<" - "<<A2<<" in Vau)"<<endl;
  #endif
  #ifdef PlotIn_eV
    cout<<"\t\t              ("<<A1<<" - "<<A2<<" in angle)"<<endl;
  #endif
  cout<<"\t\t        Energy :: "<<energy1<<" - "<<energy2<<endl;
  #ifdef PlotIn_au
    cout<<"\t\t              ("<<E1<<" - "<<E2<<" in Vau)"<<endl;
  #endif
  #ifdef PlotIn_eV
    cout<<"\t\t              ("<<E1<<" - "<<E2<<" in eV)"<<endl;
  #endif

  cout<<endl<<"\t\tDegree of freedom = "<<DOF<<endl;
  cout<<endl<<endl<<"\t\tPARAMETER        RESULT          SIGMA PARM."<<endl;
  cout<<"\t\t---------      ------      ----------"<<endl<<endl;
```

```
  int k=0;
  for (int i=0;i<6;i++)
    for (int j=0;j<2;j++) {
      cout<<"\t\tBeta("<<j<<","<<i<<")"<<setw(21)<<parm[k]<<setw(24)<<SigmaParm[k]<<endl;
      k++;
    }

  cout<<"\t\tBeta(6)"<<setw(23)<<parm[12]<<setw(24)<<SigmaParm[12]<<endl;
  cout<<"\t\tBeta(8)"<<setw(23)<<parm[13]<<setw(24)<<SigmaParm[13]<<endl;
  cout<<"\t\tBeta(10)"<<setw(22)<<parm[14]<<setw(24)<<SigmaParm[14]<<endl;
  cout<<endl<<"\t\tAngle Center"<<setw(18)<<parm[21]<<setw(24)<<SigmaParm[21]<<endl;
  cout<<"\t\tEnergy Center"<<setw(17)<<parm[22]<<setw(24)<<SigmaParm[22]<<endl;
  cout<<endl<<"\t\t Ecusp      (eV) = "<<setw(17)<<parm[32]<<" ("
          <<setw(15)<<SigmaParm[32]<<")"<<endl;
  cout<<"\t\t  Tilt      (%) = "<<setw(17)<<parm[20]<<" ("<<setw(15)<<SigmaParm[20]<<")"<<endl;
  cout<<"\t\t   TW       (%) = "<<setw(17)<<parm[23]<<" ("<<setw(15)<<SigmaParm[23]<<")"<<endl;
  cout<<"\t\t  FWHM      (%) = "<<setw(17)<<parm[24]<<" ("<<setw(15)<<SigmaParm[24]<<")"<<endl;
  cout<<"\t\t ARFin     (rad) = "<<setw(17)<<parm[25]<<" ("<<setw(15)<<SigmaParm[25]<<")"<<endl;
  cout<<"\t\tARFout     (rad) = "<<setw(17)<<parm[26]<<" ("<<setw(15)<<SigmaParm[26]<<")"<<endl;
  cout<<"\t\t ASTEP     (rad) = "<<setw(17)<<parm[27]<<" ("<<setw(15)<<SigmaParm[27]<<")"<<endl;
  cout<<"\t\t BKGDa (count/ch) = "<<setw(17)<<parm[33]<<"("          <<setw(15)<<SigmaParm[33]<<")"
<<endl;
  cout<<"\t\t BKGDb    (count) = "<<setw(17)<<parm[34]<<" ( " <<setw(15)<<SigmaParm[34]<<")"<<endl;
  cout<<"\t\t SLICE      (ch) = "<<setw(17)<<parm[35]<<" ("<<setw(15)<<SigmaParm[35]<<")"<<endl;
  cout<<"\t\tEshift     (eV) = "<<setw(17)<<model.Eshift<<endl;
  cout<<endl<<"\t\tMinimum Reduced Chi Square : "<<ChiSqr<<endl;
  cout<<"\t\tProbability of exceeding minCHI2 : "<<pchi2<<endl;
  if (Exclude)  cout<<"\t\tEXCLUDED  cusp region: ("<<NrOfEx<<" points removed)"<<endl;
  else  cout<<"\t\tINCLUDED  cusp region: "<<endl;

  cout<<"\t\tIteration stopped at "<<iteration<<endl;
  if (fLamda>1.0e5)  cout<<"\t\tfLamda blew up beyond 1.0e5"<<endl;
  else  cout<<"\t\tFinal fLamda = "<<fLamda<<endl;
  cout<<endl<<"\t.This version is using ";
    #ifdef PillBoxARF
      cout<<" a PILLBOX shape angular, and "<<endl;
    #endif
    #ifdef TrapeZoidalARF
      cout<<" a TRAPEZOIDAL shape angular, and "<<endl;
    #endif
    #ifdef TrapezoidalERF
      cout<<"\t a TRAPEZOIDAL shape energy resolution functions."<<endl;
    #endif
    #ifdef GaussianERF
      cout<<"\t a GAUSSIAN shape energy resolution functions."<<endl;
    #endif
  cout<<"\t\tDate: "<<asctime(tblock)<<endl;

  char *ans=new char[10],SaveF[50]; *ans='n';
  cout<<endl<<"\t\tSave the result?"<<endl<<"\t\t";  cin>>ans;

  /*========== Write to the file ==========*/
```

```
ofstream op;
if ((*ans=='Y')||(*ans=='y')) {
  cout<<endl<<"\t\tEnter the full filename:"<<endl;   gets(SaveF);
  op.open(SaveF,ios::out);
}
else op.open("result.dat",ios::out);

op.precision(7);
op<<endl<<endl<<"\t\t       Data file = "<<argv[1]<<endl;
op<<"\t\t       Parameter file = "<<argv[2]<<endl;
op<<"\t\t       Peak Maximum = "<<MAX<<endl;
op<<"\t\t       Fit Array Size :"<<endl;
op<<"\t\t          Angle :: "<<angle1<<" - "<<angle2<<endl;
op<<"\t\t                ("<<A1<<" - "<<A2<<" in Vau)"<<endl;
op<<"\t\t          Energy :: "<<energy1<<" - "<<energy2<<endl;
op<<"\t\t                ("<<E1<<" - "<<E2<<" in Vau)"<<endl;
op<<endl<<"\t\tDegree of freedom = "<<DOF<<endl;
op<<endl<<endl;
op<<"\t\tPARAMETER          RESULT          SIGMA PARM."<<endl;
op<<"\t\t---------          ------          ----------"<<endl<<endl;

int kk=0;
for (i=0;i<6;i++)
  for (int j=0;j<2;j++) {
    op<<"\t\tBeta("<<j<<","<<i<<")"<<setw(21)<<parm[kk]<<setw(24)<<SigmaParm[kk]<<endl;
    kk++;
  }

op<<"\t\tBeta(6)"<<setw(23)<<parm[12]<<setw(24)<<SigmaParm[12]<<endl;
op<<"\t\tBeta(8)"<<setw(23)<<parm[13]<<setw(24)<<SigmaParm[13]<<endl;
op<<"\t\tBeta(10)"<<setw(22)<<parm[14]<<setw(24)<<SigmaParm[14]<<endl;
op<<endl<<"\t\tAngle Center"<<setw(18)<<parm[21]<<setw(24)<<SigmaParm[21]<<endl;
op<<"\t\tEnergy Center"<<setw(17)<<parm[22]<<setw(24)<<SigmaParm[22]<<endl;
op<<endl<<"\t\t Ecusp      (eV) = "<<setw(17)<<parm[32]<<" ("
<<setw(15)<<SigmaParm[32]<<")"<<endl;
op<<"\t\t  Tilt       (%) = "<<setw(17)<<parm[20]<<" ("<<setw(15)<<SigmaParm[20]<<")"<<endl;
op<<"\t\t   TW        (%) = "<<setw(17)<<parm[23]<<" ("<<setw(15)<<SigmaParm[23]<<")"<<endl;
op<<"\t\t  FWHM       (%) = "<<setw(17)<<parm[24]<<" ("<<setw(15)<<SigmaParm[24]<<")"<<endl;
op<<"\t\t ARFin     (rad) = "<<setw(17)<<parm[25]<<" ("<<setw(15)<<SigmaParm[25]<<")"<<endl;
op<<"\t\tARFout     (rad) = "<<setw(17)<<parm[26]<<" ("<<setw(15)<<SigmaParm[26]<<")"<<endl;
op<<"\t\t ASTEP     (rad) = "<<setw(17)<<parm[27]<<" ("<<setw(15)<<SigmaParm[27]<<")"<<endl;
op<<"\t\t BKGDa (count/ch) = "<<setw(17)<<parm[33]<<" ("<<setw(15)<<SigmaParm[33]<<")"<<endl;
op<<"\t\t BKGDb    (count) = "<<setw(17)<<parm[34]<<" ("<<setw(15)<<SigmaParm[34]<<")"<<endl;
op<<"\t\t SLICE      (ch) = "<<setw(17)<<parm[35]<<" ("<<setw(15)<<SigmaParm[35]<<")"<<endl;
op<<"\t\tEshift      (eV) = "<<setw(17)<<model.Eshift<<endl;
op<<endl<<"\t\tMinimum Reduced Chi Square : "<<ChiSqr<<endl;
op<<"\t\tProbability of exceeding minCHI2 : "<<pchi2<<endl;
if (Exclude)
  op<<"\t\tEXCLUDED  cusp region: ("<<NrOfEx<<"points removed)"<<endl;
else
  op<<"\t\tINCLUDED  cusp region: "<<endl;
```

```
  op<<"\t\tIteration stopped at "<<iteration<<endl;
  if (fLamda>1.0e5)  op<<"\t\tfLamda blew up beyond 1.0e5"<<endl;
  else  op<<"\t\tFinal fLamda = "<<fLamda<<endl;
  op<<endl<<"\t.This version is using ";
  #ifdef PillBoxARF
    op<<" a PILLBOX shape angular, and "<<endl;
  #endif
  #ifdef TrapeZoidalARF
    op<<" a TRAPEZOIDAL shape angular, and "<<endl;
  #endif
  #ifdef TrapezoidalERF
    op<<"\t a TRAPEZOIDAL shape energy resolution functions."<<endl;
  #endif
  #ifdef GaussianERF
    op<<"\t a GAUSSIAN shape energy resolution functions."<<endl;
  #endif
  op<<"\t\tDate:  "<<asctime(tblock)<<endl;

  delete[] ans;  delete[] tblock;
}


/*=============================================================================
   FUNCTION:  GnuPlotData()
   OPERATION:  Make datafiles for Gnuplot with source file and fitted parameter values.
=============================================================================*/
void GnuPlotData(char *argv[],float **Mdata,float *parm,class Range R,int &ASize,int &ESize,
                 class Model model)
{
  char *Sgnu=new char[50],*Fgnu=new char[50],*Rgnu=new char[50];
  strcpy(Sgnu,argv[1]);  strcat(Sgnu,".SGNU");  ofstream Source(Sgnu,ios::out);
  strcpy(Fgnu,argv[1]);  strcat(Fgnu,".FGNU");  ofstream Fit(Fgnu,ios::out);
  strcpy(Rgnu,argv[1]);  strcat(Rgnu,".RGNU");  ofstream Residue(Rgnu,ios::out);
  Source.precision(6);  Fit.precision(6);  Residue.precision(6);

  float Vangle,Venergy,fitDataBase=0.0;
  cout<<endl<<"\t\tWriting GnuPlot data ....."<<endl;
  (model.*ModelF)(parm,0,0);   // Parameter initialization

  int iflag=-1;
  cout<<"0                         25                        50                        75
              100"<<endl;
  cout<<" └                        ┴                         ┴                         ┴
                  ┘ "<<endl;
  for (int i=0;i<ASize;i++) {
    float I=(float)i*76.0/(float)(ASize-1);
    if ((int)I!=iflag)  {for (int k=iflag;k<(int)I;k++) cputs("▒"); iflag=(int)I;}
    for (int j=0;j<ESize;j++) {
      model.Conversion(Vangle,Venergy,i,j);
      Source<<Venergy<<" "<<Vangle<<" "<<Mdata[i][j]<<endl;
      if ((i<R.Angle1)||(i>R.Angle2)||(j<R.Energy1)||(j>R.Energy2)) {
        Fit<<Venergy<<" "<<Vangle<<" "<<fitDataBase<<endl;
```

```
        Residue<<Venergy<<" "<<Vangle<<" "<<Mdata[i][j]<<endl;
      }
      else {
        Fit<<Venergy<<" "<<Vangle<<" "<<(model.*ModelF)(parm,i,j)<<endl;
        Residue<<Venergy<<" "<<Vangle<<" "<<Mdata[i][j]-(model.*ModelF)(parm,i,j)<<endl;
      }
    }
    Source<<endl;  Fit<<endl;  Residue<<endl;
  }
  Fit.close();  Source.close();  Residue.close();
  delete[] Sgnu;  delete[] Fgnu;  delete[] Rgnu;
}


/*=============================================================================
   FUNCTION: void MemWarn()
   OPERATION: Assign the address of 'MemWarn()' to be called if 'new' fails to allocate a requested
              amount of memory.
  =============================================================================*/
void MemWarn()
{
  cerr<<endl<<"Cannot allocate!";  exit(-2);
}


/* FitParm.cpp */
#include "fitT.h"
/*=============================================================================
   CONSTRUCTOR: parmM::parmM()
  =============================================================================*/
parmM::parmM(char **argv)
{
  set_new_handler(MemWarn);
  InFile.open(argv[2],ios::in);
  if (!InFile) {ErrDisplay(3); exit(2);}
  int LineNo;
  InFile.getline(inbuf,81);
  istrstream ins(inbuf,strlen(inbuf));
  ins>>LineNo>>nrOfParm;
  InFile.seekg(0,ios::beg);
  ans=new char[10];  parm=new float[nrOfParm];  parmBound=new float*[nrOfParm];
  for (int i=0;i<nrOfParm;i++) parmBound[i]=new float[2];
  for (i=0;i<6;i++) remark[i]=new char[81];
  DeltaParm=new float[nrOfParm];  Control=new int[nrOfParm];  header=new long[nrOfParm+6];
}


/*=============================================================================
   DESTRUCTOR: parmM::~parmM()
  =============================================================================*/
parmM::~parmM()
{
```

```
  delete[] parm;  delete[] parmBound;  delete[] DeltaParm;  delete[] Control;
  delete[] header;  delete[] ans;  InFile.close();
}


/*=============================================================================
  FUNCTION:  parmM::GetParm()
=============================================================================*/
void parmM::GetParm()
{
  int lineNo;  cout<<endl;
  for (int i=0;i<nrOfParm+6;i++) {
    InFile.getline(inbuf,81);
    header[i]=strlen(inbuf);
    istrstream ins(inbuf,strlen(inbuf));
    if (i==0) {ins>>lineNo>>nrOfParm>>ws;  ins.getline(*remark,81);}
    else if (i==1) {ins>>lineNo>>nrOfIter>>ws;  ins.getline(*(remark+1),81);}
    else if (i==2) {ins>>lineNo>>Convergence>>ws;  ins.getline(*(remark+2),81);}
    else if (i==3) ins.getline(*(remark+3),81);
    else if (i==4) ins.getline(*(remark+4),81);
    else if (i==5) ins.getline(*(remark+5),81);
    else {ins>>lineNo>>parm[i-6]>>parmBound[i-6][0]>>parmBound[i-6][1]>>DeltaParm[i-6]
          >>Control[i-6];
    }
    cout<<inbuf<<endl;
  }
  InFile.seekg(0,ios::beg);
}


/*=============================================================================
  FUNCTION:  parmM::ChangeParm()
=============================================================================*/
void parmM::ChangeParm()
{
  cout<<endl;  GetParm();
  int SkipLine;

  REPEAT:
  do {
    cout<<endl<<"Enter the line number (0 to EXIT): ";    cin>>SkipLine;
  } while ((SkipLine==4)||(SkipLine==5)||(SkipLine==6)||(SkipLine<0)||(SkipLine>(6+nrOfParm)));

  long Skip=0;
  for (int i=0;i<(SkipLine-1);i++) Skip+=header[i]+2;
  if (SkipLine==0) exit(-3);
  InFile.seekg(Skip,ios::beg);  InFile.getline(inbuf,81);

  FIELD:
  cout<<inbuf<<endl;
  if (SkipLine<4) {
    switch(SkipLine) {
```

```
    float ground;   int Ground;
    case 1: {cout<<endl<<"This line has only one field. Change? ";
             REDO1:
             cin>>ans;
             if ((*ans=='N')||(*ans=='n')) break;
             else if ((*ans=='Y')||(*ans=='y')) {
               cout<<"Enter new value : ";  cin>>Ground;  nrOfParm=Ground;  break;
             }
             else {cout<<"Enter (y|Y) or (n|N)! "; goto REDO1;}}
    case 2: {cout<<endl<<"This line has only one field. Change? ";
             REDO2:
             cin>>ans;
             if ((*ans=='N')||(*ans=='n')) break;
             else if ((*ans=='Y')||(*ans=='y')) {
               cout<<"Enter new value : ";  cin>>Ground;  nrOfIter=Ground;  break;
             }
             else {cout<<"Enter (y|Y) or (n|N)! "; goto REDO2;}}
    case 3: {cout<<endl<<"This line has only one field. Change? ";
             REDO3:
             cin>>ans;
             if ((*ans=='N')||(*ans=='n')) break;
             else if ((*ans=='Y')||(*ans=='y')) {
               cout<<"Enter new value : ";  cin>>ground;  Convergence=ground;  break;}
             else {cout<<"Enter (y|Y) or (n|N)! "; goto REDO3;}}
    default: break;
  } /* end switch */
} /* end if */

else {
  long field;
  cout<<endl<<"Enter field (1-5 for Parameter-Control, 0 to restart): ";   cin>>field;
  switch(field) {
    float newValue;  int NewValue;
    case 0: goto REPEAT;
    case 1: {cout<<"\t\tparm["<<SkipLine-7<<"] = "<<parm[SkipLine-7]<<endl;
             cout<<"Enter new value : ";  cin>>newValue;  parm[SkipLine-7]=newValue; break;}
    case 2: {cout<<"\t\tparmBound["<<SkipLine-7<<"][0] = "<<parmBound[SkipLine-7][0]<<endl;
             cout<<"Enter new value : "; cin>>newValue;  parmBound[SkipLine-7][0]=newValue;  break;}
    case 3: {cout<<"\t\tparmBound["<<SkipLine-7<<"][1] = "<<parmBound[SkipLine-7][1]<<endl;
             cout<<"Enter new value : "; cin>>newValue; parmBound[SkipLine-7][1]=newValue;  break;}
    case 4: {cout<<"\t\tDeltaParm["<<SkipLine-7<<"] = "<<DeltaParm[SkipLine-7]<<endl;
            cout<<"Enter new value : "; cin>>newValue;  DeltaParm[SkipLine-7]=newValue;  break;}
    case 5: {cout<<"\t\tControl["<<SkipLine-7<<"] = "<<Control[SkipLine-7]<<endl;
             cout<<"Enter new value : "; cin>>NewValue; Control[SkipLine-7]=NewValue;  break;}
    default: {cout<<"Field should be 1 to 5"<<endl; goto FIELD;}
  } /* end switch */
} /* end else */

cout<<endl<<" 1"<<setw(15)<<nrOfParm<<" \t\t"<<*remark<<endl;
cout<<" 2"<<setw(15)<<nrOfIter<<" \t\t"<<*(remark+1)<<endl;
cout<<" 3"<<setw(15)<<Convergence<<" \t\t"<<*(remark+2)<<endl;
cout<<*(remark+3)<<endl;  cout<<*(remark+4)<<endl;  cout<<*(remark+5)<<endl;
```

```
  for (i=0;i<nrOfParm;i++)
    cout<<setw(2)<<i+7<<setw(15)<<parm[i]<<setw(15)<<parmBound[i][0]<<setw(15)<<parmBound[i][1]
        <<setw(15)<<DeltaParm[i]<<setw(10)<<Control[i]<<endl;
  cout<<endl<<"Another line? ";

  LINE:
  cin>>ans;
  if ((*ans=='Y')||(*ans=='y')) goto REPEAT;
  else if ((*ans=='N')||(*ans=='n')) ;
  else {cout<<"Enter (y|Y) or (n|N)! "; goto LINE;}
  InFile.seekg(Skip,ios::beg);
}


/*-------------------------------------------------------------------------
   FUNCTION: parmM::WriteParm()
  -------------------------------------------------------------------------*/
void parmM::WriteParm(char *argv[],char ParmFile[50],int iteration)
{
  int dec,sign;
  char *str;
  switch(iteration) {
    case (0): {cout<<endl<<"\t\tEnter full parameter filename: "<<endl;  gets(ParmFile);break;}
    default: {strcpy(ParmFile,"iter");  str=fcvt(iteration,0,&dec,&sign);
              strcat(ParmFile,str);  strcat(ParmFile,".prm");  break;}
  }

  ofstream outf;
  outf.open(ParmFile,ios::out);
  outf.setf(ios::showpoint);
  outf.precision(7);
  outf<<" 1"<<setw(15)<<nrOfParm<<" \t\t"<<*remark<<endl;
  outf<<" 2"<<setw(15)<<nrOfIter<<" \t\t"<<*(remark+1)<<endl;
  outf<<" 3"<<setw(15)<<Convergence<<" \t\t"<<*(remark+2)<<endl;
  outf<<*(remark+3)<<endl;
  outf<<*(remark+4)<<endl;
  outf<<*(remark+5)<<endl;

  for (int i=0;i<nrOfParm;i++)
    outf<<setw(2)<<i+7<<setw(15)<<parm[i]<<setw(15)<<parmBound[i][0]<<setw(15)<<parmBound[i][1]
        <<setw(15)<<DeltaParm[i]<<setw(10)<<Control[i]<<endl;
  outf<<endl<<endl<<"\tThis parameter file comes with data file, "<<argv[1]<<endl;
  outf.close();
}
```

/* fitT.h */

```
#ifndef __fitT_h
#define __fitT_h 1
```

```
#include <iostream.h>
#include <stdlib.h>
#include <fstream.h>
#include <strstrea.h>
#include <stdio.h>
#include <math.h>
#include <iomanip.h>
#include <conio.h>
#include <string.h>
#include <new.h>
#include <time.h>
#include <float.h>
#include "Bevington.h"   /* Include Bevington.lib */
#include "Model.h"

class fitT
{
  private:
    /* NrOfParm    : Number of parameters of the model function
     * DOF0,DOF    : Degree of Freedom
     * angle1,2    : Angle-fitting range
     * Control     : Pointer to parameter control arrary
     * energy1,2   : Energy-fitting range
     * Asize,Esize : Modified angle and energy sizes of data          */
    int NrOfParm,DOF0,DOF,*Control,angle1,angle2,energy1,energy2,Asize,Esize;
    /* MIN         : Minimum of fitting data
     * MAX         : Maximum of fitting data
     * POS         : Position at maximum point
     * iNrOfParm   : Number of parameters with 'CONTROL'=-1,0,1, or 2
     * flag1,2     : Flags for assigning parameter without 'CONTROL'=-2, or -3
     * Exclude     : Number of excluded data points from fitting      */
    int MIN,MAX,POS[2],iNrOfParm,*flag1,*flag2,NrOfEx,Exclude;
    /* zval        : Pointer to pointer to fitting data
     * weight,weight1 : Pointer to pointer to standard deviation of data
     * Wm          : Pointer to pointer to temporary weight
     * parmB       : Pointer to pointer to parameter boundary array
     * SigmaParm   : Pointer to uncertainty in parameter          */
    float **zval,**weight,**weight1,**Wm,**parmB,*SigmaParm;

  public:
    /* Constructor
       . nrParmC : Number of parameters of the model function
       . Mdata   : Pointer to pointer to fitting data
       . Range R : Class for fitting ranges
       . ControlC : Pointer to parameter control value
       . parmBC : Pointer to pointer to parameter boundary array      */
    fitT(int nrParmC,float **Mdata,class Range R,float **parmBC,int *ControlC,int ASize,int ESize);
    /* Destructor: When an object of a class goes out of scope, C++ automatically calls that object's class
                   destructor if it has one.                          */
    ~fitT();
    /* CurFit()  : Calculates minimum reduced chi-square
       . Model model: Class for model function
```

```
        . parm     : Pointer to parameter array
        . DeltaParm : Pointer to increment in each parameter
        . fLamda   : Proportion of gradient search included
        . ChiSqr   : Minimum reduced Chi Square                    */
    void CurFit(class Model model,float *parm,float *DeltaParm,float &fLamda,float &ChiSqr);
    /* FindChiSq() : Evaluate reduced Chi Square for fit to data
        . Model model : Class for model function
        . parm        : Pointer to parameter array
        . ChiSqr      : Minimum reduced Chi Square                 */
    void FindChiSq(class Model model,float *parm,float &ChiSqr);

    /* FDeriv()  : Evaluate derivatives of function
        . Model model : Class for model function
        . parm        : Pointer to parameter array
        . AngleCh     : Angular coordinate
        . EnergyCh    : Energy coordinate
        . DeltaParm   : Pointer to increment in each parameter
        . deriv       : Pointer to derivatives w.r.t. each parameter    */
    void FDeriv(class Model model,float *parm,float AngleCh,float EnergyCh,float *DeltaParm,float *deriv);
    /* Result()  : Shows the result of the fitting on screen and file
        . Model model : Class for model function
        . argv[]      : Pointer to argument array
        . ChiSqr      : Minimum reduced chi-square
        . iteration   : Iteration number
        . fLamda      : Factor LAMDA which controls interpolation
        . parm        : Pointer to parameter array                 */
    void Result(char *argv[],float &ChiSqr,float *parm,int &iteration,float &fLamda,class Model model);
};
class parmM {
  private:
    char inbuf[81],*ans,*remark[6];
    long *header;
  public:
    ifstream InFile;
    int nrOfParm,nrOfIter,*Control;
    float Convergence,*parm,**parmBound,*DeltaParm;
    parmM(char **argv);
    ~parmM();
    void GetParm();
    void ChangeParm();
    void WriteParm(char **argv,char ParmFile[50],int iteration);
};

class Range {
  public:
    int Angle1,Angle2,Energy1,Energy2;
    Range();
    ~Range(){}
};

void SetSize(char **argv,int &ASize,int &ESize);
void GetData(char **argv,float **Mdata,int &ASize,int &ESize);
```

```
void GnuPlotData(char *argv[],float **Mdata,float *parm,class Range R,int &ASize,int &ESize,
                 class Model model);
void MemWarn();
void ErrDisplay(int Q);
void Display(int P);
//#define Data6464       // Default or CS data file
#define Data64128       // MS data file
//#define Data128128     // HR data file
#endif /* __fitT_h */
```

## /* model.cpp */

```
#include "Fitt.h"
/*=============================================================================
   CONSTRUCTOR: Model::Model()
=============================================================================*/
Model::Model(int *control,int ASize,int ESize)
{
  set_new_handler(MemWarn);

  Asize=ASize; Esize=ESize;
  LegendreCoeff=new double*[11];      // 0 through 10th order
  for (int i=0;i<11;i++) LegendreCoeff[i]=new double[11];
  Legendre();
  Kmax=FindKmax(control);
  ansWm=new char[10]; *ansWm='y';
  cout<<endl<<"\tFit Cusp? "; cin>>ansWm;
  Wm=new float*[Asize];
  for (i=0;i<Asize;i++)  Wm[i]=new float[Esize];
  BetaParm=new float[20];

  // Make range Ve[-Esize/2] through Ve[3*Esize/2],
  // Also the same range for Ee, RHO, Fe, Ge, and FGe matrix.
  Ve=new float[2*Esize];  Ve+=Esize/2;
  RHO=new float[2*Esize];  RHO+=Esize/2;
  Fe=new float*[11];  Ge=new float*[11];  FGe=new float**[11];
  for (i=0;i<11;i++) {
    Fe[i]=new float[2*Esize];  Fe[i]+=Esize/2;
    Ge[i]=new float[2*Esize];  Ge[i]+=Esize/2;
    FGe[i]=new float*[11];
    for (int j=0;j<11;j++) {FGe[i][j]=new float[2*Esize];  FGe[i][j]+=Esize/2;}
  }

  aMatrix=new int*[3];              // Arb. matrix for Beta parm.
  for (i=0;i<3;i++) aMatrix[i]=new int[11];
  BICO=new float*[11];              // Binomial Coefficient
  for (i=0;i<11;i++) BICO[i]=new float[11];
  DataF();

  Iconv=new float[2*Esize]; Iconv+=Esize/2;  oValue=new float[Esize];
```

```
  sIntegral=new float*[2]; lIntegral=new float*[2];  Aangle=new float[Asize];
  for (i=0;i<2;i++) {sIntegral[i]=new float[15]; lIntegral[i]=new float[15];}
  IntgOfZ=new double[31]; // 2*( # of Beta parm=15)+1
                          // IntgOfZ+=-1 or -10 or -100 are not guaranteed
                          // IntgOfZ+= 31 or 100 are not guaranteed
  IntgOfZ+=15;            // Make range, IntgOfZ[-15] through IntgOfZ[15]
}


/*-------------------------------------------------------------------------
   DESTRUCTOR: Model::~Model()
  -------------------------------------------------------------------------*/
Model::~Model()
{
  delete[] ansWm;  delete[] Wm;   delete[] aMatrix;  delete[] BICO;  delete[] oValue;  delete[] Ve;
  delete[] RHO;    delete[] Fe;   delete[] Ge;       delete[] FGe;   delete[] sIntegral;  delete[] lIntegral;
  delete[] IntgOfZ; delete[] Iconv; delete[] Aangle;  delete[] BetaParm;  delete[] LegendreCoeff;
}


/*-------------------------------------------------------------------------
   FUNCTION: Model::FindKmax()
  -------------------------------------------------------------------------*/
int Model::FindKmax(int *Control)
{
  // Find Kmax : We are only considering Beta parameter array[15].
  int iKmax=0;
  if ( ((Control[3] !=-2)&&(Control[3] !=-3)) || ((Control[2] !=-2)&&(Control[2] !=-3)) )  iKmax=1;
  if ( ((Control[5] !=-2)&&(Control[5] !=-3)) || ((Control[4] !=-2)&&(Control[4] !=-3)) )  iKmax=2;
  if ( ((Control[7] !=-2)&&(Control[7] !=-3)) || ((Control[6] !=-2)&&(Control[6] !=-3)) )  iKmax=3;
  if ( ((Control[9] !=-2)&&(Control[9] !=-3)) || ((Control[8] !=-2)&&(Control[8] !=-3)) )  iKmax=4;
  if ( ((Control[11] !=-2)&&(Control[11] !=-3)) || ((Control[10] !=-2)&&(Control[10] !=-3)) )  iKmax=5;
  if ( (Control[12] !=-2)&&(Control[12] !=-3) )  iKmax=6;
  if ( (Control[13] !=-2)&&(Control[13] !=-3) )  iKmax=8;
  if ( (Control[14] !=-2)&&(Control[14] !=-3) )  iKmax=10;
  return (iKmax);
}


/*-------------------------------------------------------------------------
   FUNCTION: float Access()
   OPERATION: Return the corresponding value of model function in (angle,energy) coordinate.
  -------------------------------------------------------------------------*/
float Model::Access(float *parm,float angleCh,float energyCh)
{
  // Prohibits the underflow exception of the coprocessor
  _control87(EM_UNDERFLOW,EM_UNDERFLOW);
  float DDCSLab;

  for (int i=0;i<20;i++)  BetaParm[i]=parm[i];
  TILT     = parm[20];          // % slope in x-angle [%]
  ACEN     = parm[21];          // Center channel in x-angle [ch]
```

```
ECEN    = parm[22];          // Center channel in energy [ch]
TW      = parm[23];          // % of Energy [%]
FWHM    = parm[24];          // % of Energy [%]
ARFin   = parm[25];          // Angular RF inner diameter [radian]
ARFout  = parm[26];          // Angular RF outer diameter [radian]
ASTEP   = parm[27];          // Angular step [radian]
ORIGIN  = parm[28];          // DAC origin voltage [Vdac]
SLOPE   = parm[29];          // DAC slope voltage [Vdac]
Aconst  = parm[30];          // Analyzer Constant [eV/Vdac]
Bconst  = parm[31];          // Analyzer Constant [eV]
ECUSP   = parm[32];          // Expected cusp energy [eV]
BKGDa   = parm[33];          // slope [count/ch]
BKGDb   = parm[34];          // count [count]
SLICE   = parm[35];          // channels in y-angle [ch]
Estep=SLOPE*Aconst/Qau;      // Energy step [au]
Eshift=(ORIGIN+ECEN*SLOPE)*Aconst+Bconst-ECUSP; // Cusp energy shift [eV]
ECUSP+=Eshift;               // Real cusp energy [eV]
ECUSP/=Qau;                  // Real cusp energy [au]
Vp=sqrt(2.0*ECUSP);          // Vproj [au]

// Calculate corresponding channels for angle and energy resolutions
ChOfERF=0.01*FWHM*ECUSP/Estep;  ChOfARF=ARFin/ASTEP;

// Calculate energy response spectrum
int NrOfERF;                 // Number of points in energy resolution func.
float cEnergy=((energyCh*SLOPE+ORIGIN)*Aconst+Bconst)/Qau; // Energy [au]
NrOfERF=EnergyRF(cEnergy);
float *nValue=new float[NrOfERF]; // Corresponding value at each point
for (i=0;i<NrOfERF;i++)  nValue[i]=oValue[i];

/*=========================
      CONSTRUCTION
  =========================*/
float E,Dfact;
for (i=(energyCh-NrOfERF/2);i<(energyCh+NrOfERF-NrOfERF/2);i++) {
  E=(((float)i*SLOPE+ORIGIN)*Aconst+Bconst)/Qau;    // Energy [au]
  if (E>=0) {Ve[i]=sqrt(2.0*E); RHO[i]=Ve[i]/Vp; MakeFGe(i);}
}
Dfact=-2.0*Ge[1][(int)energyCh];

float Theta,SUM,deltaR;
int NrOfCh=0;
#ifdef PillBoxARF
  rOfARF=0.5*ARFin;          // Radius of angular resolution function
#endif
#ifdef TrapeZoidalARF
  deltaR=(ARFout-ARFin)/2.0/NrOfARF;
#endif

NrOfYSliceConv=ceil(SLICE/2); // Maximum number of y-SLICE convolution
if (NrOfYSliceConv<1)  NrOfYSliceConv=1;
if ((int)SLICE%2==1) Rangle=ASTEP*fabs(angleCh-ACEN);      // SLICE: odd
```

```
else Rangle=ASTEP*sqrt(pow(fabs(angleCh-ACEN),2.0)+0.25);  // SLICE: even

do {
 /*-------------------------
   Angle convolution (Figure B.1)
   -------------------------*/
 for (i=(energyCh-NrOfERF/2);i<(energyCh-NrOfERF/2+NrOfERF);i++) {
  #ifdef TrapeZoidalARF
    float Asum=0.0;
    rOfARF=0.5*ARFout;    // Radius of angular resolution function
    for (int m=0;m<NrOfARF;m++) {
  #endif

  if (Rangle<=0.00001*rOfARF)  Iconv[i]=PI*(Evaluation(i,rOfARF)-Evaluation(i,0.0));
  else {
    if (Rangle<=rOfARF) {
      Theta=rOfARF-Rangle;  Iconv[i]=PI*(Evaluation(i,Theta)-Evaluation(i,0.0));
   dAngle=2.0*Rangle;
    }
    else {                    // Rangle>rOfARF
      Theta=Rangle-rOfARF;   Iconv[i]=0.0; dAngle=2.0*rOfARF;
    }
    dAngle/=(float)NrOfConeSector;
    for (int k=0;k<=NrOfConeSector;k++) {
      Tangle[k]=AngleRF('P',Theta,0);  Dsector[k]=Evaluation(i,Theta);  Theta+=dAngle;
    }
    Iconv[i]+=AngleRF('L',0.0,1)*(Dsector[1]-Dsector[0])+AngleRF('R',0.0,NrOfConeSector-1)*
              (Dsector[NrOfConeSector]-Dsector[NrOfConeSector-1]);
    for (k=1;k<NrOfConeSector-1;k++)   Iconv[i]+=AngleRF('C',0.0,k)*(Dsector[k+1]-Dsector[k]);
  } // End of 'else'

  #ifdef TrapeZoidalARF
      Asum+=Iconv[i]/NrOfARF;   // Iconv[i]*(Probability=1/NrOfARF)
      rOfARF-=deltaR;
    }
    Iconv[i]=Asum*Dfact;
#endif
#ifdef PillBoxARF
    Iconv[i]*=Dfact;
#endif
} // End of 'for' loop

/*-------------------------
   Energy convolution (Figure B.1)
   -------------------------*/
SUM=0.0;
for (int l=0;l<NrOfERF;l++)  SUM+=nValue[l]*Iconv[(int)energyCh+l-NrOfERF/2];
Aangle[NrOfCh]=SUM; NrOfCh++;
if ((int)SLICE%2==1)              // SLICE: odd
  Rangle=ASTEP*sqrt(pow(fabs(angleCh-ACEN),2.0)+pow((float)NrOfCh,2.0));
else                              // SLICE: even
  Rangle=ASTEP*sqrt(pow(fabs(angleCh-ACEN),2.0)+
```

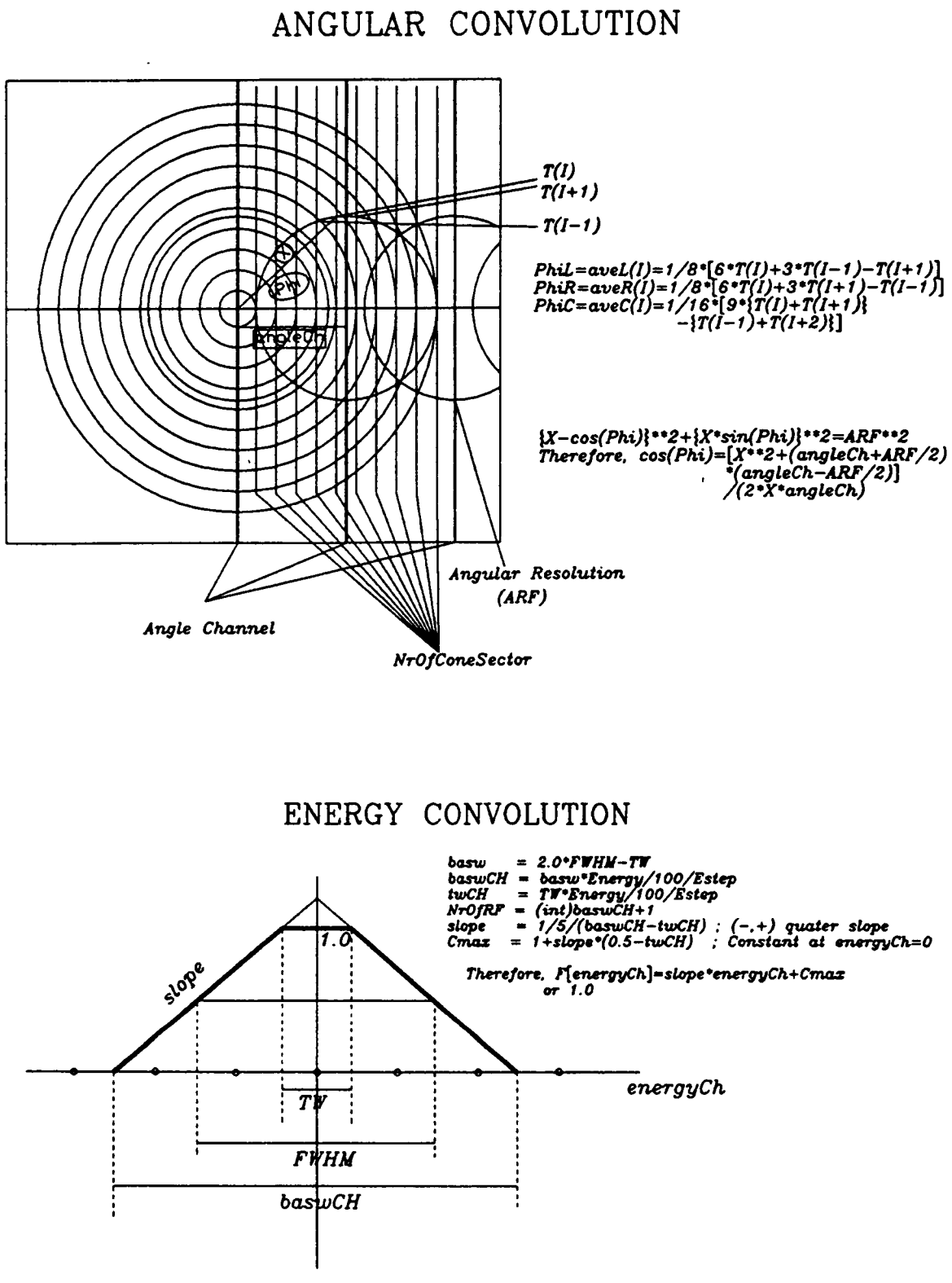



**Figure B.1**: Angular and energy resolution functions for the fitting routine.

```
            pow(((float)NrOfCh-0.5),2.0));
  } while (NrOfCh<NrOfYSliceConv);
  SUM=WidthConv();                    // Width Convolution

  // Add in angular tile to correct for non-constant efficiency in angle
  float TiltF=(angleCh-float(Asize-1)/2.0)*(TILT/100.0);  TiltF+=1.0;

  DDCSLab=SUM*(TiltF);  DDCSLab*=BetaParm[0];
  DDCSLab+=BKGDa*(energyCh-(Esize-1)/2.0)+BKGDb;  //Background

  delete[] nValue;
  return(DDCSLab);
}


/*=============================================================================
   FUNCTION:  Model::Mweight()
  ============================================================================*/
void Model::Mweight(float **Wm)
{
  for (int i=0;i<Asize;i++)
    for (int j=0;j<Esize;j++) {
      if ( (fabs((float)i-ACEN)<=ChOfARF/2.0)&&(fabs((float)j-ECEN)<=ChOfERF/2.0)&&
         ((*ansWm=='n')||(*ansWm=='N')) )
        Wm[i][j]=0.0;
      else  Wm[i][j]=1.0;
    }
}


/*=============================================================================
   FUNCTION:  Model::WidthConv()
  ============================================================================*/
float Model::WidthConv()
{
  float sum=0.0;
  if ((int)SLICE%2==1)                // SLICE: odd
    for (int i=0;i<NrOfYSliceConv;i++) { if (i==0) sum=Aangle[0];  else sum+=2.0*Aangle[i];}
  else                                // SLICE: even
    for (int i=0;i<NrOfYSliceConv;i++)  sum+=2.0*Aangle[i];
  return (sum);
}


/*=============================================================================
   FUNCTION:  Model::Legendre()
  ============================================================================*/
void Model::Legendre()
{
  int L,N;
  for (L=0;L<11;L++)  for (N=0;N<11;N++)  LegendreCoeff[L][N]=0.0;
  LegendreCoeff[0][0]=LegendreCoeff[1][1]=1.0;
```

```
  for (L=2;L<11;L++) {
    LegendreCoeff[L][0]=-(L-1.0)/L*LegendreCoeff[L-2][0];
    for (N=1;N<11;N++)
      LegendreCoeff[L][N]=((2.0*L-1.0)*LegendreCoeff[L-1][N-1]-(L-1.0)*LegendreCoeff[L-2][N])/L;
  }
}


/*-------------------------------------------------------------------------
   FUNCTION: Model::Evaluation()
-------------------------------------------------------------------------*/
inline float Model::Evaluation(int I,float ThetaL)
{
  // Prohibits the underflow exception of the coprocessor
  _control87(EM_UNDERFLOW,EM_UNDERFLOW);

  float Eval;
  // Return 0 at v'=0.0 evaluation
  if ((fabs(RHO[I]-1.0)<1.0e-05)&&(fabs(ThetaL)<1.0e-06)) return(Eval=0.0);
   // Create IntgOfZ(N): Integral of z^(n-1) where z=v'/v_p
  //              i.e. IntgOfZ(1)=z; IntgOfZ(0)=ln(z)
  IntgOfZ[1]=sqrt(1.0-2.0*RHO[I]*cos(ThetaL)+pow(RHO[I],2.0));
  if (fabs(IntgOfZ[1])<1.0e-37) return(Eval=0.0);  // Cusp in special case
  IntgOfZ[0]=log(IntgOfZ[1]);  IntgOfZ[-1]=1.0/IntgOfZ[1];

  int i,j;
  for (i=2;i<=Kmax+2;i++)         // Rest of z^(n) terms
    IntgOfZ[i]=IntgOfZ[i-1]*IntgOfZ[1];
  for (i=2;i<=Kmax-1;i++)         // Rest of 1/z^(n) terms
    IntgOfZ[-i]=IntgOfZ[-i+1]*IntgOfZ[-1];
  for (i=-Kmax+1;i<=Kmax+2;i++)    // Incorporate exponential-constant
    if (i!=0)                     // (log term has none)
      IntgOfZ[i]/=(float)i;

  ////////------- Integration of Multipoles ----------////////
  if (fabs(RHO[I]-1.0)<1.0e-06) {  // Do simpler integration code if v_e=v_p (RHO=1.0)
    for(i=0;i<=min(5,Kmax);i++) {  // for Kmax<6
      lIntegral[0][i]=0.0;         // n=0 indice
      for (j=0;j<=i;j++) lIntegral[0][i]+=LegendreCoeff[i][j]*FGe[0][j][I]*IntgOfZ[j+1];
      lIntegral[1][i]=0.0;         // n=1 indice
      for (j=0;j<=i;j++) lIntegral[1][i]+=LegendreCoeff[i][j]*FGe[0][j][I]*IntgOfZ[j+2];
      lIntegral[1][i]*=Vp;
    }
    if (Kmax>=6)                   // for high order multipoles
      for (i=6;i<=Kmax;i+=2) {lIntegral[0][i]=0.0;      // n=0 indice
        for (j=0;j<=i;j++) lIntegral[0][i]+=LegendreCoeff[i][j]*FGe[0][j][I]*IntgOfZ[j+1];
        lIntegral[1][i]=0.0;       // n=1 indice
        for (j=0;j<=i;j++) lIntegral[1][i]+=LegendreCoeff[i][j]*FGe[0][j][I]*IntgOfZ[j+2];
        lIntegral[1][i]*=Vp;
      }
  } // End of RHO=1.0 special case
```

```
  else {
    // Create sIntegral[n][k] for integral of [x^(k)*z^(n)*dz] where x=f/z+g*z
    for (j=0;j<=min(5,Kmax);j++) { // x^(k) for k<6
      sIntegral[0][j]=0.0;  sIntegral[1][j]=0.0;
      for (i=0;i<=j;i++) {
        sIntegral[0][j]+=FGe[i][j][I]*IntgOfZ[j-2*i+1];   // n=0 indice
        sIntegral[1][j]+=FGe[i][j][I]*IntgOfZ[j-2*i+2];   // n=1 indice
      }
    }
    if (Kmax>=6)
      for (j=6;j<=Kmax;j+=2) {     // high order maximum
        sIntegral[0][j]=0.0;  sIntegral[1][j]=0.0;
        for (i=0;i<=j;i++) {
          sIntegral[0][j]+=FGe[i][j][I]*IntgOfZ[j-2*i+1];   // n=0 indice
          sIntegral[1][j]+=FGe[i][j][I]*IntgOfZ[j-2*i+2];   // n=1 indice
        }
      }
    // Create lIntegral[n][k] for integral of [P_k(x)*z^(n)*dz] * v_p^(n)
    //    (P_k is the Legendre polynomial of order k)
    for (i=0;i<=min(5,Kmax);i++) { // for Kmax<6
      lIntegral[0][i]=0.0;       // n=0 indice
      for (j=0;j<=i;j++)  lIntegral[0][i]+=LegendreCoeff[i][j]*sIntegral[0][j];
      lIntegral[1][i]=0.0;       // n=1 indice
      for (j=0;j<=i;j++)  lIntegral[1][i]+=LegendreCoeff[i][j]*sIntegral[1][j];
      lIntegral[1][i]*=Vp;
    }
    if (Kmax>=6)                 // for high multipoles
      for (i=6;i<=Kmax;i+=2) {
        lIntegral[0][i]=0.0;       // n=0 indice
        for (j=0;j<=i;j++) lIntegral[0][i]+=LegendreCoeff[i][j]*sIntegral[0][j];
        lIntegral[1][i]=0.0;       // n=1 indice
        for (j=0;j<=i;j++) lIntegral[1][i]+=LegendreCoeff[i][j]*sIntegral[1][j];
        lIntegral[1][i]*=Vp;
      }
  } // End of else
  // Now create the normalized sum
  float Amp=BetaParm[0];  BetaParm[0]=1.0;  Eval=0.0;
  for (j=0;j<=min(5,Kmax);j++)     // for k<6
    for (i=0;i<=1;i++) {lIntegral[i][j]*=BetaParm[aMatrix[i][j]]*Amp;  Eval+=lIntegral[i][j];}
  if (Kmax>=6)                     // for higher order multipoles
    for (j=6;j<=Kmax;j+=2)
      for (i=0;i<=1;i++) {lIntegral[i][j]*=BetaParm[aMatrix[i][j]]*Amp;  Eval+=lIntegral[i][j];}
  BetaParm[0]=Amp;
  return (Eval);
}

/*=============================================================================
  FUNCTION:  Model::MakeFGe()
=============================================================================*/
void Model::MakeFGe(int I)
```

```
{
  Fe[0][I]=1.0; Ge[0][I]=1.0; Fe[1][I]=0.5*(pow(RHO[I],2.0)-1.0); Ge[1][I]=-0.5;
  for (int i=2;i<=Kmax;i++) {Fe[i][I]=Fe[i-1][I]*Fe[1][I]; Ge[i][I]=Ge[i-1][I]*Ge[1][I];}
  // Create matrix
  for (i=0;i<=min(5,Kmax);i++)  for (int j=0;j<=i;j++) FGe[j][i][I]=BICO[j][i]*Fe[j][I]*Ge[i-j][I];
  if (Kmax>=6)
    for (i=6;i<=Kmax;i+=2)  for (int j=0;j<=i;j++) FGe[j][i][I]=BICO[j][i]*Fe[j][I]*Ge[i-j][I];
}


#ifdef TrapezoidalERF
/*-------------------------------------------------------------------------
   FUNCTION: Model::EnergyRF()
   OPERATION: Produce trapezoidal energy resolution function. (Figure B.1)
-------------------------------------------------------------------------*/
inline int Model::EnergyRF(float &Energy)
{
  // Prohibits the underflow exception of the coprocessor
  _control87(EM_UNDERFLOW,EM_UNDERFLOW);

  int NrOfRF;
  float basw,baswCH,twCH,slope,Cmax;
  basw=2.0*FWHM-TW;
  baswCH=basw*Energy/(100.0*Estep);     // Base in channel
  twCH=TW*Energy/(100.0*Estep);         // Top in channel
  NrOfRF=(int)baswCH+1;

  if ((baswCH-twCH)<1.0e-5) baswCH=1.0e-5+twCH;  // Extreme case
  slope=1.0/0.5/(baswCH-twCH);                   // (-,+) quater slope
  Cmax=1.0+slope*0.5*twCH;                       // Constant at X=0; (0,Cmax)
  if ((NrOfRF%2)==1) {                           // Odd case
    int X;
    for (int i=0;i<=NrOfRF/2;i++) {
      X=-NrOfRF/2+i;  oValue[i]=slope*X+Cmax;
      if ((oValue[i]>1.0)&&((int)twCH!=0)) oValue[i]=1.0;
    }
    for (i=NrOfRF-1;i>=NrOfRF/2+1;i--)           // Make a symmetry along X=0
      oValue[i]=oValue[NrOfRF-1-i];
  }

  if ((NrOfRF%2)==0) {                           // even case
    float X;
    for (int i=0;i<=NrOfRF/2-1;i++) {
      X=-(float)NrOfRF/2.0+(float)i+0.5;  oValue[i]=slope*X+Cmax;
      if ((oValue[i]>1.0)&&((int)twCH!=0)) oValue[i]=1.0;
    }
    for (i=NrOfRF-1;i>=NrOfRF/2;i--)             // Make a symmetry along X=0
      oValue[i]=oValue[NrOfRF-1-i];
  }
  float sum=0.0;                                 // Normalization
  for (int i=0;i<NrOfRF;i++)  sum+=oValue[i];
  for (i=0;i<NrOfRF;i++)  oValue[i]/=sum;
```

```
  return (NrOfRF);
}
#endif


#ifdef GaussianERF
/*-------------------------------------------------------------------------
   FUNCTION: Model::EnergyRF()
   OPERATION: Produce Gaussian shape energy resolution function f(ε)=exp(-4ln2*(ε/dE)^2) where
              dE=ECUSP*FWHM
  -------------------------------------------------------------------------*/
inline int Model::EnergyRF(float &Energy)
{
  // Prohibits the underflow exception of the coprocessor
  _control87(EM_UNDERFLOW,EM_UNDERFLOW);
  float Efwhm=Energy*FWHM/100.0;              // FWHM energy
  int NrOfRF=(int)(2.0*Efwhm/Estep);          // Gaussian width = 4*Efwhm
  NrOfRF*=2; NrOfRF++;
  for (int i=0;i<NrOfRF;i++) {
    int X=-NrOfRF/2+i;  oValue[i]=exp(-4.0*log(2.0)*pow((float)X*Estep/Efwhm,2.0));
  }
  float sum=0.0;                              // Normalization
  for (i=0;i<NrOfRF;i++)  sum+=oValue[i];
  for (i=0;i<NrOfRF;i++)  oValue[i]/=sum;
  return (NrOfRF);
}
#endif


/*-------------------------------------------------------------------------
   FUNCTION: Model::AngleRF()  (Figure B.1)
  -------------------------------------------------------------------------*/
inline float Model::AngleRF(char Ref,float Theta,int NrOfS=0)
{
  // Prohibits the underflow exception of the coprocessor
  _control87(EM_UNDERFLOW,EM_UNDERFLOW);
  float AngleOfPHI;
  switch (Ref) {
    case ('P'): {   // Calculate 1/2 of the range of the PHI variablefor a given position of a circle in angle space
                AngleOfPHI=acos(min(1.0,max(-1.0,
                            (pow(Theta,2.0)+(Rangle+rOfARF)*(Rangle-rOfARF))/
                            (2.0*Theta*Rangle))));
                break;}
    case ('L'): {   // Left averaged quadratic interpolation to find a good 'average' value for dPhi
                AngleOfPHI=(6.0*Tangle[NrOfS]+3.0*Tangle[NrOfS-1]-Tangle[NrOfS+1])/8.0;
                break;}
    case ('R'): {   // Right averaged quadratic interpolation to find a good 'average' value for dPhi
                AngleOfPHI=(6.0*Tangle[NrOfS]+3.0*Tangle[NrOfS+1]-Tangle[NrOfS-1])/8.0;
                break;}
    case ('C'): {   // Center averaged quadratic interpolation to find a good 'average' value for dPhi
                AngleOfPHI=(9.0*(Tangle[NrOfS]+Tangle[NrOfS+1])-
                            (Tangle[NrOfS-1]+Tangle[NrOfS+2]))/16.0;
```

```
                break;}
    default: break;
  }
  return (AngleOfPHI);
}


/*-------------------------------------------------------------------------
   FUNCTION: Model::DataF()
-------------------------------------------------------------------------*/
void Model::DataF()
{
  aMatrix[0][0]= 0;   aMatrix[0][1]= 2;   aMatrix[0][2]= 4;   aMatrix[0][3]= 6;
  aMatrix[0][4]= 8;   aMatrix[0][5]=10;  aMatrix[0][6]=12;  aMatrix[0][7]=35;
  aMatrix[0][8]=13;  aMatrix[0][9]=35;  aMatrix[0][10]=14;
  aMatrix[1][0]= 1;   aMatrix[1][1]= 3;   aMatrix[1][2]= 5;   aMatrix[1][3]= 7;
  aMatrix[1][4]= 9;   aMatrix[1][5]=11;  aMatrix[1][6]=35;  aMatrix[1][7]=35;
  aMatrix[1][8]=35;  aMatrix[1][9]=35;  aMatrix[1][10]=35;
  aMatrix[2][0]=15;  aMatrix[2][1]=16;  aMatrix[2][2]=17;  aMatrix[2][3]=18;
  aMatrix[2][4]=19;  aMatrix[2][5]=35;  aMatrix[2][6]=35;  aMatrix[2][7]=35;
  aMatrix[2][8]=35;  aMatrix[2][9]=35;  aMatrix[2][10]=35;
  BICO[0][0]= 1;  BICO[0][1]= 1;  BICO[0][2]= 1;  BICO[0][3]= 1;  BICO[0][4]= 1;
                  BICO[0][5]= 1;  BICO[0][6]= 1;  BICO[0][7]= 1;  BICO[0][8]= 1;
                  BICO[0][9]= 1;  BICO[0][10]= 1;
  BICO[1][0]= 1;  BICO[1][1]= 1;  BICO[1][2]= 2;  BICO[1][3]= 3;  BICO[1][4]= 4;
                  BICO[1][5]= 5;  BICO[1][6]= 6;  BICO[1][7]= 7;  BICO[1][8]= 8;
                  BICO[1][9]= 9;  BICO[1][10]=10;
  BICO[2][0]= 1;  BICO[2][1]= 1;  BICO[2][2]= 1;  BICO[2][3]= 3;  BICO[2][4]= 6;
                  BICO[2][5]=10;  BICO[2][6]=15;  BICO[2][7]=21;  BICO[2][8]=28;
                  BICO[2][9]=36;  BICO[2][10]=45;
  BICO[3][0]= 1;  BICO[3][1]= 1;  BICO[3][2]= 1;  BICO[3][3]= 1;  BICO[3][4]= 4;
                  BICO[3][5]=10;  BICO[3][6]=20;  BICO[3][7]=35;  BICO[3][8]=56;
                  BICO[3][9]=84;  BICO[3][10]=120;
  BICO[4][0]= 1;  BICO[4][1]= 1;  BICO[4][2]= 1;  BICO[4][3]= 1;  BICO[4][4]= 1;
                  BICO[4][5]= 5;  BICO[4][6]=15;  BICO[4][7]=35;  BICO[4][8]=70;
                  BICO[4][9]=126; BICO[4][10]=210;
  BICO[5][0]= 1;  BICO[5][1]= 1;  BICO[5][2]= 1;  BICO[5][3]= 1;  BICO[5][4]= 1;
                  BICO[5][5]= 1;  BICO[5][6]= 6;  BICO[5][7]=21;  BICO[5][8]=56;
                  BICO[5][9]=126; BICO[5][10]=252;
  BICO[6][0]= 1;  BICO[6][1]= 1;  BICO[6][2]= 1;  BICO[6][3]= 1;  BICO[6][4]= 1;
                  BICO[6][5]= 1;  BICO[6][6]= 1;  BICO[6][7]= 7;  BICO[6][8]=28;
                  BICO[6][9]=84;  BICO[6][10]=210;
  BICO[7][0]= 1;  BICO[7][1]= 1;  BICO[7][2]= 1;  BICO[7][3]= 1;  BICO[7][4]= 1;
                  BICO[7][5]= 1;  BICO[7][6]= 1;  BICO[7][7]= 1;  BICO[7][8]= 8;
                  BICO[7][9]=36;  BICO[7][10]=120;
  BICO[8][0]= 1;  BICO[8][1]= 1;  BICO[8][2]= 1;  BICO[8][3]= 1;  BICO[8][4]= 1;
                  BICO[8][5]= 1;  BICO[8][6]= 1;  BICO[8][7]= 1;  BICO[8][8]= 1;
                  BICO[8][9]= 9;  BICO[8][10]=45;
  BICO[9][0]= 1;  BICO[9][1]= 1;  BICO[9][2]= 1;  BICO[9][3]= 1;  BICO[9][4]= 1;
                  BICO[9][5]= 1;  BICO[9][6]= 1;  BICO[9][7]= 1;  BICO[9][8]= 1;
                  BICO[9][9]= 1;  BICO[9][10]=10;
  BICO[10][0]= 1; BICO[10][1]= 1; BICO[10][2]= 1; BICO[10][3]= 1; BICO[10][4]= 1;
```

```
                BICO[10][5]= 1; BICO[10][6]= 1; BICO[10][7]= 1; BICO[10][8]= 1;
                BICO[10][9]= 1; BICO[10][10]= 1;
}


/*-------------------------------------------------------------------------
   FUNCTION: Model::Conversion()
  -------------------------------------------------------------------------*/
void Model::Conversion(float &Vangle,float &Venergy,float angleCh,float energyCh)
{
  #ifdef PlotIn_au
    float E=((energyCh*SLOPE+ORIGIN)*Aconst+Bconst)/Qau;    // Energy [au]
    Venergy=sqrt(2.0*E);                                    // Velocity [au]
    Vangle=Venergy*tan(ASTEP*(angleCh-ACEN));               // Energy [au]
  #endif

  #ifdef PlotIn_eV
    Vangle=ASTEP*(angleCh-ACEN)*180.0/PI;                   // Angle [degree]
    Venergy=(energyCh*SLOPE+ORIGIN)*Aconst+Bconst;          // Energy [eV]
  #endif
}
```

## /* Model.h */

```
#ifndef __Model_h
#define __Model_h 1
#define PI 3.14159265358979323846
#define VC 137.03604          // Velocity of light in a.u.
#define Kev 0.454             // Vstep(V)=Kev*Estep(eV) : analyzer constant
#define Qau 27.21             // Estep(eV)=Qau*Estep(au)
#define Kau 12.21729          // Vstep(V)=Kau*Estep(au) : analyzer constant
#define NrOfConeSector 9      // # of Theta cone sectors, ideal if infinitive
#define NrOfARF 5             // # of angular resolution function slices
#define PillBoxARF            // Pillbox angular resolution function
//#define TrapeZoidalARF      // Trapezoidal angular resolution function
#define TrapezoidalERF        // Trapezoidal energy resolution function
//#define GaussianERF         // Gaussian energy resolution function

class Model {
  private:
    char *ansWm;
    int Asize,Esize,Kmax,**aMatrix,NrOfYSliceConv;
    double **LegendreCoeff,*IntgOfZ;
    float **Wm,**BICO,Vp,Estep,*Aangle,dAngle, rOfARF,Rangle,Tangle[NrOfConeSector+1],
          Dsector[NrOfConeSector+1],*Ve,*RHO,**Fe,**Ge,***FGe,
          TILT,ACEN,ECEN,TW,FWHM,ARFin,ARFout,ASTEP,ORIGIN,SLOPE,
          ECUSP,BKGDa,BKGDb,Aconst,Bconst,SLICE,
          *BetaParm,ChOfARF,ChOfERF,*oValue,**sIntegral,**lIntegral,*Iconv;
    inline int EnergyRF(float &Energy);
    inline float AngleRF(char Ref,float Theta,int NrOfS);
    inline float Evaluation(int I,float Theta);
```

```
    void Legendre();
    void DataF();
    void MakeFGe(int I);
    int FindKmax(int *Control);
    float WidthConv();

  public:
    float Eshift;
    Model(int *control,int Asize,int Esize);
    ~Model();
    float Access(float *parm,float angleCh,float energyCh);
    void Mweight(float **Wm);
    void Conversion(float &Vangle,float &Venergy,float angleCh,float energyCh);
};
#define PlotIn_eV       // GnuPlot unit ([angle],[eV])
//#define PlotIn_au     // GnuPlot unit ([au],[au])
#endif /* __Model_h */
```

## /* pchisq.cpp */

```
#include "Bevington.h"
/*------------------------------------------------------------------------------
   FUNCTION: float pchisq()
   OPERATION: Evaluate probability for exceeding CHI square.
  ------------------------------------------------------------------------------*/
float pchisq(int &NFREE,float &CHISQR)
{
  int i,even,max;
  long double z,term,sum;
  float free,fl,pwr,pchi2;

  if (NFREE<=0) {pchi2=0.0; goto END;}
  free=NFREE; z=CHISQR*free/2.0; even=2*(NFREE/2);
  if (NFREE<=even) {
    BACK:
    max=NFREE/2; term=1.0; sum=0.0;
    for (i=1; i<=max; i++) {fl=i; sum+=term; term=term*z/fl;}
    pchi2=sum*exp(-z);
  }
  else {
    if(z>25.0) {z=CHISQR*(free-1.0)/2.0; goto BACK;}
    pwr=free/2.0; term=1.0; sum=term/pwr;
    for (i=1; i<=1000; i++) {
      fl=i; term=-term*z/fl; sum+=term/(pwr+fl);
      if (fabs(term/sum)<=0.00001) {break;}
    }
    pchi2=1.0-pow(z,pwr)*sum/Gamma(pwr);
  }
```

```
  END:
  return(pchi2);
}
```

## /* Gamma.cpp */

```
#include "Bevington.h"
/*-------------------------------------------------------------------------
   FUNCTION: float Gamma()
   OPERATION: Calculate the gamma function for integers and half-integers.
 -------------------------------------------------------------------------*/
float Gamma(float &x)
{
  int i,n;
  double prod,sum,f1;
  float xn,GammA;

  /* Integerize argument */
  n=x-0.25; xn=n;
  if ((x-xn)>0.75) {/* Argument is integer */ GammA=factor(n); goto END;}

  /* Argument is half-integer */
  prod=1.77245385;
  if (n<=0) {GammA=prod; goto END;}
  if (n<=10) {
    for (i=1; i<=n; i++) {f1=i; prod*=f1-0.5;}
    GammA=prod; goto END;
  }

  sum=0.0;
  for (i=11; i<=n; i++) {f1=i; sum+=log10(f1-0.5);}
  GammA=prod*639383.8623*exp(sum);
  END:
  return(GammA);
}
```

## /* factor.cpp */

```
#include "Bevington.h"
/*-------------------------------------------------------------------------
   FUNCTION: float factor()
   OPERATION: Calculates factorial function for integers
 -------------------------------------------------------------------------*/
float factor(int &n)
{
  int i;
  double f1,sum;
  float factor=1.0;

  if (n<=1) {goto END;}
```

```
  if (n<=10) {    /* n less than 11 */
    for (i=2; i<=n; i++) {f1=i;  factor*=f1;}
    goto END;
   }

   /* n greater than 10 */
   sum=0.0;
   for (i=11; i<=n; i++) {f1=i;  sum+=log10(f1);}
   factor=3628800.0*exp(sum);

   END:
   return(factor);
}
```

## /* determ.cpp */

```
#include "Bevington.h"
/*------------------------------------------------------------------------------
   FUNCTION: float determ()
   OPERATION: Calculate the determinant of a square matrix. This subprogram destroys the input matrix
              array dimension statement valid for Norder up to 35.
------------------------------------------------------------------------------*/
float determ(float array[][Cols],int Norder)
{
  int i,j,k,k1;
  float save,Determ=1.0;
  for (k=0; k<Norder; k++) {

  /* Interchange columns if diagonal element is zero */
    if (array[k][k]==0.0) {
      for (j=k; j<Norder; j++) if (array[k][j]!=0.0) {goto AWAY;}
      Determ=0.0;  return (Determ);
      AWAY:
      for (i=k; i<Norder; i++) {save=array[i][j];  array[i][j]=array[i][k];  array[i][k]=save;}
      Determ=-Determ;
    }
    /* Subtract row k from lower rows to get diagonal matrix */
    Determ*=array[k][k];
    if (k<Norder) {
      k1=k+1;
      for (i=k1; i<Norder; i++)
        for (j=k1; j<Norder; j++) array[i][j]+=-array[i][k]*array[k][j]/array[k][k];
    }
  }
  return (Determ);
}
```

## /* MatInv.cpp */

```
#include "Bevington.h"
```

```
/*=============================================================================
   FUNCTION: void MatInv()
  ============================================================================*/
void MatInv(float **array,int &norder,float &det)
{
  int i,j,k,l;
  float amax,save,ik[35],jk[35];
  det=1.0;
  for (k=0; k<norder; k++) {
    amax=0.0;
    /* Find largest element array[i][j] in rest of matrix */
    CASE1:
    for (i=k; i<norder; i++)
      for (j=k; j<norder; j++)
        if (fabs(amax)<=fabs(array[i][j])) {amax=array[i][j]; ik[k]=i; jk[k]=j;}

    /* Interchange rows and columns to put amax in array[k][k] */
    if (amax==0.0) {det=0.0; goto END;}
    i=ik[k];
    if (i<k) goto CASE1;  if (i==k) goto CASE2;
    for (j=0; j<norder; j++) {save=array[k][j]; array[k][j]=array[i][j]; array[i][j]=save;}
    CASE2:
    j=jk[k];
    if (j<k) goto CASE1;  if (j==k) goto CASE3;
    for (i=0; i<norder; i++) {save=array[i][k];  array[i][k]=array[i][j];  array[i][j]=save;}

    /* Accumulate elements of inverse matrix */
    CASE3:
    for (i=0; i<norder; i++) {if (i!=k)  array[i][k]=-array[i][k]/amax;}
    for (i=0; i<norder; i++)
      for (j=0; j<norder; j++)
        if (i!=k)  if (j!=k)  array[i][j]+=array[i][k]*array[k][j];
    for (j=0; j<norder; j++) {if (j!=k)  array[k][j]/=amax;}
    array[k][k]=1.0/amax;  det*=amax;
  }

  /* Restore ordering of matrix */
  for (l=0; l<norder; l++) {
    k=norder-l-1;  j=ik[k];
    if (j<=k) goto CASE4;
    for (i=0; i<norder; i++) {save=array[i][k];  array[i][k]=-array[i][j];  array[i][j]=save;}
    CASE4:
    i=jk[k];
    if (i>k)
      for (j=0; j<norder; j++) {save=array[k][j];  array[k][j]=-array[i][j];  array[i][j]=save;}
  }
  END:;
}
```

```
/* Bevington.h */
#ifndef __Bevington_h
#define __Bevington_h 1
#include <math.h>
#define Cols 3
float pchisq(int &,float &);
float Gamma(float &);
float factor(int &);
float determ(float [][Cols],int);
void MatInv(float **array,int &norder,float &det);
#endif /* Bevington_H */
```

# Vita

Jong Yeon Lim was born in Andong, Korea, on March 11, 1959. He attended elementary school there, and graduated from Dongsung High School in Seoul in February 1978. The next March he entered Kyung-Hee University in Seoul. In 1981 he joined military service, and served in army as a soldier for three years.

In 1985, he entered the bachelor's program in Mathematics at the University of Tennessee, Martin, transferred to the University of Tennessee at Knoxville, and graduated in May 1990 with a Bachelor of Science degree in Physics with High Honor.

In September of that year he entered the Graduate School of the University of Tennessee, Knoxville. He has worked on the Doctor of Philosophy degree with a major in Physics since 1991, under the supervision of Professor Stuart B. Elston within the University of Tennessee Atomic Physics Group, working on a variety of problems in ion-surface collisions.

The author is a member of Phi Kappa Phi and the American Physical Society. Mr. Lim will be employed by the department of Chemistry of the University of Tennessee after graduation.