ELECTRON TRANSFER IN ION-ATOM COLLISIONS

by

LAURA NORMAN TUNNELL

B.S., East Texas State University, 1976

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Physics

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1979

Approved:

Chi-Dey Lin

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

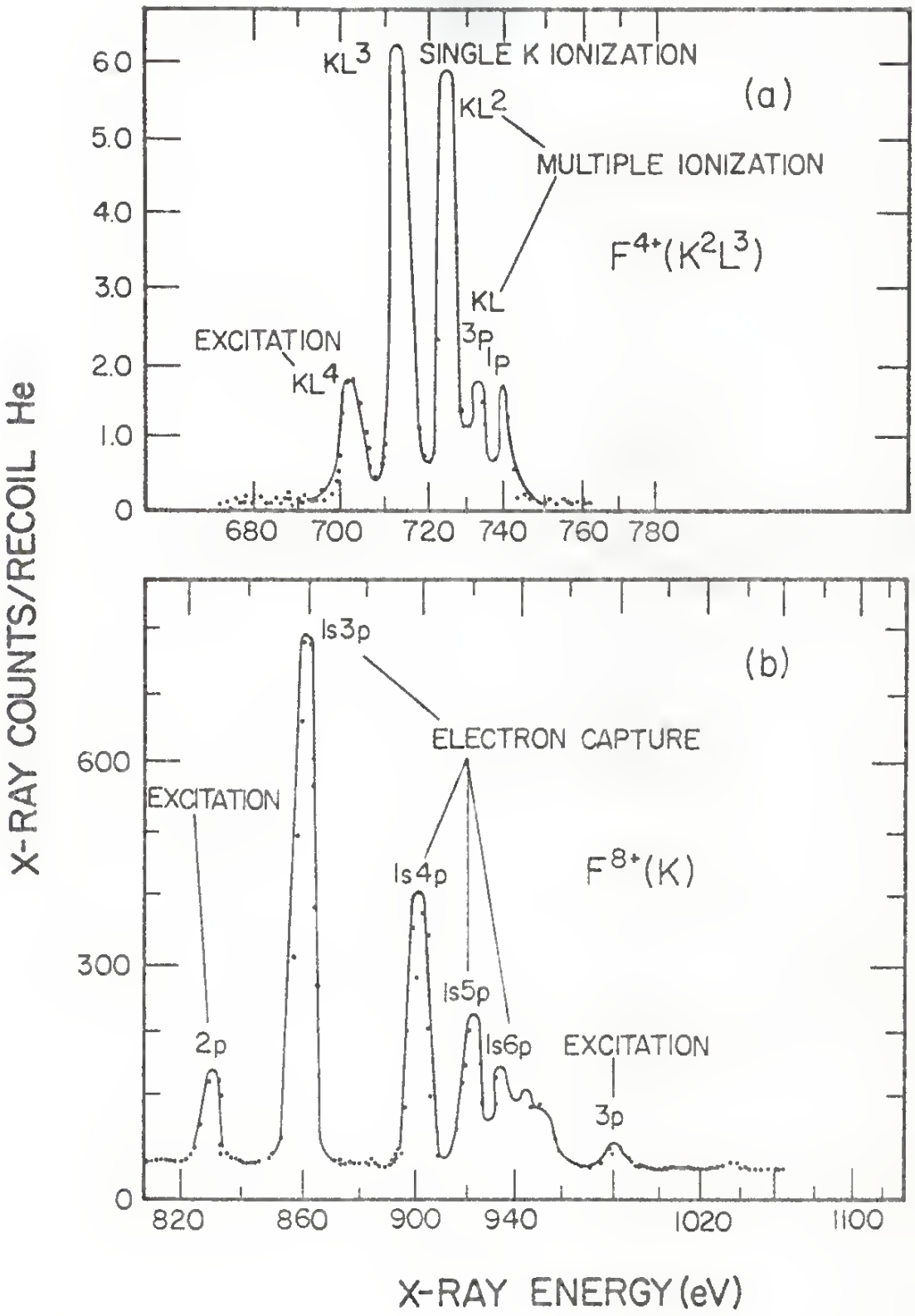
The author wishes to express appreciation to Dr. C. D. Lin for his patience and guidance throughout the course of this work. She is grateful to the U. S. Department of Energy, Division of Chemical Sciences for financial support under contract no. EY-76-S-02-2753. Use of the computer facilities at Argonne National Laboratory and the assistance of Drs. K. T. Lu and K. T. Cheng are also appreciated. The author wishes to thank Richard Vore for his technical assistance and Rhonda Born for the painstaking job of typing this thesis.

CHAPTER I: INTRODUCTION

Electron capture has been a subject of interest for both experimentalists and theoreticians in recent years. The transfer of an electron from the bound state of one system to the bound state of another is of fundamental interest; it is the very basis of many chemical processes. An understanding of charge exchange is needed in order to explain the bulk behavior of plasmas in thermonuclear reactions. It is also an important inelastic process occurring in ion-atom collisions.

Along with the widespread use of tandem Van de Graaff accelerators in recent years, ion-atom collisions have been investigated by experimentalists over a spectrum of projectile velocities and charge states. In a violent ion-atom collision electrons in both the target and projectile can undergo a variety of single or multiple events. These processes are generally classified as excitation, ionization, or charge transfer. To illustrate this point Figs. I.1a and I.1b show the K-shell x-ray spectra of fluorine ions resulting from the bombardment of neutral helium by F^{4+} and F^{8+}, respectively, at a projectile energy of 15 Mev.[1] In addition to single excitation and ionization events Fig. I.1a displays prominent features at higher x-ray energies which are the result of multiple ionization. Fig. I.1b exhibits peaks due to single excitation similar to those of Fig. I.1a, as well as structure from excited states of F^{7+} formed during the collision by the transfer of a target electron to the projectile.

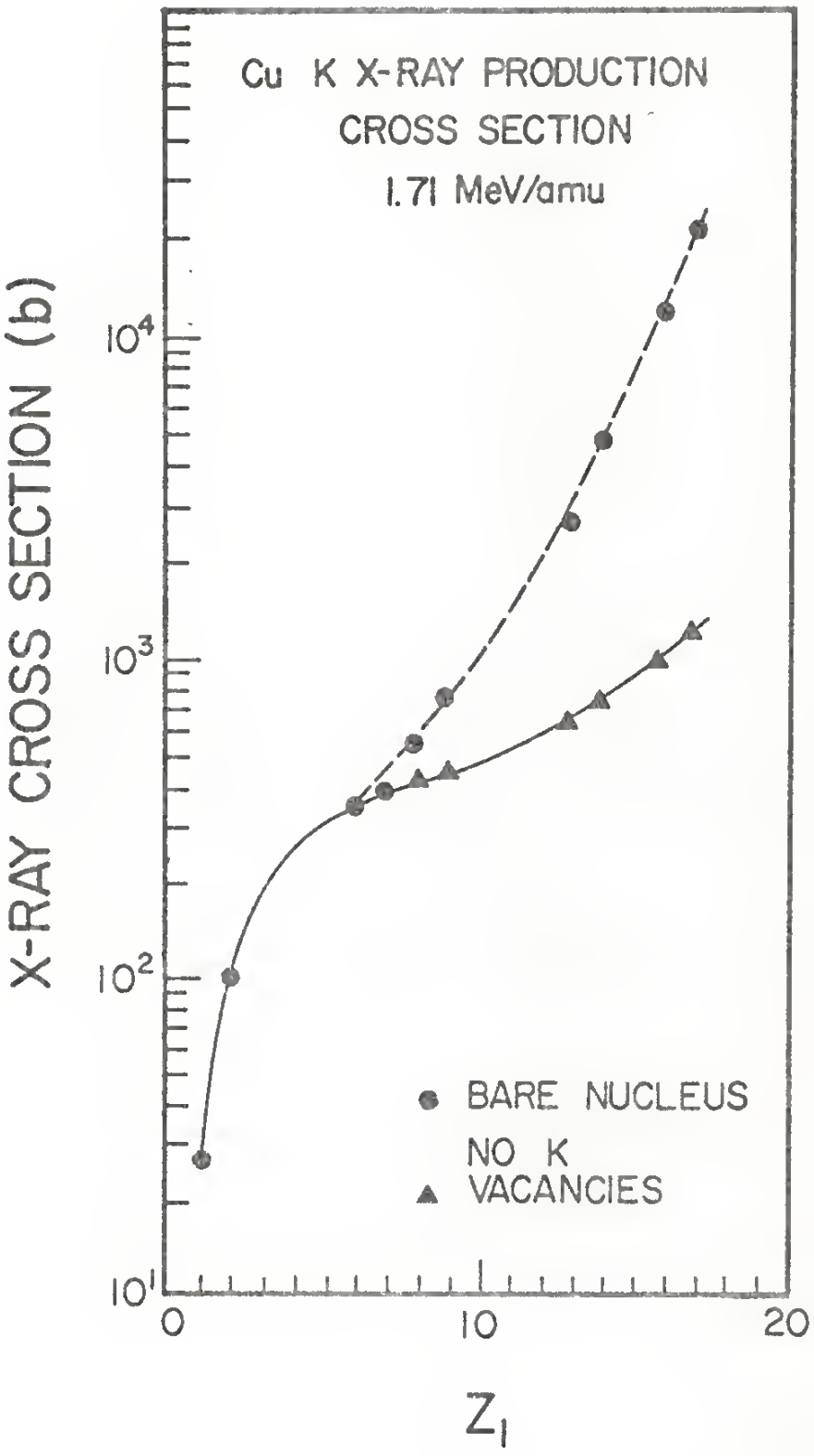
Figure I.1: X-ray spectra resulting from the bombardment of helium by (a), F^{4+} and (b), F^{8+} at an incident energy of 15 Mev. The data are from the work of Richard, et. al.[1].


X-RAY COUNTS/RECOIL He
6.0
5.0
4.0
3.0
2.0
1.0
0
680
700
720
740
760
780
KL3
SINGLE K IONIZATION
KL2
(a)
MULTIPLE IONIZATION
F4+(K2L3)
KL
3P
1P
EXCITATION
KL4
600
300
0
820
860
900
940
1020
1100
ls3p
(b)
ELECTRON CAPTURE
EXCITATION
ls4p
F8+(K)
ls5p
2p
ls6p
EXCITATION
3p
X-RAY ENERGY (eV)

Direct Coulomb ionization is known to be the primary mechanism for inner shell vacancy production in targets bombarded by lighter ions. As the nuclear charge of the projectile increases, however, the electron capture mechanism is found to be important as well. Fig. I.2 shows the Cu K x-ray production cross section as a function of projectile atomic number for velocities corresponding to 1.71 Mev/amu.[2] Two types of projectiles are used, one with and one without K shell vacancies. The K x-rays in the target follow as the result of vacancies created by excitation, ionization, or capture of a Cu K shell electron. The contribution from each of these mechanisms to the total production of target K vacancies is approximately the same for both types of projectiles with the exception of the K-K capture process. The latter can occur only if the impinging ion has a K vacancy. As the collision becomes more symmetric, the K-K transfer process becomes more important to the production of target K vacancies.

Over the years various theoretical models have been proposed to explain vacancy production in ion-atom collisions. The electron promotion model of Fano and Lichten[3] has been very successful in providing a qualitative description for collision velocities which are much less than the characteristic orbital velocity of the electron under observation. This theory has recently been put in quantitative form by Briggs and Macek.[4] The First Born theory has been adequate to describe the excitation and ionization mechanisms of asymmetric systems at higher collision velocities.[5] A description of the charge transfer mechanism, however, is much more complex.[5,6,7]

Figure 1.2: Cu K x-ray production as a function of projectile charge at projectile velocities corresponding to 1.71 Mev/amu. As the atomic number increases, the data exhibit a pronounced distinction between projectiles with and without K shell vacancies. The experimental points are from Gardner, et. al.[2].


Cu K X-RAY PRODUCTION
CROSS SECTION
1.71 MeV/amu
X-RAY CROSS SECTION (b)
10^4
10^3
10^2
10^1
0
10
20
Z_1
BARE NUCLEUS
NO K
VACANCIES

Like ionization and excitation, the charge transfer process was originally described by the First Born Approximation. From the beginning questions arose concerning the presence of an internuclear interaction in the electronic transition amplitude. It was argued that an interaction between the colliding nuclei could not directly affect an electronic transition except to deflect the projectile. This is the reasoning which led to the Oppenheimer[8], Brinkman, and Kramers (OBK) Approximation.[9] Other authors, however, preferred to retain the internuclear term. Bates and Dalgarno[10] argued that, in some way, it compensated for the nonorthogonality between the initial and final state wavefunctions. (The appearance of this term can be traced to the fact that this nonorthogonality was not formally recognized.) Jackson and Schiff (JS)[11] believed that retaining the internuclear interaction improved the convergence of the Born series. The JS method appeared to be successful for the simple transfer process

$$P + H(1s) \rightarrow H(1s) + P \qquad \text{I.1}$$

while the OBK results were an order of magnitude too high. Despite its success for K-K capture by protons on hydrogen, the straightforward generalization of the JS method to arbitrary systems again failed.[11,12,13]

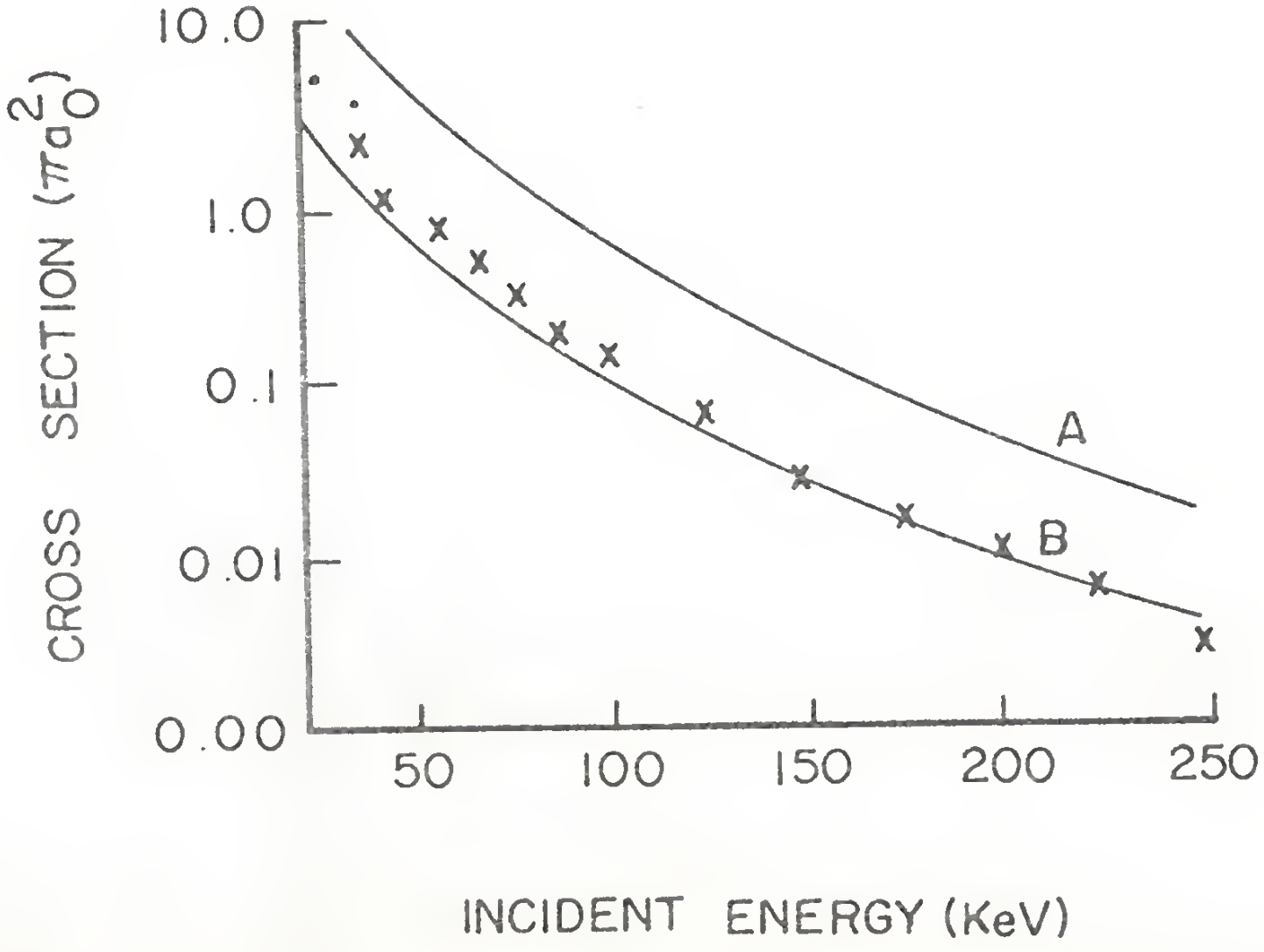
In 1958 D. R. Bates reformulated the electron transfer problem.[14] Whereas the First Born theory solved the time dependent Schrödinger equation by means of a perturbation expansion, Bates employed a truncated eigenfunction expansion to solve the time dependent Schrödinger equation. Unlike the various Born formulations, Bates' Two State Atomic Expansion

(TSAE) method took formal cognizance of the nonorthogonality between the initial and final states of a captured electron. The resulting expression for the capture probability is easily shown to be independent of any internuclear interaction. Thus, the difficulty formerly associated with this term was resolved. When minimal approximations are made, the TSAE expression is equivalent to the Distorted Wave Born Approximation (DWBA)[15] of Bassel and Gerjuoy. The TSAE results simulate the JS and OBK expressions when further approximations are made. Results of the OBK, the JS, and the TSAE methods are compared with data in Fig. I.3 for K-K capture by fast protons incident on atomic hydrogen. The two curves, labeled A and B, are results of the OBK and JS[16] methods respectively. For this case the TSAE[17] and JS results are indistinguishable to the scale drawn.

In this work the TSAE method has been generalized to study single electron transfer in multielectron ion-atom collisions. The multielectron systems are described within the independent particle approximation in order to avoid undue mathematical complications. For collision velocities of interest the motion of the nuclei can be treated within the impact parameter approximation. This is a semi-classical method in which the nuclear motion is treated classically and the electronic motion, quantum mechanically. The projectile is deflected very little by the target at these velocities; thus straight line trajectories are adopted.

The earlier work of this type[20] generalized the TSAE method to transfer processes involving multielectron systems and was applied to K-K capture by fast protons on carbon, nitrogen, oxygen, neon, and argon targets. Because K-shell electronic motion is dominated by the influence of the

Figure I.3: The total cross section for electron capture by protons from atomic hydrogen. Curve A results from the Brinkman-Kramers approximation; curve B, the Two State and the Born approximation. The latter two are indistinguishable to the scale shown and are drawn as one. Experimental data are from refs. no. 18 and 19.


10.0
1.0
0.1
0.01
0.00
CROSS SECTION (πa_0^2)
50
100
150
200
250
A
B
INCIDENT ENERGY (KeV)

nucleus rather than the aggregate influence of the passive electrons, it was reasoned that a screened hydrogenic wavefunction and potential would give an adequate description of the multielectron targets. Results of these calculations agreed reasonably well with experimental data.

As opposed to the description of inner shell capture processes, a description of outer shell capture requires a more complex atomic model. Whereas a K-shell electron's interaction with the nucleus is much stronger than its interaction with the other electrons, the influence of neighboring electrons on an outer shell electron is comparable to that of the nucleus. In this work it was assumed that the most important effect of the passive electrons on the electron under observation was to provide screening of the nuclear potential. This screening, described by a Herman-Skillman screening function,[21] enabled a study of transfer from outer shells. Comparison with earlier K-K calculations was made as well in order to verify the validity of the simple atomic model used to describe inner shell capture processes.

In Chapter II the details of the TSAE method are given. Chapter III describes the numerical techniques employed in this work to perform the calculations. A discussion of the results is given in Chapter IV and Chapter V summarizes the work. Appendix I contains the derivations of the relevant formulas. The computer coding written to perform the calculations is listed in Appendix II and the publication connected with this work is given in Appendix III. Atomic units will be used.

CHAPTER II: DESCRIPTION OF THE TSAE METHOD

In this study the single electron transfer problem is treated within the independent particle approximation. Only one electron is considered active. The presence of the passive electrons is acknowledged through the screening of the nuclear potential.

A. Derivation of the Coupled Equations. In the impact parameter approximation the electronic wavefunction satisfies the time dependent Schrödinger equation

$$i \frac{\partial}{\partial t} \Psi(\vec{r},t) = H(t)\ \Psi(\vec{r},t) \qquad \text{II.1}$$

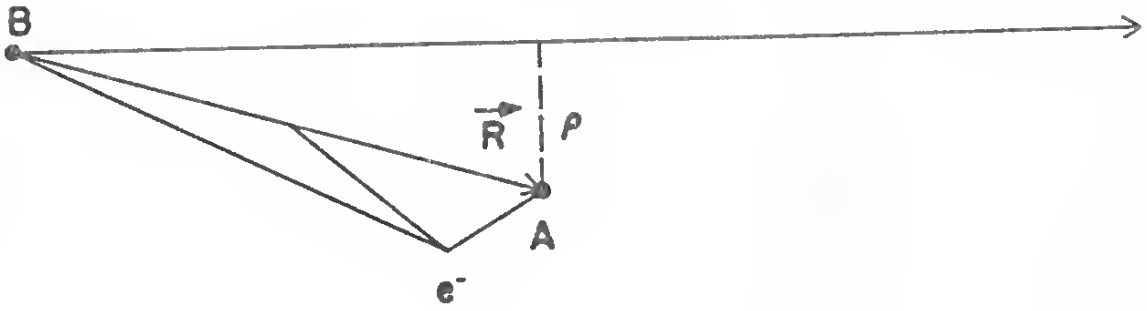
where $H(t)$ is assumed to have the form

$$H(t) = -1/2\ \nabla^2 + V_A(\vec{r}_A) + V_B(\vec{r}_B) \qquad \text{II.2}$$

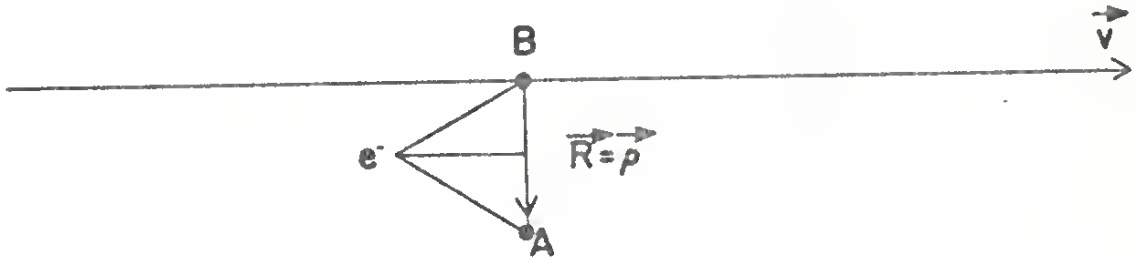
The time dependence of the Hamiltonian occurs through the change of the internuclear distance as the collision progresses. $\vec{r}_A(\vec{r}_B)$ connects the electron to the target (projectile) as shown in Fig. II.1. $V_A(V_B)$ is the potential experienced by the electron at infinite internuclear separation when it is bound to nucleus A(B).

Eqn. II.1 can be conveniently solved by the method of eigenfunction expansions. The type of basis set to be chosen in a truncated expansion depends on the ratio of the projectile's velocity to the characteristic orbital velocity of the electron under consideration. When this ratio is small, a molecular basis set is appropriate. Atomic basis sets are used for moderate to high collision velocities. In describing the charge cap-

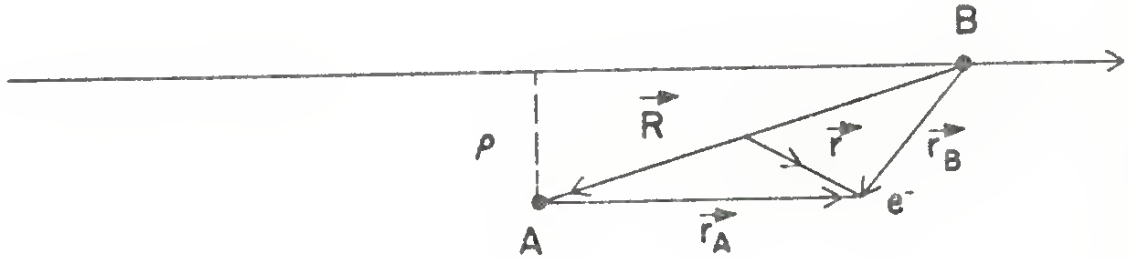
Figure II.1: The coordinate system used to calculate electron capture probabilities within the impact parameter formulation. ρ is the impact parameter, $\vec{R}$ is the internuclear line joining the target, A, to the projectile, B, and $\vec{r}_A(\vec{r}_B)$ connects nucleus A(B) with the electron. $\vec{r}$ is the position vector of the electron with respect to the origin of this coordinate system, the mid-point of the internuclear axis.



t = 0



t > 0



ture process, it is convenient to use an expansion centered about both the target and the projectile. This insures that each term in the expansion will be an eigenfunction of H in the limit of infinite internuclear separation. With this in mind, the electronic wavefunction can be written

$$\Psi(\vec{r}, t) = \sum_i a_i(t)\, \psi_i(\vec{r}_A)\, e^{-i\varepsilon_i^A t} + \sum_j b_j(t)\, \psi_j(\vec{r}_B)\, e^{-i\varepsilon_j^B t} \qquad \text{II.3}$$

Each ψ_i (ψ_j) is a product of a stationary state wavefunction centered about nucleus A(B) and a 'plane-wave-like' phase factor. These phase factors, necessary to insure translational invariance of the system, represent the momentum that the electron has by virtue of being bound to one or the other of the two nuclei at infinite internuclear separation.

$$\psi_i(\vec{r}_A) = \phi_i(\vec{r}_A)\, e^{i\left[\frac{\vec{v}}{2} \cdot \vec{r} - 1/2(\frac{v}{2})^2 t\right]}$$
$$\psi_j(\vec{r}_B) = \phi_j(\vec{r}_B)\, e^{-i\left[\frac{\vec{v}}{2} \cdot \vec{r} + 1/2(\frac{v}{2})^2 t\right]} \qquad \text{II.4}$$

where $\vec{v}$ is the collision velocity and $\vec{r}$, the position of the electron with respect to the origin, (the midpoint of the internuclear axis, R, as shown in Fig. II.1).

$\phi_i(r_A)$ and $\phi_j(r_B)$ are stationary eigenstates satisfying

$$\left[-1/2\nabla_A^2 + V_A(\vec{r}_A) - \varepsilon_i^A\right] \phi_i(\vec{r}_A) = 0$$
$$\left[-1/2\nabla_B^2 + V_B(\vec{r}_B) - \varepsilon_j^B\right] \phi_j(\vec{r}_B) = 0 \qquad \text{II.5}$$

In the Two State Approximation only one term from each sum in Eqn. II.3

is retained, those representing the initial and final state of a captured electron. (Henceforth, the subscripts i and j will be replaced by A and B.) This truncated version of the wavefunction is substituted into the Schrödinger equation, II.1. The result is

$$i[\dot{a}\ \psi_A\ e^{-i\varepsilon_A t} + \dot{b}\ \psi_B\ e^{-i\varepsilon_B t}]$$
$$= a\ V_B\ \psi_A\ e^{-i\varepsilon_A t} + b\ V_A\ \psi_B\ e^{-i\varepsilon_B t} \qquad \text{II.6}$$

The overlap of this equation is taken with $\psi_A^* e^{i\varepsilon_A t}$ and $\psi_B^* e^{i\varepsilon_B t}$, respectively, resulting in the final set of coupled equations

$$i(a + S_{AB}\dot{b}) = H_{AA}a + H_{AB}b$$
$$i(S_{BA}\dot{a} + \dot{b}) = H_{BA}a + H_{BB}b, \qquad \text{II.7}$$

where $S_{AB} = S_{BA}^*$ is the overlap between the initial and final states

$$S_{AB} = \int d\tau\ \phi_A^*\ \phi_B\ e^{i(\vec{v}\cdot\vec{r} + wt)} \qquad \text{II.8}$$

and $w = \varepsilon_A - \varepsilon_B$. The diagonal and off-diagonal elements of H are the direct and exchange elements of the interaction matrix.

$$H_{AA} = \int d\tau\ \phi_A^*\ V_B\ \phi_A$$
$$H_{BB} = \int d\tau\ \phi_B^*\ V_A\ \phi_B$$
$$H_{AB} = \int d\tau\ \phi_A^*\ V_A\ \phi_B\ e^{+i(\vec{v}\cdot\vec{r} + wt)} \qquad \text{II.9}$$
$$H_{BA} = \int d\tau\ \phi_B\ V_B\ \phi_A\ e^{-i(\vec{v}\cdot\vec{r} + wt)}$$

The integrations in the above matrix elements are over the electronic

coordinates. Eqns. II.7 can be further simplified by the unitary transformation

$$a(t) = A(t)\ \exp(-i\int^{t} dt'\alpha(t')) \qquad \text{II.10}$$

$$b(t) = B(t)\ \exp(-i\int^{t} dt'\beta(t'))$$

where

$$(1 - |S|^2)\ \alpha(t) = H_{AA} - S_{AB}\ H_{BA} \qquad \text{II.11}$$

$$(1 - |S|^2)\ \beta(t) = H_{BB} - S_{BA}\ H_{AB}$$

The resulting set of equations now read

$$i\ \dot{A}(t) = X_{AB}\ B(t) \qquad \text{II.12}$$

$$i\ \dot{B}(t) = X_{BA}\ A(t)$$

where

$$(1 - |S|^2)\ X_{AB} = (H_{AB} - S_{AB}H_{BB})e^{+i\delta} \qquad \text{II.13}$$

$$(1 - |S|^2)\ X_{BA} = (H_{BA} - S_{BA}H_{AA})e^{-i\delta}$$

and

$$\delta(t) = \int^{t} dt'\ [\alpha(t') - \beta(t')] \qquad \text{II.14}$$

Eqns. II.11 are to be solved subject to the initial conditions

$$A(-\infty) = 1 \qquad \text{II.15}$$

$$B(-\infty) = 0$$

The total capture cross section at projectile energy, E, is given by

$$Q(E) = 2\pi \int_0^{\infty} \rho d\rho\ P(\rho) \qquad \text{II.16}$$

where ρ is the impact parameter and P, the probability for single electron transfer. The functional form of P depends upon the number of equivalent electrons available for capture. For a one electron target

$$P(\rho) = p(\rho) = |B(\infty)|^2 \qquad \text{II.17}$$

In multielectron targets there are two equivalent electrons for every set of principal, orbital, and magnetic quantum numbers. p is the probability that an electron is transferred and 1-p is the probability that it is not. Therefore, the probability for only one of the two equivalent electrons to be captured is given by

$$P(\rho) = 2p(1-p) \qquad \text{II.18}$$

where the factor of two arises because there are two possible arrangements for a single electron to be transferred.

B. Comparison of the TSAE Expression With Various Born Theories. Eqns. II.12 are exact within the TSAE approximation. When the capture probability is expected to be small, A(t) can be set equal to one for all t and B(∞) obtained by first order perturbation

$$B(\infty) = -i\int_{-\infty}^{\infty} dt' \, X_{BA}(t') \qquad \text{II.19}$$

where $X_{BA}(t)$ can be written explicitly as II.20

$$X_{BA} = \frac{e^{-i(wt + \delta)}}{1 - |S|^2} \int d\tau \, \phi_B^* \{V_B - \langle \phi_A | V_B | \phi_A \rangle\} \phi_A \, e^{-i \vec{v} \cdot \vec{r}}$$

This expression for the capture amplitude, unlike other Born theories, is independent of any internuclear term or other constant potential added to the original Hamiltonian, Eqn. II.2. The addition of an internuclear

term would have the effect of replacing V_B in the previous expression by $V_B + V(R)$, where $V(R)$ is the internuclear interaction. The inner bracketed terms in Eqn. II.20 would become

$$\begin{aligned} & V_B + V(R) - \langle \phi_A | V_B + V(R) | \phi_A \rangle \\ & = V_B + V(R) - \langle \phi_A | V_B | \phi_A \rangle - \langle \phi_A | \phi_A \rangle V(R) \\ & = V_B - \langle \phi_A | V_B | \phi_A \rangle \end{aligned} \qquad \text{II.21}$$

as before. This stems from the fact that formal recognition has been taken of the nonzero overlap between the initial and final state wavefunctions at small internuclear separation.

When δ is neglected and the denominator of Eqn. II.20 set equal to unity, the TSAE expression is equivalent to the Distorted Wave Born Approximation (DWBA) of Bassel and Gerjuoy. Both are characterized by a potential term, $\langle \phi_A | V_B | \phi_A \rangle$, in addition to the projectile-electron interaction, V_B. However, it must be recognized that the physical interpretation of this term is entirely different. In the DWBA, the term $\langle \phi_A | V_B | \phi_A \rangle$ arises because the distortion of the projectile is included in the formalism. In Eqn. II.20 this term arises from proper treatment of the nonorthogonality between the initial and final state wavefunction. If the nonorthogonality is ignored, Eqn. II.20 can be justified by identifying the interaction for charge transfer as $V_B - \langle \phi_A | V_B | \phi_A \rangle$. This fictitious potential is sometimes called the ' Bates - Born ' potential.[5] Eqn. II.20 can be compared to other Born theories as well. For the K-K capture process of protons on hydrogen $w = \delta = 0$. When the denominator of Eqn. II.20 is approximated by unity and $\langle \phi_A | V_B | \phi_A \rangle$ by

its large R limit, -1/R, Eqns. II.19 and II.20 reduce to the result of Jackson and Schiff. If $<\phi_A|V_B|\phi_A>$ is set equal to zero the OBK expression is recovered.

CHAPTER III: NUMERICAL METHOD

In order to solve Eqns. II.12 within the independent electron approximation an appropriate local potential for the active electron must be obtained, the matrix elements defined by Eqns. II.9 must be evaluated, and the coupled equations must be numerically integrated. Sections III.A, III.B, and III.C describe the techniques employed in this work to do each step.

A. Potential and Wavefunction. A local potential in a multielectron atom can be expressed as

$$V(r) = -\frac{Z}{r} U(r) \qquad \text{III.1}$$

where the screening function, U(r) has the limiting forms

$$\begin{aligned} U(r) &\to 1 \qquad r \to 0 \\ U(r) &\to I/Z \qquad r \to r_0 \end{aligned} \qquad \text{III.2}$$

r_0 is roughly the size of the atom and (I-1) is its charge. In this work the potential and subsequent wavefunctions were obtained by fitting the Herman-Skillman[21] screening function to the form

$$Z\,U(r) = I + (Z - I)\, p(r)\, e^{-\lambda_v r} \qquad \text{III.3}$$

where

$$p(r) = 1 + c_1 r + c_2 r^2 + c_3 r^3$$

This particular form was chosen for its correct asymptotic behavior and its compatibility to the techniques employed for the evaluation of the

matrix elements.

The screening function drops rapidly from 1 at the origin and smooths out to I/Z as r approaches the size of the atom. These characteristics can be adequately reproduced if the parameter λ_v and the coefficients c_i are well chosen. A proper choice for λ_v insures the sharp decline of the screening function in the small r region. The c_i are chosen to fit the intermediate region of r.

In order to determine λ_v the $r \to 0$ limit of Eqn. III.3 is considered. In this limit $p \to 1$ and the resulting expression can be rearranged to give

$$- \lambda_v \, r = \ell n \left| \frac{UZ - I}{Z - I} \right| \qquad \text{III.4}$$

λ_v is obtained by calculating the slope of the right hand side of the above equation. Eqn. III.3 is then linearized and the coefficients of the polynomial determined by the least squares fitting procedure.

The angular dependence of the wavefunctions for a central potential such as Eqn. III.1 are the spherical harmonics, $Y_{\ell m}(\Omega)$. The radial dependence can be conveniently expressed as a sum of Slater type orbitals.

$$P_{n\ell}(r) = \sum_i A_{ni} \, r^{n_i} \, e^{-\alpha_i r} \qquad \text{III.5}$$

where the parameters n_i and α_i were chosen from the work of Clementi and Roetti.[22] The remaining parameters, A_{ni}, were left free to absorb any necessary adjustments. These, as well as the eigenenergies were obtained by diagonalization of the radial Schrödinger Equation

$$H_\ell(r) \; P_{n\ell}(r) = E_{n\ell} \, P_{n\ell}(r) \qquad \text{III.6}$$

The details of this derivation are given in Appendix I.A.

Fitting the Herman-Skillman screening function in this manner gives energy eigenvalues which are very close to both the Hartree-Fock and original Herman-Skillman numbers. As a typical example Table III.1 lists the orbital energies and Table III.2, the wavefunction parameters for the bound states of neutral argon. A comparison is made between the results of this method and the afore-mentioned works. As can be seen, the agreement between the orbital energies is very good for all values of n and ℓ. A plotted comparison of the original and fitted Herman-Skillman wavefunctions reveals few discernible differences. The coding written to perform these calculations is listed in Appendix II.A.

B. Matrix Elements. The two centered matrix elements in Eqns. II.9 can be conveniently evaluated using prolate spheriodal coordinates λ, μ, and ϕ defined by

$$\lambda = \frac{r_A + r_B}{R} \qquad \mu = \frac{r_A - r_B}{R} \qquad \text{III.7}$$

ϕ is the azimuthal angle. These coordinates have ranges

$$1 \leq \lambda \leq \infty \qquad -1 \leq \mu \leq 1 \qquad 0 \leq \phi \leq 2\pi \qquad \text{III.8}$$

and volume element $d\tau = R^3/8(\lambda^2 - \mu^2)\, d\lambda d\mu d\phi$. Other quantities of interest are

$$r_\gamma \cos\theta_\gamma = R/2\ (\lambda\mu \pm 1) \qquad r_\gamma \sin\theta_\gamma = R/2\ \sqrt{(\lambda^2-1)(1-\mu^2)} \qquad \text{III.9}$$

$$\vec{v} \cdot \vec{r} = 1/2 v^2 t\lambda\mu + v_0/2\ \sqrt{(\lambda^2-1)(1-\mu^2)}$$

where the subscript γ represents either A or B. The upper signs are taken for A and the lower signs, for B. These derivations can be found in the monograph by McDowell and Coleman.[5]

The diagonal matrix elements appearing in Eqns. II.9 can be evaluated analytically. The integration over ϕ normalizes to unity. The two remaining integrals are a linear combination of products of incomplete gamma functions. As an example, the diagonal matrix elements for the K-K transfer of a hydrogen electron to a proton are

$$H_{AA} = H_{BB} = -R^2/2 \ (f_1 g_0 + g_1 f_0)$$

where III.10

$$g_n = \int_1^\infty d\lambda \ \lambda^n \ e^{-R\lambda}$$

$$f_n = \int_{-1}^1 d\mu \ \mu^n$$

The off-diagnonal terms are much more complex. In spite of the difficulties which arise because of the factor $e^{\pm i \vec{v} \cdot \vec{r}}$, two of the three integrals can be done analytically. The integration over ϕ is done with the help of Bessel's integral

$$2\pi \ i^{|m|} J_{|m|}(\alpha) = \int_o^{2\pi} d\phi \ e^{i[\alpha \cos \phi + m\phi]} \qquad \text{III.11}$$

and, over μ[23]

$$2i^{\ell - m} j_\ell(T) \ P_\ell^m (\cos X) = \int_{-1}^1 d\mu \ J_m(T \sin \chi \sqrt{1-\mu^2}) P_\ell^m(\mu) \ e^{i\mu T \cos X} \qquad \text{III.12}$$

where j_ℓ is the spherical Bessel function and P_ℓ^m is the associated Legendre polynomial. The integration over λ is done by making a simple change of variable and using Gauss-Laguerre quadrature. For the process mentioned above the off-diagonal matrix elements are

$$S_{AB} = S^*_{BA} = R^3/2 \int_1^\infty d\lambda \, \delta(\lambda) e^{-R\lambda}$$
$$H_{AB} = H_{BA} = -R^2 \int_1^\infty d\lambda \, h(\lambda) e^{-R\lambda} \qquad \text{III.13}$$

where

$$\delta(\lambda) = [\lambda^2 - 1/3] \, j_0(T) + 2/3 \, j_2(T) \, P_2 (\cos \chi)$$

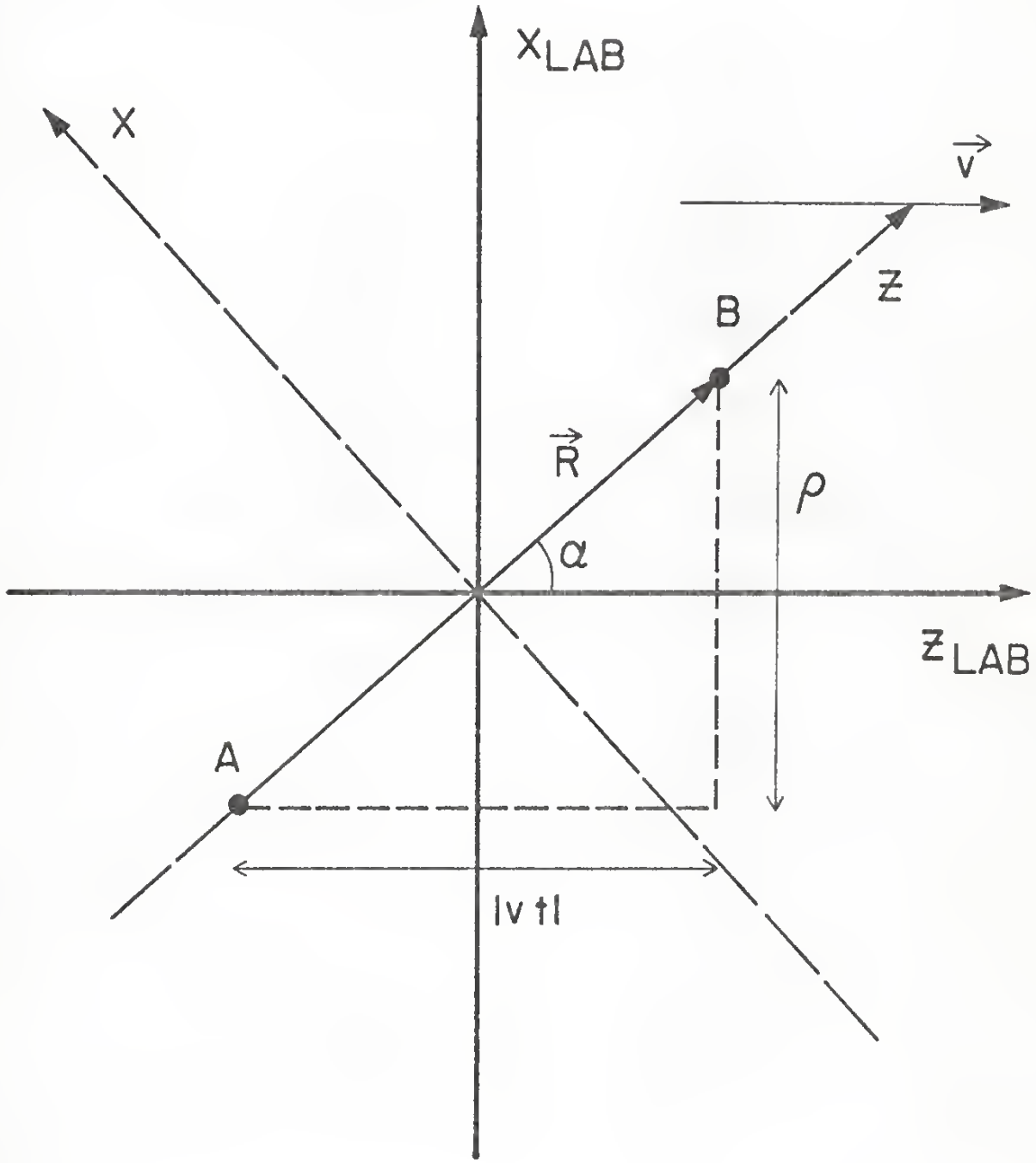
$$h(\lambda) = j_0(T) - i \, j_1(T) \, P_1 (\cos \chi)$$

$$T = v/2 \, [\lambda^2 R^2 - \rho^2]^{1/2} \qquad \cos \chi = \lambda v \, t \, [\lambda^2 R^2 - \rho^2]^{-1/2}$$

(S, by definition, is Hermitian. In general, however, H is neither symmetric nor Hermitian.)

An additional complexity is manifested for processes involving non-spherical wavefunctions. The Schrödinger Equation, II.1, is written in terms of the laboratory coordinates. In the lab system, the Z-axis is defined to be in the direction of the projectile's incident velocity. The matrix elements, however, are derived in a coordinate system which defines the Z-axis to be along the internuclear line, R, as shown in Fig. II.1. Since R changes angle continuously with respect to the projectile's velocity vector during the course of the collision, the two coordinate systems rotate with respect to each other as shown in Fig. III.1. This rotation can be handled in one of two ways; either the coordinate

Figure III.1: The rotation of the laboratory coordinates with respect to the coordinate system used to perform the calculations as shown in Figure II.1. In this illustration $\sin \alpha = \rho/R$ and $\cos \alpha = vt/R$.


X_LAB
X
v
z
B
R
ρ
α
z_LAB
A
|v t|

system or the wavefunctions in Eqns. II.1 can undergo a rotational transformation. In this work the latter method has been chosen.

Firstly, the stationary state part of the wavefunctions in Eqn. II.6 can be expressed as

$$\phi_\gamma(r_\gamma) = G(r_\gamma)\ F(x_\gamma, y_\gamma, z_\gamma) \qquad \text{III.14}$$

where the subscript γ represents either A or B. From Fig. III.1 it is apparent that

$$\begin{pmatrix} x_\gamma \\ y_\gamma \\ z_\gamma \end{pmatrix} = \begin{pmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \qquad \text{III.15}$$

Thus, to account for this rotation x_γ and z_γ should be expressed in terms of x and z. The y components are perpendicular to the plane of scattering; thus, they do not contribute to the capture cross section and need not be considered. For spherically symmetric states, F = 1 in Eqn. III.14. Accordingly, capture processes involving only states with spherical symmetry do not exhibit the effects of this rotation and it can be ignored.

In Appendix I.B matrix elements for transfer processes of the form

$$B^+ + A(n\lambda) \rightarrow B(n'\ell) + A^+ \qquad \text{III.16}$$

have been derived. Program listings for the specific cases of $\ell = 0$ and $\ell = 1$ are given in Appendices II.B and II.C.

C. Numerical Integration of the Coupled Equations. Eqns. II.12 are solved in one of two ways; when the capture probability is expected

to be less than 1/10, A(t) can be set equal to unity and the solution found by repeated iteration. For larger capture probabilities the coupled equations are integrated directly by the Gill-Runga-Kutta method. The coding written for the iterative method is listed in Appendices II.B and II.C. Appendix II.B includes the coding used for the direct integration method.

CHAPTER IV: RESULTS

The Two State Atomic Expansion method described in the previous chapters has been applied to the study of electron transfer cross sections in ion-atom collisions. The discussions in this chapter are divided into three parts; A, gross features of the capture cross section as functions of projectile velocity; B, the sensitivity of the calculations to the type of atomic model used; and C, outer shell capture. Comparison of the theoretical results with experimental data is presented in Sections B and C.

A. Gross Features of Electron Transfer Cross Sections. Massey's criterion states that the electron capture cross section peeks at projectile velocities approximately equal to the characteristic orbital velocity of the active electron. To elucidate this point Eqns. II.12 are rewritten in slightly different form

$$\begin{aligned} i\,\dot{A} &= X_{AB}\,B \\ i\,\dot{B} &= X_{BA}\,A \end{aligned} \qquad \text{IV.1}$$

where

$$\begin{aligned} [1 - |S|^2]X_{AB} &= \int d\tau\, \phi_A^* \,[V_A - \langle \phi_B | V_A | \phi_B \rangle]\phi_B e^{i[\vec{v}\cdot\vec{r} + W(R)/v]} \\ [1 - |S|^2]X_{BA} &= \int d\tau\, \phi_B^* \,[V_B - \langle \phi_A | V_B | \phi_A \rangle]\phi_A e^{-i[\vec{v}\cdot\vec{r} + W(R)/v]} \end{aligned} \qquad \text{IV.2}$$

and

$$W(R) = \int_b^R \frac{x\,dx}{\sqrt{x^2 - b^2}}\,[V_A - V_B] \qquad \text{IV.3}$$

$$U_A = \varepsilon_A + \langle\phi_A|V_B|\phi_A\rangle$$
$$U_B = \varepsilon_B + \langle\phi_B|V_A|\phi_B\rangle \qquad \text{IV.4}$$

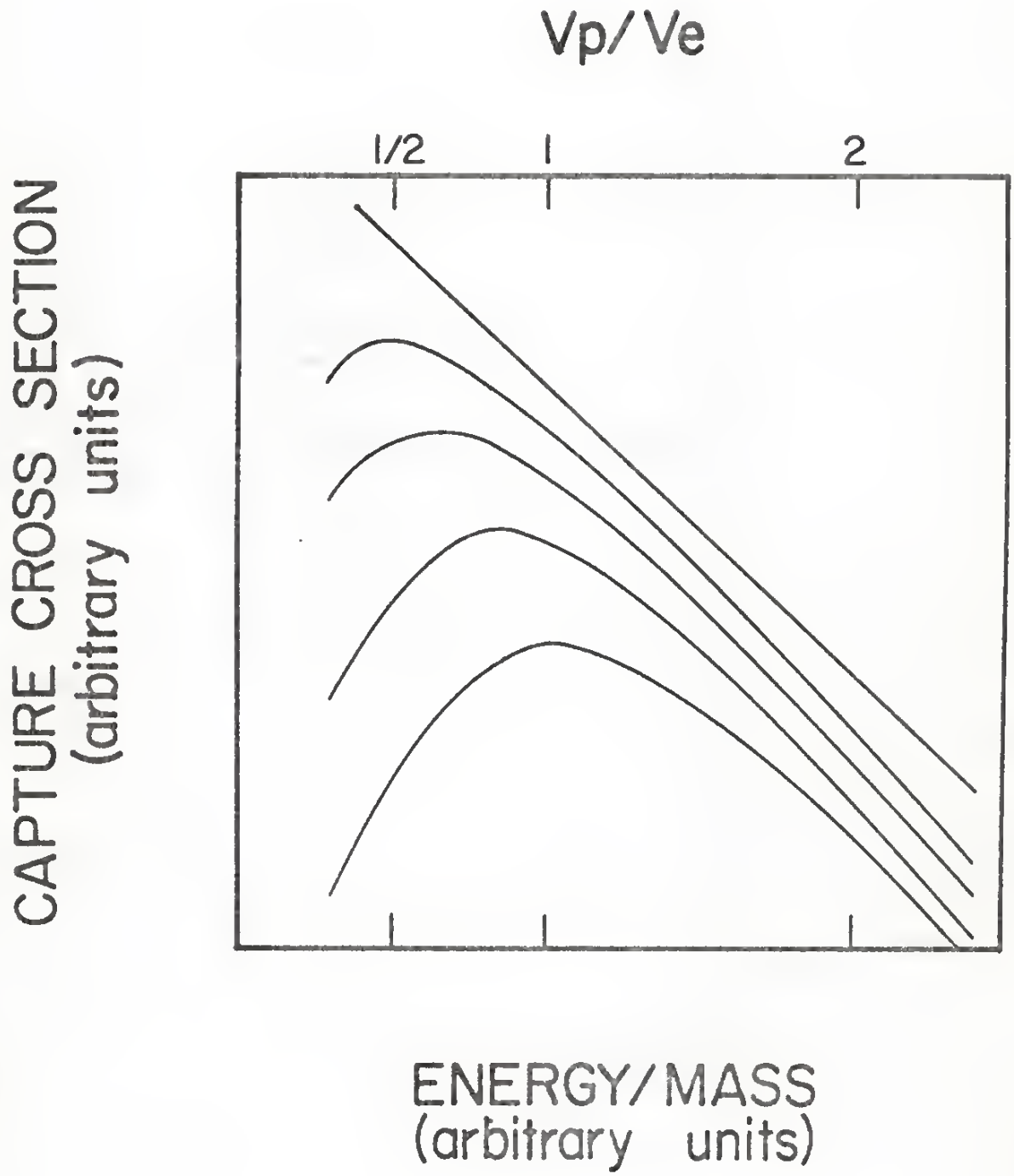
From Eqns. IV.2 it can be seen that the velocity dependence of the coupling matrix elements, X_{AB} and X_{BA}, occurs primarily through the exponential factors, $e^{\pm i(\vec{v} \cdot \vec{r} + W(R)/v)}$. At high collision energies their magnitude is greatly reduced by cancellation in the integrand due to oscillation of the exponential, $e^{\pm i(\vec{v} \cdot \vec{r})}$. This factor is responsible for the rapid decrease of the transfer cross section with increasing projectile velocity. The magnitude of $X_{AB}(X_{BA})$ is reduced at low collision energies through the oscillation of $e^{\pm i(W(R)/v)}$. The cross section peak occurs at v_0, the velocity at which oscillations from the two terms add destructively. Taking r~R, a rough estimate can be made for v_0.

$$\vec{v}_0 \cdot \vec{r} \simeq W(R)/v_0$$
$$v_0 R \sim (\varepsilon_A - \varepsilon_B) R/v_0 \qquad \text{IV.5}$$
$$v_0 \sim \sqrt{|\varepsilon_A - \varepsilon_B|}$$

Fig. IV.4 illustrates the general behavior of the electron transfer cross section as a function of projectile energy per amu, (or equivalently, projectile velocity), for varying types of colliding systems. For symmetrically resonant transfer the cross section does not peak: it exhibits a monotonic decrease with increasing projectile energy. For nonresonant transfer the cross section peak shifts to higher collision velocities with decreasing symmetry of the system. This behavior is consistent with Eqn. IV.5.

The high velocity damping factor, $e^{\pm i(\vec{v} \cdot \vec{r})}$, is not very sensitive

Figure IV.1: Gross features of the electron capture cross section as functions of projectile energy per amu, (or equivalently, projectile velocity). Resonantly symmetric collisions exhibit a monotonic increase with decreasing energy as typified by the uppermost curve. As the colliding system becomes less symmetric, the peak of the capture cross section shifts toward the energy corresponding to $v_p/v_e = 1$ where $v_p(v_e)$ is the velocity of the projectile (electron).


Vp/Ve
1/2
1
2
CAPTURE CROSS SECTION
(arbitrary units)
ENERGY/MASS
(arbitrary units)

to the type of atomic model used in the description of the multielectron atom. As can be seen from its definition, however, the low velocity damping factor, $e^{\pm i(W(R)/v)}$, is extremely model dependent. U_A and U_B in Eqn. IV.4 can, in fact, be identified as the 'potential curves' defined within this theory. They represent the distortion of the electron cloud in the initial (final) state by the projectile (target) nucleus. In anticipation of the following discussion, therefore, the calculations can be expected to show sensitivity to the atomic model at low collision velocities.

B. Comparison of K-K Results Using Different Atomic Models. The TSAE method has been applied to the description of K-K capture processes for bare projectiles on multielectron targets. The potential of the bare projectile is given by $V_B = -Z_B/r_B$. Within the independent electron approximation, the target potential can be expressed in several ways. In the early work of Lin et. al. [20] a screened hydrogenic potential, $V_A = -Z_A^*/r_A$, (where $Z_A^* = Z_A - 5/16$), was used. This potential includes the mutual screening of the K shell electrons and the corresponding wavefunction is known to represent the actual 1s orbital very well. However, the screening of the outer electrons is not acknowledged by this treatment; thus, the corresponding energy, $\varepsilon_A = -Z_A^{*2}/2$, is quite different from the experimental value. It was recognized in the earlier work that this discrepancy would not arise with the use of a proper multielectron theory. Therefore, experimental K shell binding energies were substituted for ε_A. It should be noted, however, that by choosing ε_A and V_A inconsistently the unitarity condition of the calculation was destroyed.

To ascertain the importance of these approximations on the capture

probability a comparison between this work and the earlier one is made. The Herman-Skillman potential differs from the hydrogenic potential primarily by its inclusion of outer as well as inner shell screening. In Fig. IV.2 the hydrogenic, (dashed lines), and Herman-Skillman, (solid lines), potentials for neutral argon are compared. The differences are most significant in the region outside the K shell radius. The corresponding 1s wavefunction for each potential is compared in Fig. IV.3. The reason for the apparent agreement between the wavefunctions can be attributed to the approximately hydrogenic behavior of the Herman-Skillman potential within the region of the 1s amplitude.

The statement was made in Section IV.A that discrepancies between calculations using different atomic models should occur at low collision velocities due to the increased importance of the factor, $e^{\pm i(W(R)/v)}$. By considering the above comments and the V_A dependence of W(R), it can be seen that these discrepancies are caused by the differing behavior of the hydrogenic and Herman-Skillman potentials. Fig. IV.4 compares the electron transfer cross sections calculated in the two models for two widely varying systems, P + Ar and F^{9+} + Ar. As anticipated in the comments at the end of Section IV.A, the agreement between the models is good in the high energy region. At low collision energies, however, significant discrepancies exist, particularly for F^{9+} + Ar. To further illustrate this point the calculated values of the transfer cross sections in the two models are listed in Table IV.1. Though the agreement does improve with increasing projectile velocity, the disagreement in the low energy region is substantial.

The discrepancy between the two models is generally small for very

Figure IV.2: Comparison of the screened hydrogenic and Herman-Skillman potentials for neutral argon. It should be noted that the former is much stronger than the latter, particularly in the large r region.

Figure IV.3: Comparison of the screened hydrogenic and Herman-Skillman wavefunctions of argon. (The screened charge is taken to be 17.6875). Unlike their corresponding potentials, the agreement between the wavefunctions is very good. The K shell radius, r_K, is indicated.

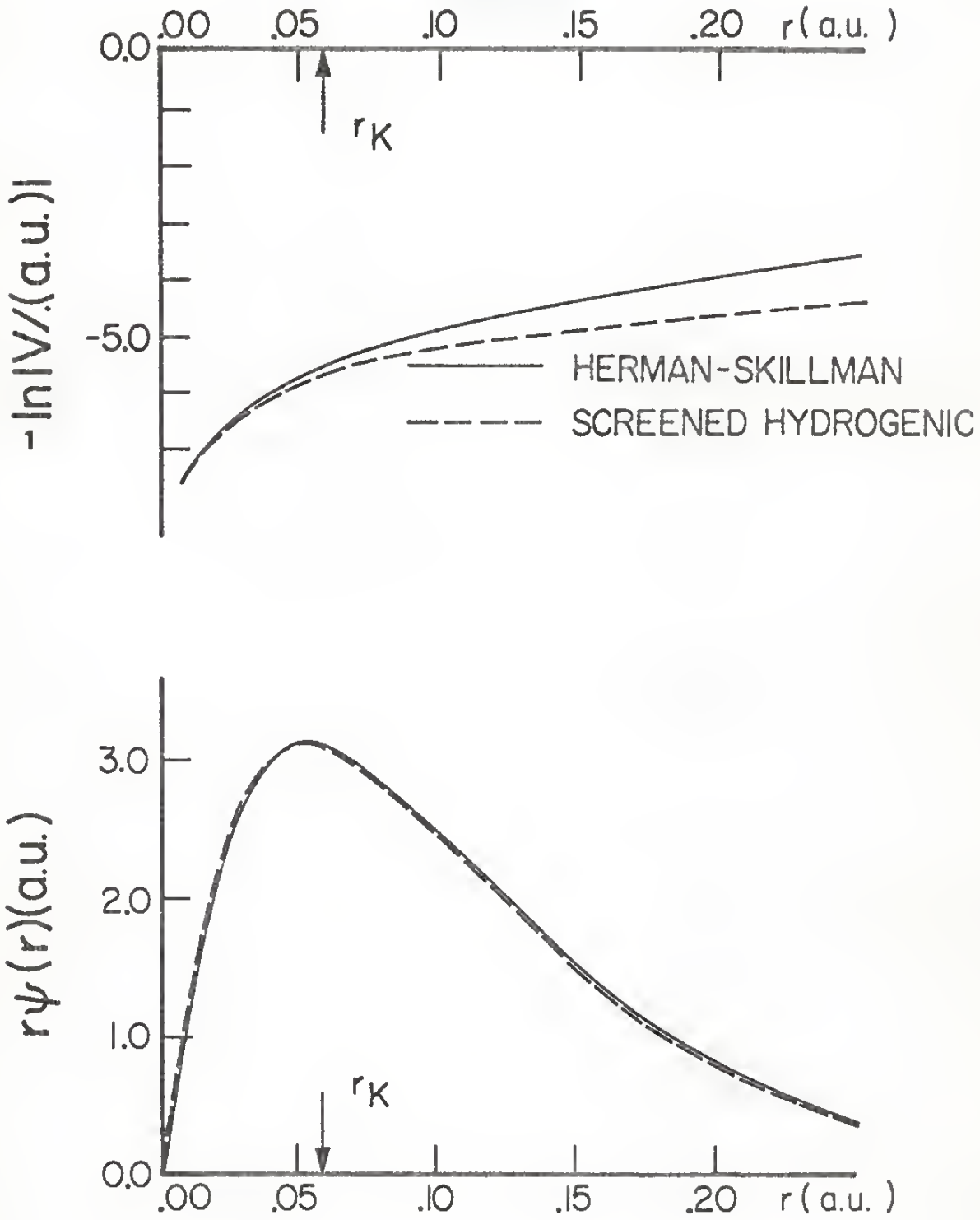
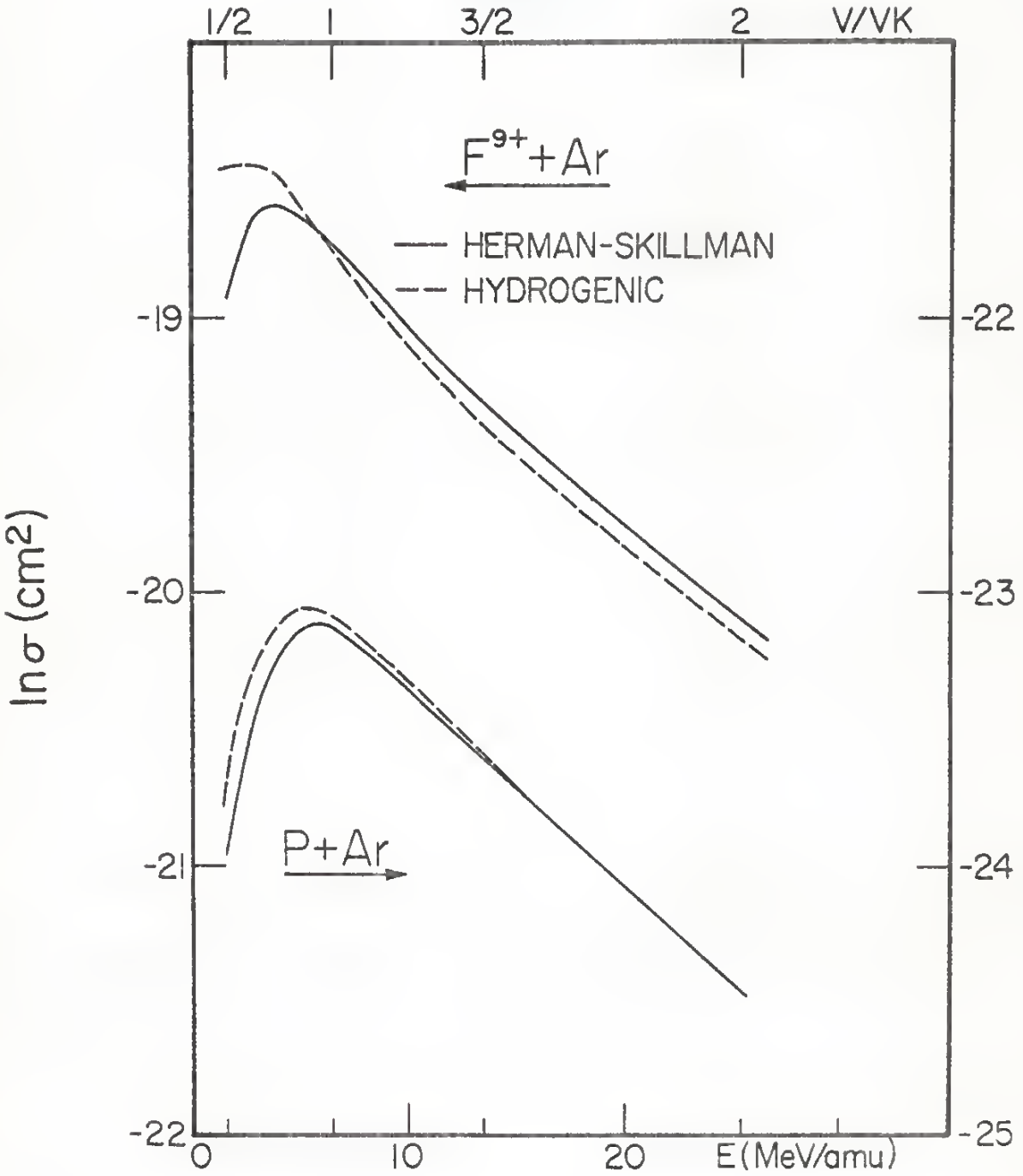

0.0
.00
.05
.10
.15
.20
r(a.u.)
r_K
-5.0
$-\ln|V/(a.u.)|$
HERMAN-SKILLMAN
SCREENED HYDROGENIC
3.0
2.0
1.0
0.0
$r\psi(r)(a.u.)$
r_K
.00
.05
.10
.15
.20
r(a.u.)

Figure IV.4: K-K cross sections for electron transfer as functions of collision energy for F^{9+} + Ar and P + Ar calculated using the Herman-Skillman and hydrogenic models. (v is the collision velocity and v_K is the characteristic orbital velocity of the K shell electron.)

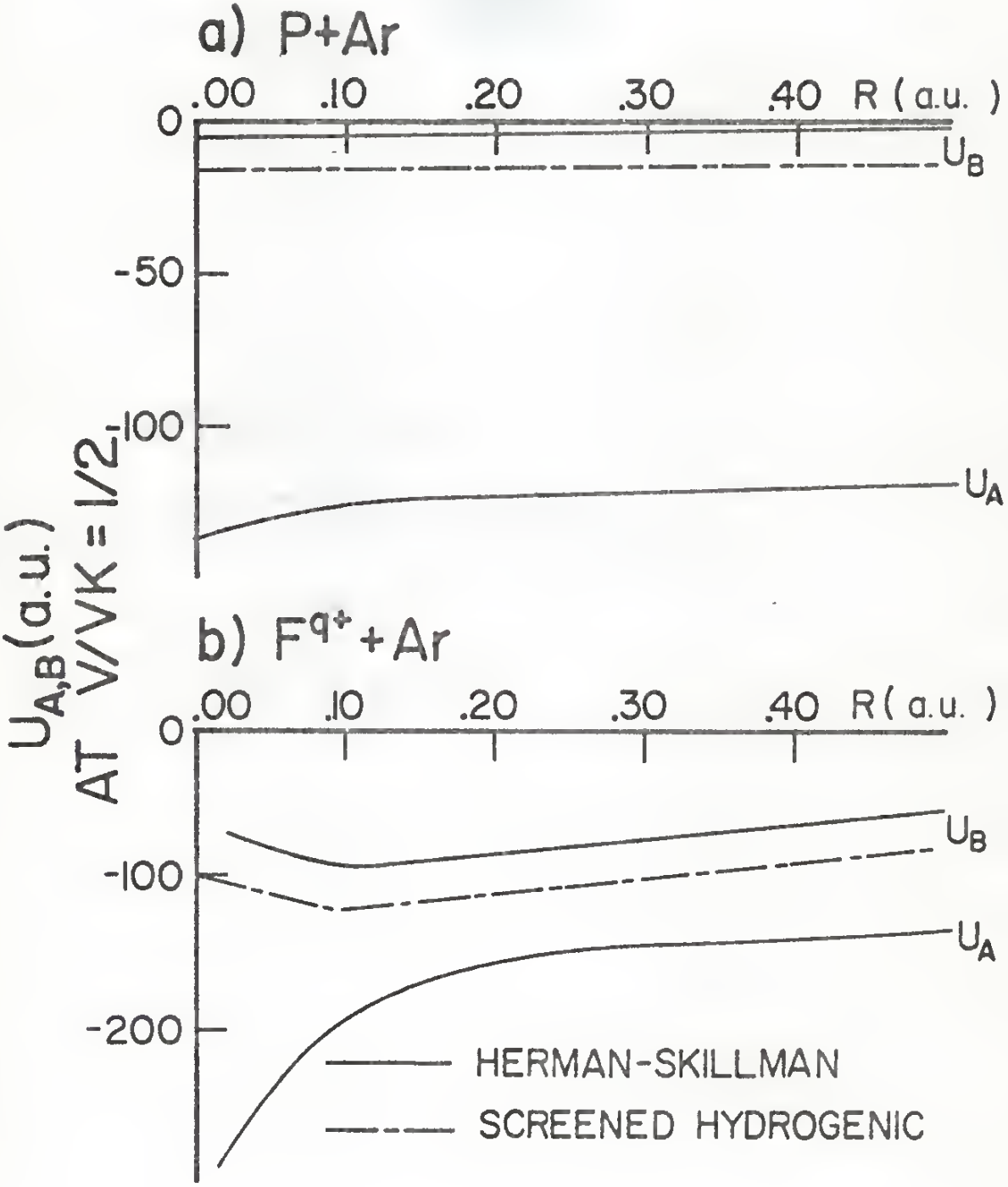

1/2
1
3/2
2
V/VK
F9+ + Ar
HERMAN-SKILLMAN
HYDROGENIC
P + Ar
ln σ (cm2)
-19
-20
-21
-22
-22
-23
-24
-25
0
10
20
E (MeV/amu)

asymmetric systems such as P + Ar. As the symmetry of the system increases, however, the differences are significant. The reason for this can be elucidated by considering Fig. IV.5. The potential curves, U_A and U_B, are plotted for P + Ar and F^{9+} + Ar using both the hydrogenic and Herman-Skillman models. The discrepancies between the models is much more significant for F^{9+} + Ar than P + Ar. This is due to the increased importance of terms containing V_A for the former system than the latter.

Table. IV.1 lists theoretical cross sections for several other systems along with experimental data. Though the Herman-Skillman is a more realistic potential, these results do not show improvement with experimental measurements. The Herman-Skillman calculations were done under the assumption that the active electron experiences the potential of a neutral target. It is known that the target becomes multiply ionized during the course of the collision. Table IV.2 compares several Herman-Skillman calculations for F^{9+} + Ar^{n+} for varying values of n. As evidenced by these results, the capture cross section does not exhibit a strong dependence on the final charge state of the target. This point is further discussed in Chapter V.

Though it is important to use a more realistic model for more symmetric systems, the computer time involved in such a calculation poses practical difficulties. It therefore remained desirable to find a simpler way to give an adequate description of the multielectron system. As evidenced by Figs. IV.2 and IV.3, the most significant difference between the two models occurs in the behavior of the multielectron potential, V_A. The idea was therefore proposed of using a 'hybrid' model— a screened hydrogenic wavefunction with a Herman-Skillman potential. This proved to

Figure IV.5: The potential curves for F^{9+} + Ar and P + Ar in the Herman-Skillman and hydrogenic models at an incident energy corresponding to $v/v_K = 1/2$. U_A is indistinguishable for the two models to the scale shown and are drawn as one.


a) P+Ar
.00
.10
.20
.30
.40
R (a.u.)
0
-50
-100
U_B
U_A
$U_{A,B}$(a.u.)
AT V/VK = 1/2
b) F^{q+}+Ar
.00
.10
.20
.30
.40
R(a.u.)
0
-100
-200
U_B
U_A
HERMAN-SKILLMAN
SCREENED HYDROGENIC

be quite successful. For the particular case of F^{9+} + Ar at a collision velocity of $v = v_K/2$, the agreement with the full Herman-Skillman calculation was better than 2% and the computer time reduced by more than 80%. In view of these considerations it is clear that the most practical way of insuring an adequate description of K-K capture processes involving multielectron atoms is to use a screened hydrogenic wavefunction, (with screened charge $Z^* = Z-5/16$), and a Herman-Skillman description of the potential.

C. Outer Shell Capture. The Herman-Skillman model provides a satisfactory description of the potential in the large r region of a multielectron atom, thus enabling a study of electron capture from outer shells. Section IV.C discusses charge transfer from the outermost shells of neon, argon, and krypton to the K shell of hydrogen. The capture cross sections for these systems can be expected to be important because the energy defects between the initial and final states are small.

There are many studies of this type for low collision energies which are based on the MO theory. Except for the simplistic OBK method,[5] however, there are no theoretical investigations for more energetic collisions. The assumptions on which the OBK theory is based are invalid for systems such as these because the capture probabilities are not small and the potential experienced by an outer shell electron is not Coulombic. This is an effort to describe outer shell electron transfer in which more realistic assumptions are made.

It was emphasized in Section IV.A that charge transfer cross sections are sensitive to both the energy defect between the initial and final states as well as the velocity of the incomimg projectile. In low

energy collisions between protons and argon atoms, for example, the capture process is dominated by electron transfer from the Ar(3p) to the H(1s) state. Capture to excited states of the projectile is much less important, as is capture from more tightly bound states of the target, because the energy defects are larger. In faster collisions, however, outer shell electrons have very little time to react to the field of the impinging ion; thus, the more energetic L and K shell electrons are transferred. Because the energy defects are larger for these processes, the magnitude of the total capture cross sections are reduced.

Table IV.3 lists the electron transfer cross sections from the K, L, and M shells of argon to the K shell of hydrogen. Comparison between these results and the available experimental data is shown in Fig. IV.6. At low collision velocities capture from the Ar(3p) state, (dashed line), dominates the total electron transfer cross section, (dot-dashed line), entirely masking the Ar(3s), (dashed line), contribution. This is consistent with the foregoing discussion concerning energy defects. As the collision energy increases, L shell capture begins to take over. Again, most of the capture occurs from the p state though the s state contribution is not negligible. At still higher proton velocities transfer from the K shell begins to be important. The fine details of this study warrant improvement; however, the overall agreement with experimental data is satisfactory.

In Figs. IV.7a, IV.7b, and IV.7c the impact parameter dependence of the weighted capture probability, $2P(1-P)b$, is illustrated for the transfer of K, L, and M shell electrons. These figures indicate some general trends. The maxima of the weighted probability for a given state moves in to smaller impact parameters with increasing projectile velocity. Accom-

Figure IV.6: The energy dependence of the capture cross section for electron transfer from the K, L, and M shells of argon atoms to the K shell of hydrogen. Calculated total cross sections from each shell are indicated, (— - — -), along with individual subshell contributions, (— — — —). The experimental data, (————), from the K and L shells are from Macdonald et. al.[28] and Rodbro et. al.[29].

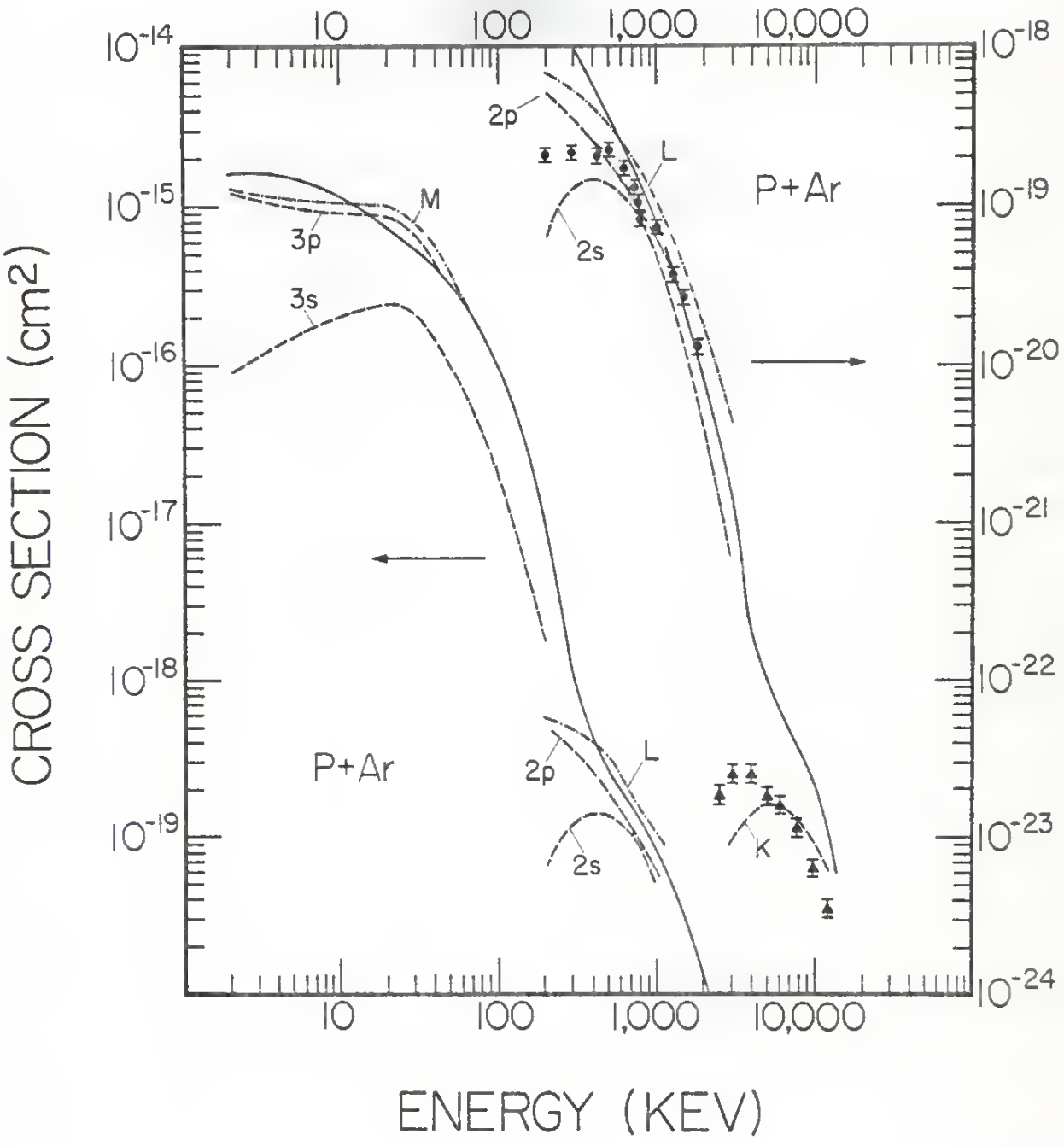
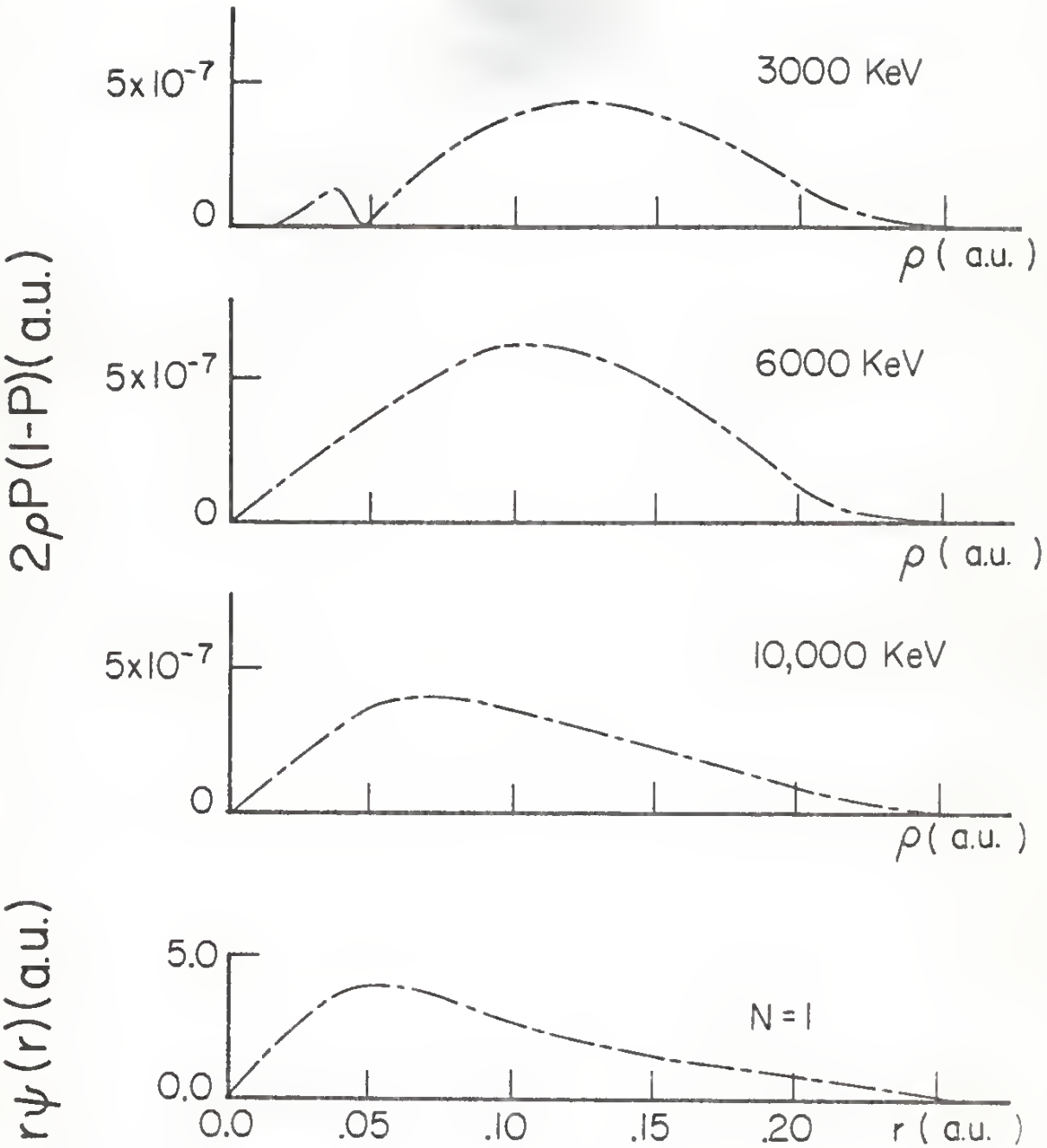
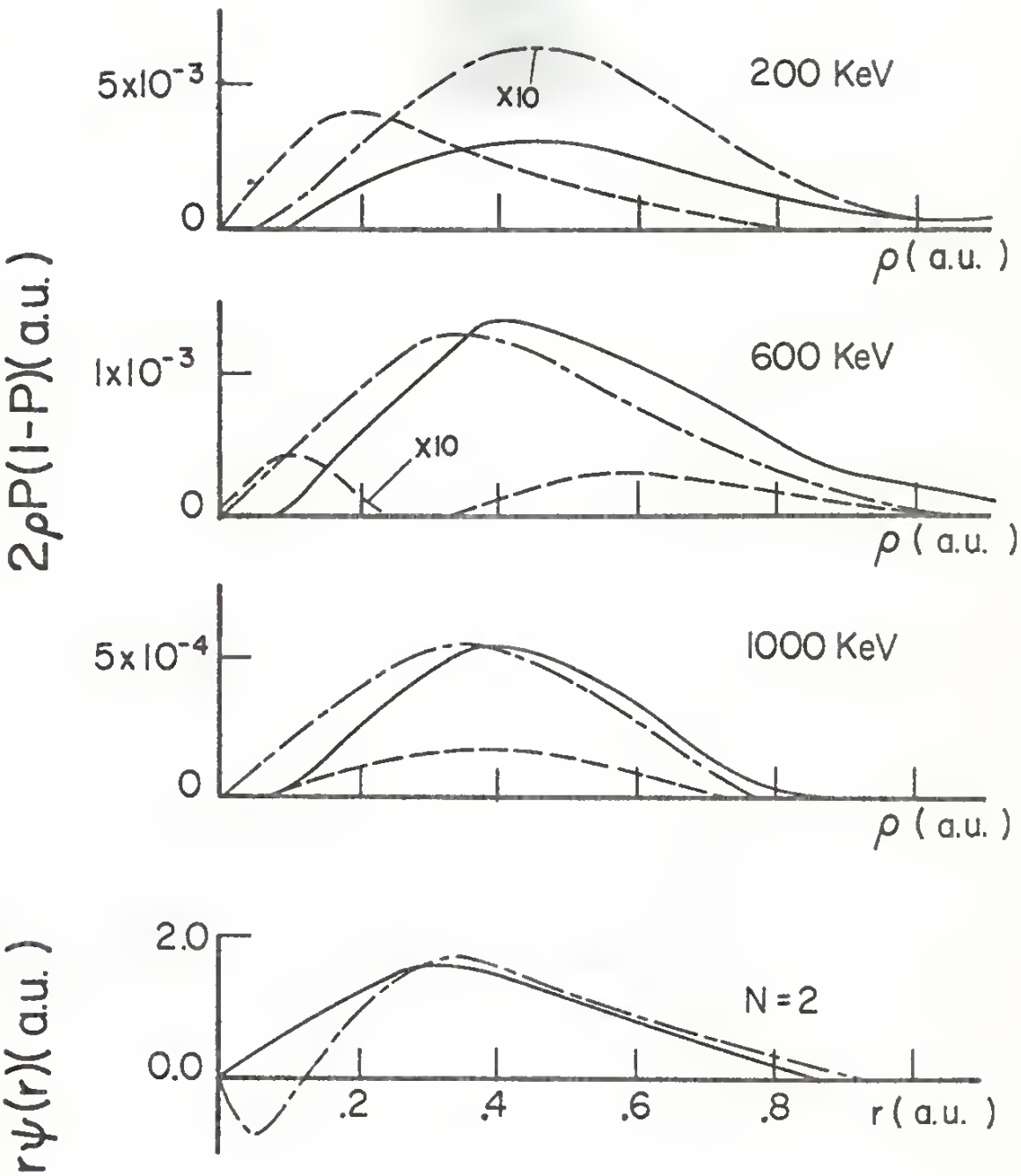
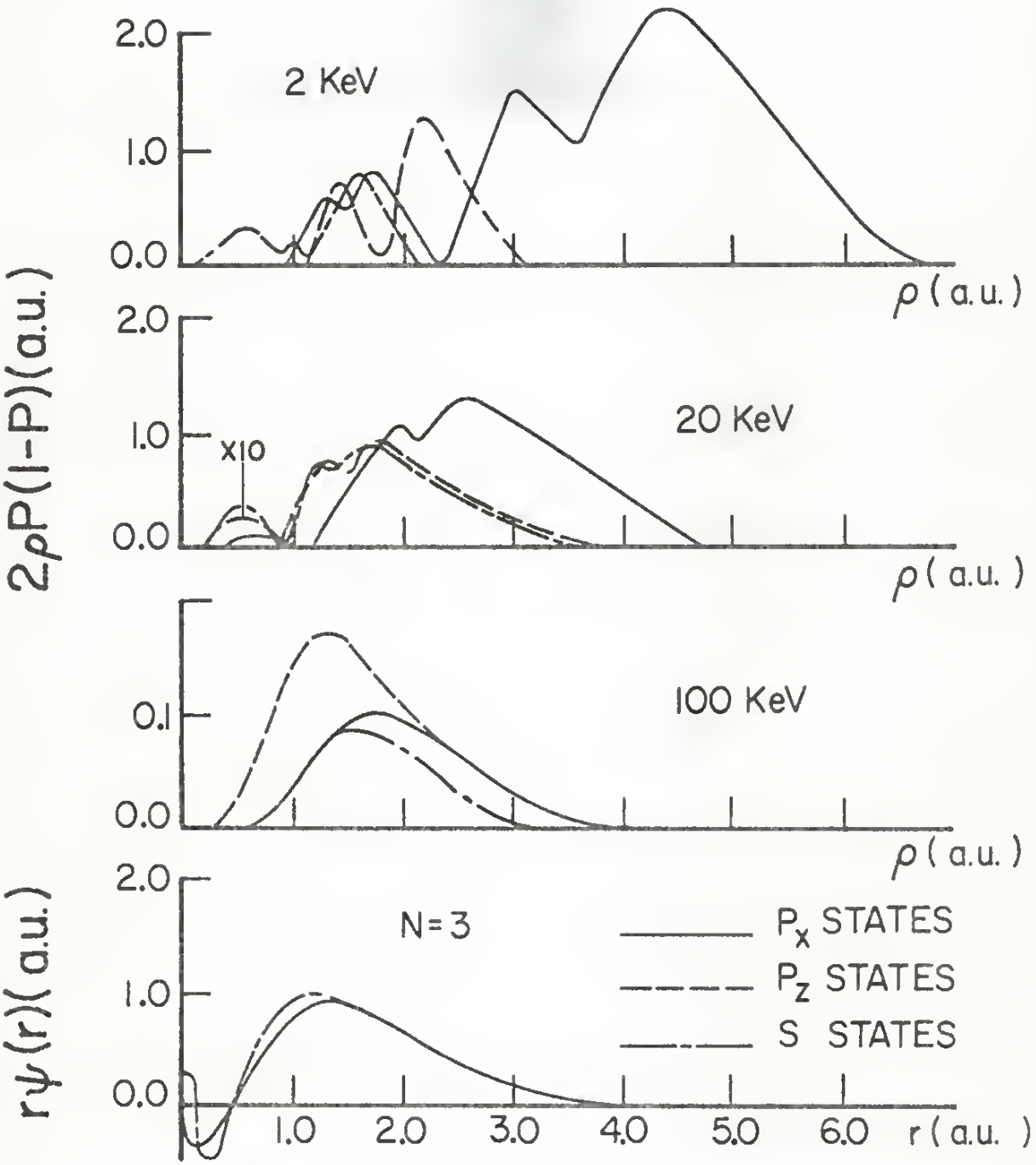

CROSS SECTION (cm²)
ENERGY (KEV)
10
100
1,000
10,000
10⁻¹⁴
10⁻¹⁵
10⁻¹⁶
10⁻¹⁷
10⁻¹⁸
10⁻¹⁹
10⁻¹⁸
10⁻¹⁹
10⁻²⁰
10⁻²¹
10⁻²²
10⁻²³
10⁻²⁴
P+Ar
P+Ar
2p
2s
L
M
3p
3s
2p
2s
L
K

Figure IV.7: The weighted capture probability, $2P(1-P)\rho$, for electron transfer from the N = 3, 2, and 1 states of argon to the N = 1 state of hydrogen. The radial distribution of the target wavefunction for each corresponding orbital is illustrated in the lower figures.


$2\rho P(1-P)$(a.u.)
5×10^{-7}
0
3000 KeV
ρ (a.u.)
5×10^{-7}
0
6000 KeV
ρ (a.u.)
5×10^{-7}
0
10,000 KeV
ρ (a.u.)
$r\psi(r)$(a.u.)
5.0
0.0
N = 1
0.0
.05
.10
.15
.20
r (a.u.)


5x10⁻³
0
x10
200 KeV
ρ (a.u.)
1x10⁻³
0
x10
600 KeV
ρ (a.u.)
5x10⁻⁴
0
1000 KeV
ρ (a.u.)
2ρP(1-P)(a.u.)
2.0
0.0
N = 2
.2
.4
.6
.8
r (a.u.)
rψ(r)(a.u.)

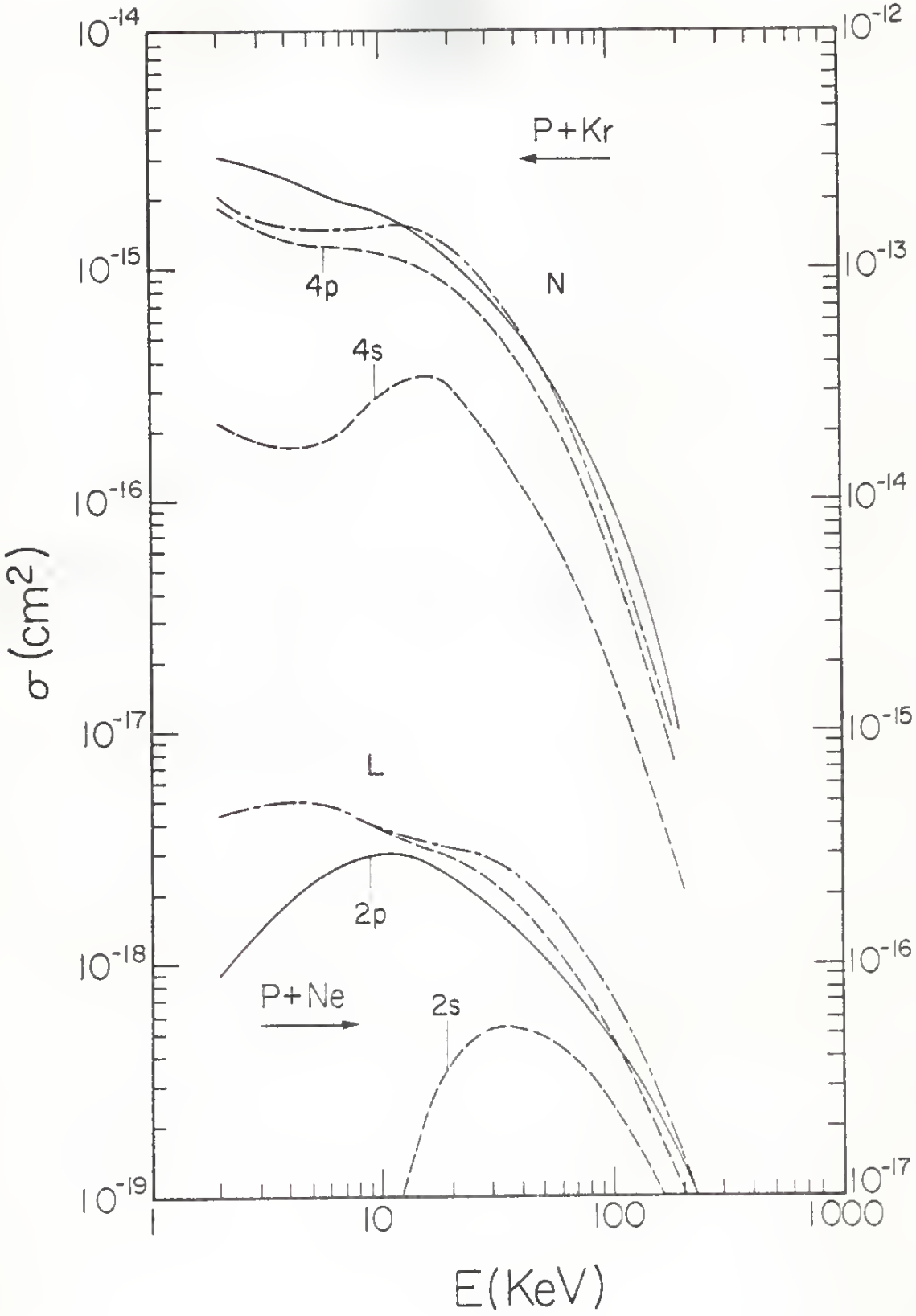

2 KeV
20 KeV
100 KeV
x10
N=3
P_x STATES
P_z STATES
S STATES
$2\rho P(1-P)$(a.u.)
$r\psi(r)$(a.u.)
ρ (a.u.)
r (a.u.)
2.0
1.0
0.0
0.1
1.0
2.0
3.0
4.0
5.0
6.0

panied by this is the disappearance of oscillation in the probability function.

Fig. IV.8 presents results from a similar study of electron transfer from the outermost shells of neon and krypton to the K shell of hydrogen. The subshell contributions are listed in Table IV.4 for selected collision energies. The agreement with experimental data is good for P + Kr though the shape of the curve is somewhat questionable. This is not the case for P + Ne. The disagreement between theoretical and experimental results is substantial, particularly for proton energies less than 10 kev. Capture from the L shell of neon is a much more asymmetric process than the other two systems discussed in this section. Usage of the single particle approximation is known to be a limitation in this formulation as is the retainment of only two states in the multistate expansion, Eqn. II.3. Chapter V discusses these points in further detail.

A preliminary study of the Z_B dependence of electron transfer from the L shell of argon to the K shell of various ions has been made. The OBK theory predicts these cross sections to be scaled by Z_B^5 when Z_B/Z_A is small and the projectile velocity is large. Table IV.5 presents the results of this study for three collision velocities. These calculations are not in accord with the Z_B^5 scaling estimate nor do they exhibit any such simplistic Z_B dependence.

Figure IV.8: The energy dependence of the charge transfer cross section from the outermost shells of neon and krypton to the K shell of hydrogen. The total transfer cross section from each shell is indicated, (—— - —— -), along with the individual subshell contributions, (— — — —). The experimental data, (————), are from the compilation by Tawara and Russek[30].


10^{-14}
10^{-15}
10^{-16}
10^{-17}
10^{-18}
10^{-19}
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{-16}
10^{-17}
P+Kr
4p
N
4s
L
2p
P+Ne
2s
σ (cm^2)
10
100
1000
E(KeV)

CHAPTER V: DISCUSSION AND SUMMARY

In this work the Two State Atomic Expansion method has been applied to the study of electron capture within the independent particle approximation. A Herman-Skillman model has been used to describe charge transfer processes involving multielectron ions. Wide ranges of collision energies have been covered in order to observe the shell dependence of the total capture cross section. Two different models, the Herman-Skillman and the screened hydrogenic, were compared for the description of the K-K transfer process. It was observed that, except for very asymmetric systems, the capture probabilities are sensitive to the type of atomic model used for the multielectron atoms. Prior to this there are few serious attempts to describe charge transfer from outer shells and it is hoped that a more in depth study will emerge from this preliminary work.

A comment on the experimental data is in order. Particularly for the K-K transfer process, comparison of theoretical and experimental results is difficult. The experimental values in Table IV.1, for instance, were deduced from either x-ray or Auger cross sections. A single fluorescence yield or Auger rate was assumed in determining the total vacancy production. This assumption is open to question. For example, it has been shown by Tawara et. al.[31] that in F^{q+} + Si collisions the fluorescence yield for Si K x-rays changes from $1.7w_0$ to $1.9w_0$ with the removal of 5 and 6 target electrons, respectively. (w_0 is the fluorescence yield for neutral Silicon.) In F^{9+} + Ar collisions it was estimated that 6-8 target electrons are removed with the production of one K shell vacancy.

This discussion implies that, theoretically, ionization and inner shell capture should be considered simultaneously. Such a formulation, however, is impossible at present.

The preliminary study of outer shell capture done in this work is the first of its kind. Both the atomic and scattering models are as simple as possible without being unrealistic. However, improvements in the description of capture from both outer and inner shells can be made.

A. Atomic Model. The primary defects of the atomic model are due to the single particle approximation. Because outer shell capture occurs at low projectile velocities, usage of antisymmetrized wavefunctions would probably improve the agreement with experimental data. The effect of the neighboring electrons on the active electron may be comparable to the perturbation caused by the impinging ion as well, particularly for low collision velocities. (For example, this may be important in the capture of neon L shell electrons by protons.) Thus, the formulation of a many electron theory of charge transfer is warranted. The difficulties involved in such a calculation, however, are formidable.

B. Scattering Model. The major approximation made in the scattering model is the retainment of only two terms in the multistate expansion, Eqn. II.3. For instance, the initial and final state wavefunctions in very asymmetric collisions are severely different in size; therefore, it may be necessary to include intermediate states in the multistate expansion for the purpose of 'filling the gaps'. This type of approach should improve the description of a process such as the K-K transfer of argon electrons to protons.

As is evidenced by Figs. IV.7, the impact parameters important to

the transfer process become small with increasing projectile velocity. The Two State Approximation is not adequate to describe this region well. At high collision velocities, therefore, the inclusion of states which have amplitude in the small impact parameter region should improve the results. For the description of outer shell electron transfer this would mean the inclusion of lower orbital states and for transfer of inner shell electrons, the inclusion of pseudostates.

REFERENCES

1. Richard, et. al. (to be published)
2. Gardner, et. al. (to be published)
3. U. Fano and W. Lichten, Phys. Rev. Lett. 14, 627 (1965).
4. J. S. Briggs and J. H. Macek, J. Phys. B 5, 579 (1972).
   See also the review by J. S. Briggs in Rep. Prog. Phys. 39, 217 (1976).
5. M.R.C. McDowell and J. P. Coleman, Introduction to the Theory of Ion-Atom Collisions (North-Holland, Amsterdam, 1970).
6. See the review by R. A. Mapleton, Theory of Charge Exchange (Wiley-Interscience, New York, 1972).
7. B. H. Bransden, Rep. Prog. Phys. 35, 949 (1972).
8. J. R. Oppenheimer, Phys. Rev. 31, 349 (1928).
9. H. C. Brinkman and H. A. Kramers, Proc. Acad. Sci. (Amsterdam) 33, 973 (1930).
10. D.R. Bates and A. Dalgarno, Proc. Phys. Soc. Lond. A 65, 919 (1952).
11. J. D. Jackson and H. Schiff, Phys. Rev. 89, 359 (1953).
12. Y. B. Band, Phys. Rev. Lett. 37, 634 (1976).
13. A. Halpern and J. Law, Phys. Rev. A 12, 1776 (1975).
14. D. R. Bates, Proc. R. Soc. A 274, 294 (1958).
15. R. H. Bassel and E. Gerjuoy, Phys. Rev. 117, 749, (1960).
16. R. McCarroll, Proc. R. Soc. A 264, 547 (1961).
17. R. A. Mapleton, Phys. Rev. 126, 1477 (1962).
18. J. P. Coleman and S. A. Trelease, Atom. Molec. Phys. 1, 172 (1968).
19. A. B. Wittkower, G. Ryding and H. B. Gilbody, Proc. Phys. Soc. 89, 541 (1966).
20. C. D. Lin, S. C. Soong, and L. N. Tunnell, Phys. Rev. A 17, 1646 (1978).
21. F. Herman and S. Skillman, Atomic Structure Calculations (Prentice-Hall, New Jersey, 1963).
22. E. Clementi and C. Roetti, Atomic Data and Nuclear Data Tables 14, 177 (1974).
23. G. N. Watson, Treatise on the Theory of Bessel Functions (Cambridge Univ. Press, Cambridge, 1944).
24. C. W. Woods, R. L. Kauffman, K. A. Jamison, N. Stolterfoht and P. Richard, Phys. Rev. A 13, 1358-69 (1976).

25. T. G. Winter and N. F. Lane, Phys. Rev. A, (January 1978).
26. F. Hopkins, N. Cue and V. Dutkiewica, Phys. Rev. A 12, 1710-2 (1976a).
27. F. Hopkins, R. Breen, A. R. Wittlemore, N. Cue, V. Dutkiewica and R. Chaturvedi, Phys. Rev. A 13 74-85 (1976b).
28. J. R. Macdonald, C. L. Cocke, and W. W. Eidson, Phys. Rev. Lett. 32, 648 (1974).
29. M. Rodbro, E. Pederson, and J. R. Macdonald: Proc. X ICPEAC, p. 48, Paris (1977).
30. Compilation from Tawara and Russek, Rev. Mod. Phys. 45, 178 (1973).
31. H. Tawara, P. Richard, J. R. Macdonald, R. Dillingham, and P. Pepmiller; to be published in Phys. Rev. A (1979).

TABLE III. 1

COMPARISON OF ORBITAL ENERGIES
FOR THE BOUND STATES OF ARGON

ORBITAL	HARTREE-FOCK*	HERMAN-SKILLMAN**	THIS WORK
1s	- 118.61	- 116.28	- 117.77
2s	- 12.32	- 11.44	- 11.11
3s	- 1.28	- 1.05	- 1.07
2p	- 9.57	- 9.10	- 9.01
3p	- .59	- .53	- .56

* ref. no. 22

** ref. no. 21

TABLE III. 2

EXPONENTS AND EXPANSION
COEFFICIENTS FOR NEUTRAL ARGON*

EXPONENTS	1s	2s	3s
1s 18.01640	.97349(.97824)	.27635(.28011)	.08634(.09449)
3s 22.04650	.01684(.01148)	.00289(.00069)	.00186(.00042)
3s 16.08250	.02422(.02659)	-.03421(-.03907)	-.01540(-.01478)
3s 11.63570	-.00114(-.00685)	-.33229(-.35409)	-.10236(-.12777)
3s 7.70365	.00123(.00259)	-.65828(-.62148)	-.27614(-.28171)
3s 4.87338	-.00039(-.00122)	-.06834(-.08899)	-.11879(-.10256)
3s 3.32987	.00010(.00054)	.00623(.14479)	.68436(.74965)
3s 2.02791	-.00003(-.00011)	-.00174(-.00198)	.52050(.45017)

EXPONENTS	2p	3p
2p 9.05477	.64116(.68865)	-.17850(-.20725)
4p 15.54410	.00865(.00241)	-.00812(-.00801)
4p 12.39770	.04186(.03307)	.00520(.01045)
4p 8.56120	.31735(.28159)	-.10986(-.10940)
4p 5.94658	.09642(.09553)	.10944(.15359)
4p 3.42459	.00003(-.00285)	.56149(.59541)
4p 1.96709	.00053(.00101)	.46314(.39726)
4p 1.06717	-.00013(-.00029)	.02951(.02429)

* The entries to the left of the parenthesis are the expansion coefficients, A_i in Eqn. III.5, from reference no. 21. Those in parenthesis are the expansion coefficients obtained in this work by the procedure described in Section III.A.

TABLE IV. 1

K-K CAPTURE CROSS SECTIONS FOR+
BARE PROJECTILES ON NOBLE GASES

PROJECTILE + TARGET	E(MEV)	V/V_K	σ_H*	σ_{HS}**	σ_{EXP}
N^{7+} + Ne	14	.79	368	284	355[a]
	19	.92	343	223	350
N^{7+} + Ar	14.7	.42	5.1	1.4	3.2[b]
	26.3	.56	12.6	3.4	13.2
F^{9+} + Ar	20	.42	23.1	5.6	9.7[c]
	30	.52	34.7	10.0	29.0
	36	.57	38.6	16.0	30.4
	46	.64	40.4	22.0	47.7
	56	.71	38.7	26.0	53.6
	66	.77	35.0	27.0	48.8
	80	.85	30.0	27.0	--
	86	.88	28.0	--	--
	114	1.01	20.0	21.0	--
F^{9+} + Kr	46	.30	.074	.029	.010
	56	.33	.079	.037	.037
	66	.36	.075	.040	.064
	76	.39	.068	.038	.061
Cl^{17+} + Kr	100	.33	1.55	0.52	0.60[d]
	120	.36	1.40	0.59	1.15
	140	.39	1.62	0.60	1.90
	160	.42	2.50	0.60	3.80

\+ Units are 10^{-20} cm^2 per target electron.
* Calculations done using a hydrogenic model.
** Calculations done using a Herman-Skillman model.
a) Woods et. al. (1976)
b) Woods et. al. (1973)
c) Hopkins et. al. (1976a)
d) Hopkins et. al. (1976b)

TABLE IV. 2

TARGET IONICITY DEPENDENCE OF
K-K CAPTURE CROSS SECTIONS FOR F^{9+} + Ar^{n+}*

V/V_k	Target Ionicity 0^+	7^+	10^+	13^+
.42	5.2	5.8	5.6	5.3
.57	14.5	16.1	15.5	13.6
.77	24.9	25.8	25.5	24.3
1.01	18.8	19.0	18.9	18.7

*Units are 10^{-20} cm^2 per target electron.

TABLE IV. 3

SUBSHELL CAPTURE CROSS SECTIONS PER
TARGET ATOM FOR PROTONS ON ARGON (cm^2)*

TARGET	ENERGY(KEV)	σ_s	σ_{px}	σ_{pz}	σ_T
Ar(n=3)	2.0	8.9(-17)	9.7(-16)	2.4(-16)	1.3(-15)
	5.0	1.5(-16)	7.0(-16)	3.0(-16)	1.1(-15)
	10.0	2.0(-16)	6.4(-16)	2.7(-16)	1.1(-15)
	20.0	2.4(-16)	4.8(-16)	2.8(-16)	1.0(-15)
	50.0	1.0(-16)	1.4(-16)	1.4(-16)	3.8(-16)
	75.0	2.2(-17)	3.2(-17)	4.1(-17)	9.5(-17)
	100.0	1.0(-17)	1.5(-17)	2.4(-17)	4.9(-17)
	200.0	8.8(-19)	1.7(-18)	4.1(-18)	6.7(-18)
Ar(n=2)	200.0	5.8(-20)	3.2(-19)	2.9(-19)	6.7(-19)
	400.0	1.4(-19)	2.2(-19)	3.4(-20)	4.0(-19)
	600.0	1.2(-19)	1.3(-19)	3.3(-21)	2.5(-19)
	1000.0	4.5(-20)	4.6(-20)	1.4(-20)	1.1(-19)
	3000.0	5.7(-22)	1.5(-21)	2.9(-21)	5.0(-21)

*The numbers in paranthesis are the exponents of the cross sections.

TABLE IV. 4

SUBSHELL CAPTURE CROSS SECTIONS PER TARGET ATOM FOR PROTONS ON KRYPTON AND NEON (cm^2)*

TARGET	ENERGY(KEV)	σ_s	σ_{px}	σ_{pz}	σ_T
K_r(n=4)	2.0	2.1(-16)	1.2(-15)	6.2(-16)	2.0(-15)
	5.0	1.7(-16)	8.9(-16)	3.9(-16)	1.5(-15)
	12.5	3.2(-16)	7.3(-16)	4.1(-16)	1.5(-15)
	25.0	2.6(-16)	4.0(-16)	3.2(-16)	9.7(-16)
	50.0	7.2(-17)	1.3(-16)	1.5(-16)	3.5(-16)
	100.0	2.1(-17)	2.2(-17)	4.0(-17)	8.0(-17)
	200.0	2.2(-18)	2.0(-18)	4.5(-18)	8.7(-18)
N_e(n=2)	2.0	---	1.6(-16)	2.7(-16)	4.3(-16)
	5.0	1.4(-18)	3.5(-16)	1.3(-16)	4.8(-16)
	10.0	4.4(-18)	3.4(-16)	2.8(-17)	3.7(-16)
	30.0	5.3(-17)	2.1(-16)	2.1(-17)	2.9(-16)
	60.0	4.3(-17)	8.0(-17)	2.2(-17)	1.5(-16)
	100.0	2.3(-17)	3.1(-17)	1.6(-17)	7.0(-17)

*The numbers in paranthesis are the exponents of the cross sections.

TABLE IV. 5

Z_p DEPENDENCE OF THE ARGON (N=2) → Z_p (N=1)
CAPTURE CROSS SECTIONS PER TARGET ELECTRON (Cm^2)*

E(MEV/AMU)	Z_p	σ_{2s}	σ_{2px}	σ_{2pz}	σ_T	σ_{T/z^5}
200	1	5.8(-20)	3.2(-19)	2.9(-19)	6.7(-19)	6.7(-19)
	2	2.3(-18)	9.6(-18)	1.9(-18)	1.4(-17)	4.4(-19)
	3	1.5(-17)	3.0(-17)	3.5(-18)	4.8(-17)	2.0(-19)
	4	3.4(-17)	3.6(-17)	1.4(-17)	8.4(-17)	8.2(-20)
	5	3.5(-17)	3.7(-17)	2.3(-17)	9.5(-17)	3.0(-20)
400	1	1.4(-19)	2.2(-19)	3.4(-20)	4.0(-19)	4.0(-19)
	2	2.0(-18)	4.9(-18)	7.0(-20)	7.0(-18)	2.2(-19)
	3	8.2(-18)	1.7(-17)	3.7(-18)	2.9(-17)	1.2(-19)
	4	1.5(-17)	2.7(-17)	1.5(-17)	5.7(-17)	5.6(-20)
	5	1.6(-17)	2.7(-17)	2.1(-17)	6.4(-17)	2.0(-20)
600	1	1.2(-19)	1.3(-19)	3.3(-21)	2.5(-19)	2.5(-19)
	2	1.2(-18)	2.6(-18)	2.6(-19)	4.1(-18)	1.3(-19)
	3	4.2(-18)	9.8(-18)	3.8(-18)	1.8(-17)	7.4(-20)
	4	7.2(-18)	1.7(-17)	1.2(-17)	3.7(-17)	3.6(-20)
	5	8.3(-18)	2.0(-17)	1.7(-17)	4.5(-17)	1.4(-20)

*The numbers in paranthesis are the exponents of the cross sections.

APPENDIX I

A. Potential and Wavefunctions. This section outlines the procedure used to determine the wavefunction parameters, A_i, in Eqn. III.5 and the corresponding energy eigenvalue. Within the independent electron approximation, the wavefunction for the active electron is written

$$\psi_{n\ell m}(\vec{r}) = \frac{P_{n\ell}(r)}{r} Y_{\ell m}(\Omega) \qquad \text{A.1}$$

$P_{n\ell}(r)$ satisfies $\{H_\ell - E_{n\ell}\} P_{n\ell} = 0$ where

$$H_\ell(r) = -\frac{1}{2}\frac{d^2}{dr^2} + \frac{\ell(\ell+1)}{2r^2} + V(r) \qquad \text{A.2}$$

The first part of program HERMAN in App. II.A fits the potential of Herman and Skillman to the form

$$-rV(r) = I + (Z-I)e^{-hr}\left\{1 + \sum_{k=1}^{3} c_k r^k\right\} \qquad \text{A.3}$$

$P_{n\ell}(r)$ is expressed as a sum of Slater type orbitals

$$P_{n\ell}(r) = \sum_i A_i \chi_i \qquad \text{A.4}$$

The column matrix, A, is obtained from the solution of the eigenvalue equation

$$S^{-1}HA = EA \qquad \text{A.5}$$

where S and H are defined as

$$S_{ij} = \langle \chi_i | \chi_j \rangle \qquad \text{A.6}$$

$$H_{ij} = \langle \chi_i | H_e | \chi_j \rangle \qquad \text{A.7}$$

Eqn. A.5 is inconvenient to solve because the basis functions, $\{\chi_i\}$, are not orthogonal. Though programs are available to diagonalize matrices of the form $A^{-1}B$, Eqn. A.5 can also be solved by expressing the $\{\chi_i\}$ in terms of an orthonormal basis set, $\{\phi_i\}$

$$\chi_i = \sum_j a_{ij} \phi_j \qquad \text{A.8}$$

where the a_{ij} are determined by the procedure of Schmidt orthogonalization. Both the χ_i and ϕ_i are of the form

$$\chi_i = N_i r^{n_i} e^{-\zeta_i r} \qquad \text{A.9}$$

where the parameters n_i and α_i are taken from the tables by Clementi and Roetti.[22] The basic integrals to be solved are

$$\begin{aligned} S_{ij} &= \langle \chi_i | \chi_j \rangle \\ &= N_i N_j \int_0^\infty dr\, r^{n_i+n_j} e^{-(\alpha_i+\alpha_j)r} \\ &= N_i N_j \frac{\Gamma(n+1)}{\alpha^{n+1}} \end{aligned} \qquad \text{A.10}$$

where $n = n_i + n_j$ and $\alpha = \alpha_i + \alpha_j$.

$$\begin{aligned} H_{ij} &= \langle \chi_i | H_\ell | \chi_j \rangle \\ &= \langle \chi_i | -\tfrac{1}{2}\frac{d^2}{dr^2} + \frac{\ell(\ell+1)}{2r^2} + V(r) | \chi_j \rangle \\ &= H^1_{ij} + H^2_{ij} + H^3_{ij} \end{aligned} \qquad \text{A.11}$$

H^1_{ij}, H^2_{ij}, and H^3_{ij} are defined

$$\begin{aligned} H^1_{ij} &= \langle \chi_i | -\frac{1}{2}\frac{d^2}{dr^2} | \chi_j \rangle \\ &= -\frac{1}{2}\left\langle \frac{d}{dr}\chi_i \middle| \frac{d}{dr}\chi_j \right\rangle \\ &= \frac{N_i N_j}{-2}\left\{ n_i n_j \frac{\Gamma(n-1)}{\alpha^{n-1}} \right. \\ &\quad \left. -(n_i\alpha_j + n_j\alpha_i)\frac{\Gamma(n)}{\alpha^n} + \alpha_i\alpha_j \frac{\Gamma(n+1)}{\alpha^{n+1}} \right\} \end{aligned} \qquad \text{A.12}$$

$$H^2_{ij} = \langle \chi_i \left| \frac{l(l+1)}{2r^2} \right| \chi_j \rangle \qquad \text{A.13}$$

$$= N_i N_j \frac{l(l+1)}{2} \frac{\Gamma(n-1)}{\alpha^{n-1}}$$

$$H^3_{ij} = \langle \chi_i | V(r) | \chi_j \rangle \qquad \text{A.14}$$

$$= -\langle r^{-1} \chi_i \left| I + (Z-I) e^{\lambda_v r} \left\{ 1 + \sum_{k=1}^{3} c_k r^k \right\} \right| \chi_j \rangle$$

$$= -N_i N_j \left\{ I \frac{\Gamma(n)}{\alpha^n} + (Z-I) \left[\frac{\Gamma(n)}{\alpha'^n} + \sum_{k=1}^{3} c_k \frac{\Gamma(n+k)}{\alpha'^{n+k}} \right] \right\}$$

where $\alpha' = \alpha + \lambda_v$. Evaluation of these matrices and the diagonalization of Eqn. A.5 is performed in the second part of program HERMAN listed in App. II.A.

B. Evaluation of the Matrix Elements. The two centered matrix elements appearing in Eqn. II.9 can be evaluated in prolate spheroidal coordinates. The procedure used to derive the off-diagonal terms of both the overlap and interaction matrices is similar because of the common factor $e^{i \vec{v} \cdot \vec{r}}$. These matrix elements are evaluated in Appendix I.B.1. The derivation of the diagonal terms is given in Appendix I.B.2.

B.1. Off-Diagonal Terms. The following definitions necessary for the evaluation of the off-diagonal matrix elements in Eqns. II.9 can be found in the monograph by McDowell and Coleman.[5]

$$\vec{\rho} \cdot \vec{r} = \tfrac{1}{2} \rho^2 t \lambda\mu + \rho\rho/2 \sqrt{\lambda^2-1}\sqrt{1-\mu^2}\cos\phi \qquad \text{A.15}$$

$$r_A = R/2(\lambda+\mu)$$

$$r_B = R/2(\lambda-\mu)$$

$$r_B \cos\theta_B = R/2(\lambda\mu-1)$$

$$r_B \sin\theta_B = R/2\sqrt{\lambda^2-1}\sqrt{1-\mu^2}$$

It is convenient to define the function

$$4\pi G_{KL}(\delta_+\delta_-) = \int_1^\infty d\lambda\, \lambda^K (\lambda^2-1)^{m/2} e^{-\lambda\delta_+} \qquad \text{A.16}$$

$$\times \int_{-1}^{1} d\mu\, \mu^L (1-\mu^2)^{m/2} \int_0^{2\pi} d\phi\, (-)^m e^{im\phi} e^{i\vec{\rho}\cdot\vec{r} - \mu\delta_-}$$

where

$$i\vec{\nu}\cdot\vec{\pi} - \mu\delta_- = i\,T\sin\chi\sqrt{1-\mu^2}\cos\phi + i\mu T\cos\chi \qquad \text{A.17}$$

$$T\cos\chi = \tfrac{1}{2}\nu^2\lambda + i\,\delta_-$$

$$T\sin\chi = \tfrac{1}{2}\nu\rho\sqrt{\lambda^2+1}$$

Using Bessel's integral

$$2\pi i^{|m|} J_{|m|}\left(T\sin\chi\sqrt{1-\mu^2}\right) \qquad \text{A.18}$$

$$= \int_0^{2\pi} d\phi\, e^{i\left\{T\sin\chi\sqrt{1-\mu^2}\cos\phi + m\phi\right\}}$$

the integration over ϕ is easily performed. $\mu^L(-)^m(1-\mu^2)^{m/2}$ is expanded in an associated Legendre series

$$\mu^L(-)^m(1-\mu^2)^{m/2} = \sum_{l=m}^{L+m} \frac{2l+1}{2}\, a_{Ll}\, P_l^m(\mu) \qquad \text{A.19}$$

where $a_{Ll} = 0$ for $L+l$ odd. For $L+l$ even

$$a_{Ll} = \frac{1}{2^m}\,\frac{\Gamma\left(\frac{1+L}{2}\right)}{\Gamma\left(\frac{2+L+m-l}{2}\right)}\,\frac{\Gamma\left(\frac{2+L}{2}\right)}{\Gamma\left(\frac{3+L+m+l}{2}\right)} \qquad \text{A.20}$$

The μ integration is done with the help of an identity given by Watson[24].

$$2i^{l-m} j_l(T) P_l^m(\cos\chi) = \int_{-1}^{1} d\mu\, J_m(T \sin\chi \sqrt{1-\mu^2}) P_l^m(\mu) e^{i\mu T\cos\chi} \qquad \text{A.21}$$

The resulting expression is the same regardless of whether m is positive or negative.

$$G_{KL}(\delta_+ \delta_-) = \sum_{l=m}^{L+m} i^l a_{Ll} \frac{2l+1}{2} \int_1^\infty d\lambda\, e^{-\lambda\delta_+} (\lambda^2-1)^{m/2} \lambda^K j_l(T) P_l^m(\cos\chi) \qquad \text{A.22}$$

The integration over λ is done by making a simple change of variables and using Gauss-Laguerre quadrature.

In this work only processes of the form

$$B^+ + A(ns) \rightarrow B(n'l) + A^+ \qquad \text{A.23}$$

and their time reversed counterparts have been considered. Though the programs in Appendix II.B and II.C are written for the specific cases of $l = 0$ and $l = 1$, it is simplest to derive the matrix elements for the more general processes, Eqn. A.23 and then specialize.

The initial and final state wavefunctions for the reaction, A.23 are

$$\psi^A_{n_A 00} = \frac{P_{n_A}(r_A)}{r_A} Y_{00}(\Omega_A) \qquad \text{A.24}$$

$$\psi^{B}_{n_B l m} = \frac{P_{n_B}(r_B)}{r_B} Y_{lm}(\Omega_B) \qquad \text{A.25}$$

where

$$P^{\chi}_{n_\chi}(r_\chi) = \sum_k C_{n_\chi k} r^{n_k} e^{-\delta_k r_\chi} \qquad n_\chi \leq n_k \qquad \text{A.26}$$

χ stands for either A or B. The standard series representation is used for $Y_{lm}(\Omega_B)$.

$$Y_{lm}(\Omega_B) = \frac{(-)^m e^{im\phi}}{\sqrt{4\pi}} (\sin\theta_B)^m \sum_{k=m}^{\frac{l-m}{2}} b^k_{lm} (\cos\theta_B)^{l-m-2k} \qquad \text{A.27}$$

where

$$b^k_{lm} = \frac{\sqrt{2l+1}}{2^l l!} \frac{\sqrt{(l-m)!}}{\sqrt{(l+m)!}} \binom{l}{k} \frac{(-)^k (2l-2k)!}{(l-m-2k)!}$$

Using the definitions A.15 and repeated applications of the binomial expansion to r_A , r_B , and $\cos\theta_B$, the expression for the wavefunctions are cast into the form

$$\sqrt{4\pi}\, r_A \psi^{A}_{n_A 00} = \sum_i A_i \left(R/2\right)^{n_i} \sum_{p=0}^{n_i} \binom{n_i}{p} \lambda^{n_i - p} \mu^p e^{-\alpha_i r_A} \qquad \text{A.28}$$

$$\sqrt{4\pi}\, r_B \psi^B_{n_B l m} = \sum_j B_j \left[\frac{R}{2}\right]^{n_j} \sum_{k=0}^{\frac{l-m}{2}} b^k_{lm} \sum_{g=0}^{n_1} \binom{n_1}{g} \sum_{r=0}^{n_2} \binom{n_2}{r} (-)^{n_2-r+g}$$
$$\lambda^{n_1+r-g} \mu^{r+g} \{(\lambda^2-1)(1-\mu^2)\}^{m/2} (-)^m e^{im\phi} e^{-\beta_j r_B} \qquad \text{A.29}$$

where

$$n_1 = n_j - l + 2k$$
$$n_2 = l - m - 2k$$

It is convenient as well to express V_A and V_B in powers of λ and μ.

$$-r_A V_A = I_A + (Z_A - I_A)\, e^{-\lambda^A_\nu r_A} \sum_{\Lambda=0}^{3} c^A_\Lambda \left[\frac{R}{2}\right]^\Lambda \sum_{\Lambda'=0}^{\Lambda} \binom{\Lambda}{\Lambda'} \lambda^{\Lambda-\Lambda'} \mu^{\Lambda'}$$
$$-r_B V_B = I_B + (Z_B - I_B)\, e^{-\lambda^B_\nu r_B} \sum_{t=0}^{3} c^B_t \left[\frac{R}{2}\right]^t \sum_{t'=0}^{t} \binom{t}{t'} \lambda^{t-t'} (-\mu)^{t'} \qquad \text{A.30}$$

where $c^A_0 = c^B_0 = 1$.

Substitution of Eqns. A.28 - A.30 into Eqns. II.9 gives for the off-diagonal terms

$$S_{AB} = R/2\, F^0_{00} \qquad \text{A.31}$$
$$H_{AB} = -F^A_{10}$$
$$H^*_{BA} = -F^B_{01}$$

where

$$F^{\chi}_{n_1' n_2'} = \sum_i A_i \sum_j B_j \left[R/2\right]^{n_{1i}+n_j} \qquad \text{A.32}$$

$$\times \sum_{k=0}^{\frac{L-m}{2}} \omega_{lm}^{k} \sum_{p=0}^{n_{1i}-n_1'} \binom{n_{1i}-n_1'}{p} \sum_{g=0}^{n_1-n_2'} \binom{n_1-n_2'}{g} \sum_{r=0}^{n_2} \binom{n_2}{r} (-)^{n_2+r+\zeta}$$

$$\times \left\{ I_\chi G_{N-n_1'-n_2', M}(\delta_+ \delta_-) + (-1)^{\chi} I_\chi \sum_{A=0}^{3} C_A^{\chi} \left[R/2\right]^A \sum_{t=0}^{A} \binom{A}{t} \pi_\chi^t G_{N'-n_1'-n_2', M}(\delta_+^\chi \delta_-^\chi) \right\}$$

and

$$\pi_A = 1 \qquad I_0 = 1$$

$$\pi_B = -1 \qquad C_A^0 = 0 \quad A = 0, 1, 2, 3$$

$$\delta_\pm = R/2(\alpha_i \pm \beta_j) \qquad N = n_1 + r - g + n_{1i} - p \qquad \text{A.33}$$

$$\delta_+^\chi = \delta_+ + \lambda_v^\chi R/2 \qquad M = p + r + g$$

$$\delta_-^A = \delta_- + \lambda_v^A R/2 \qquad N' = N + A - t$$

$$\delta_-^B = \delta_- - \lambda_v^B R/2 \qquad M' = M + t$$

B.2 Diagonal Terms. It is convenient to define the function

$$I_{N_1 N_2}(\delta_+ \delta_-) = \frac{1}{4\pi}\int_0^{2\pi} d\phi \int_1^\infty d\lambda \lambda^{N_1} e^{-\lambda\delta_+} \int_{-1}^{1} d\mu \mu^{N_2} e^{-\mu\delta_-} \qquad \text{A.34}$$

$$= \frac{1}{2} g_{N_1}(\delta_+) f_{N_2}(\delta_-)$$

where

$$g_n(\alpha) = \int_1^\infty d\lambda\, \lambda^n e^{-\alpha\lambda} \qquad \text{A.35}$$

$$f_n(\alpha) = \int_{-1}^{1} d\mu\, \mu^n e^{-\alpha\mu}$$

The g n's and fn's satisfy the recursion relations

$$\alpha g_n = (n-1) g_{n-1} + e^{-\alpha} \qquad \text{A.36}$$

$$\alpha f_n = (n-1) f_{n-1} - (-)^n e^{\alpha} - e^{-\alpha}$$

where

$$\alpha g_0 = e^{-\alpha} \qquad \text{A.37}$$

$$\alpha f_0 = e^{\alpha} - e^{-\alpha}$$

Using Eqns. A.34 and A.28-29, it is straightforward to derive the analytical expression for the diagonal terms.

$$H_{AA} = -\sum_{ii'} A_i A_{i'} \left[R/2\right]^{n_i+n_{i'}} \sum_{p'=0}^{n_{i'}} \binom{n_{i'}}{p'} \sum_{p=0}^{n_i-1} \binom{n_i-1}{p} \sum_{3}^{A} \times \left\{ I_B h_{N_1 N_2}(\delta_+ \delta_-) + (Z_B - I_B) \sum_{A=0}^{3} C_A^B \left[R/2\right]^A \sum_{\Delta'=0}^{A} \binom{A}{\Delta'} (-)^{\Delta'} h_{N_1+A-\Delta', N_2+\Delta'}(\delta_+^B \delta_-^B) \right\} \quad \text{A.38}$$

where

$$\delta_\pm = R/2(\alpha_i \pm \alpha_{i'}) \qquad N_2 = p + p'$$
$$\delta_\pm^B = \gamma_\pm \pm \lambda_V^B R/2 \qquad N_1 = n_i + n_{i'} - 1 - N_2$$

A.39

$$H_{BB} = -\sum_{jj'} B_j B_{j'} \left[R/2\right]^{n_j+n_{j'}} \sum_{kk'=0}^{\frac{l-m}{2}} b_{lm}^k b_{lm}^{k'} \sum_{g=0}^{n_{j'}} \binom{n_{j'}}{g} \sum_{g'=0}^{n_j-1} \binom{n_j-1}{g'} \sum_{r=0}^{n_2} \binom{n_2}{r} \sum_{r'=0}^{n_2'} \binom{n_2'}{r'} \sum_{\Delta=0}^{m} \binom{m}{\Delta} (-)^{\Delta} \times \sum_{\Delta'=0}^{m} \binom{m}{\Delta'} \left\{ I_A h_{N_1 N_2}(\gamma_+ \gamma_-) + (Z_A - I_A) \sum_{t=0}^{3} C_t^A \left[R/2\right]^t \sum_{t'=0}^{t} \binom{t}{t'} (-)^{t'} h_{N_1'+t-t', N_2'+t'}(\gamma_+^A \gamma_-^A) \right\}$$

where

$$\gamma_\pm = R/2(\beta_j \pm \beta_{j'}) \qquad n_2 = l - m - 2k$$
$$\gamma_\pm^A = \gamma_\pm \pm \lambda_V^A R/2 \qquad n_2' = l - m - 2k'$$
$$n_1 = n_j - l + 2k \qquad N_1 = n_1 + n_1' + r + r' - g - g' - 1 + 2(m-\Delta)$$
$$n_1' = n_{j'} - l + 2k' \qquad N_2 = g + g' + r + r' + 2\Delta'$$

The coding written to perform these calculations is listed in Appendices II.B and II.C.

APPENDIX II

Appendix II lists the computer coding written to perform the calculations of Appendix I. Appendix II.A begins on p. 79, Appendix II.B, on p. 89; and Appendix II.C begins on p. 108.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

                                 HERMAN

PURPOSE:  THIS PROGRAM DIAGONALIZES THE HAMILTONIAN FOR ANY
          POTENTIAL WHICH CAN BE FITTED TO THE FORM

          V(R) = (1+(Z-I)*(1+F(R))*EXP(-OLM*R))/Z              EQN. 1

          WHERE F(R) IS A POLYNOMIAL OF ORDER THREE.
          THE RADIAL WAVEFUNCTIONS ARE EXPRESSED AS SUMS OF
          SLATER TYPE ORBITALS.

          P(R) = SUM(I) (C(I)*X(I))                            EQN. 2

          EACH ORBITAL IS OF THE FORM

          X(I) = NORM*(R**N)*EXPH-A*R)                         EQN. 3

          WHERE NORM IS A NORMALIZATION FACTOR. THE THE SCREENING
          FUNCTION CORRESPONDING TO EQN. I  IS READ IN AS WELL AS
          THE PARAMETERS A AND N IN EQN. 3.  THE PROGRAM PUNCHES OUT
          OLM, THE COEFFICIENTS IN F(R), AND THE
          INFORMATION NECESSARY FOR THE DESCRIPTION OF THE WAVEFUNCTION.


CARD 1:   (LABEL(I),I=1,10), Z, RM, NUMTIM, NU, NZ  (10A4,2F5.0,6I5)
          LABEL:  PROGRAM TITLE
          Z:  CHARGE OF THE NUCLEUS
          NUMTIM:  THE NUMBER OF L VALVUES FOR WHICH THE DIAGONALIZATION
                   IS PERFORMED
          IZ:  THE IONICITY OF THE ATOM
          NU:  THE NUMBER OF VALUES OF THE SCREENING FUNCTION TO BE READ
CARD 2:   (U(I),I=1,NU)  (8F10.5)
          U:  VALUES OF THE SCREENING FUNCTION
              FOR A HYDROGENIC POTENTIAL, ONE BLANK CARD
              SHOULD BE READ IN.
          .

THE NEXT CARDS ARE INPUTTED IN SETS OF THREE. THERE ARE NUMTIM SETS.

CARD A:   NTRMS, L  (16I5)
          NTRMS:  THE UPPER LIMIT OF THE SUM IN EQN. 2
          L:  THE ORBITAL ANGULAR MOMENTUM QUANTUM NUMBER
CARD B:   (NOP(I),I=1,NTRMS)
          NOP(I):  THE N  IN EQN. 3
CARD C:   (AXP(I),I=1,NTRMS)
          AXP(I):  THE A  IN EQN. 3


FOR EACH L VALUE, THERE WILL BE X NUMBER OF BOUND STATES. FOUR CARDS
WILL BE OUTPUTTED FOR EACH STATE BEGINNING WITH THE ONE HAVING
THE MOST NEGATIVE ENERGY EIGENVALUE.

CARD AA:   EVAL, D, (CV(J),J=1,3), DNZ
           EVAL:  ENERGY EIGENVALUE FOR A GIVEN STATE.
           D:  THE FLOATING POINT EQUIVALENT OF NTRMS
           OLM:  DEFINED IN EQN. 1.
           CV:  THE COEFFICIENTS IN F(R) OF EQN. 1
           DNZ:  THE FLOATING POINT EQUIVALENT OF NZ
CARD BB:   IMAGE OF CARD B
CARD CC:   IMAGE OF CARD C
CARD DD:   (EVCT(K,I),I=1,NTRMS)
           THE C(I) IN EQN. 2 FOR A GIVEN STATE K

CARD IMAGES OF THE OUTPUT ARE PRINTED IN THE PROGRAM AFTER THE LISTING
OF THE WAVEFUNCTIONS FOR EVERY L VALUE. EACH SET OF FOUR CARDS ARE THEN
USED AS INPUT FOR ANSBNS AND ANSBNP.

                     WRITTEN BY LAURA TUNNELL (1976)


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/WF/ACF(20),AXP(20),ANRM(20),NOP(20),LOP(20),L,NTRMS
      COMMON/HSV/CV(10),DLM,Z,ZI,RM,U(225),NV,NU
      COMMON/EGN/A(225),G(225),X(225),EVAL(20),EVCT(12,12),NM
      DIMENSION LABEL(20)
    1 FORMAT(16I5)
    2 FORMAT(8F10.5)
    3 FORMAT(10A4,2F5.0,8I5)
    4 FORMAT(///10A4,'  Z =',F3.0/)
    5 FORMAT(///'1',10A4,'  L =',I2/)
  220 FORMAT(///' '/)


      NV=3
      DLM=0.000
      DO 20 I=1,10
   20 CV(I)=0.000
      READ(5,3)  (LABEL(I),I=1,10),Z,RM,NUMTIM,NU,NZ
      IF(RM.EQ.0.000)  RM=1.000
      IF(NUMTIM.EQ.0)  NUMTIM=1
      IF(NU.EQ.0)  NU=1
      ONZ=NZ
      ZI=ONZ+1
      READ(5,2)  (U(I),I=1,NU)
      PRINT 220
      PRINT 200
  200 FORMAT('   (LABEL(I),I=1,10),Z,RM,NUMTIM,NU,NZ          ')
      PRINT 3,   (LABEL(I),I=1,10),Z,RM,NUMTIM,NU,NZ
      PRINT 210
  210 FORMAT('   (U(I),I=1,NU) ')
      PRINT 2,   (U(I),I=1,NU)
      PRINT 220
      PRINT 4, (LABEL(I),I=1,10),Z
      IF(U(1).EQ.1.000)  CALL VFCTN
      DO 12 N=1,NUMTIM
      READ(5,1)  NTRMS,L
      READ(5,1)  (NOP(I),I=1,NTRMS)
      READ(5,2)  (AXP(I),I=1,NTRMS)
      PRINT 220
      PRINT 221
  221 FORMAT('   NTRMS,L      ')
      PRINT 1,   NTRMS,L
      PRINT 222
  222 FORMAT('   (NOP(I),I=1,NTRMS)     ')
      PRINT 1,   (NOP(I),I=1,NTRMS)
      PRINT 223
  223 FORMAT('   (AXP(1),I=1,NTRMS)  ')
      PRINT 2,   (AXP(I),I=1,NTRMS)
      PRINT 220
      DO 10 J=1,NTRMS
      A1=(2.000*AXP(J))**(2*NOP(J)+1)
      A2=DGAMMA(DFLOAT(2*NOP(J)+1))
   10 ANRM(J)=DSQRT(A1/A2)
      PRINT 5, (LABEL(I),I=1,10),L
      CALL COEF
      DO 12 II=1,NTRMS
      I=(1+NTRMS)-II
      IF(EVAL(I).GT.0.000)  GO TO 12
      EVAL1=-EVAL(I)
      D=DFLOAT(NTRMS)
      PUNCH 2,   EVAL1,D,DLM,(CV(J),J=1,3),ONZ
      PUNCH 1, (NOP(J),J=1,NTRMS)
      PUNCH 2, (AXP(J),J=1,NTRMS)
      PUNCH 2, (EVCT(I,J),J=1,NTRMS)
      PRINT 2,   EVAL1,D,DLM,(CV(J),J=1,3),ONZ
      PRINT 1, (NOP(J),J=1,NTRMS)
      PRINT 2, (AXP(J),J=1,NTRMS)
      PRINT 2, (EVCT(I,J),J=1,NTRMS)
   12 CONTINUE

      STOP
      END

      SUBROUTINE COEF
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/WF/ACF(20),AXP(20),ANRM(20),NOP(20),LOP(20),L,NTRMS
      COMMON/EGN/A(225),G(225),X(225),EVAL(20),EVCT(12,12),NM
      DIMENSION  WFCTN(80,12)
      DIMENSION ALP(12,12),DLP(12,12),H(12,12),R(12,12)
      COMMON/HSV/CV(10),DLM,Z,ZI,WS,G(225),NV,N7
      DIMENSION RF(20),VF(20),J(20)
      DIMENSION   RI(80)
    3 FORMAT(10(1X,1PD12.5))
    6 FORMAT(10(1X,1PD12.5))
```

```
      NM=NTRMS
      DO 10 I=1,NTRMS
      DO 10 J=1,NTRMS
      NO=NOP(I)+NOP(J)+1
      N1=NO-1
      N2=N1-1
      AA=AXP(I)+AXP(J)
      RF(1)=1.000/AA
      DO 11 K=1,NO
      KK=K-1
   11 RF(K+1)=(DFLOAT(KK+1)/AA)*RF(K)
      H1=NOP(I)*NOP(J)*AF(N2)-(NOP(I)*AXP(J)+NOP(J)*AXP(I))*RF(N1)
     . +AXP(I)*AXP(J)*RF(NO)
      H2=DFLOAT(2*(L+1))*RF(N2)
      VA=ALH+AA
      VF(1)=1.000/VA
      NNV=NV+NO
      DO 12 K=1,NNV
      KK=K-1
   12 VF(K+1)=(DFLOAT(KK+1)/VA)*VF(K)
      NN1=NO+1
      NN1=NNV+1
      H3=0.000
      DO 13 K=1,NV
   13 H3=H3+CV(K)*VF(N1+K)
      H3=Z(*RF(N1)+(Z-Z1)*(VF(N1)+H3)
      AN=ANRM(I)*ANRM(J)
      H(I,J)=AN*(H1+H2-2.000*H3)/2.000
      R(I,J)=AN*RF(NO)
      H(J,I)=H(I,J)
      R(J,I)=R(I,J)
   10 CONTINUE
      ALP(1,1)=1.000
      DO 30 N=2,NTRMS
      ALP(N,N)=1.000
      N1=N-1
      ONRM=1.000
      DO 36 I=1,N1
      A(I)=0.000
      DO 35 J=1,I
   35 A(I)=A(I)+ALP(I,J)*R(J,N)
   36 ONRM=ONRM-A(I)*A(I)
      ONRM=DSQRT(DABS(ONRM))
      DO 21 K=1,N1
      ALP(K,N)=0.000
      ALP(N,K)=0.000
      DO 21 I=K,N1
   21 ALP(N,K)=ALP(N,K)-A(I)*ALP(I,K)
      DO 30 K=1,N
   30 ALP(N,K)=ALP(N,K)/ONRM
      DO 70 I=1,NTRMS
      DO 70 J=1,I
      OLP(I,J)=0.000
      DO 70 K=1,I
      EVCT(K,J)=0.000
      DO 75 L=1,J
      EVCT(K,J)=EVCT(K,J)+H(K,L)*ALP(J,L)
   75 CONTINUE
      OLP(I,J)=OLP(I,J)+ALP(I,K)*EVCT(K,J)
   70 CONTINUE
      PRINT 5
    5 FORMAT(//'  HAMILTON(AN  '/)
      K=0
      DO 80 I=1,NTRMS
      PRINT 6, (OLP(I,J),J=1,I)
      DO 80 J=1,I
      K=K+1
   80 A(K)=OLP(I,J)
      NM=NTRMS*NTRMS
      CALL EIGEN(NTRMS,0)
      NN=NTRMS*NTRMS
      NMAX=1
      KK=0
      LL=0
      DO 90 J=1,NTRMS
      LL=LL+J
      EVAL(J)=A(LL)
      IF(EVAL(J).GT.0.000)  NMAX=J
      DO 90 I=1,NTRMS
   90 OLP(J,I)=X(1+(J-1)*NTRMS)
      N1=NMAX+1
      DO 92 I=1,NTRMS
      DO 92 J=1,NTRMS
      EVCT(I,J)=0.000
      DO 193 K=1,NTRMS
```

```
      EVCT(I,J)=EVCT(I,J)+DLP(I,K)*ALP(K,J)
  193 CONTINUE
   92 CONTINUE
      PRINT 91
   91 FORMAT(//'     ENERGY EIGENVALUES      '//)
      PRINT 6, (EVAL(I),I=1,NTRMS)
      PRINT 93
   93 FORMAT(/'     COEFFICIENTS FOR BASIS SET     '/)
      DO 94 I=1,NTRMS
   94 PRINT 3,  (EVCT(J,I),J=1,NTRMS)
      S=0.0D0
      P=DARCOS(-1.0D0)
      FMU=(2.0D0*DLOG(3.0D0*P)-DLOG(Z)-7.0D0*DLOG(2.0D0))/3.0D0
      FMU=DEXP(FMU)
      DO 106 J=1,8
      SA=2.0D0**(J-1)
      SA=SA*1.0D-02
      DO 106 I=1,10
      IJ=I+(J-1)*1.0D+01
      A(IJ)=S
      R1(IJ)=FMU*S
  106 S=S+SA
      NU=IJ-1
      N2=NTRMS-NMAX
      PRINT 210
      DO 96 K=1,NU
      FLM=DLM*R1(K)
      IF(FLM.GE.1.40D+02) DD=0.0D0
      IF(FLM.LT.1.40D+02) DD=DEXP(-FLM)
      U1=R1(K)*(CV(1)+R1(K)*(CV(2)+R1(K)*CV(3)))
      U(K)=(Z(+(Z-Z1)*DD*(1.0D0+U1))/Z
      DO 96 J=N1,NTRMS
      WFCTN(K,J)=0.0D0
      DO 96 I=1,NTRMS
      W=0.0D0
      IF((AXP(I)*R1(K)).LE.1.20D+02) W=DEXP(-AXP(I)*R1(K))
      WFCTN(K,J)=WFCTN(K,J)+ EVCT(J,I)*ANRM(I)*(R1(K)**NDP(I))*W
   96 CONTINUE
      PRINT 103
  103 FORMAT(//'  X      R(X)        U(X)        P(X)        '//)
      A1=0.0D0
      DO 97 I=1,NU
   97 PRINT 3, A(I),R1(I),U(I),A1,(WFCTN(I,J),J=N1,NTRMS)

      N1=NTRMS-1
      PRINT 210
      SSUM=0.0D0
      NU1=NU-1
      DO 220 I=2,NU1
      X1=R1(I-1)
      X2=R1(I)
      X3=R1(I+1)
      Y1=WFCTN(I-1,NTRMS)
      Y2=WFCTN(I,NTRMS)
      Y3=WFCTN(I+1,NTRMS)
      Y1=Y1*Y1
      Y2=Y2*Y2
      Y3=Y3*Y3
  220 SSUM=SSUM+SUM(X1,X2,X3,Y1,Y2,Y3)
      PRINT 225, SSUM
  225 FORMAT(/'  SSUM= ', 1PD12.5//)
  210 FORMAT(/' '/)
      RETURN
      END

      REAL FUNCTION SUM*8(X1,X2,X3,Y1,Y2,Y3)
      IMPLICIT REAL*8 (A-H,O-Z)
      IF(DABS((X3-X2)-(X2-X1)).LT.1.0E-8) GO TO 10
      A=Y1/((X1-X2)*(X1-X3))
      B=Y2/((X2-X3)*(X2-X1))
      C=Y3/((X3-X1)*(X3-X2))
      D=A+B+C
      E=-A*(X2+X3)-B*(X1+X3)-C*(X2+X1)
      F=A*X2*X3+B*X1*X3+C*X1*X2
      SUM=(D*(X2**2+X1*(X2+X1))/3.0D0+E*(X2+X1)/2.0D0+F)*(X2-X1)
      RETURN
   10 D=X2-X1
      SUM=D*(5*Y1+8*Y2-Y3)/12.0D0
      RETURN
      END

      SUBROUTINE VFCTN
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/HSV/CV(10),DLM,Z,Z1,RMAX,U(225),NV,N
      DIMENSION LABEL(20),ERR(50)
```

```
      DIMENSION R(225),X(225),UU(225)
      DIMENSION  A(225,10), B(10,10), C(225), F(225)
      DIMENSION  IIK(3),IKK(3), SS(3),ZS(3), OC(10,3)
      DIMENSION SGMA(50),ZI(50),G(225),CV(10,6)
    3 FORMAT(IC(1X,1P012.5))
  305 FORMAT(/' ')


      XMAX=1.0D0
      NC=3
      ZZ=1.000/Z
      S=0.000
      PI=DARCOS(-1.000)
      OMU=(2.0D0*DLOG(3.000*PI)-DLOG(Z)-7.000*DLOG(2.000))/3.0D0
      OMU=DEXP(OMU)
      NU=0
      DO 6 J=1,12
      SA=2.000**(J-1)
      SA=SA*1.0D-02
      DO 6 I=1,10
      IJ=I+(J-1)*10
      R(IJ)=OMU*S
      X(IJ)=S
      S=S+SA
      IF(NU.NE.0)  GO TO 6
      IF(R(IJ).GE.XMAX)  NU=IJ
    6 CONTINUE
      NNU=IJ
      IF(NU.EQ.0)  NU=IJ
      DO 5 I=N,NU
    5 U(I)=ZZ
       DETERMINE  DLM
      S=0.000
      U1=0.000
      DO 32 I=2,NU
      R1=R(I-1)
      R2=R(I)
      U2=(U(I)*Z-ZI)/(Z-ZI)
      U2=DLOG(DABS(U2))
      SZ=(U2-U1)/(R2-R1)
      S=S+SZ
      IF(U(I).LE.2.000/Z)   GO TO 33
   32 U1=U2
   33 ZZ=-S/DFLOAT(I-1)
      DO 7 I=1,NU
      DLM=ZZ*R(I)
      IF(DLM.GT.1.40+02)  GO TO 8
      DLM=DEXP(DLM)
    7 UU(I)=(U(I)*Z-ZI)*DLM/(Z-ZI)-1.000
    8 NR=I-1
      CONSTRUCT BASIS VECTORS
      DO 10 I=1,NNU
      DO 10 J=1,NC
      A(I,J)=R(I)**J
   10 CONTINUE
      PERFORM MULTIPLICATION PROCEDURE
      DO 20 I=1,NC
      DO 20 J=I,NC
      B(I,J)=0.000
      DO 21 K=1,NR
   21 B(I,J)=B(I,J)+A(K,I)*A(K,J)
      B(J,I)=B(I,J)
      F(I)=0.000
      DO 22 K=1,NR
   22 F(I)=F(I)+A(K,I)*UU(K)
   20 CONTINUE
      DO 30 J=1,NC
      DO 30 I=1,NC
      IJ=I+(J-1)*NC
      C(IJ)=B(I,J)
   30 CONTINUE
      NNC=NC*NC
      II=NC
      KK=II
      SOLVE EQUATIONS FOR THE COEFFICIENTS
      CALL DGELG(F,C,NC,NNC,1,1.0E-03,IER)
      SGMA(II)=0.000
      CS=1.000
      SGG=0.000
      DO 40 I=1,NNU
      OLM=ZZ*R(I)
      IF(OLM.GE.1.400+02) DD=0.000
      IF(OLM.LT.1.400+02) DD=DEXP(-OLM)
      GG=0.000
      DO 50 J=1,NC
```

```
      CV(J,II)=F(J)
   50 GG=GG+F(J)*A(I,J)
      G(I)=(ZI+(Z-ZI)*DO*(1.000+GG))/Z
      IF(I.GE.NU)  GO TO 40
      SGMA(II)=SGMA(II)+((U(I)-GG)**2)*CS
   40 CONTINUE
      PRINT 309
  309 FORMAT(/////'    FIT  TO  THE  HERMAN-SKILLMAN  POTENTIAL  '/)
      PRINT 315, RMAX,OMU,ZZ
  315 FORMAT(' RMAX= ',1PD12.5,' OMU= ',1PD12.5,' OLM= ',1PD12.5)
      PRINT 310
  310 FORMAT(/'     X.....R.....       HM-SK.....FITTED.....       '/)
      DO 311 I=1,NNU
  311 PRINT 3, X(I),R(I),XR,U(I),G(I)
      PRINT 301
  301 FORMAT(//'    COEFFICIENTS FOR THE POTENTIAL ')
      PRINT 3,   (CV(J,KK),J=1,NC)
      PRINT 305
      OLM=ZZ
      DO 320 I=1,3
  320 OV(I)=CV(I,KK)
 5000 CONTINUE
      RETURN
      END
```

```
                                                                      OELG  10
C*************************************************************      OELG  20
C                                                                     OELG  30
C   SUBROUTINE OGELG                                                  OELG  40
C                                                                     OELG  50
C   PURPOSE                                                           OELG  60
C      TO SOLVE A GENERAL SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS.    OELG  70
C                                                                     OELG  80
C   USAGE                                                             OELG  90
C      CALL OGELG(R,A,M,N,EPS,IER)                                    OELG 100
C                                                                     OELG 110
C   DESCRIPTION OF PARAMETERS                                         OELG 120
C      R      - DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX        OELG 130
C               (DESTROYED). ON RETURN R CONTAINS THE SOLUTIONS       OELG 140
C               OF THE EQUATIONS.                                     OELG 150
C      A      - DOUBLE PRECISION M BY M COEFFICIENT MATRIX            OELG 160
C               (DESTROYED).                                          OELG 170
C      M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.                OELG 180
C      N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.                OELG 190
C      EPS    - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS      OELG 200
C               RELATIVE TOLERANCE FOR TEST ON LOSS OF                OELG 210
C               SIGNIFICANCE.                                         OELG 220
C      IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS            OELG 230
C               IER=0  - NO ERROR,                                    OELG 240
C               IER=-1 - NO RESULT BECAUSE OF M LESS THAN 1 OR        OELG 250
C                        PIVOT ELEMENT AT ANY ELIMINATION STEP        OELG 260
C                        EQUAL TO 0,                                  OELG 270
C               IER=K  - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI-     OELG 280
C                        CANCE INDICATED AT ELIMINATION STEP K+1,     OELG 290
C                        WHERE PIVOT ELEMENT WAS LESS THAN OR         OELG 300
C                        EQUAL TO THE INTERNAL TOLERANCE EPS TIMES    OELG 310
C                        ABSOLUTELY GREATEST ELEMENT OF MATRIX A.     OELG 320
C                                                                     OELG 330
C   REMARKS                                                           OELG 340
C      INPUT MATRICES R AND A ARE ASSUMED TO BE STORED COLUMNWISE     OELG 350
C      IN M*N RESP. M*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN       OELG 360
C      SOLUTION MATRIX R IS STORED COLUMNWISE TOO.                    OELG 370
C      THE PROCEDURE GIVES RESULTS IF THE NUMBER OF EQUATIONS M IS    OELG 380
C      GREATER THAN 0 AND PIVOT ELEMENTS AT ALL ELIMINATION STEPS     OELG 390
C      ARE DIFFERENT FROM 0. HOWEVER WARNING IER=K - IF GIVEN -       OELG 400
C      INDICATES POSSIBLE LOSS OF SIGNIFICANCE. IN CASE OF A WELL     OELG 410
C      SCALED MATRIX A AND APPROPRIATE TOLERANCE EPS, IER=K MAY BE    OELG 420
C      INTERPRETED THAT MATRIX A HAS THE RANK K. NO WARNING IS        OELG 430
C      GIVEN IN CASE M=1.                                             OELG 440
C                                                                     OELG 450
C   SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                     OELG 460
C      NONE                                                           OELG 470
C                                                                     OELG 480
C   METHOD                                                            OELG 490
C      SOLUTION IS DONE BY MEANS OF GAUSS-ELIMINATION WITH            OELG 500
C      COMPLETE PIVOTING.                                             OELG 510
C                                                                     OELG 520
C*************************************************************      OELG 530
C                                                                     OELG 540
      SUBROUTINE OGELG(R,A,M,MM,N,EPS,IER)
C                                                                     OELG 560
C                                                                     OELG 570
      DOUBLE PRECISION PIV,TB,TOL,PIVI
      DOUBLE PRECISION A(MM),R(M)
      IF(M)23,23,1                                                   OELG 600
```

```
                                                                        DELG 610
      SEARCH FOR GREATEST ELEMENT IN MATRIX A                           DELG 620
    1 IER=0                                                             DELG 630
      PIV=0.00                                                          DELG 640
      MM=M*M                                                            DELG 650
      NM=N*M                                                            DELG 660
      DO 3 L=1,MM                                                       DELG 670
      TB=DABS(A(L))                                                     DELG 680
      IF(TB-PIV)3,3,2                                                   DELG 690
    2 PIV=TB                                                            DELG 700
      I=L                                                               DELG 710
    3 CONTINUE                                                          DELG 720
      TOL=EPS*PIV                                                       DELG 730
      A(I) IS PIVOT ELEMENT. PIV CONTAINS THE ABSOLUTE VALUE OF A(I).   DELG 740
                                                                        DELG 750
                                                                        DELG 760
      START ELIMINATION LOOP                                            DELG 770
      LST=1                                                             DELG 780
      DO 17 K=1,M                                                       DELG 790
                                                                        DELG 800
      TEST ON SINGULARITY                                               DELG 810
      IF(PIV)23,23,4                                                    DELG 820
    4 IF(IER)7,5,7                                                      DELG 830
    5 IF(PIV-TOL)6,6,7                                                  DELG 840
    6 IER=K-1                                                           DELG 850
    7 PIVI=1.00/A(I)                                                    DELG 860
      J=(I-1)/M                                                         DELG 870
      I=I-J*M-K                                                         DELG 880
      J=J+1-K                                                           DELG 890
      I+K IS ROW-INDEX, J+K COLUMN-INDEX OF PIVOT ELEMENT               DELG 900
                                                                        DELG 910
      PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R      DELG 920
      DO 8 L=K,NM,M                                                     DELG 930
      LL=L+I                                                            DELG 940
      TB=PIVI*R(LL)                                                     DELG 950
      R(LL)=R(L)                                                        DELG 960
    8 R(L)=TB                                                           DELG 970
                                                                        DELG 980
      IS ELIMINATION TERMINATED                                         DELG 990
      IF(K-M)9,18,18                                                    DELG1000
                                                                        DELG1010
      COLUMN INTERCHANGE IN MATRIX A                                    DELG1020
    9 LEND=LST+M-K                                                      DELG1030
      IF(J)12,12,10                                                     DELG1040
   10 II=J*M                                                            DELG1050
      DO 11 L=LST,LEND                                                  DELG1060
      TB=A(L)                                                           DELG1070
      LL=L+II                                                           DELG1080
      A(L)=A(LL)                                                        DELG1090
   11 A(LL)=TB                                                          DELG1100
                                                                        DELG1110
      ROW INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A               DELG1120
   12 DO 13 L=LST,MM,M                                                  DELG1130
      LL=L+I                                                            DELG1140
      TB=PIVI*A(LL)                                                     DELG1150
      A(LL)=A(L)                                                        DELG1160
   13 A(L)=TB                                                           DELG1170
                                                                        DELG1180
      SAVE COLUMN INTERCHANGE INFORMATION                               DELG1190
      A(LST)=J                                                          DELG1200
                                                                        DELG1210
      ELEMENT REDUCTION AND NEXT PIVOT SEARCH                           DELG1220
      PIV=0.00                                                          DELG1230
      LST=LST+I                                                         DELG1240
      J=0                                                               DELG1250
      DO 16 II=LST,LEND                                                 DELG1260
      PIVI=-A(II)                                                       DELG1270
      IST=II+M                                                          DELG1280
      J=J+1                                                             DELG1290
      DO 15 L=IST,MM,M                                                  DELG1300
      LL=L-J                                                            DELG1310
      A(L)=A(L)+PIVI*A(LL)                                              DELG1320
      TB=DABS(A(L))                                                     DELG1330
      IF(TB-PIV)15,15,14                                                DELG1340
   14 PIV=TB                                                            DELG1350
      I=L                                                               DELG1360
   15 CONTINUE                                                          DELG1370
      DO 16 L=K,NM,M                                                    DELG1380
      LL=L+J                                                            DELG1390
   16 R(LL)=R(LL)+PIVI*R(L)                                             DELG1400
   17 LST=LST+M                                                         DELG1410
      END OF ELIMINATION LOOP                                           DELG1420
                                                                        DELG1430
                                                                        DELG1440
      BACK SUBSTITUTION AND BACK INTERCHANGE                            DELG1450
```

```
   18 IF(M-1)23,22,19                                                   DELGI460
   19 IST=MM+M                                                          DELGI470
      LST=M+I                                                           DELGI480
      DO 21 I=2,M                                                       DELGI490
      II=LST-I                                                          DELGI500
      IST=IST-LST                                                       DELGI510
      L=IST-M                                                           DELGI520
      L=A(L)+.500                                                       DELGI530
      DO 21 J=II,NM,M                                                   DELGI540
      TB=R(J)                                                           DELGI550
      LL=J                                                              DELGI560
      DO 20 K=IST,MM,M                                                  DELGI570
      LL=LL+1                                                           DELGI580
   20 TB=TB-A(K)*R(LL)                                                  DELGI590
      K=J+L                                                             DELGI600
      R(J)=R(K)                                                         DELGI610
   21 R(K)=TB                                                           DELGI620
   22 RETURN                                                            DELGI630
                                                                        DELGI640
                                                                        DELGI650
      ERROR RETURN                                                      DELGI660
   23 IER=-2                                                            DELGI670
      RETURN                                                            DELGI680
      END                                                               DELGI690

                                                                        EIGE  10
      ..................................................................EIGE  20
                                                                        EIGE  30
         SUBROUTINE EIGEN                                               EIGE  40
                                                                        EIGE  50
         PURPOSE                                                        EIGE  60
            COMPUTE EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC     EIGE  70
            MATRIX                                                      EIGE  80
                                                                        EIGE  90
         USAGE                                                          EIGE 100
            CALL EIGEN(A,R,N,MV)                                        EIGE 110
                                                                        EIGE 120
         DESCRIPTION OF PARAMETERS                                      EIGE 130
            A - ORIGINAL MATRIX (SYMMETRIC), DESTROYED IN COMPUTATION.  EIGE 140
                RESULTANT EIGENVALUES ARE DEVELOPED IN DIAGONAL OF      EIGE 150
                MATRIX A IN DESCENDING ORDER.                           EIGE 160
            R - RESULTANT MATRIX OF EIGENVECTORS (STORED COLUMNWISE,     EIGE 170
                IN SAME SEQUENCE AS EIGENVALUES)                        EIGE 180
            N - ORDER OF MATRICES A AND R                               EIGE 190
            MV- INPUT CODE                                              EIGE 200
                      0   COMPUTE EIGENVALUES AND EIGENVECTORS          EIGE 210
                      1   COMPUTE EIGENVALUES ONLY (R NEED NOT BE       EIGE 220
                          DIMENSIONED BUT MUST STILL APPEAR IN CALLING  EIGE 230
                          SEQUENCE)                                     EIGE 240
                                                                        EIGE 250
         REMARKS                                                        EIGE 260
            ORIGINAL MATRIX A MUST BE REAL SYMMETRIC (STORAGE MODE=1)   EIGE 270
            MATRIX A CANNOT BE IN THE SAME LOCATION AS MATRIX R         EIGE 280
                                                                        EIGE 290
         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  EIGE 300
            NONE                                                        EIGE 310
                                                                        EIGE 320
         METHOD                                                         EIGE 330
            DIAGONALIZATION METHOD ORIGINATED BY JACOBI AND ADAPTED     EIGE 340
            BY VON NEUMANN FOR LARGE COMPUTERS AS FOUND IN 'MATHEMATICALEIGE 350
            METHODS FOR DIGITAL COMPUTERS', EDITED BY A. RALSTON AND    EIGE 360
            H.S. WILF, JOHN WILEY AND SONS, NEW YORK, 1962, CHAPTER 7   EIGE 370
                                                                        EIGE 380
      ..................................................................EIGE 390
                                                                        EIGE 400
      SUBROUTINE EIGEN(N,MV)
      DOUBLE PRECISION A,R,U,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,          EIGE 5
     1                 COSX2,SINCS,RANGE,AAAA,EVAL
      COMMON/EGN/A(225),U(225),R(225),EVAL(20),AAAA(12,12),NM10
                                                                        EIGE 640
         GENERATE IDENTITY MATRIX                                       EIGE 650
                                                                        EIGE 660
    5 RANGE=1.0D-6
      IF(MV-1) 10,25,10                                                 EIGE 680
   10 IQ=-N                                                             EIGE 690
      DO 20 J=1,N                                                       EIGE 700
      IQ=IQ+N                                                           EIGE 710
      DO 20 I=1,N                                                       EIGE 720
      IJ=IQ+I                                                           EIGE 730
      R(IJ)=0.0                                                         EIGE 740
      IF(I-J) 20,15,20                                                  EIGE 750
   15 R(IJ)=1.0                                                         EIGE 760
   20 CONTINUE                                                          EIGE 770
                                                                        EIGE 780
         COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX)             EIGE 790
```

```
                                                                      EIGE 800
 25 ANORM=0.0D0
    DO 35 I=1,N                                                       EIGE 820
    DO 35 J=I,N                                                       EIGE 830
    IF(I-J) 30,35,30                                                  EIGE 840
 30 IA=I+(J*J-J)/2                                                    EIGE 850
    ANORM=ANORM+A(IA)*A(IA)                                           EIGE 860
 35 CONTINUE                                                          EIGE 870
    IF(ANORM) 165,165,40                                              EIGE 880
 40 ANORM=1.414*DSQRT(ANORM)
    ANRMX=ANORM*RANGE/DFLOAT(N)
                                                                      EIGE 910
       INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR               EIGE 920
                                                                      EIGE 930
    IND=0                                                             EIGE 940
    THR=ANORM                                                         EIGE 950
 45 THR=THR/DFLOAT(N)
 50 L=1                                                               EIGE 970
 55 M=L+1                                                             EIGE 980
                                                                      EIGE 990
       COMPUTE SIN AND COS                                            EIGE1000
                                                                      EIGE1010
 60 MQ=(M*M-M)/2                                                      EIGE1020
    LQ=(L*L-L)/2                                                      EIGE1030
    LM=L+MQ                                                           EIGE1040
 62 IF(DABS(A(LM))-THR) 130,65,65
 65 IND=1                                                             EIGE1060
    LL=L+LQ                                                           EIGE1070
    MM=M+MQ                                                           EIGE1080
    X=0.5*(A(LL)-A(MM))                                               EIGE1090
 68 Y=-A(LM)/DSQRT(A(LM)*A(LM)+X*X)
    IF(X) 70,75,75                                                    EIGE1110
 70 Y=-Y                                                              EIGE1120
 75 SINX=Y/DSQRT(2.0*(1.0+(DSQRT(1.0-Y*Y))))
    SINX2=SINX*SINX                                                   EIGE1140
 78 COSX=DSQRT(1.0-SINX2)
    COSX2=COSX*COSX                                                   EIGE1160
    SINCS =SINX*COSX                                                  EIGE1170
                                                                      EIGE1180
       ROTATE L AND M COLUMNS                                         EIGE1190
                                                                      EIGE1200
    ILQ=N*(L-1)                                                       EIGE1210
    IMQ=N*(M-1)                                                       EIGE1220
    DO 125 I=1,N                                                      EIGE1230
    IQ=(I*I-I)/2                                                      EIGE1240
    IF(I-L) 80,115,80                                                 EIGE1250
 80 IF(I-M) 85,115,90                                                 EIGE1260
 85 IM=I+MQ                                                           EIGE1270
    GO TO 95                                                          EIGE1280
 90 IM=M+IQ                                                           EIGE1290
 95 IF(I-L) 100,105,105                                               EIGE1300
100 IL=I+LQ                                                           EIGE1310
    GO TO 110                                                         EIGE1320
105 IL=L+IQ                                                           EIGE1330
110 X=A(IL)*COSX-A(IM)*SINX                                           EIGE1340
    A(IM)=A(IL)*SINX+A(IM)*COSX                                       EIGE1350
    A(IL)=X                                                           EIGE1360
115 IF(IND-1) 120,125,120                                             EIGE1370
120 ILR=ILQ+1                                                         EIGE1380
    IMR=IMQ+I                                                         EIGE1390
    X=R(ILR)*COSX-R(IMR)*SINX                                         EIGE1400
    R(IMR)=R(ILR)*SINX+R(IMR)*COSX                                    EIGE1410
    R(ILR)=X                                                          EIGE1420
125 CONTINUE                                                          EIGE1430
    X=2.0*A(LM)*SINCS                                                 EIGE1440
    Y=A(LL)*COSX2+A(MM)*SINX2-X                                       EIGE1450
    X=A(LL)*SINX2+A(MM)*COSX2+X                                       EIGE1460
    A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)                     EIGE1470
    A(LL)=Y                                                           EIGE1480
    A(MM)=X                                                           EIGE1490
                                                                      EIGE1500
       TESTS FOR COMPLETION                                           EIGE1510
                                                                      EIGE1520
       TEST FOR M = LAST COLUMN                                       EIGE1530
                                                                      EIGE1540
130 IF(M-N) 135,140,135                                               EIGE1550
135 M=M+1                                                             EIGE1560
    GO TO 60                                                          EIGE1570
                                                                      EIGE1580
       TEST FOR L = SECOND FROM LAST COLUMN                           EIGE1590
                                                                      EIGE1600
140 IF(L-(N-1)) 145,150,145                                           EIGE1610
145 L=L+1                                                             EIGE1620
    GO TO 55                                                          EIGE1630
150 IF(IND-1) 160,155,160                                             EIGE1640
```

```
155 IND=0                                                              EIGE1650
    GO TO 50                                                           EIGE1660
                                                                       EIGE1670
       COMPARE THRESHOLD WITH FINAL NORM                               EIGE1680
                                                                       EIGE1690
160 IF(THR-ANRMX) 165,165,45                                           EIGE1700
                                                                       EIGE1710
       SORT EIGENVALUES AND EIGENVECTORS                               EIGE1720
                                                                       EIGE1730
165 IQ=-N                                                              EIGE1740
    DO 185 I=1,N                                                       EIGE1750
    IQ=IQ+N                                                            EIGE1760
    LL=I+(I*I-I)/2                                                     EIGE1770
    JQ=N*(I-2)                                                         EIGE1780
    DO 185 J=I,N                                                       EIGE1790
    JQ=JQ+N                                                            EIGE1800
    MM=J+(J*J-J)/2                                                     EIGE1810
    IF(A(LL)-A(MM)) 170,185,185                                        EIGE1820
170 X=A(LL)                                                            EIGE1830
    A(LL)=A(MM)                                                        EIGE1840
    A(MM)=X                                                            EIGE1850
    IF(MV-1) 175,185,175                                               EIGE1860
175 DO 180 K=1,N                                                       EIGE1870
    ILR=IQ+K                                                           EIGE1880
    IMR=JQ+K                                                           EIGE1890
    X=R(ILR)                                                           EIGE1900
    R(ILR)=R(IMR)                                                      EIGE1910
180 R(IMR)=X                                                           EIGE1920
185 CONTINUE                                                           EIGE1930
    RETURN                                                             EIGE1940
    END                                                                EIGE1950
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

                                 ANSBNS

      PURPOSE:  TO COMPUTE CAPTURE CROSS SECTIONS FOR PROCESSES OF THE FORM
                    B+    +   A(NS)   ->   B(N'S)   +   A+
                OR THEIR TIME REVERSED COUNTERPARTS.


      CARD 1:   NLG   (16I5)
                NUMBER OF INTEGRATION POINTS FOR GAUSS-LAGUERRE QUADRATURE
      CARD 2:   (X(IX),WX(IX),IX=1,NLG)  (4D20.15)
                ABCISSAS AND WEIGHTING FACTORS FOR GAUSS-LAGUERRE QUADRATURE
      CARD 3:   (LABEL(I),I=1,20)  (20A4)
                PROGRAM TITLE
      CARD 4:   Z1, Z2, RMAX, FMB  (8F10.5)
                Z1:  CHARGE OF THE TARGET
                Z2:  CHARGE OF THE PROJECTILE
                RMAX:  INTERNUCLEAR DISTANCE FOR WHICH REACTION BEGINS TO OCCUR
                       (FOR K-K CAPTURE RMAX=30)
                FMB:  FMB=0  IF ENERGY IS IN KEV/AMU
                FMB:  FMB= MASS OF THE PROJECTILE IF ENERGY IS IN KEV
      CARD 5:   NCODE, ICODE, JCODE, IPNCH, IRED, LIMIT, MM, IPNT1,
                IPNT2, NITP, NDAB, NCAB, IEL   (16I5)
                NCODE:  NCODE=0  NON-ITERATIVE SOLUTION OF COUPLED EQUATIONS
                        IF NCODE=0  LIMIT, MM, IPNT1, IPNT2, NITP, NCAB, NCAB
                        NEED NOT BE DEFINED
                        NCODE=1  ITERATIVE SOLUTION OF COUPLED EQUATIONS
                ICODE=1  IF INTEGRATION OVER IMPACT PARAMETERS IS PERFORMED
                JCODE=1  IF   UA, UB, XAB, XBA  ARE PRINTED
                IPNCH=1  IF MATRIX ELEMENTS ARE PUNCHED
                IRED=1  IF MATRIX ELEMENTS ARE READ IN
                LIMIT:  NUMBER OF ITERATIONS NEEDED
                        FOR  V/VE=1/2,1,2;  LIMIT=6,4,3
                MM:  10*NN+1  IS THE NUMBER OF TIME INTEGRATION POINTS
                     USED IN THE ITERATIVE SOLUTION
                IPNT1=1  FOR EVERY NITP INTERPOLATED MATRIX ELEMENTS
                         TO BE PRINTED
                IPNT2=1  FOR EVERY NDAB POINTS OF EVERY ITERATIVE SOLUTION
                         TO BE PRINTED
                NCAB:  EVERY NCAB POINTS OF THE FINAL SOLUTION ARE PRINTED IF
                       NCAB IS NOT DEFINED IT IS AUTOMATICALLY SET EQUAL TO 1
                IEL=1  FOR A ONE ELECTRON TARGET
      CARD 6:   NI, NEV, NRO, NA, NB, NT, NP   (16I5)
                NI:  6*NI+1  IS THE NUMBER OF TIME INTEGRATION POINTS
                NEV:  NUMBER OF ENERGIES
                NRO:  NUMBER OF IMPACT PARAMETERS
                NA (NB):  IS THE PRINCIPAL QUANTUM NUMBER FOR THE
                          INITIAL (FINAL) STATE AND NEED BE DEFINED ONLY
                          IF A HYDROGENIC MODEL IS USED.
                NT (NP):  NT=1 (NP=1)  IF A NON-HYDROGENIC MODEL IS USED FOR
                          THE INITIAL (FINAL) STATE. IF NT=1 (NP=1) THE 4 CARDS
                          FROM PROGRAM HERMAN CONTAINING THE INFORMATION
                          FOR THE INITIAL (FINAL) STATE MUST FOLLOW CARD 8
                          ( FOR THE FINAL STATE THE 4 CARDS FOLLOW EITHER
                          CARD 8 OR CARD 12, WHICH EVER IS LAST.)
      CARD 7:   (EINPT(I),I=1,NEV)  (8F10.5)
                ENERGY POINTS IN KEV
      CARD 8:   (EB(I),I=1,NRO)  (8F10.5)
                IMPACT PARAMETERS IN A.U.
                IF NT=1 THE INITIAL STATE INFORMATION FOLLOWS
                IF NP=1 THE FINAL STATE INFORMATION FOLLOWS

                          WRITTEN BY LAURA TUNNELL (1978)


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 SS,A11,A12,A21,A22,SBA(150)
      COMPLEX*16 SAB,HAB,HBA,CI,PRMP
      COMMON/MATRX/SAB(150),HAB(150),HBA(150),HAA(150),HBB(150)
      COMMON/IMPCT1/V,RHO,PM,T,PI,NT,NNT,NNT1
      COMMON/IMPCT2/R(150),TIME(150),Z1,Z2
      COMMON/IMPCT3/EB(20),EINPT(10),NRO,NEV,LEV,IPB
     ,,PRB(10,20),3RPB(10,20),PRMP(10,20)
      COMMON/LAGU/X(32),WX(32),NLG
      COMMON/PTNTL/DLMV(3),CV(3,3),AZ,BZ
      COMMON/WFA/EALP,ZA,ACOEF(10),AXP(10),NXA(10),NA
```

```
      COMMON/WFB/EBTA,ZB,BCOEF(10),BXP(10),NXB(10),NB
      COMMON/OPTN/LIMIT,MM,IPNCH,IREO,IPN11,IPNT2,NITP,NOAB,NCAB
      DIMENSION LABEL(20),XXX(25),YYY(25),SGMA(10),Y(4)
  100 FORMAT(1615)
  200 FORMAT(8F10.3)
  300 FORMAT(20A4)
  400 FORMAT('1',////25X,20A4///)
  249 FORMAT(4E20.15)
    3 FORMAT(10(1X,1P012.5))
      CI=OCMPLX(0.000,1.000)
      PF1=1.000
      AZ=0.000
      BZ=0.000

      READ(5,100) NLG
      READ(5,249) (X(IX),WX(IX),IX=1,NLG)
      READ(5,300) ILABEL(I),I=1,20)
      READ(5,200) Z1,Z2,RMAX.FMB
      IF(RMAX.EQ.0.000)  RMAX=40.000
      IF(FMB.EQ.0.000)  FMB=1.000
      READ(5,100)  NCODE,ICODE,JCODE,IPNCH,IREO,
     .LIMIT,MM,IPNT1,IPNT2,NITP,NCAB,NCAB,IEL
      IF(IEL.NE.0)  PF1=0.000
      IF(NCAB.EQ.0)  NCAB=5
      IF(NOAB.EQ.0)  NCAB=5
      IF(NITP.EQ.0)  NITP=5
      READ(5,100)  NI,NEV,NRO,NA,NB,NT,NP,IFILE
      READ(5,200) (EIMPT(I),I=1,NEV)
      READ(5,200) I   EB(I),I=1,NRO)
      IF(NA.EQ.0)  NA=1
      IF(NB.EQ.0)  NB=1
      PRINT 411
  411 FORMAT('  NLG,    (X(IX),WX(IX),IX=1,NLG)  ')
      PRINT 100,  NLG
      PRINT 3,    (X(IX),WX(IX),IX=1,NLG)
      PRINT 414
  414 FORMAT('     Z1,Z2,RMAX,FMB  ')
      PRINT 200,  Z1,Z2,RMAX,FMB
      PRINT 413
  413 FORMAT('      NCODE,ICODE,JCODE,IPNCH,IRED,
     .LIMIT,MM,IPNT1,IPNT2,NITP,NOAB,NCAB   ')
      PRINT 100,   NCODE,ICODE,JCODE,IPNCH,IREO,
     .LIMIT,MM,IPNT1,IPNT2,NITP,NCAB,NCAB
      PRINT 415
  415 FORMAT('      NI,NEV,NRO,NA,NB,NT,NP,IFILE  ')
      PRINT 100,   NI,NEV,NRO,NA,NB,NT,NP,IFILE
      PRINT 416
  416 FORMAT('     (EIMPT(I),I=1,NEV)                          ')
      PRINT 200,  (EIMPT(I),I=1,NEV)
      PRINT 417
  417 FORMAT('     (EB(I),I=1,NRC                ')
      PRINT 200,  (EB(I),I=1,NRC)
      FNA=OFLOAT(NA)
      FNB=OFLOAT(NB)
      EALP=Z1*Z1/FNA/FNA/2.000
      EBTA=Z2*Z2/FNB/FNB/2.000
      ACOEF(1)=1.000
      BCOEF(1)=1.000
      ACOEF(2)=-DSQRT(3.000)
      BCOEF(2)=-DSQRT(3.000)
      ACOEF(3)=DSQRT(2.5000)
      BCOEF(3)=DSQRT(2.5000)
      DO 20 I=1,3
      NXA(I)=I
      NXB(I)=I
      AXP(I)=Z1/FNA
      BXP(I)=Z2/FNB
      OLMVII)=0.000
      DO 20 J=1,3
   20 CV(I,J)=0.000
      IF(NT.EQ.0)  GO TO 22
      READ(5,200) EALP,FA,OLMV(2),(CV(I,2),I=1,3),AZ
      NA=FA
      READ(5,100) (NXA(I),I=1,NA)
      READ(5,200) (AXP(I),I=1,NA)
      READ(5,200) (ACOEF(I),I=1,NA)
   22 CONTINUE
      IF(NP.EQ.0)  GO TO 24
      READ(5,200) EBTA,FB,OLMV(3),(CV(I,3),I=1,3),BZ
      NB=FB
      READ(5,100) (NXB(I),I=1,NB)
      READ(5,200) (BXP(I),I=1,NB)
      READ(5,200) (BCOEF(I),I=1,NB)
   24 CONTINUE
      AZ=AZ+1.000
```

```
      BZ=BZ+1.000
      FA=OFLOAT(NA)
      FB=OFLOAT(NB)
      PRINT 423
  423 FORMAT('    EALP,NA,OLMV(2),(CV(I,2),I=1,3)  ')
      PRINT 200,  EALP,FA,OLMV(2),(CV(I,2),I=1,3)
      PRINT 425
  425 FORMAT('    (NXA(I),I=1,NA)       ')
      PRINT 100,  (NXA(I),I=1,NA)
      PRINT 419
  419 FORMAT('    (AXP(I),I=1,NA)       ')
      PRINT 200,  (AXP(I),I=1,NA)
      PRINT 421
  421 FORMAT('    (ACOEF(I),I=1,NA)               ')
      PRINT 200,  (ACOEF(I),I=1,NA)
      PRINT 424
  424 FORMAT('    EBTA,NB,OLMV(3),(CV(I,3),I=1,3)       ')
      PRINT 200,  EBTA,FB,OLMV(3),(CV(I,3),I=1,3)
      PRINT 418
  418 FORMAT('    (NXB(I),I=1,NB)       ')
      PRINT 100,  (NXB(I),I=1,NB)
      PRINT 420
  420 FORMAT('    (BXP(I),I=1,NB)                 ')
      PRINT 200,  (BXP(I),I=1,NB)
      PRINT 422
  422 FORMAT('    (BCOEF(I),I=1,NB)                ')
      PRINT 200,  (BCOEF(I),I=1,NB)


      PI=DARCOS(-1.000)
      TWOPI=2.000*PI*((.5290-08)**2)
      ZA=Z1
      ZB=Z2
      Z2=ZA
      IF(ZB.LT.ZA)  Z2=ZB
      Z1=Z2
      EALP=EALP/Z2/Z2
      EBTA=EBTA/Z2/Z2
      NNT=3*NI
      NNT1=NNT+1
      NT=2*NNT+1
      DO 1000 LEV=1,NEV
      EKEV=E(MPT(LEV)
      V=DSQRT(EKEV/FMB)/5.000
      SETUP TIME VECTOR
      TIME(NNT1)=0.0
      IF(NI.EQ.0) GO TO 1005
      TIME(NT)=RMAX/IZA+ZBI/V
      HT=TIME(NT)/7.000/OFLOAT(NI)
      DO 10 J=1,3
      JI=(J-1)*NI
      DO 12 I=1,NI
      JII=I
      IF(J.GT.1) JII=(*(2**(J-1))
      TIME(NNT1+J(+()=TIME(NNTI+J(I+JII*HT
   12 CONTINUE
   10 CONTINUE
      DO 14 (=1,NNT
      TIME(NNT1-I)=-TIME(NNT1+I)
   14 CONTINUE
 1005 CONTINUE
      DO 1001 IPB=1,NRO
      RHO=EB(IPBI
      R(NNTI)=RHO
      IF(N1.EQ.0) GO TO 1006
      DO 15 I=1,NNT
      R(NNTI+I)=DSQRI(RHO*RHO+V*V*TIME(NNT1+I)*TIME(NNTI+I))
      R(NNT1-I)=R(NNTI+I)
   15 CONTINUE

      IF((REQ.EQ.I) GO TO 50
 1006 CONTINUE
      CALL CMATRX
      GO TO 60
   50 CONTINUE
      DO 51 I=1,NNT1
      READ(5,52) TIME(II,HAA(I),HBB(II
      READ(5,52) SAB(I),HAB(II,HBA(I)
   52 FORMAI(6(IPD12.5))
   51 CONTINUE
   60 CONTINUE
      IF(IPNCH.NE.1)  GO TO 61
      DO 62 I=1,NNTI
      PUNCH 52,  TIME(I),HAAI(I),HBB(I)
   62 PUNCH 52,  SAB(I),HAB(II,HBA(I)
```

```
  61 CONTINUE
     DO 55 I=1,NNT1
     II=NT-(I-1)
     TIME(II)=TIME(I)*Z2*Z2
     TIME(I1)=-TIME(I)
     R(II)=R(I)*Z2
     R(I1)=R(1)
     SAB(II)=DCONJG(SAB(I))
     HAB(II)=DCONJG(HAB(I))
     HBA(II)=DCONJG(HBA(I))
     HAA(II)=HAA(I)
     HBB(II)=HBB(I)
  55 CONTINUE
     EKEV=EKEV/Z2/Z2
     V=V/Z2
     RHO=RHO*Z2
     VR=DSQRT(EKEV/EALP/FMB/27.21060-03/1836.0)
     PRINT 400,(LABEL(I),I=1,20)
     PRINT 91
  91 FORMAT('   TIME...R      SAB...HAB...HBA    HAA...HBB   '//)
     DO 92 IT=1,NNTI
  92 PRINT 93, TIME(IT),R(IT),SAB(IT),HAB(IT),HBA(IT),HAA(IT),HBB(IT)
  93 FORMAT(2(1X,F10.5),5X,3(1PD11.4,1X,1PD11.4,2X),3X,2(1PD11.4,1X))
     PRINT 94, Z2,EKEV,V,VR,RHO
  94 FORMAT(//'  SCALING FACTOR =',F6.2,' SCALED QUANTITIES ; E(KEV)=',
    .1PD11.4,' V =',1PD9.2,' V/VK =',1PD9.2,' RHO =',1PD9.2//)
     IF(NCODE.EQ.1)  CALL COUPEQ(NT)
     IF(NCODE.NE.0)  GO TO 96
     PRINT 97
  97 FORMAT('1',///' TIME...R      CA...CB...PRCB...UNIT  '//)
     Y(1)=1.000
     Y(2)=0.000
     Y(3)=0.000
     Y(4)=0.000
     CALL DIFEQ(TIME(1),(TIME(1)+TIME(2))/2.00C,Y,0.2000,4,1.0-6)
     DO 90 IT=2,NT
     CALL DEQ(TIME(I),TIME(IT),Y,HH,4,1.0-6)
     PROB=Y(3)*Y(3)+Y(4)*Y(4)
     UNIT=PROB+Y(1)*Y(1)+Y(2)*Y(2)-1.000
     UNIT=UNIT+2.000*SAB(IT)*DCMPLX(Y(1),-Y(2))*DCMPLX(Y(3),Y(4))
    ./CDEXP(CI*TIME(IT)*(EALP-EBTA))
  90 PRINT 98, TIME(IT),R(IT),(Y(I),I=1,4),PROB,UNIT
  98 FORMAT(2(2X,F10.5),6X,2(1PD10.3,1X,1PD10.3,2X),2X,2(1PD10.3,1X))
     PRB(LEV,IPB)=PROB
     BRPB(LEV,IPB)=PROB*EE(IPB)
     PRMP(LEV,IPB)=DCMPLX(Y(3),Y(4))
     PRINT 99, PRB(LEV,IPB),BRPB(LEV,IPB),EIMPT(LEV),EB(IPB)
  99 FORMAT(/' PROB =',1PD9.2,' PROB X RHO =',1PD9.2,' AT E(KEV) =',
    .F10.2,' RHO =',F8.6//)
  96 CONTINUE
     IF(JCODE.EQ.0)  GO TO 83
     PRINT 81
  81 FORMAT('1',///'TIME.....R      A11.....A12.....A21.....A22'/)
     DO 82 I=NNT1,NT
     SBA(I)=DCONJG(SAB(I))
     SS=1.000-SAB(I)*SBA(I)
     A11=-EALP+(HAA(I)-SAB(I)*HBA(I))/SS
     A12=(HAB(I)-SAB(I)*HBB(I))/SS
     A21=(HBA(I)-SBA(I)*HAA(I))/SS
     A22=-EBTA+(HBB(I)-SBA(I)*HAB(I))/SS
     PRINT 3, TIME(I),R(I),A11,A12,A21,A22
  82 CONTINUE
  83 CONTINUE
     EKEV=EKEV*Z2*Z2
     V=V*Z2
     RHO=RHO/Z2
     DO 56 I=1,NNT1
     II=NT-(I-1)
     TIME(I)=TIME(I)/Z2/Z2
     TIME(II)=-TIME(I)
     R(I)=R(1)/Z2
     R(II)=R(I)
  56 CONTINUE
1001 CONTINUE
     IF(ICODE.EQ.0) GO TO 1000
     DO 85 I=1,NRO
     XXX(I)=EB(I)
     PF2=1.000-PFI*PRB(LEV,I)
  85 YYY(I)=BRPB(LEV,I)*PF2
     CALL AREA(XXX,YYY,NRO,AA)
     SGMA(LEV)=TWOP(*AA
1000 CONTINUE
  101 FORMAT(10X,16I5)
  201 FORMAT(10X,8F10.5)
  333 FORMAT(10X,9(1X,1PD12.5))
```

```
      DO 402 IJK=I,4
      PRINT 301, (LABEL(I),I=1,20),IFILE
      PRINT 411
      PRINT 101,  NLG
      PRINT 333,  (X(IX),WX(IX),IX=1,NLG)
      PRINT 414
      PRINT 201,  ZA,ZB,RMAX,FMB
      PRINT 413
      PRINT 101,   NCODE,ICODE,JCODE,IPNCH,IREQ,
     .LIM(T,MM,IPNT1,IPNT2,NITP,NGAB,NGAB
      PRINT 415
      PRINT 101,   NI,NEV,NRO,NA,NB,NT,NP,(FILE
      PRINT 416
      PRINT 201,  (EIMPT(I),I=1,NEV)
      PRINT 417
      PRINT 201,  (EB(I),I=I,NRC)
  301 FORMAT('1',10X,20A4,'   FILE NO. =',I6)
      PRINT 401, (LABEL(I),I=I,20)
  401 FORMAT(' ',////25X,20A4///)
      DO 10000 LEV=1,NEV
      PRINT 10001
10001 FORMAT:10X,'      RHO      PROB.AMP....PROB....PROB X RHO'//)
      DO 10002 IPB=1,NRO
10002 PRINT 10003, EB(IPB),PRMP(LEV,IPB),PRB(LEV,IPB),BRPB(LEV,IPB)
10003 FORMAT(10X,F10.5,5X,2(1PD11.4,1X),2(2X,1PD11.4))
      PRINT 10004, SGMA(LEV),EIMPT(LEV)
10004 FORMAT(/20X,'  TOTAL CROSS SECTION/ELECTRON(CM**2) =',1PD9.2,
     .       '  AT E(KEV) =',0PF12.5///)
10000 CONTINUE
  402 CONTINUE
      STOP
      END
      SUBROUTINE CMATRX
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX*16 C,C1,T,X,TCOSX,TSINX
      COMPLEX*16  SABI,HABI,HBAI, SABJ,HABJ,HBAJ, SABX,HABX,HBAX
      COMPLEX*16  SABP,HABP,HBAP, SABQ,HABQ,HBAQ, SAB,HAB,HBA
      COMPLEX*16 GSAB(15,15),GHAB(15,15),GHBA(15,15)
      COMPLEX*16 FLM(3,15),GLM(3,15),PL(15),JL(15)
      COMMON/IMPCT1/V,RHO,RM,T1,P(,NT,NNT,NNT1
      COMMON/IMPCT2/RI(150),TIME(150),Z1,Z2
      COMMON/IMPCT3/EB(20),EIMPT(10),NRO,NEV,LEV,IPB
      COMMON/MATRX/SAB(150),HAB(150),HBA(150),HAA(150),HBB(150)
      COMMON/LAGU/XX(32),WX(32),NLG
      COMMON/PINTL/OLMV(3),CV(3,3),AZ,BZ,Z(3)
      COMMON/WFA/EALP,ZA,ACOEF(10),AXP(10),NXA(10),NA
      COMMON/WFB/EBTA,ZB,BCOEF(10),BXP(10),NXB(10),NB
      COMMON/CONST/A(15,15),DF(20),DG(20)
      DIMENSION VV(5,3),P(3),DPCS(3),DNEG(3),CLM(3),DPX(3),NM(3)


      NA1=NXA(NA)+1
      NB1=NXB(NB)+1
      NMX=NA1+NB1
      NM3=NMX+3
      C=DCMPLX(0.000,0.000)
      CI=DCMPLX(0.000,1.000)
      CR=DCMPLX(1.000,0.000)
      PL(1)=CR
      GMA2=V*RHO/2.000
      P(1)=0.000
      P(2)=1.000
      P(3)=-1.000
      Z(1)=0.000
      Z(2)=ZA-AZ
      Z(3)=ZB-BZ
      OLMV(1)=0.000
      DO 14 J=1,3
   14 CV(J,1)=0.000
      IF(IPB.GT.1)  GO TO 499
      IF(LEV.GT.1)  GO TO 499

      DF(1)=0.000
      DF(2)=1.000
      DG(1)=1.000
      DG(2)=1.000
      DO 15 I=2,19
      DF(I+1)=DF(I)+1.000
   15 DG(I+1)=DF(I+1)*DG(I)
      DO 16 I=1,NA
      NI=2*NXA(I)+I
      A1=(2.000*AXP(I))**NI/DG(NI)
   16 ACOEF(I)=ACOEF(I)*DSQRT(DABS(A1))
      DO 17 J=1,NB
```

```
      NJ=2*NXB(J)+1
      B1=(2.000*BXP(J))**NJ/DG(NJ)
   17 BCOEF(J)=BCOEF(J)*DSQRT(DABS(B1))

      A(1,1)=1.000
      A(2,2)=1.000/3.000
      DO 20 K=2,NM3,2
      AK=DF(K+1)
      A(K+1,1)=1.000/(AK+1.000)
      KK=K+1
      AKK=DF(KK+1)
      A(KK+1,2)=1.000/(AKK+2.000)
      DO 20 L=2,KK,2
      AL=DF(L+1)
      A(K+1,L+1)=-(2.000+AK-AL)*A(K+1,L-1)/(1.000+AK+AL)
      LL=L+1
      ALL=DF(LL+1)
      A(KK+1,LL+1)=-(2.000+AKK-ALL)*A(KK+1,LL-1)/(1.000+AKK+ALL)
   20 CONTINUE


  499 CONTINUE
      DO 500 IT=1,NNT1
      GMA1=V*V*TIME(IT)/2.000
      R2=R1(IT)/2.000
      SABI=C
      HABI=C
      HBAI=C
      DO 510 I=1,NA
      NP=NXA(I)+1
      SABJ=C
      HABJ=C
      HBAJ=C
      DO 520 J=1,NB
      NQ=NXB(J)+1
      NMM=NP+NQ+1
      OPOS1=(AXP(I)+BXP(J))*R2
      ONEG1=(AXP(I)-BXP(J))*R2
      SABX=C
      HABX=C
      HBAX=C
      DO 530 IX=1,NLG

      DO 9 M=1,3
      OPOS(M)=OPOS1+DLMV(M)*R2
      ONEG(M)=ONEG1+DLMV(M)*R2*P(M)
      DLM(M)=1.000+XX(IX)/OPOS(M)
      DX=0.000
      IF(DPOS(M).LT.1.30+02)  DX=DEXP(-OPOS(M))
      OPX(M)=DX/OPOS(M)
      DN=0.000
      IF(DABS(ONEG(M)).LT.1.30+02)  DN=DNEG(M)
      TCOSX=DCMPLX(GMA1*DLM(M),DN)
      TSINX=DCMPLX(DSQRT(DLM(M)*DLM(M)-1.0D0)*GMA2,0.0D0)
      RD=R2*DLM(M)
      ZZ=Z(M)
      VV(2,M)=ZZ*(1.0D0+RD*(CV(1,M)+RD*(CV(2,M)+RD*CV(3,M))))
      VV(3,M)=P(M)*ZZ*R2*(CV(1,M)+RD*(2.0D0*CV(2,M)+RD*3.0D0*CV(3,M)))
      VV(4,M)=ZZ*R2*R2*(CV(2,M)+RD*3.0D0*CV(3,M))
      VV(5,M)=P(M)*ZZ*R2*R2*R2*CV(3,M)
      IF((M.GT.1).AND.(DLMV(M).LE.1.0D-08)) GO TO 91
      T=CDSQRT(TSINX*TSINX+TCOSX*TCOSX)
      X=TCOSX/T
      JL(1)=CDSIN(T)/T
      JL(2)=(JL(1)-CDCOS(T))/T
      PL(2)=X
      FLM(M,1)=A(1,1)*JL(1)*PL(1)
      FLM(M,2)=A(2,2)*JL(2)*PL(2)*3.0D0*CI
      DO 10 N=2,NMM,2
      FN=DF(N+1)
      JL(N+1)=(2.000*FN-1.0D0)*JL(N)/T-JL(N-1)
      PL(N+1)=((2.000*FN-1.000)*X*PL(N)-(FN-1.0D0)*PL(N-1))/FN
      FLM(M,N+1)=A(N+1,1)*JL(1)*PL(1)
      NN=N+1
      FNN=DF(NN+1)
      JL(NN+1)=(2.000*FNN-1.000)*JL(NN)/T-JL(NN-1)
      PL(NN+1)=((2.000*FNN-1.000)*X*PL(NN)-(FNN-1.000)*PL(NN-1))/FNN
      FLM(M,NN+1)=CI*3.000*A(NN+1,2)*JL(2)*PL(2)
      DO 10 L=2,NN,2
      AL=DF(L+1)
      FLM(M,N+1)=FLM(M,N+1)+(2.000*AL+1.000)
     *           *A(N+1,L+1)*JL(L+1)*PL(L+1)
      LL=L+1
      ALL=DF(LL+1)
      FLM(M,NN+1)=FLM(M,NN+1)+CI*(2.000*ALL+1.000)
```

```
     *            *A(NN+1,LL+1)*JL(LL+1)*PL(LL+1)
  10 CONTINUE
     GO TO 9
  91 CONTINUE
     N1=NMM+1
     DO 92 N=1,N1
  92 FLM(M,N)=FLM(1,N)
   9 CONTINUE
 100 CONTINUE

     NMX=NP+NQ-2
     DO 35 K=1,NMX
     DO 36 L=1,NMX
     GSAB(K,L)=OPX(1)*FLM(1,L)
     GHAB(K,L)=AZ*GSAB(K,L)+OPX(2)*(VV(2,2)*FLM(2,L)+VV(3,2)*FLM(2,L+1)
    *          +VV(4,2)*FLM(2,L+2)+VV(5,2)*FLM(2,L+3))
     GHBA(K,L)=BZ*GSAB(K,L)+OPX(3)*(VV(2,3)*FLM(3,L)+VV(3,3)*FLM(3,L+1)
    *          +VV(4,3)*FLM(3,L+2)+VV(5,3)*FLM(3,L+3))
  36 CONTINUE
     GSAB(K,NMX+1)=OPX(1)*FLM(I,NMX+I)
     DO 37 L=1,3
  37 OPX(L)=QLM(L)*OPX(L)
  35 CONTINUE
     NMX=NMX+1
     DO 38 L=1,NMX
  38 GSAB(NMX,L)=OPX(I)*FLM(1,L)



     SABP=C
     HABP=C
     HBAP=C
     DO 540 IP=1,NP
     SABQ=C
     HABQ=C
     HBAQ=C
     QI=1.000
     DO 550 IQ=1,NQ
     NI=NXA(I)+NXB(J)-IP-IQ
     N2=IP+IQ
     CCQ=DG(NQ)/DG(IQ)/DG(NQ-IQ+1)*QI
     SABQ=SABQ+GSAB((NI+2)+1,(N2-2)+1)*CCQ
     IF(NI.EQ.-2) GO TO 551
     HABQ=HABQ+GHAB((NI+1)+1,(N2-2)+1)*CCQ
     HBAQ=HBAQ+GHBA((NI+1)+1,(N2-2)+1)*CCQ*(DF(NQ)-DF(IQ))/DF(NQ)
 551 CONTINUE
 550 QI=-QI
     CCP=DG(NP)/DG(IP)/DG(NP-IP+1)
     SABP=SABP+SABQ*CCP
     HABP=HABP+HABQ*CCP*(DF(NP)-DF(IP))/DF(NP)
     HBAP=HBAP+HBAQ*CCP
 540 CONTINUE
     SABX=SABX+SABP*WX(IX)
     HABX=HABX+HABP*WX(IX)
     HBAX=HBAX+HBAP*WX(IX)
 530 CONTINUE
     CCJ=BCOEF(J)*(R2**(NXA(I)+NXB(J)))
     SABJ=SABJ+SABX*CCJ
     HABJ=HABJ+HABX*CCJ
     HBAJ=HBAJ+HBAX*CCJ
 520 CONTINUE
     CCI=ACOEF(I)
     SABI=SABI+SABJ*CCI
     HABI=HABI+HABJ*CCI
     HBAI=HBAI+HBAJ*CCI
 510 CONTINUE
     SAB(IT)=SABI*R2
     HAB(IT)=-HABI/Z2/Z2
     HBA(IT)=DCONJG(-HBAI)/Z2/Z2
 500 CONTINUE
                .

     CALL DIAG(ACOEF,AXP,NXA,NA,3,HAA)
     CALL DIAG(BCOEF,BXP,NXB,NB,2,HBB)



     RETURN
     END

     SUBROUTINE DIAG(COEF,CXP,NXC,NC,NV,HH)
     IMPLICIT REAL*8(A-H,O-Z)
     COMMON/IMPCTI/V,RHO,RM,T1,PI,NT,NNT,NNT1
     COMMON/IMPCT2/R1(150),TIME(150),Z1,Z2
     COMMON/PTNTL/OLMV(3),VC(3,3),AZ,BZ,Z(3)
```

```
      COMMON/CCNST/W(15,15),OF(20),OG(20)
      DIMENSION COEF(NC),CXP(NC),NXC(NC)
      DIMENSION F(20),FF(20),G(20),GG(20)
      DIMENSION CV(3),HH(150)

      C=0.000
      ZZ1=BZ
      IF(NV.EQ.2)  ZZ1=AZ
      ZZ2=Z(NV)
      OLM=OLMV(NV)
      CV(1)=VC(1,NV)
      CV(2)=VC(2,NV)
      CV(3)=VC(3,NV)
      DO 100 IT=1,NNT1
      R2=R1(IT)/2.000
      HI=C
      DO 200 I=1,NC
      HJ=C
      NI=NXC(I)
      NII=NI+1
      DO 300 J=1,NC
      HP=C
      NJ=NXC(J)
      NH=NI+NJ
      NM3=NM+3+1
      A=(CXP(I)+CXP(J))*R2
      B=A
      AA=A+OLM*R2
      BB=B-OLM*R2
      GX=C
      GGX=C
      FFX=C
      IF((2.000*B).LE.1.200+02)  GX=DEXPI-2.000*B)
      IF((2.000*BB).LE.1.200+02)  GGX=DEXP(-2.0D0*BB)
      IF((BB-AA).LE.1.00+02)  FFX=DEXP(BB-AA)
      F(I)=I.0C0/A
      FF(1)=FFX/AA
      G(1)=(1.000-GX)/B
      GG(I)=(1.0D0-GGX)/BB
      GM=-I.000
      NM3=NM3+2
      DO 13 K=2,NM3
      F(K)=(OF(K)*F(K-I)+I.0D0)/A
      FF(K)=(OF(K)*FF(K-I)+FFX)/AA
      G(K)=(OF(K)*G(K-1)+GM-GX)/B
      GG(K)=(OF(K)*GG(K-I)+GM-GGX)/BB
   13 GM=-GM
      DO 400 IP=I,NII
      HQ=C
      DO 500 IQ=1,NJ
      N=NM+2-IP-IQ
      N=IP+IQ-I
      QO=FF(N)*GG(M)
      Q1=CV(1)*(FF(N+1)*GG(M)-FF(N)*GG(M+I))
      Q2=CV(2)*(FF(N+2)*GG(M)-2.0D0*FF(N+1)*GG(M+I)+FF(N)*GG(M+2))
      Q3=CV(3)*(FF(N+3)*GG(M)-3.0D0*(FF(N+2)*GG(M+1)
     *   -FF(N+1)*GG(M+2))-FF(N)*GG(M+3))
      Q=ZZI*F(N)*G(M)+ZZ2*(QO+R2*(Q1+R2*(Q2+R2*Q3)))
      CQ=DGINJ)/DGI IQ)/DGINJ-IO+1)
      HQ=HQ+O*CQ
  500 CONTINUE
      CP=DG(NII)/DG(IP)/DGINI-IP+2)
      HP=HP+HQ*CP
  400 CONTINUE
      CJ=COEF(J)*(R2**NM)
      HJ=HJ+HP*CJ
  300 CONTINUE
      CI=COEF(I)
      HI=HI+HJ*CI
  200 CONTINUE
      HH(IT)=-HI/2.000/Z2/Z2
  100 CONTINUE

      RETURN
      END

      SUBROUTINE AREA(X,Y,N,AA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(25),Y(25)
      INDEX=0
      AA=0.
      NN=N+I
      Y(NN)=0.
      X(NN)=2*X(N)-X(N-1)
      X1=0.
```

```
      Y1=0.
      I=0
   70 CONTINUE
      X2=X(I+1)
      Y2=Y(I+1)
      X3=X(I+2)
      Y3=Y(I+2)
      D1=(Y1-Y2)/(X1-X2)
      D2=(Y2-Y3)/(X2-X3)
      A=(D1-D2)/(X1-X3)
      B=D1-A*(X1+X2)
      C=Y1-A*X1*X1-B*X1
      IF(INDEX.NE.0) X1=X2
      AA=AA+A*(X3**3-X1**3)/3+B*(X3*X3-X1*X1)/2+C*(X3-X1)
      IF((I+2).GE.NN) RETURN
      X1=X3
      Y1=Y3
      I=I+2
      IF((I+2).GT.NN) GO TO 80
      GO TO 70
   80 I=I-1
      X1=X2
      Y1=Y2
      INDEX=1
      GO TO 70
      END
      SUBROUTINE DER(T,Y,YP)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 SS,A,B,ADOT,BDOT,ENT,CMI
      COMPLEX*16 S12,S21,H12,H21,SAB,SBA,HAB,HBA
      COMMON/MATRX/S12(150),H12(150),H21(150),H11(150),H22(150)
      COMMON/IMPCT1/V,RHC,RM,TZ,PI,NT,NNT,NNT1
      COMMON/WFA/EALP,ZA
      COMMON/WFB/EBTA,ZB
      COMMON/IMPCT2/R(150),TIME(150)
      DIMENSION Z(150),ARG( 5),VAL( 5),Y(4),YP(4)

      CMI=DCMPLX(0.0D0,-1.0D0)
      DO 301 I=1,NT
  301 Z(I)=S12(I)
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DALI(T,ARG,VAL,Z1,5,1.E-5,IER)
      DO 302 I=1,NT
  302 Z(I)=S12(I)*CMI
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DALI(T,ARG,VAL,Z2,5,1.E-5,IER)
      SAB=DCMPLX(Z1,Z2)
      DO 303 I=1,NT
  303 Z(I)=H12(I)
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DALI(T,ARG,VAL,Z1,5,1.E-5,IER)
      DO 304 I=1,NT
  304 Z(I)=H12(I)*CMI
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DALI(T,ARG,VAL,Z2,5,1.E-5,IER)
      HAB=DCMPLX(Z1,Z2)
      DO 305 I=1,NT
  305 Z(I)=H21(I)
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DALI(T,ARG,VAL,Z1,5,1.E-5,IER)
      DO 306 I=1,NT
  306 Z(I)=H21(I)*CMI
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DALI(T,ARG,VAL,Z2,5,1.E-5,IER)
      HBA=DCMPLX(Z1,Z2)
      DO 307 I=1,NT
  307 Z(I)=H11(I)
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DALI(T,ARG,VAL,Z1,5,1.E-5,IER)
      HAA=Z1
      DO 308 I=1,NT
  308 Z(I)=H22(I)
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DALI(T,ARG,VAL,Z2,5,1.E-5,IER)
      HBB=Z2
      SBA=DCONJG(SAB)
      ENT=DCMPLX(0.0D0,T*(EBTA-EALP))
      SS=(1.000-SAB*SBA)*(0.000,1.000)
      SS=1.000/SS
      A=DCMPLX(Y(1),Y(2))
      B=DCMPLX(Y(3),Y(4))
      ADOT=SS*(A*(HAA-SAB*HBA)+B*(HAB-SAB*HBB)*CDEXP(ENT))
      BDOT=SS*(B*(HBB-SBA*HAB)+A*(HBA-SBA*HAA)*CDEXP(-ENT))
      YP(1)=ADOT
      YP(2)=ADOT*CMI
```

```
      YP(3)=BOOT
      YP(4)=BOCI*CMI
      RETURN
      END
      SUBROUTINE COUPEQ(N)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 GG,SUM,A,X,ENT,U,CMI,PRMP
      COMPLEX*16 XAB(600),XBA(600)
      COMPLEX*16 DA(600),DB(600),DELTA(600)
      COMPLEX*16 S12,S21,H12,H21
      COMPLEX*16 SAB(600),SBA(600),HAB(600),HBA(600)
      COMPLEX*16 ALP(600),BTA(600),AA(600),BB(600),CA(600),CB(600)
      COMPLEX*16 Y1,Y2,Y3
      COMMON/MATRX/S12(150),H12(150),H21(150),H11(150),H22(150)
      COMMON/OPTN/LIMIT,MM,IPNCH,IRED,IPNT1,IPNT2,NITP,NOAB,NCAB
      DIMENSION HAA(600),HBB(600)
      DIMENSION ARG(10),VAL(10),TIME(600),R(600),Z(150)
      COMMON/IMPCT1/V,RHO
      COMMON/IMPCT2/RA(150),T(150)
      COMMON/IMPCT3/EB(20),EIMPT(10),NRO,NEV,LEV,IPB
     ,,PRB(10,20),BRPB(10,20),PRMP(10,20)
      COMMON/WFA/EALP,ZA
      COMMON/WFB/EBTA,ZB


      CMI=DCMPLX(0.000,-1.000)
      SET UP TIME VECTOR
      T1=T(I)
      H=-16.000*T1/(31.000*MM)
      K=0
      MM1=MM+1
   92 CONTINUE
      DO 91 I=1,MM1
   91 TIME(K*MM+I)=T1+(I-1)*H
      K=K+1
      IF(K.GT.4) GO TO 93
      T1=TIME(K*MM+1)
      H=H/2.000
      GO TO 92
   93 CONTINUE
      NNT=5*MM
      NNT1=NNT+1
      NT=2*NNT+1
      DO 98 I=1,NNT1
      I1=NT-(I-1)
      TIME(I1)=-TIME(I)
      R(I)=DSQRT(RHO**2+(V*TIME(I))**2)
      R(I1)=R(I)
   98 CONTINUE
      EKEV=25*V*V

      DO 500 K=1,NNT1
      DO 301 I=1,N
  301 Z(I)=S12(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DALI (TIME(K),ARG,VAL,Z1,5,1.E-5,IER)
      DO 302 I=1,N
  302 Z(I)=S12(I)*CMI
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DALI (TIME(K),ARG,VAL,Z2,5,1.E-5,IER)
      SAB(K)=DCMPLX(Z1,Z2)
      DO 303 I=1,N
  303 Z(I)=H12(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DALI (TIME(K),ARG,VAL,Z1,5,1.E-5,IER)
      DO 304 I=1,N
  304 Z(I)=H12(I)*CMI
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DALI (TIME(K),ARG,VAL,Z2,5,1.E-5,IER)
      HAB(K)=DCMPLX(Z1,Z2)
      DO 305 I=1,N
  305 Z(I)=H21(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DALI (TIME(K),ARG,VAL,Z1,5,1.E-5,IER)
      DO 306 I=1,N
  306 Z(I)=H21(I)*CMI
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DALI (TIME(K),ARG,VAL,Z2,5,1.E-5,IER)
      HBA(K)=DCMPLX(Z1,Z2)
      DO 307 I=1,N
  307 Z(I)=H11(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DALI (TIME(K),ARG,VAL,Z1,5,1.E-5,IER)
      HAA(K)=Z1
      DO 308 I=1,N
```

```
308 Z(I)=H22(I)
    CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
    CALL DALI (TIME(K),ARG,VAL,Z2,5,1.E-5,IER)
    HBB(K)=Z2
500 CONTINUE

    IF(IPNT1.EQ.0) GO TO 11
    PRINT 520
520 FORMAT(' ',////50X,'INTERPOLATED MATRIX ELEMENTS'///)
    PRINT 521
521 FORMAT('   TIME...R   SAB...HAB...HBA...HAA...HBB  '//)
    DO 507 I=1,NNTI,NITP
507 PRINT 508,TIME(I),R(I),SAB(I),HAB(I),HBA(I),HAA(I),HBB(I)
508 FORMAT(2(1X,F10.5),5X,4(1PD11.4,1X,1PD11.4,2X))
 11 CONTINUE

    DO 600 I=1,NNT1
    II=NT-(I-1)
    SAB(II)=DCONJG(SAB(I))
    SBA(I)=DCONJG(SAB(I))
    SBA(II)=DCONJG(SBA(I))
    HAB(II =DCONJG(HAB(I))
    HBA(II)=DCONJG(HBA(I))
    HAA(II)=HAA(I)
    HBB(II)=HBB(I)

    SS=1.000-SAB(I)*SBA(I)
    AA(I)=(HAA(I)-SAB(I)*HBA(I))/SS
    AA(II)=DCONJG(AA(I))
    BB(I)=(HBB(I)-SBA(I)*HAB(I))/SS
600 BB(II)=DCONJG(BB(I))


    NT1=NT-1
    DELTA(1)=(0.000,0.000)
    ALP(1)=(0.000,0.000)
    BTA(1)=(0.000,0.000)
    DO 700 J=2,NT1
    X1=TIME(J-1)
    X2=TIME(J)
    X3=TIME(J+1)
    Y1=AA(J-1)
    Y2=AA(J)
    Y3=AA(J+1)
    ALP(J)=ALP(J-1)+SUM(X1,X2,X3,Y1,Y2,Y3)*
    Y1=BB(J-1)
    Y2=BB(J)
    Y3=BB(J+1)
    BTA(J)=BTA(J-1)+SUM(X1,X2,X3,Y1,Y2,Y3)
700 DELTA(J)=ALP(J)-BTA(J)
    X3=TIME(NT)
    X2=TIME(NT1)
    X1=TIME(NT1-1)
    Y1=AA(NT)
    Y2=AA(NT1)
    Y3=AA(NT1-1)
    ALP(NT)=ALP(NT1)+SUM(X1,X2,X3,Y1,Y2,Y3)
    Y1=BB(NT)
    Y2=BB(NT1)
    Y3=BB(NT1-1)
    BTA(NT)=BTA(NT1)+SUM(X1,X2,X3,Y1,Y2,Y3)
    DELTA(NT)=ALP(NT)-BTA(NT)

    DO 800 K=1,NT
    DELTA(K)=DELTA(K)-TIME(K)*(EALP-EBTA)
    SS=1.000-SAB(K)*SBA(K)
    Y3=CDEXP(DELTA(K)/CM1)
    XAB(K)=(HAB(K)-SAB(K)*HBB(K))*CM1*Y3/SS
    XBA(K)=(HBA(K)-SBA(K)*HAA(K))*CM1/Y3/SS
    DA(K)=(1.000,0.000)
800 DB(K)=(0.000,0.000)


    SS3=0.000
    DO 1000 K1=1,LIMIT
    DA(1)=(1.000,0.000)
    DB(1)=(0.000,0.000)
    DO 900 J=2,NT1
    X1=TIME(J-1)
    X2=TIME(J)
    X3=TIME(J+1)
    Y1=XBA(J-1)*DA(J-1)
    Y2=XBA(J)*DA(J)
    Y3=XBA(J+1)*DA(J+1)
900 DB(J)=DB(J-1)+SUM(X1,X2,X3,Y1,Y2,Y3)
```

```
      X3=TIME(NT)
      X2=TIME(NT1)
      X1=TIME(NT1-1)
      Y1=XBA(NT)*DA(NT)
      Y2=XBA(NT1)*DA(NT1)
      Y3=XBA(NT1-1)*DA(NT1-1)
      DB(NT)=DB(NT1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      DO 950 J=2,NT1
      X1=TIME(J-1)
      X2=TIME(J)
      X3=TIME(J+1)
      Y1=XAB(J-1)*DB(J-1)
      Y2=XAB(J)*DB(J)
      Y3=XAB(J+1)*DB(J+1)
  950 DA(J)=DA(J-1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      X3=TIME(NT)
      X2=TIME(NT1)
      X1=TIME(NT1-1)
      Y1=XAB(NT)*DB(NT)
      Y2=XAB(NT1)*DB(NT1)
      Y3=XAB(NT1-1)*DB(NT1-1)
      DA(NT)=DA(NT1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      IF(IPMT2.EQ.0)  GO TO 13
      PRINT 21, KI
   21 FORMAT('1    ITERATION=',I3/)
      PRINT 125
  125 FORMAT(4X,4X,'TIME',25X,'DA',30X,'DB'/)
      DO 22 I=1,NT,NOAB
   22 PRINT 23, TIME(I),DA(I),DB(I)
   23 FORMAT(2X,F12.6,4X,4E16.7)
   13 CONTINUE

      SS1=0.000
      SS2=0.000
      DO 960 I=1,NT
      SS1=SS1+CDABS(DA(I)-DB(I))
  960 SS2=SS2+CDABS(DB(I))
      SS3=SS1/SS2
 1000 CONTINUE


 2000 CONTINUE
      PRINT 3
    3 FORMAT(///' R    UA..UB  XAB..XBA    CA..CB..PROB..UNITARITY'//)
      DO 50 I=1,NT,NCAB
      CA(I)=DA(I)*CDEXP((0.000,-1.000)*ALP(I))
      CB(I)=DB(I)*CDEXP((0.000,-1.000)*BTA(I))
      PROB=CB(I)*DCONJG(CB(I))
      ENN=TIME(I)*(EALP-EBTA)
      ENT=DCMPLX(0.000,-ENN)
      UNIT=PROB+CA(I)*DCONJG(CA(I))
      U=DCONJG(CA(I))*CB(I)*SAB(I)*CDEXP(ENT)
      UNIT=UNIT+U+DCONJG(U)
      UNIT=1.DCO-UNIT
      SS=1.000-SAB(I)*SBA(I)
      X1=(HAA(I)-SAB(I)*HBA(I))/SS-EALP
      X2=(HBB(I)-SBA(I)*HAB(I))/SS-EBTA
      Y1=(HAB(I)-SAB(I)*HBB(I))/SS
      Y2=(HBA(I)-SBA(I)*HAA(I))/SS
   50 PRINT 60, R(I),X1,X2,Y1,Y2,CA(I),CB(I),PROB,UNIT
   60 FORMAT(1X,F9.3,12(1X,1PD9.2))


      BP=PROB*EB(IPB)
      PRINT 229,BP,EB(IPB),EIMPT(LEV)
  229 FORMAT('/'     PROB*RHO=',E12.5,'  AT RHO=',F8.4,'   EKEV=',F10.3/)
      PRMP(LEV,IPB)=CB(NT)
      PRB(LEV,IPB)=PROB
      BRPB(LEV,IPB)=BP
      RETURN
      END
      SUBROUTINE DATSM

                                                                      DTSM  10
     .................................................................DTSM  20
                                                                      DTSM  30
        SUBROUTINE DATSM                                              DTSM  50
                                                                      DTSM  60
        PURPOSE                                                       DTSM  70
           NOIM POINTS OF A GIVEN TABLE WITH MONOTONIC ARGUMENTS ARE  DTSM  80
           SELECTED AND ORDERED SUCH THAT                             DTSM  90
           ABS(ARG(I)-X).GE.ABS(ARG(J)-X) IF I.GT.J.                  DTSM 100
                                                                      DTSM 110
        USAGE                                                         DTSM 120
           CALL DATSM (X,Z,F,IROW,ICOL,ARG,VAL,NO(M)
```

```
C                                                                       OTSM 130
C     DESCRIPTION OF PARAMETERS                                         OTSM 140
C        X      - DOUBLE PRECISION SEARCH ARGUMENT.                     OTSM 150
C        Z      - DOUBLE PRECISION VECTOR OF ARGUMENT VALUES (DIMEN-    OTSM 160
C                 SION IROW). THE ARGUMENT VALUES MUST BE STORED IN     OTSM 170
C                 INCREASING OR DECREASING SEQUENCE.                    OTSM 180
C        F      - IN CASE ICOL=1, F IS THE DOUBLE PRECISION VECTOR      OTSM 190
C                 OF FUNCTION VALUES (DIMENSION IROW).                  OTSM 200
C                 IN CASE ICOL=2, F IS A DOUBLE PRECISION IROW BY 2     OTSM 210
C                 MATRIX. THE FIRST COLUMN SPECIFIES VECTOR OF FUNC-    OTSM 220
C                 TION VALUES AND THE SECOND VECTOR OF DERIVATIVES.     OTSM 230
C        IROW   - THE DIMENSION OF VECTOR Z AND OF EACH COLUMN          OTSM 240
C                 IN MATRIX F.                                          OTSM 250
C        ICOL   - THE NUMBER OF COLUMNS IN F (I.E. 1 OR 2).             OTSM 260
C        ARG    - RESULTING DOUBLE PRECISION VECTOR OF SELECTED AND     OTSM 270
C                 ORDERED ARGUMENT VALUES (DIMENSION NDIM).             OTSM 280
C        VAL    - RESULTING DOUBLE PRECISION VECTOR OF SELECTED         OTSM 290
C                 FUNCTION VALUES (DIMENSION NDIM) IN CASE ICOL=1.      OTSM 300
C                 IN CASE ICOL=2, VAL IS THE DOUBLE PRECISION VECTOR    OTSM 310
C                 OF FUNCTION AND DERIVATIVE VALUES (DIMENSION          OTSM 320
C                 2*NDIM) WHICH ARE STORED IN PAIRS (I.E. EACH FUNC-    OTSM 330
C                 TION VALUE IS FOLLOWED BY ITS DERIVATIVE VALUE).      OTSM 340
C        NDIM   - THE NUMBER OF POINTS WHICH MUST BE SELECTED OUT OF    OTSM 350
C                 THE GIVEN TABLE (Z,F).                                OTSM 360
C                                                                       OTSM 370
C     REMARKS                                                           OTSM 380
C        NO ACTION IN CASE IROW LESS THAN 1.                            OTSM 390
C        IF INPUT VALUE NDIM IS GREATER THAN IROW, THE PROGRAM          OTSM 400
C        SELECTS ONLY A MAXIMUM TABLE OF IROW POINTS.  THEREFORE THE    OTSM 410
C        USER OUGHT TO CHECK CORRESPONDENCE BETWEEN TABLE (ARG,VAL)     OTSM 420
C        AND ITS DIMENSION BY COMPARISON OF NDIM AND IROW, IN ORDER     OTSM 430
C        TO GET CORRECT RESULTS IN FURTHER WORK WITH TABLE (ARG,VAL).   OTSM 440
C        THIS TEST MAY BE DONE BEFORE OR AFTER CALLING                  OTSM 450
C        SUBROUTINE DATSM.                                              OTSM 460
C        SUBROUTINE DATSM ESPECIALLY CAN BE USED FOR GENERATING THE     OTSM 470
C        TABLE (ARG,VAL) NEEDED IN SUBROUTINES DALI, DAHI, AND DACFI.   OTSM 480
C                                                                       OTSM 490
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                     OTSM 500
C        NONE                                                           OTSM 510
C                                                                       OTSM 520
C     METHOD                                                            OTSM 530
C        SELECTION IS DONE BY SEARCHING THE SUBSCRIPT J OF THAT         OTSM 540
C        ARGUMENT, WHICH IS NEXT TO X (BINARY SEARCH).                  OTSM 550
C        AFTERWARDS NEIGHBOURING ARGUMENT VALUES ARE TESTED AND         OTSM 560
C        SELECTED IN THE ABOVE SENSE.                                   OTSM 570
C                                                                       OTSM 580
C     ..................................................................OTSM 590
C                                                                       OTSM 600
C                                                                       OTSM 620
C                                                                       OTSM 630
      SUBROUTINE DATSM(X,Z,F,IROW,ICOL,ARG,VAL,NDIM)
      DOUBLE PRECISION X,Z,F,ARG,VAL
      DIMENSION Z(IROW),F(IROW),ARG(NDIM),VAL(NDIM)
                                                                        OTSM 660
C     CASE IROW=1 IS CHECKED OUT
      IF(IROW-1)23,21,1
    1 N=NDIM                                                            OTSM 690
                                                                        OTSM 700
C     IF N IS GREATER THAN IROW, N IS SET EQUAL TO IROW.                OTSM 710
      IF(N-IROW)3,3,2
    2 N=IROW                                                            OTSM 730
                                                                        OTSM 740
C     CASE IROW.GE.2                                                    OTSM 750
C     SEARCHING FOR SUBSCRIPT J SUCH THAT Z(J) IS NEXT TO X.            OTSM 760
    3 IF(Z(IROW)-Z(1))5,4,4
    4 J=IROW                                                            OTSM 780
      I=1                                                               OTSM 790
      GOTO 6                                                            OTSM 800
    5 I=IROW                                                            OTSM 810
      J=I                                                               OTSM 820
    6 K=(J+I)/2                                                         OTSM 830
      IF(X-Z(K))7,7,8                                                   OTSM 840
    7 J=K                                                               OTSM 850
      GOTO 9                                                            OTSM 860
    8 I=K                                                               OTSM 870
    9 IF(IABS(J-I)-1)10,10,6                                            OTSM 880
   10 IF(DABS(Z(J)-X)-DABS(Z(I)-X))12,12,11                             OTSM 890
   11 J=I                                                               OTSM 900
                                                                        OTSM 910
C     TABLE SELECTION                                                   OTSM 920
   12 K=J                                                               OTSM 930
      JL=0                                                              OTSM 940
      JR=0                                                              OTSM 950
      DO 20 I=1,N                                                       OTSM 960
      ARG(I)=Z(K)                                                       OTSM 970
```

```
   IF(ICOL-1)14,14,13                                                   DTSM 980
13 VAL(2*I-1)=F(K)                                                      DTSM 990
   KK=K+IROW                                                            DTSM1000
   VAL(2*I)=F(KK)                                                       DTSM1010
   GOTO 15                                                              DTSM1020
14 VAL(1)=F(K)                                                          DTSM1030
15 JJR=J+JR                                                             DTSM1040
   IF(JJR-IROW)16,18,18                                                 DTSM1050
16 JJL=J-JL                                                             DTSM1060
   IF(JJL-1)19,19,17                                                    DTSM1070
17 IF(DABS(Z(JJR+I)-X)-DABS(Z(JJL-1)-X))19,19,18                        DTSM1080
18 JL=JL+1                                                              DTSM1090
   K=J-JL                                                               DTSM1100
   GOTO 20                                                              DTSM1110
19 JR=JR+1                                                              DTSM1120
   K=J+JR                                                               DTSM1130
20 CONTINUE                                                             DTSM1140
   RETURN                                                               DTSM1150
                                                                        DTSM1160
   CASE IROW=I                                                          DTSM1170
21 ARG(1)=Z(1)                                                          DTSM1180
   VAL(1)=F(1)                                                          DTSM1190
   IF(ICOL-2)23,22,23                                                   DTSM1200
22 VAL(2)=F(2)                                                          DTSM1210
23 RETURN                                                               DTSM1220
   END                                                                  DTSM1230
                                                                        DALI  10
C*******************************************************************DALI  20
C                                                                       DALI  30
C     SUBROUTINE DALI                                                   DALI  40
C                                                                       DALI  50
C     PURPOSE                                                           DALI  60
C        TO INTERPOLATE FUNCTION VALUE Y FOR A GIVEN ARGUMENT VALUE     DALI  70
C        X USING A GIVEN TABLE (ARG,VAL) OF ARGUMENT AND FUNCTION       DALI  80
C        VALUES.                                                        DALI  90
C                                                                       DALI 100
C     USAGE                                                             DALI 110
C        CALL DALI (X,ARG,VAL,Y,NDIM,EPS,IER)                           DALI 120
C                                                                       DALI 130
C     DESCRIPTION OF PARAMETERS                                         DALI 140
C        X      - DOUBLE PRECISION ARGUMENT VALUE SPECIFIED BY INPUT.   DALI 150
C        ARG    - DOUBLE PRECISION INPUT VECTOR (DIMENSION NDIM) OF     DALI 160
C                 ARGUMENT VALUES OF THE TABLE (NOT DESTROYED).         DALI 170
C        VAL    - DOUBLE PRECISION INPUT VECTOR (DIMENSION NDIM) OF     DALI 180
C                 FUNCTION VALUES OF THE TABLE (DESTROYED).             DALI 190
C        Y      - RESULTING INTERPOLATED DOUBLE PRECISION FUNCTION      DALI 200
C                 VALUE.                                                DALI 210
C        NDIM   - AN INPUT VALUE WHICH SPECIFIES THE NUMBER OF          DALI 220
C                 POINTS IN TABLE (ARG,VAL).                            DALI 230
C        EPS    - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS      DALI 240
C                 UPPER BOUND FOR THE ABSOLUTE ERROR.                   DALI 250
C                 FOR THE ABSOLUTE ERROR.                               DALI 260
C        IER    - A RESULTING ERROR PARAMETER.                          DALI 270
C                                                                       DALI 280
C     REMARKS                                                           DALI 290
C        (1) TABLE (ARG,VAL) SHOULD REPRESENT A SINGLE-VALUED           DALI 300
C            FUNCTION AND SHOULD BE STORED IN SUCH A WAY, THAT THE      DALI 310
C            DISTANCES ABS(ARG(I)-X) INCREASE WITH INCREASING           DALI 320
C            SUBSCRIPT I. TO GENERATE THIS ORDER IN TABLE (ARG,VAL),    DALI 330
C            SUBROUTINES DATSG, DATSM OR DATSE COULD BE USED IN A       DALI 340
C            PREVIOUS STAGE.                                            DALI 350
C        (2) NO ACTION BESIDES ERROR MESSAGE IN CASE NDIM LESS          DALI 360
C            THAN 1.                                                    DALI 370
C        (3) INTERPOLATION IS TERMINATED EITHER IF THE DIFFERENCE       DALI 380
C            BETWEEN TWO SUCCESSIVE INTERPOLATED VALUES IS              DALI 390
C            ABSOLUTELY LESS THAN TOLERANCE EPS, OR IF THE ABSOLUTE     DALI 400
C            VALUE OF THIS DIFFERENCE STOPS DIMINISHING, OR AFTER       DALI 410
C            (NDIM-1) STEPS. FURTHER IT IS TERMINATED IF THE            DALI 420
C            PROCEDURE DISCOVERS TWO ARGUMENT VALUES IN VECTOR ARG      DALI 430
C            WHICH ARE IDENTICAL. DEPENDENT ON THESE FOUR CASES,        DALI 440
C            ERROR PARAMETER IER IS CODED IN THE FOLLOWING FORM         DALI 450
C              IER=0 - IT WAS POSSIBLE TO REACH THE REQUIRED            DALI 460
C                      ACCURACY (NO ERROR).                             DALI 470
C              IER=1 - IT WAS IMPOSSIBLE TO REACH THE REQUIRED          DALI 480
C                      ACCURACY BECAUSE OF ROUNDING ERRORS.             DALI 490
C              IER=2 - IT WAS IMPOSSIBLE TO CHECK ACCURACY BECAUSE      DALI 500
C                      NDIM IS LESS THAN 3, OR THE REQUIRED ACCURACY    DALI 510
C                      COULD NOT BE REACHED BY MEANS OF THE GIVEN       DALI 520
C                      TABLE. NDIM SHOULD BE INCREASED.                 DALI 530
C              IER=3 - THE PROCEDURE DISCOVERED TWO ARGUMENT VALUES     DALI 540
C                      IN VECTOR ARG WHICH ARE IDENTICAL.               DALI 550
C                                                                       DALI 560
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                     DALI 570
C        NONE                                                           DALI 580
C                                                                       DALI 590
```

```
      METHOD                                                            DALI 600
         INTERPOLATION IS DONE BY MEANS OF AITKENS SCHEME OF            DALI 610
         LAGRANGE INTERPOLATION. ON RETURN Y CONTAINS AN INTERPOLATED   DALI 620
         FUNCTION VALUE AT POINT X, WHICH IS IN THE SENSE OF REMARK     DALI 630
         (3) OPTIMAL WITH RESPECT TO GIVEN TABLE. FOR REFERENCE, SEE    DALI 640
         F.B.HILDEBRAND, INTRODUCTION TO NUMERICAL ANALYSIS,            DALI 650
         MCGRAW-HILL, NEW YORK/TORONTO/LONDON, 1956, PP.49-50.          DALI 660
                                                                        DALI 670
     .....DALI 680
                                                                        DALI 690
      SUBROUTINE DALI(X,ARG,VAL,Y,NDIM,EPS,IER)
                                                                        DALI 710
                                                                        DALI 720
      DOUBLE PRECISION ARG,VAL,X,Y,H
      DIMENSION ARG(NDIM),VAL(NDIM)
      IER=2                                                             DALI 750
      DELT2=0.                                                          DALI 760
      IF(NDIM-1)9,7,1                                                   DALI 770
                                                                        DALI 780
      START OF AITKEN-LOOP                                              DALI 790
    1 DO 6 J=2,NDIM                                                     DALI 800
      DELT1=DELT2                                                       DALI 810
      IEND=J-1                                                          DALI 820
      DO 2 I=I,IEND                                                     DALI 830
      H=ARG(I)-ARG(J)                                                   DALI 840
      IF(DABS(H).LT.1.D0-6) GO TO 13                                    DALI 850
    2 VAL(J)=(VAL(I)*(X-ARG(J))-VAL(J)*(X-ARG(I)))/H                    DALI 860
      DELT2=DABS(VAL(J)-VAL(IEND))                                      DALI 870
      IF(J-2)6,6,3                                                      DALI 880
    3 IF(DELT2-EPS)10,10,4                                              DALI 890
    4 IF(J-8)6,5,5                                                      DALI 900
    5 IF(DELT2-DELT1)6,11,11                                            DALI 910
    6 CONTINUE                                                          DALI 920
      END OF AITKEN-LOOP                                                DALI 930
                                                                        DALI 940
    7 J=NDIM                                                            DALI 950
    8 Y=VAL(J)                                                          DALI 960
    9 RETURN                                                            DALI 970
                                                                        DALI 980
      THERE IS SUFFICIENT ACCURACY WITHIN NDIM-1 ITERATION STEPS        DALI 990
   10 IER=0                                                             DALI1000
      GOTO 8                                                            DALI1010
                                                                        DALI1020
      TEST VALUE DELT2 STARTS OSCILLATING                               DALI1030
   11 IER=1                                                             DALI1040
   12 J=IEND                                                            DALI1050
      GOTO 8                                                            DALI1060
                                                                        DALI1070
      THERE ARE TWO IDENTICAL ARGUMENT VALUES IN VECTOR ARG             DALI1080
   13 IER=3                                                             DALI1090
      GOTO 12                                                           DALI1100
      END                                                               DALI1110
      COMPLEX FUNCTION SUM*16(X1,X2,X3,Y1,Y2,Y3)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 Y1,Y2,Y3,A,B,C,D,E,F
      IF(DABS((X3-X2)-(X2-X1)).LT.1.0E-8) GO TO 10
      A=Y1/((X1-X2)*(X1-X3))
      B=Y2/((X2-X3)*(X2-X1))
      C=Y3/((X3-X1)*(X3-X2))
      D=A+B+C
      E=-A*(X2+X3)-B*(X1+X3)-C*(X2+X1)
      F=A*X2*X3+B*X1*X3+C*X1*X2
      SUM=D*(X2**2+X1*(X2+X1))/3.000+E*(X2+X1)/2.000+F)*(X2-X1)
      RETURN
   10 D=X2-X1
      SUM=D*(5*Y1+8*Y2-Y3)/12.000
      RETURN
      END
      SUBROUTINE DIFEQ(X0,XND,Y0,HH,NN,EE)
      .....
      .WRITTEN BY L.AUER, 11/14/67. MODIFIED FOR IBM 370 BY A.ENDAL    .
      .3/13/74.                                                        .
      .                                                                .
      .PURPOSE                                                         .
      .     SIMULTANEOUS SOLUTION OF NN FIRST ORDER DIFFERENTIAL       .
      .     EQUATIONS IN ONE INDEPENDENT VARIABLE AND NN DEPENDENT     .
      .     VARIABLES. NN MAY BE UP TO 100.                            .
      .                                                                .
      .ARGUMENTS                                                       .
      .     X0 = STARTING VALUE FOR INDEPENDENT VARIABLE               .
      .     XND = VALUE OF INDEPENDENT VARIABLE AT WHICH VALUES FOR    .
      .           DEPENDENT VARIABLES ARE TO BE RETURNED               .
      .     Y0 = ARRAY OF DIMENSION NN WHICH WILL CONTAIN (INITIAL     .
      .          VALUES OF DEPENDENT VARIABLES UPON ENTRY              .
      .        = ARRAY OF FINAL VALUES OF DEPENDENT VARIABLES UPON     .
```

```
C.          RETURN                                                  .
C.      NH = INITIAL STEP SIZE TO BE USED. STEP SIZE IS INTERNALLY  .
C.           MODIFIED TO PRESERVE ACCURACY AND INCREASE SPEED       .
C.      EE = MAX. PERMISSIBLE TRUNCATION ERROR (SHOULD BE 1.E-5 TO  .
C.           1.E-8), OR                                             .
C.         = RELATIVE ERROR IF NEGATIVE                            .
C.......................................................................
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N
      COMMON/INTEG/X,N,NN,I1,I2,I3,I4
C.......................................................................
C.COMMON/INTEG/ IS A COMMON AREA USED TO COMMUNICATE WITH OTHER        .
C.     SUBROUTINES IN THIS PACKAGE.                                    .
C.......................................................................
      COMMON/SCRATC/M(100),T(100),TP(100),D1(100),D2(100)
C.......................................................................
C.COMMON/SCRATC/ IS A SCRATCH AREA WHICH MAY BE REUSED ELSEWHERE.      .
C.     NOTE THAT N IS OF TYPE REAL*8                                   .
C.......................................................................
      COMMON/YS/Y(100,4),YP(100,4),A(4)
C.......................................................................
C.COMMON/YS/ IS A SCRATCH AREA WHICH MAY BE REUSED ELSEWHERE. THE      .
C.     SIZE OF THESE ARRAYS AND THOSE IN COMMON/SCRATC/ SET THE        .
C.     UPPER LIMIT ON NN.                                              .
C.......................................................................
      DIMENSION YO(1)
C.......................................................................
C.SUBPROGRAMS NEEDED                                                   .
C.     SUBROUTINE HERMIT - SUPPLIED IN THE PACKAGE                     .
C.     SUBROUTINE IUP    - SUPPLIED IN THE PACKAGE                     .
C.     SUBROUTINE GILL   - SUPPLIED IN THE PACKAGE                     .
C.                         CONTAINS ENTRY POINT GILL2                  .
C.     SUBROUTINE DER (X,Y,YP) - SUBROUTINE WHICH GENERATES THE NN.    .
C.                               DERIVATIVES OF THE DEPENDENT          .
C.                               VARIABLES WITH RESPECT TO THE         .
C.                               INDEPENDENT VARIABLE. IT MUST BE      .
C.                               SUPPLIED WITH THE VALUE OF NN         .
C.                               WHEN WRITTEN BY THE USER.             .
C.                                                                     .
C.     ARGUMENTS FOR DER                                               .
C.          X = VALUE OF INDEPENDENT VARIABLE                          .
C.          Y = ARRAY OF VALUES OF DEPENDENT VARIABLES                 .
C.          YP = ARRAY OF VALUES OF DERIVATIVES. Y AND YP MUST BE      .
C.               OF DIMENSION NN                                       .
C.                                                                     .
C.ENTRY DEQ (XO,XNO,YO,HH,NN,EE)                                       .
C.     THIS IS PROVIDED TO CONTINUE THE INTEGRATION FROM THE POINT.    .
C.     WHERE THE PREVIOUS CALL TO DIFEQ LEFT OFF, WHICH IN             .
C.     GENERAL MAY NOT BE THE LAST XNO.                                .
C.     THE MEANING OF THE ARGUMENTS ARE THE SAME AS IN THE MAIN        .
C.     ENTRY, THOUGH XO, YO, AND HH ARE NOT USED AND NEED NOT BE       .
C.     DEFINED                                                         .
C.......................................................................
 9999 CONTINUE
C.......................................................................
C.METHOD                                                               .
C.     HAMMING'S METHOD PREDICTOR-CORRECTOR STARTED BY THREE GILL-     .
C.     RUNGA-KUTTA STEPS AND INTERNAL HALVING AND DOUBLING OF STEP     .
C.     SIZE TO PRESERVE ACCURACY AND INCREASE SPEED                    .
C.     THE STEP-HALVING IS DONE BY 4-POINT HERMITE INTERPOLATION       .
C.                                                                     .
C.ADDITIONAL DOCUMENTATION AVAILABLE                                   .
C.......................................................................

      N = NN
      H = HH
      HMIN = DABS (HH) / 4096.
      R = XO
      I1 = 1
      I2 = 2
      I3 = 3
      I4 = 4
      DO 1 I = 1 , N
    1     Y(I,1) = YO(I)
      CALL DER (X , YO , YP(I,1))
      CALL GILL (Y(I,1) , Y(I,2) , YP(I,2))
C.......................................................................
C.X IS UPPED WITHIN GILL                                               .
C.......................................................................
      CALL GILL2 (Y(I,2) , Y(I,3) , YP(I,3))
      CALL GILL2 (Y(I,3) , Y(I,4) , YP(I,4))
C.......................................................................
C.WE NOW HAVE THE VALUES AND THE DERIVATIVES AT THE FIRST 4 POINTS.     .
C.AND ARE READY TO USE THE HAMMING'S PREDICTOR-CORRECTOR. THESE         .
C.RUNGA-KUTTA STEPS ARE ALWAYS COMPUTED TO RESTART THE SOLUTION         .
```

```
C*****************************************************************
2 DO 3 I = 1 , N
3       M(I) = 0.

  ENTRY DEC (X0 , XND , Y0 , HH , NN , EE)
  EPS = DABS (EE)
  IF (X .GE. XND) GO TO 14

4 DO 5 I = 1 , N
        T(I) = Y(I,I1) + 4. * H * (2. * YP(I,I4) - YP(I,I3) + 2. *
*        YP(I,I2)) / 3.
C*****************************************************************
C       .D2 IS THE MODIFIED PREDICTOR                             .
C*****************************************************************
5       D2(I) = T(I) + .92561983 * M(I)
  CALL DER (X+H , D2 , TP)
  DO 6 I = 1 , N
        D1(I) = (19. * Y(I,I4) - Y(I,I2)) + 3. * H * (TP(I) + 2. *
*        YP(I,I4) - YP(I,I3))) / 8.
6       M(I) = D1(I) - T(I)
C*****************************************************************
C  .CHECK WHETHER THE STEP SIZE SHOULD BE HALVED OR DOUBLED        .
C*****************************************************************
  EMIN = 0.
  DO 7 I = 1 , N
C*****************************************************************
C       .CHECK THAT THE RELATIVE ACCURACY CRITERION IS BEING      .
C       .SATISFIED                                                .
C*****************************************************************
        E = 9. * DABS (M(I)) / 121.
        IF (EE .LT. 0.) E = 2. * E / DABS (D1(I) + T(I) + 1.D-65)
        IF (E .GT. EPS) GO TO 9

7       EMIN = DMAXI (E , EMIN)
C*****************************************************************
C  .THE LAST STEP WAS OF SATISFACTORY ACCURACY SO THE SOLUTION MAY .
C  .BE UPDATED                                                    .
C*****************************************************************
  DO 8 I = 1 , N
8       Y(I,I1) = D1(I) - .07438016 * M(I)
C*****************************************************************
C  .MODIFY THE FINAL SOLUTION INSTEAD OF ITERATING                .
C*****************************************************************
  X = X + H
  CALL DER (X , Y(1,I1) , YP(1,I1))
  CALL IUP
  IF (X .GE. XND) GO TO 14

  IF (EMIN .GT. 0.001 * EPS) GO TO 4

  GO TO 11
C*****************************************************************
C  .STEP HALVING                                                  .
C*****************************************************************
9 CALL HERMIT (X-0.5*H , D2)
  CALL HERMIT (X-1.5*H , D1)
  DO 10 I = 1 , N
        M(I) = M(I) / 32.
        Y(I,I2) = Y(I,I3)
        YP(I,I2) = YP(I,I3)
        Y(I,I1) = D1(I)
10      Y(I,I3) = D2(I)
  CALL DER (X-0.5*H , D2 , YP(I,I3))
  CALL DER (X-1.5*H , D1 , YP(I,I1))
  H = H / 2.
  IF (DABS (H) .GT. HMIN) GO TO 4

  PRINT 100,X,H
100 FORMAT (' STEP SIZE H HAS BEEN HALVED TOO OFTEN AT X=',1PD12.5,/,
*     ' PRESENT STEP SIZE IS H=',D12.5)
  PRINT 101
101 FORMAT (10X,'N',9X,'Y(I1)',8X,'YP(I1)',9X,'Y(I2)',8X,'YP(I2)',
*     9X,'Y(I3)',8X,'YP(I3)',9X,'Y(I4)',8X,'YP(I4)')
  PRINT 102,(M(I),Y(I,I1),YP(I,I1),Y(I,I2),YP(I,I2),Y(I,I3),YP(I,I3)
*     ,Y(I,I4),YP(I,I4),I=1,N)
102 FORMAT (1P9D14.6)
  GO TO 4
C*****************************************************************
C  .STEP DOUBLING                                                 .
C*****************************************************************
11 DO 12 I = 1 , N
        Y(I,I3) = Y(I,I2)
12      YP(I,I3) = YP(I,I2)
  H = 2. * H
  CALL GILL (Y(I,I4) , Y(I,I1) , YP(I,I1))
```

```
      CALL IUP
      CALL GILL2 (Y(1,I4) , Y(1,I1) , YP(1,I1))
      CALL IUP
      DO 13 I = 1 , N
   13      H(I) = D.
      ************************************************************
      .ONCE MORE CHECK THE TERMINATION CONDITION                  .
      ************************************************************

   14 CALL HERMIT (XND , YO)
      IF (X .LT. XND) GO TO 2

      RETURN
      END
      SUBROUTINE HERMIT (XO , YO)
      ************************************************************
      .THIS RDUTINE PERFORMS THE HERMITE INTERPOLATION FOR STEP HALVING.
      .IN DIFEQ. SEE DIFEQ FOR ADDITIONAL DOCUMENTATION.          .
      ************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L,LP
      COMMON/INTEG/X,H,N,I1,I2,I3,I4
      COMMON/YS/Y(100,4),YP(100,4),A(4)
      DIMENSION YD(1)

      A(I4) = X
      A(I3) = X - H
      A(I2) = A(I3) - H
      A(I1) = A(I2) - H
      DO 1 I = 1 , N
    1      YD(I) = 0.
      DO 4 I = 1 , 4
           L = 1.
           LP = 0.
           DO 2 J = 1 , 4
                IF (I .EQ. J) GO TO 2

                L = L * (XO - A(J)) / (A(I) - A(J))
                LP = LP + 1. / (A(I) - A(J))

    2           CONTINUE
           DO 3 K = 1 , N
                YD(K) = YD(K) + ((1. - 2. * (XO - A(I)) * LP) * Y(K,I)
     $           + (XO - A(I)) * YP(K,I)) * L * L
    3           CONTINUE
    4      CONTINUE

      RETURN
      END
      SUBROUTINE IUP
      ************************************************************
      .THIS SUBROUTINE UPDATES THE INDICES IN COMMON/INTEG/ WHEN CALLED.
      .BY DIFEQ. SEE DIFEQ FOR ADDITIONAL DOCUMENTATION.          .
      ************************************************************
      REAL*8 X , H
      COMMON/INTEG/X,H,N,I1,I2,I3,I4

      I1 = MOD (I1 , 4) + 1
      I2 = MOD (I2 , 4) + 1
      I3 = MOD (I3 , 4) + 1
      I4 = MOD (I4 , 4) + 1

      RETURN
      END
      SUBROUTINE GILL (YD , YI , YF)
      ************************************************************
      .THIS RDUTINE PERFORMS THE GILL-RUNGA-KUTTA STEPS WHEN CALLED BY .
      .DIFEQ. SEE DIFEQ FOR ADDITIONAL DOCUMENTATION.             .
      ************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/INTEG/X,H,N,I1,I2,I3,I4
      DATA CONS/0.70710678118654700/
      DIMENSION YO(1),YI(1),YP(1)
      COMMON/SCRATC/T1(100),T2(100),T3(100),T4(100),S(100)
      ************************************************************
      .THIS IS A GILL'S METHOD OF INTEGRATION.                    .
      ************************************************************
      CALL DER (X , YD , T1)

      ENTRY GILL2 (YD , YI , YP)
      DO 1 I = 1 , N
           T1(I) = H * T1(I)
    1      S(I) = YD(I) + 0.5 * T1(I)
      CALL DER (X+0.5*H , S , T2)
      DO 2 I = 1 , N
```

```
      T2(I) = H * T2(I)
2     S(I) = YD(I) + (CONS - 0.5) * T1(I) + (1.0 - CONS) * T2(I)
  CALL DER (X+0.5*H , S , T3)
  DO 3 I = 1 , N
      T3(I) = H * T3(I)
3     S(I) = YD(I) - CONS * T2(I) + (1.0 + CONS) * T3(I)
  X = X + H
  CALL DER (X , S , T4)
  DO 4 I = 1 , N
      Y1(I) = YD(I) + (T1(I) + 2. * (1. - CONS) * T2(I) + 2. *
     $     (1. + CONS) * T3(I) + H * T4(I)) / 6.
4     CONTINUE
  CALL DER (X , Y1 , YP)
  DO 5 I = 1 , N
5     T1(I) = YP(I)

  RETURN
  END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

                                ANSBNP

   PURPOSE:  TO COMPUTE CAPTURE CROSS SECTIONS FOR PROCESSES OF THE FORM
                   B+   +   A(NS)   ->   B(N'P)   +   A+
             OR THEIR TIME REVERSED COUNTERPARTS.


   CARD 1:   NLG   (16I5)
             NUMBER OF INTEGRATION POINTS FOR GAUSS-LAGUERRE QUADRATURE
   CARD 2:   (X(IX),WX(IX),IX=1,NLG)  (4D20.15)
             ABCISSAS AND WEIGHTING FACTORS FOR GAUSS-LAGUERRE QUADRATURE
   CARD 3:   (LABEL(I),I=1,20)  (20A4)
             PROGRAM TITLE
   CARD 4:   Z1, Z2, RMAX, FMB  (8F10.5)
             Z1:  CHARGE OF THE TARGET
             Z2:  CHARGE OF THE PROJECTILE
             RMAX:  INTERNUCLEAR DISTANCE FOR WHICH REACTION BEGINS TO OCCUR
                    (FOR K-K CAPTURE RMAX=30)
             FMB:  FMB=0  IF ENERGY IS IN KEV/AMU
             FMB:  FMB= MASS OF THE PROJECTILE IF ENERGY IS IN KEV
   CARD 5:   NCODE, ICODE, JCODE, IPNCH, IRED, LIMIT, MM, IPNT1,
             IPNT2, NITP, NOAB, NCAB, IEL   (16I5)
             NCODE:  NCODE=0  NON-ITERATIVE SOLUTION OF COUPLED EQUATIONS
                     IF NCODE=0  LIMIT, MM, IPNT1, IPNT2, NITP, NCAB, NCAB
                     NEED NOT BE DEFINED
                     NCODE=1  ITERATIVE SOLUTION OF COUPLED EQUATIONS
             ICODE=1  IF INTEGRATION OVER IMPACT PARAMETERS IS PERFORMED
             JCODE=1  IF   UA, UB, XAB, XBA  ARE PRINTED
             IPNCH=1  IF MATRIX ELEMENTS ARE PUNCHED
             IRED=1  IF MATRIX ELEMENTS ARE READ IN
             LIMIT:  NUMBER OF ITERATIONS NEEDED
                     FOR  V/VE=1/2,1,2;  LIMIT=6,4,3
             MM:  10*MM+1  IS THE NUMBER OF TIME INTEGRATION POINTS
                  USED IN THE ITERATIVE SOLUTION
             IPNT1=1  FOR EVERY NITP INTERPOLATED MATRIX ELEMENTS
                      TO BE PRINTED
             IPNT2=1  FOR EVERY NOAB POINTS OF EVERY ITERATIVE SOLUTION
                      TO BE PRINTED
             NCAB:  EVERY NCAB POINTS OF THE FINAL SOLUTION ARE PRINTED IF
                    NCAB IS NOT DEFINED IT IS AUTOMATICALLY SET EQUAL TO 1
             IEL=1  FOR A ONE ELECTRON TARGET
   CARD 6:   NI, NEV, NRO, NA, NB, NT, NP   (16I5)
             NI:  6*NI+1  IS THE NUMBER OF TIME INTEGRATION POINTS
             NEV:  NUMBER OF ENERGIES
             NRO:  NUMBER OF IMPACT PARAMETERS
             NA (NB):  IS THE PRINCIPAL QUANTUM NUMBER FOR THE
                       INITIAL (FINAL) STATE AND NEED BE DEFINED ONLY
                       IF A HYDROGENIC MODEL IS USED.
             NT (NP):  NT=1 (NP=1)  IF A NON-HYDROGENIC MODEL IS USED FOR
                       THE INITIAL (FINAL) STATE. IF NT=1 (NP=1) THE 4 CARDS
                       FROM PROGRAM HERMAN CONTAINING THE INFORMATION
                       FOR THE INITIAL (FINAL) STATE MUST FOLLOW CARD 8
                       ( FOR THE FINAL STATE THE 4 CARDS FOLLOW EITHER
                       CARD 8 OR CARD 12, WHICH EVER IS LAST.)
   CARD 7:   (EIMPT(I),I=1,NEV)  (8F10.5)
             ENERGY POINTS IN KEV
   CARD 8:   (EB(I),I=1,NRO)  (8F10.5)
             IMPACT PARAMETERS IN A.U.
             IF NT=1 THE FINAL  STATE INFORMATION FOLLOWS
             IF NP=1 THE FINAL STATE INFORMATION FOLLOWS

                     WRITTEN BY LAURA TUNNELL (1973)


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 SAB,HAB,HBA,SBA(150),PRMP,CI
      COMPLEX*16 SABPX,HABPX,HBAPX
      COMPLEX*16 SABPZ,HABPZ,HBAPZ
      COMMON/MXX/SABPX(150),HABPX(150),HBAPX(150),HBBPX(150)
      COMMON/MZZ/SABPZ(150),HABPZ(150),HBAPZ(150),HBBPZ(150)
      COMMON/MATRX/SAB(150),HAB(150),HBA(150),HAA(150),HBB(150)
      COMMON/IMPCT1/V,RHO,RM,T,P1,NT,NNT,NNT1
      COMMON/IMPCT2/R(150),TIME(150),Z1,Z2
      COMMON/IMPCT3/EB(20),EIMPT(10),NRO,NEV,LEV,IPB
```

```
     ,,PRB(2,10,20),BRPB(2,10,20),PRMP(2,10,20)
      COMMON/LABU/X(32),WX(32),NLG
      COMMON/PTNTL/OLMV(3),CV(3,3),AZ,BZ
      COMMON/WFA/EALP,ZA,ACOEF(10),AXP(10),NXA(10),NA
      COMMON/WFB/EBTA,ZB,BCOEF(10),BXP(10),NXB(10),NB
      COMMON/OPTN/LIMIT,MM,IPNCH,IREO,IPNTI,IPNT2,NITP,NCAB,NOAB
      DIMENSION LABEL(20),XXX(25),YYY(25),SUMA(2,10),Y(4)
    3 FORMAT(10(1X,1PD12.5))
   81 FORMAT('1',///'TIME.....R      A11.....A12.....A21.....A22'/)
  100 FORMAT(16I5)
  200 FORMAT(8F10.4)
  300 FORMAT(20A4)
  400 FORMAT('1',////25X,20A4)
  249 FORMAT(4E20.15)
      AZ=0.000
      BZ=0.000
      PFI=1.000

      READ(5,100) NLG
      READ(5,249) (X(IX),WX(IX),IX=I,NLG)
      READ(5,300) (LABEL(I),I=1,20)
      READ(5,200) ZI,Z2,RMAX,FMB
      IF(RMAX.EQ.0.000)  RMAX=40.000
      IF(FMB.EQ.0.000)  FMB=1.000
      READ(5,100)  NCODE,ICODE,JCODE,IPNCH,IREO,
     .LIMIT,MM,IPNT1,IPNT2,NITP,NOAB,NCAB,IEL
      IF(IEL.NE.0)  PFI=0.000
      IF(NCAB.EQ.0)  NCAB=5
      IF(NOAB.EQ.0)  NOAB=5
      IF(NITP.EQ.0)  NITP=5
      READ(5,100)  NI,NEV,NRO,NA,NB,NT,NP,IFILE
      READ(5,200) (EIMPT(I),I=1,NEV)
      READ(5,200) (   EB(I),I=1,NRO)
      IF(NA.EQ.0)  NA=I
      IF(NB.EQ.0)  NB=2
      PRINT 411
  411 FORMAT('  NLG,     (X(IX),WX(IX),IX=1,NLG)  ')
      PRINT 100,  NLG
      PRINT 3,    (X(IX),WX(IX),IX=I,NLG)
      PRINT 414
  414 FORMAT('     ZI,Z2,RMAX,FMB  ')
      PRINT 200,  ZI,Z2,RMAX,FMB
      PRINT 413
  413 FORMAT1'      NCODE,ICODE,JCODE,IPNCH,IREO,
     .LIMIT,MM,IPNTI,IPNT2,NITP,NOAB,NCAB   ')
      PRINT 100,   NCODE,ICODE,JCODE,IPNCH,IREO,
     .LIMIT,MM,IPNT1,IPNT2,NITP,NOAB,NCAB
      PRINT 415
  415 FORMAT('      NI,NEV,NRO,NA,NB,NT,NP,IFILE  ')
      PRINT 100,   NI,NEV,NRO,NA,NB,NT,NP,IFILE
      PRINT 416
  416 FORMAT('     (EIMPT(I),I=I,NEV)   ')
      PRINT 200,  (EIMPT(I),I=1,NEV)
      PRINT 417
  417 FORMAT('     (EB(I),I=I,NRO  ')
      PRINT 200,  (EB(I),I=I,NRO)
      FNA=OFLOAT(NA)
      FNB=OFLOAT(NB)
      EALP=ZI*ZI/FNA/FNA/2.000
      EBTA=Z2*Z2/FNB/FNB/2.000
      ACOEF(1)=1.000
      ACOEF(2)=-DSQRT(3.000)
      ACOEF(3)=DSQRT(2.5000)
      BCOEF(1)=DSQRT(24.000)
      BCOEF(2)=-DSQRT(45.000)
      IF(NB.EQ.2)  BCOEF(1)=1.000
      DO 20 I=1,3
      NXA(I)=I
      NXB(I)=I+1
      AXP(I)=ZI/FNA
      BXP(I)=Z2/FNB
      OLMV(I)=0.000
      DO 20 J=1,3
   20 CV(I,J)=0.000
      NB=NB-1
      IF(NT.EQ.0)  GO TO 22
      READ(5,200) EALP,FA,OLMV(2),(CV(1,2),I=1,3),AZ
      NA=FA
      READ(5,100) (NXA(I),I=1,NA)
      READ(5,200) (AXP(I),I=1,NA)
      READ(5,200) (ACOEF(I),I=1,NA)
   22 CONTINUE
      IF(NP.EQ.0)  GO TO 24
      READ(5,200) EBTA,FB,OLMV(3),(CV(I,3),I=1,3),BZ
      NB=FB
```

```
      READ(5,100) (NXB(I),I=1,NB)
      READ(5,200) (BXP(I),I=1,NB)
      READ(5,200) (BCOEF(I),I=1,NB)
   24 CONTINUE
      FA=DFLOAT(NA)
      FB=DFLOAT(NB)
      PRINT 423
  423 FORMAT('     EALP,NA,DLMV(2),(CV(I,2),I=1,3)  ')
      PRINT 200,  EALP,FA,DLMV(2),(CV(I,2),I=1,3)
      PRINT 425
  425 FORMAT('     (NXA(I),I=1,NA)       ')
      PRINT 100,  (NXA(I),I=1,NA)
      PRINT 419
  419 FORMAT('     (AXP(I),I=1,NA)       ')
      PRINT 200,  (AXP(I),I=1,NA)
      PRINT 421
  421 FORMAT('     (ACOEF(I),I=1,NA)              ')
      PRINT 200,  (ACOEF(I),I=1,NA)
      PRINT 424
  424 FORMAT('     EBTA,NB,DLMV(3),ICV(I,3),I=1,3)        ')
      PRINT 200,  EBTA,FB,DLMV(3),(CV(I,3),I=1,3)
      PRINT 418
  418 FORMAT('     (NXB(I),I=1,NB)       ')
      PRINT 100,  (NXB(I),I=1,NB)
      PRINT 420
  420 FORMAT('     (BXP(I),I=1,NB)                ')
      PRINT 200,  (BXP(I),I=1,NB)
      PRINT 422
  422 FORMAT('     (BCOEF(I),I=1,NB)               ')
      PRINT 200,  (BCOEF(I),I=1,NB)

      AZ=AZ+1.000
      BZ=BZ+1.000
      CI=DCMPLX(0.000,1.000)
      PI=DARCOS(-1.000)
      TWOPI=2.000*PI*((.5290-08)**2)
      ZA=Z1
      ZB=Z2
      Z2=ZA
      IF(ZB.LT.ZA)  Z2=ZB
      Z1=Z2
      EALP=EALP/Z2/Z2
      EBTA=EBTA/Z2/Z2
      NNT=3*NI
      NNT1=NNT+1
      NT=2*NNT+1
      DO 1000 LEV=1,NEV
      EKEV=E(MPT(LEV)
      V=DSQRT(EKEV/FMB)/5.000
      SETUP TIME VECTOR
      TIME(NNT1)=0.0
      IF(NI.EQ.0) GO TO 1005
      TIME(NT)=RMAX/(ZA+ZB)/V
      TIME(NNT1)=0.000
      HT=TIME(NT)/7.000/DFLOAT(NI)
      DO 10 J=1,3
      JI=(J-1)*NI
      DO 12 I=1,NI
      JII=I
      IF(J.GT.1) JII=I*(2**(J-1))
      TIME(NNT1+JI+I)=TIME(NNT1+JI)+JII*HT
   12 CONTINUE
   10 CONTINUE
      DO 14 I=1,NNT
   14 TIME(NNT1-I)=-TIME(NNT1+I)
 1005 CONTINUE
      DO 1001 IPB=1,NRD
      RHO=BB(IPB)
      R(NNT1)=RHO
      IF(NI.EQ.0) GO TO 1006
      DO 15 I=1,NNT
      R(NNT1+I)=DSQRT(RHO*RHO+V*V*TIME(NNT1+I)*TIME(NNT1+I))
      R(NNT1-I)=R(NNT1+I)
   15 CONTINUE

      IF((REO.EQ.1) GO TO 50
 1006 CONTINUE
      CALL CMATRX
      GO TO 55
   50 CONTINUE
      EKEV=EKEV/Z2/Z2
      V=V/Z2
      RHO=RHO*Z2
      VR=DSQRT(EKEV/EALP/FMB/27.2100-03/(836.0)
      DO 59 II=1,2
```

```
      DO 51 I=I,NT
      READ(5,52) TIME(I),HAA(I),HBB(I)
      READ(5,52) SAB(I),HAB(I),HBA(I)
      TIME(I)=TIME(I)*Z2*Z2
   51 CONTINUE
   52 FORMAT(7E11.5)
      IF(III.EQ.1)  PRINT 91
      IF(III.EQ.2)  PRINT 61
      DO 75 IT=I,NNTI
   78 PRINT 93, TIME(IT),R(IT),SAB(IT),HAB(IT),HBA(IT),HAA(IT),HBB(ITI
   93 FORMAT(2(1X,F10.5),5X,3(1PD11.4,1X,1PD11.4,2X),3X,2(1PD11.4,1X))
      PRINT 94, Z2,EKEV,V,VR,RHO
      IF(NCODE.EQ.1)  CALL COUPEQ(NT,III)
      IF(NCODE.NE.0)  GO TO 76
      IF(III.EQ.1)  PRINT 97
      IF(III.EQ.2)  PRINT 67
      Y(1)=1.000
      Y(2)=0.000
      Y(3)=0.000
      Y(4)=0.000
      CALL DIFEQ(TIME(II,TIME(2)/2.000,Y,0.2000,4,1.0-6)
      DO 78 IT=2,NT
      CALL DEQ(TIME(I),TIME(IT),Y,HH,4,1.0-6)
      PROB=Y(3)*Y(3)+Y(4)*Y(4)
      UNIT=PROB+Y(1)*Y(1)+Y(2)*Y(2I-1.000
      UNIT=UNIT+2.000*SAB(IT)*DCMPLX(Y(1),-Y(2))*DCMPLX(Y(3),Y(4)I
     ./CDEXP(CI*TIME(IT)*(EALP-EBTA))
   78 PRINT 95, TIME(IT),R(IT),(Y(I),I=I,4),PROB,UNIT
      PRB(III,LEV,IPB)=PROB
      BRPB(III,LEV,IPB)=PROB*EB(IPB)
      PRMP(III,LEV,IPB)=DCMPLX(Y(3),Y(4))
      PRINT 99, PRB(III,LEV,IPB),BRPB(III,LEV,IPB),EIMPT(LEV),EB(IPB)
   76 CONTINUE
   59 CONTINUE
      GO TO 53
   55 CONTINUE
      EKEV=EKEV/Z2/Z2
      V=V/Z2
      RHO=RHO*Z2
      VR=DSQRT(EKEV/EALP/FMB/27.2106D-03/1836.0)
      DO 56 I=I,NNTI
      II=NT-(I-I)
      TIME(I)=TIME(I)*Z2*Z2
      TIME(II)=-TIME(I)
      SAB(I)=SABPX(I)
      HAB(I)=HABPX(I)
      HBA(I)=HBAPX(I)
      HBB(I)=HBBPX(I)
      SAB(II)=SABPX(II)
      HAB(II)=HABPX(II)
      HBA(II)=HBAPX(II)
      HBB(II)=HBBPX(II)
   56 CONTINUE
      IF(IPNCH.EQ.0)  GO TO 70
      DO 71 I=I,NNTI
      PUNCH 52,  TIME(I),HAA(I),HBB(I)
   71 PUNCH 52,  SAB(I),HAB(I),HBA(I)
   70 CONTINUE
      PRINT 400, (LABEL(I),I=I,20)
      PRINT 91
   91 FORMAT('  NPX    TIME...R      SAB...HAB...HBA    HAA...HBB    '//)
      DO 92 IT=I,NNTI
   92 PRINT 93, TIME(IT),R(IT),SAB(ITI,HAB(IT),HBA(IT),HAA(IT),HBB(IT)
      PRINT 94, Z2,EKEV,V,VR,RHO
   94 FORMAT(//'  SCALING FACTOR =',F6.2,' SCALED QUANTITIES ; E(KEV)=',
     .1PD11.4,' V =',1PD9.2,' V/VK =',1PD9.2,' RHO =',1PD9.2//)
      IF(NCODE.EQ.1)  CALL COUPEQ(NT,1)
      IF(NCODE.NE.0)  GO TO 96
      PRINT 97
   97 FORMAT('I',///' NPX     TIME...R      CA...CB...PROB...UNIT  '//)
      Y(1)=I.000
      Y(2)=0.000
      Y(3)=0.000
      YI4I=0.000
      CALL DIFEQ(TIME(I),TIME(2)/2.000,Y,0.2000,4,I.0-6)
      DO 98 IT=2,NT
      CALL DEQ(TIME(I),TIME(IT),Y,HH,4,1.0-6)
      PROB=Y(3)*Y(3)+Y(4)*Y(4)
      UNIT=PROB+Y(1)*Y(1)+Y(2)*Y(2)-1.000
      UNIT=UNIT+2.000*SAB(IT)*DCMPLX(Y(1),-Y(2))*DCMPLX(Y(3),Y(4))
     ./CDEXP(CI*TIME(IT)*(EALP-EBTA))
   98 PRINT 95, TIME(IT),R(IT),(Y(I),I=1,4),PROB,UNIT
   95 FORMAT(2(2X,F10.5),6X,2(1PD10.3,1X,1PD10.3,2X),2X,2(1PD10.3,1X))
      PRB(I,LEV,IPB)=PROB
      BRPB(I,LEV,IPB)=PROB*EB(IPB)
```

```
      PRMP(I,LEV,IPB)=DCMPLX(Y(3),Y(4))
      PRINT 99, PRB(1,LEV,IPB),BRPB(1,LEV,IPB),EIMPT(LEV),EB(IPB)
   99 FORMAT(/' PROB =',1PD9.2,' PROB X RHO =',1PD9.2,' AT E(KEV) =',
     .F10.2,' RHO =',F8.6//)
   96 CONTINUE
      IF(JCODE.EQ.0)  GO TO 84
      PRINT 81
      DO 82 I=NNT1,NT
      SBA(I)=DCONJG(SAB(I))
      SS=1.000-SAB(I)*SBA(I)
      A11=-EALP+(HAA(I)-SAB(I)*HBA(I))/SS
      A12=(HAB(I)-SAB(I)*HBB(I))/SS
      A21=(HBA(I)-SBA(I)*HAA(I))/SS
      A22=-EBTA+(HBB(I)-SBA(I)*HAB(I))/SS
      PRINT 3, TIME(I),R(I),A11,A12,A21,A22
   82 CONTINUE
   84 CONTINUE
      DO 57 I=I,NNTI
      II=NT-(I-I)
      SAB(I)=SABPZ(I)
      HAB(I)=HABPZ(I)
      HBA(I)=HBAPZ(I)
      HBB(I)=HBBPZ(I)
      SAB(II)=SABPZ(II)
      HAB(II)=HABPZ(II)
      HBA(II)=HBAPZ(II)
      HBB(II)=HBBPZ(II)
   57 CONTINUE
      IF(IPNCH.EQ.0)  GO TO 73
      DO 74 I=I,NNTI
      PUNCH 52,  TIME(I),HAB(I),HBB(I)
   74 PUNCH 32,  SAB(I),HAB(I),HBA(I)
   73 CONTINUE
      PRINT 400, (LABEL(I),I=I,20)
      PRINT 61
   61 FORMAT('  NPZ   TIME...R     SAB...HAB...HBA   HAA...HBB   '//)
      DO 62 IT=I,NNT(
   62 PRINT 93, TIME(IT),R(IT),SAB(IT),HAB(IT),HBA(IT),HAA(IT),HBB(IT)
      PRINT 94, Z2,EKEV,V,VR,RHO
      IF(NCODE.EQ.1)  CALL COUPEQ(NT,2)
      IF(NCODE.NE.0)  GO TO 66
      PRINT 67
   67 FORMAT('I',///' NPZ    TIME...R      CA...CB...PROB...UNIT  '//)
      Y(1)=I.000
      Y(2)=0.000
      Y(3)=0.000
      Y(4)=0.000
      CALL DIFEQ(TIME(1),TIME(2)/2.000,Y,0.2000,4,I.0-6)
      DO 68 IT=2,NT
      CALL DEQ(TIME(I),TIME(IT),Y,HH,4,1.0-6)
      PROB=Y(3)*Y(3)+Y(4)*Y(4)
      UNIT=PROB+Y(1)*Y(1)+Y(2)*Y(2)-I.000
      UNIT=UNIT+2.000*SAB(IT)*DCMPLX(Y(1),-Y(2))*DCMPLX(Y(3),Y(4))
     ./CDEXP(CI*TIME(IT)*(EALP-EBTA))
   68 PRINT 95, TIME(IT),R(IT),(Y(I),I=I,4),PROB,UNIT
      PRB(2,LEV,IPB)=PROB
      BRPB(2,LEV,IPB)=PROB*EB(IPB)
      PRMP(2,LEV,IPB)=DCMPLX(Y(3),Y(4))
      PRINT 99, PRB(2,LEV,IPB),BRPB(2,LEV,IPB),EIMPT(LEV),EB(IPB)
   66 CONTINUE
      IF(JCODE.EQ.0)  GO TO 85
      PRINI 81
      DO 83 I=NNT1,NT
      SBA(I)=DCONJG(SAB(I))
      SS=1.000-SAB(I)*SBA(I)
      A11=-EALP+(HAA(I)-SAB(I)*HBA(I))/SS
      A12=(HAB(I)-SAB(I)*HBB(I))/SS
      A21=(HBA(I)-SBA(I)*HAA(I))/SS
      A22=-EBTA+(HBB(I)-SBA(I)*HAB(I))/SS
      PRINT 3, TIME(I),R(I),A11,A12,A21,A22
   83 CONTINUE
   85 CONTINUE
   53 CONTINUE
      EKEV=EKEV*Z2*Z2
      V=V*Z2
      RHO=RHO/Z2
      DO 58 I=1,NNTI
      TIME(I)=TIME(I)/Z2/Z2
   58 TIME(I)=-TIME(I)
 1001 CONTINUE
      IF(ICODE.EQ.0)  GO TO 1000
      DO 87 INDX=1,2
      DO 88 I=1,NRO
      XXX(I)=EB(I)
      PF2=1.000-PF1*PRB(INDX,LEV,I)
```

```
   88 YYY(I)=BRPB(INDX,LEV,I)*PF2
      CALL AREA(XXX,YYY,NRG,AA)
   87 SGMA(INDX,LEV)=TWOPI*AA
 1000 CONTINUE


  101 FORMAT(1CX,16I5)
  201 FORMAT(10X,8F10.5)
  333 FORMAT(10X,9(IX,1PD12.5))
  301 FORMAT('1',10X,20A4,'   FILE NO. =',I6)
  401 FORMAT(' ',////25X,20A4///)
      DO 402 IJK=1,4
      PRINT 301, (LABEL(I),I=1,20),IFILE
      PRINT 411
      PRINT 101,  NLG
      PRINT 333,  (X(IX),WX(IX),IX=1,NLG)
      PRINT 414
      PRINT 201,  ZA,ZB,RMAX,EMB
      PRINT 413
      PRINT 101,   NCODE,ICODE,JCODE,IPNCH,IREO,
     .LIMIT,MM,IPNT1,IPNT2,NITP,NGAB,NCAB
      PRINT 415
      PRINT 101,   N1,NEV,NRO,NA,NB,NT,NP,IFILE
      PRINT 416
      PRINT 201,  (EIMPT(I),I=1,NEV)
      PRINT 417
      PRINT 201,  (EB(I),I=1,NRO)
      PRINT 401, (LABEL(I),I=1,20)
      DO 10000 LEV=1,NEV
      DO 404 JKL=1,2
      PRINT 10001
10001 FORMAT(10X,'  RHO      PROB.AMP....PROB....PROB X RHO'//)
      DO 10002 IPB=1,NRO
10002 PRINT 10003, EB(IPB),PRMP(JKL,LEV,IPB),PRB(JKL,LEV,IPB),
     .       BRPB(JKL,LEV,IPB)
10003 FORMAT(10X,F10.5,5X,2(1PD11.4,1X),2(2X,1PD11.4))
      IF(JKL.EQ.2)  GO TO 10005
      PRINT 10004, SGMA(JKL,LEV),EIMPT(LEV)
10004 FORMAT(/15X,'  TOTAL NPX CROSS SECTION/ELECTRON(CM**2) =',1PD9.2,
     .       '  AT E(KEV) =',F12.5///)
      GO TO 404
10005 PRINT 10006, SGMA(JKL,LEV),EIMPT(LEV)
10006 FORMAT(/15X,'  TOTAL NPZ CROSS SECTION/ELECTRON(CM**2) =',1PD9.2,
     .       '  AT E(KEV) =',F12.5///)
  404 CONTINUE
10000 CONTINUE
  402 CONTINUE


      STOP
      END

      SUBROUTINE CMATRX
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX*16 SAB,SBA,HAB,HBA
      COMPLEX*16  SABIO,HABIO,HBAIO,  SABJO,HABJO,HBAJO
      COMPLEX*16  SABI1,HABI1,HBAI1,  SABJ1,HABJ1,HBAJ1
      COMPLEX*16  SABXO,HABXO,HBAXO,  SABPO,HABPO,HBAPO
      COMPLEX*16  SABX1,HABX1,HBAX1,  SABP1,HABP1,HBAP1
      COMPLEX*16  SABQO,HABQO,HBAQO,  SABZ,HABZ,HBAZ
      COMPLEX*16  SABQ1,HABQ1,HBAQ1,  SABX,HABX,HBAX
      COMPLEX*16  SABR,HABR,HBAR
      COMPLEX*16  SABPX, HABPX, HBAPX, SABPZ, HABPZ, HBAPZ
      COMPLEX*16  C,CI,T,X,TCOSX,TSINX
      COMPLEX*16  GSAB0(15,15), GHAB0(15,15), GHBA0(15,15)
      COMPLEX*16  GSAB1(15,15), GHAB1(15,15), GHBA1(15,15)
      COMPLEX*16  FLM0(3,20), FLM1(3,20), PL(20), PM(20), JL(20)
      DIMENSION  VV(5,3),P(3),DPDS(3),DMEG(3)
      DIMENSION  DPX0(3),DPX1(3),DLM(3),NM(3)
      COMMON/IMPCT1/V,RHO,RM,TI,PI,NT,NNT,NNT1
      COMMON/IMPCT2/R1(150),TIME(150),Z1,Z2
      COMMON/IMPCT3/EB(20),EIMPT(10),NRO,NEV,LEV,IPB
      COMMON/MZZ/SABPZ(150),HABPZ(150),HBAPZ(150),HBBPZ(150)
      COMMON/MXX/SABPX(150),HABPX(150),HBAPX(150),HBBPX(150)
      COMMON/MATRX/SAB(150),HAB(150),HBA(150),HAA(150),HBB(150)
      COMMON/LAGU/XA(32),WX(32),NLG
      COMMON/WFA/EALP,ZA,ACCEF(10),AXP(10),NXA(10),NA
      COMMON/WFB/EBTA,ZB,BCCEF(10),BXP(10),NXB(10),NB
      COMMON/PTNTL/OLMV(3),VL(3,3),AZ,BZ,Z(3)
      COMMON/CONST/A(25,25),B(25,25),DF(30),DG(30)


      NAI=NXA(NA)+I
      NBI=NXB(NB)
```

```
      NMX=NA1+N31
      NM3=NMX+3
      C=DCMPLX(0.000,0.000)
      C1=DCMPLX(0.000,1.000)
      CR=DCMPLX(1.000,0.000)
      PL(1)=CR
      PM(1)=C
      D3=-DSQRT(3.000)
      GMA2=V*RHO/2.000
      P(1)=0.000
      P(2)=1.000
      P(3)=-1.000
      Z(1)=0.000
      Z(2)=2A-A2
      Z(3)=ZB-B2
      DLMV(1)=0.000
      DO 14 J=1,3
   14 CV(J,1)=C.000
      IF(IPB.GT.1)  GO TO 499
      IF(LEV.GT.1)  GO TO 499
      TIME(NNT1)=TIME(NNT1-1)/50.000
      R1(NNT1)=DSQRT(V*V*TIME(NNT1)*TIME(NNT1)+RHO*RHO)

      DF(1)=0.000
      DF(2)=1.000
      DG(1)=1.000
      DG(2)=1.000
      DO 15 I=2,29
      DF(I+1)=DF(I)+1.000
   15 DG(I+1)=DF(I+1)*DG(I)
      DO 16 I=1,NA
      NI=2*NXA(I)+1
      A1=(2.000*AXP(I))**N1/DG(NI)
   16 ACOEF(I)=ACOEF(I)*DSQRT(DABS(A1))
      DO 17 J=1,NB
      NJ=2*NXB(J)+1
      B1=(2.000*BXP(J))**NJ/DG(NJ)
   17 BCOEF(J)=BCOEF(J)*DSQRT(DABS(B1))

      A(1,1)=1.000
      A(2,2)=1.000
      B(1,2)=1.000
      B(2,3)=-1.000/3.000
      DO 20 K=2,NM3,2
      AK=DF(K+1)
      A(K+1,1)=1.000/(AK+1.000)
      B(K+1,2)=3.000/(AK+1.000)/(AK+3.000)
      KK=K+1
      AKK=DF(KK+1)
      A(KK+1,2)=3.000/(AKK+2.000)
      B(KK+1,3)=-5.000/(AKK+2.000)/(AKK+4.000)
      DO 20 L=2,KK,2
      AL=DF(L+1)
      AB=(2.000*AL+1.000)/(2.000*AL-3.000)
      A(K+1,L+1)=-(2.000+AK-AL)*A(K+1,L-1)/(1.000+AK+AL)*AB
      AL=AL+1.000
      AB=(2.000*AL+1.000)/(2.000*AL-3.000)
      B(K+1,L+2)=-(3.000+AK-AL)*B(K+1,L)/(2.000+AK+AL)*AB
      LL=L+1
      ALL=AL
      A(KK+1,LL+1)=-(2.000+AKK-ALL)*A(KK+1,LL-1)/(1.000+AKK+ALL)*AB
      ALL=ALL+1.000
      AB=(2.000*ALL+1.000)/(2.000*ALL-3.000)
      B(KK+1,LL+2)=-(3.000+AKK-ALL)*B(KK+1,LL)/(2.000+AKK+ALL)*AB
   20 CONTINUE


  499 CONTINUE
      DO 500 IT=1,NNT1
      ITT=NT-(IT-1)
      GMA1=V*V*TIME(IT)/2.000
      R2=R1(IT)/2.000
      SAB10=C
      HAB10=C
      HBA10=C
      SAB11=C
      HAB11=C
      HBA11=C
      DO 510 I=1,NA
      NP=NXA(I)+1
      SABJ0=C
      HABJ0=C
      HBAJ0=C
      SABJ1=C
      HABJ1=C
```

```
      HBAJ1=C
      DO 520 J=1,NB
      NQ=NXO(J)
      DPOS1=(AXP(I)+BXP(J))*R2
      DNEG1=(AXP(I)-BXP(J))*R2
      NMX=NP+NQ
      NMM=NP+NQ+2
      SAOX0=C
      HABX0=C
      HBAX0=C
      SABX1=C
      HABX1=C
      HBAX1=C
      DO 530 IX=1,NEG
      DO 9 M=1,3
      DPOS(M)=DPOS1+DLMV(M)*R2
      DNEG(M)=DNEG1+DLMV(M)*R2*P(M)
      DLM(M)=1.000+XX(IX)/DPOS(M)
      DM=DSQRT(DLM(M)*DLM(M)-1.000)
      DX=0.000
      IF(DPOS(M).LT.1.3D+02)  DX=DEXP(-DPOS(M))
      DPXO(M)=DX/DPOS(M)
      DPX1(M)=DX/DPOS(M)*DM
      DN=0.000
      IF(DABS(DNEG(M)).LT.1.3D+02)  DN=DNEG(M)
      TCOSX=DCMPLX(GMA1*DLM(M),DN)
      TS1NX=DCMPLX(DM*GMA2,0.000)
      RD=R2*DLM(M)
      ZZ=Z(M)
      VV(2,M)=ZZ*(1.000+RD*(CV(1,M)+RD*(CV(2,M)+RD*CV(3,M))))
      VV(3,M)=P(M)*ZZ*R2*(CV(1,M)+RD*(2.000*CV(2,M)+RD*3.000*CV(3,M)))
      VV(4,M)=ZZ*R2*R2*(CV(2,M)+RD*3.000*CV(3,M))
      VV(5,M)=P(M)*ZZ*R2*R2*R2*CV(3,M)
      IF((M.GT.1).AND.(DLMV(M).LE.1.0D-08)) GO TO 91
      T=CDSQRT(TS1NX*TS1NX+TCOSX*TCOSX)
      X=TCOSX/T

      JL(1)=CDSIN(T)/T
      JL(2)=(JL(1)-CDCOS(T))/T
      PL(2)=X
      PM(2)=COSQRT(1.000-X*X)
      JL(3)=3.000*JL(2)/T-JL(1)
      PL(3)=(3.000*X*PL(2)-PL(1))/2.000
      PM(3)=3.000*X*PM(2)-2.000*PM(1)
      FLMO(M,1)=A(1,1)*JL(1)*PL(1)
      FLM1(M,1)=B(1,2)*JL(2)*PM(2)*CI
      FLMO(M,2)=A(2,2)*JL(2)*PL(2)*C1
      FLM1(M,2)=B(2,3)*JL(3)*PM(3)
      DO 10 N=2,NMM,2
      N1=N+1
      FN1=DF(N1+1)
      JL(N1+1)=(2.000*FN1-1.000)*JL(N1)/T-JL(N1-1)
      PL(N1+1)=((2.000*FN1-1.000)*X*PL(N1)-(FN1-1.000)*PL(N1-1))/FN1
      PM(N1+1)=((2.000*FN1-1.000)*X*PM(N1)-FN1*PM(N1-1))/(FN1-1.000)
      FLMO(M,N1)=A(N1,1)*JL(1)*PL(1)
      FLM1(M,N1)=B(N1,2)*JL(2)*PM(2)*CI
      N2=N1+1
      FN2=DF(N2+1)
      JL(N2+1)=(2.000*FN2-1.000)*JL(N2)/T-JL(N2-1)
      PL(N2+1)=((2.000*FN2-1.000)*X*PL(N2)-(FN2-1.000)*PL(N2-1))/FN2
      PM(N2+1)=((2.000*FN2-1.000)*X*PM(N2)-FN2*PM(N2-1))/(FN2-1.000)
      FLMO(M,N2)=A(N2,2)*JL(2)*PL(2)*C1
      FLM1(M,N2)=B(N2,3)*JL(3)*PM(3)
      NN=N1
      DO 10 L=2,N1,2
      AL=DF(L+1)
      FLMO(M,N+1)=FLMO(M,N+1)+A(N+1,L+1)*JL(L+1)*PL(L+1)
      FLM1(M,N+1)=FLM1(M,N+1)+B(N+1,L+2)*JL(L+2)*PM(L+2)*C1
      LL=L+1
      ALL=DF(LL+1)
      FLMO(M,NN+1)=FLMO(M,NN+1)+A(NN+1,LL+1)*JL(LL+1)*PL(LL+1)*C1
      FLM1(M,NN+1)=FLM1(M,NN+1)+B(NN+1,LL+2)*JL(LL+2)*PM(LL+2)
   10 CONTINUE
      GO TO 9
   91 N1=NMM+1
      DO 92 N=1,N1
      FLMO(M,N)=FLMO(1,N)
   92 FLM1(M,N)=FLM1(1,N)
    9 CONTINUE
  100 CONTINUE

      DO 35 K=1,NMX
      DO 36 L=1,NMX
      GSABO(K,L)=DPXO(1)*FLMO(1,L)
      GHABO(K,L)=AZ*GSABO(K,L)+DPXO(2)*(VV(2,2)*FLMO(2,L)
```

```
     ,+VV(3,2)*FLM0(2,L+1)+VV(4,2)*FLM0(2,L+2)+VV(5,2)*FLM0(2,L+3))
      GHBA0(K,L)=BZ*GSAB0(K,L)+DPX0(3)*(VV(2,3)*FLM0(3,L)
     ,+VV(3,3)*FLM0(3,L+1)+VV(4,3)*FLM0(3,L+2)+VV(5,3)*FLM0(3,L+3))
      GSAB1(K,L)=DPX1(1)*FLM1(1,L)
      GHAB1(K,L)=AZ*GSAB1(K,L)+DPX1(2)*(VV(2,2)*FLM1(2,L)
     ,+VV(3,2)*FLM1(2,L+1)+VV(4,2)*FLM1(2,L+2)+VV(5,2)*FLM1(2,L+3))
      GHBA1(K,L)=BZ*GSAB1(K,L)+DPX1(3)*(VV(2,3)*FLM1(3,L)
     ,+VV(3,3)*FLM1(3,L+1)+VV(4,3)*FLM1(3,L+2)+VV(5,3)*FLM1(3,L+3))
   36 CONTINUE
      DO 35 L=1,3
      DPX0(L)=DPX0(L)*DLM(L)
   35 DPX1(L)=DPX1(L)*DLM(L)
      SABP0=0
      HABP0=0
      HBAP0=0
      SABP1=0
      HABP1=0
      HBAP1=0
      DO 540 IP=1,NP
      SABQ0=0
      HABQ0=0
      HBAQ0=0
      SABQ1=0
      HABQ1=0
      HBAQ1=0
      Q1=1.000
      DO 550 IQ=1,NQ
      N1=NP+NQ+1-IP-IQ
      N2=IP+IQ-1
      CCQ=DG(NQ)/DG(IQ)/DG(NQ-IQ+1)*Q1
      SABQ1=SABQ1+GSAB1(N1,N2)*CCQ
      IF(N1.EQ.1) GO TO 551
      HABQ1=HABQ1+GHAB1(N1-1,N2)*CCQ
      HBAQ1=HBAQ1+GHBA1(N1-1,N2)*CCQ*(DF(NQ)-DF(IQ))/DF(NQ)
  551 CONTINUE
      SABR=0
      HABR=0
      HBAR=0
      CR=1.000
      DO 545 IR=1,2
      M1=NP+NQ+IP-IP-IQ
      M2=IP+IQ+IR-2
      SABR=SABR+GSAB0(M1,M2)*CR
      IF(M1.EQ.1) GO TO 545
      HABR=HABR+GHAB0(M1-1,M2)*CR
      HBAR=HBAR+GHBA0(M1-1,M2)*CR
  545 CR=-CR
      SABQ0=SABQ0+SABR*CCQ
      HABQ0=HABQ0+HABR*CCQ
      HBAQ0=HBAQ0+HBAR*CCQ*(DF(NQ)-DF(IQ))/DF(NQ)
  550 Q1=-Q1
      CCP=DG(NP)/DG(IP)/DG(NP-IP+1)
      SABP0=SABP0+SABQ0*CCP
      HABP0=HABP0+HABQ0*CCP*(DF(NP)-DF(IP))/DF(NP)
      HBAP0=HBAP0+HBAQ0*CCP
      SABP1=SABP1+SABQ1*CCP
      HABP1=HABP1+HABQ1*CCP*(DF(NP)-DF(IP))/DF(NP)
      HBAP1=HBAP1+HBAQ1*CCP
  540 CONTINUE
      SABX0=SABX0+SABP0*WX(IX)
      HABX0=HABX0+HABP0*WX(IX)
      HBAX0=HBAX0+HBAP0*WX(IX)
      SABX1=SABX1+SABP1*WX(IX)
      HABX1=HABX1+HABP1*WX(IX)
      HBAX1=HBAX1+HBAP1*WX(IX)
  530 CONTINUE
      CCJ=BCOEF(J)*(R2**(NXA(I)+NXB(J)))
      SABJ0=SABJ0+SABX0*CCJ
      HABJ0=HABJ0+HABX0*CCJ
      HBAJ0=HBAJ0+HBAX0*CCJ
      SABJ1=SABJ1+SABX1*CCJ
      HABJ1=HABJ1+HABX1*CCJ
      HBAJ1=HBAJ1+HBAX1*CCJ
  520 CONTINUE
      CCI=-ACOEF(I)*03
      SABI0=SABI0+SABJ0*CCI
      HABI0=HABI0+HABJ0*CCI
      HBAI0=HBAI0+HBAJ0*CCI
      SABI1=SABI1+SABJ1*CCI
      HABI1=HABI1+HABJ1*CCI
      HBAI1=HBAI1+HBAJ1*CCI
  510 CONTINUE
      SABZ=SABI0*R2
      HABZ=-HABI0/Z2/Z2
      HBAZ=DCONJG(-HBAI0)/Z2/Z2
```

```
      SA8X=SA8II*R2
      HA8X=-HA8II/Z2/Z2
      HB8X=DCONJG(-HB8II)/Z2/Z2

      FORM LINEAR COMBINATIONS
      COLTA=V*TIME(IT)/R1(IT)
      SOLTA=RHC/R1(IT)
      SA8PX(IT)= SA8X*COLTA+SA8Z*SOLTA
      HA8PX(IT)= HA8X*COLTA+HA8Z*SOLTA
      HB8PX(IT)= HB8X*COLTA+HB8Z*SOLTA
      SA8PZ(IT)=-SA8X*SOLTA+SA8Z*COLTA
      HA8PZ(IT)=-HA8X*SOLTA+HA8Z*COLTA
      HB8PZ(IT)=-HB8X*SOLTA+HB8Z*COLTA
      SA8PZ(ITT)=-DCONJG(SA8PZ(IT))
      HA8PZ(ITT)=-DCONJG(HA8PZ(IT))
      HB8PZ(ITT)=-DCONJG(HB8PZ(IT))
      SA8PX(ITT)=DCONJG(SA8PX(IT))
      HA8PX(ITT)=DCONJG(HA8PX(IT))
      HB8PX(ITT)=DCONJG(HB8PX(IT))
  500 CONTINUE


      CALL HHS(ACOEF,AXP,NXA,NA,3,HAA)
      CALL HHP(BCOEF,BXP,NXB,NB,2,HB8PX,HB8PZ)


      RETURN
      END

      SUBROUTINE  HHS(COEF,CXP,NXC,NC,NV,HH)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/IMPCTI/V,RHO,RM,TI,PI,NT,NNT,NNTI
      COMMON/IMPCT2/R1(150),TIME(150),Z1,Z2
      COMMON/PTNTL/DLMV(3),VC(3,3),AZ,BZ,Z(3)
      COMMON/CONST/S(25,25),T(25,25),DF(30),DG(30)
      DIMENSION COEF(NC),CXP(NC),NXC(NC)
      DIMENSION F(30),FF(30),G(30),GG(30)
      DIMENSION CV(3),HH(150)
      C=0.000
      ZZ1=BZ
      IF(NV.EQ.2)  ZZ1=AZ
      ZZ2=Z(NV)
      DLM=DLMV(NV)
      CV(1)=VC(1,NV)
      CV(2)=VC(2,NV)
      CV(3)=VC(3,NV)
      DO 100 IT=I,NNTI
      ITT=NT-(IT-1)
      R2=R1(IT)/2.000
      HI=C
      DO 200 I=1,NC
      HJ=C
      NI=NXC(I)
      NII=NI+I
      DO 300 J=1,NC
      HP=C
      NJ=NXC(J)
      NM=NI+NJ
      A=(CXP(I)+CXP(J))*R2
      B=A
      AA=A+DLM*R2
      BB=B-DLM*R2
      NM3=NM+3+1

      GX=C
      FFX=C
      GGX=C
      IF((2.000*B).LE.1.200+02)  GX=DEXP(-2.000*B)
      IF((2.000*BB).LE.1.200+02)  GGX=DEXP(-2.000*BB)
      IF((DABS(BB-AA)).LE.1.00+02)  FFX=DEXP(BB-AA)
      F(1)=1.000/A
      FF(1)=FFX/AA
      G(1)=(1.000-GX)/B
      GG(1)=(1.000-GGX)/BB
      GM=-1.000
      DO 10 K=2,NM3
      F(K)=DF(K)*F(K-1)/A+F(1)
      FF(K)=DF(K)*FF(K-1)/AA+FF(1)
      G(K)=(DF(K)*G(K-1)+GM-GX)/B
      GG(K)=(DF(K)*GG(K-1)+GM-GGX)/BB
   10 GM=-GM

      DO 400 IP=I,NII
      HQ=C
      DO 500 IQ=1,NJ
```

```
      N=NN+2-IP-IQ
      M=IP+IQ-1
      QO=FF(N)*GG(M)
      Q1=CV(1)*(FF(N+1)*GG(N)-FF(N)*GG(N+1))
      Q2=CV(2)*(FF(N+2)*GG(M)-2.000*FF(N+1)*GG(N+1)+FF(N)*GG(N+2))
      Q3=CV(3)*(FF(N+3)*GG(M)-3.000*(FF(N+2)*GG(M+1)
     *   -FF(N+1)*GG(M+2))-FF(N)*GG(M+3))
      Q=ZZI*F(N)*G(M)+ZZ2*(QO+R2*(Q1+R2*(Q2+R2*Q3)))
      CQ=DG(NJ)/DG(IQ)/DG(NJ-IQ+1)
      HQ=HQ+Q*CQ
  500 CONTINUE
      CP=DG(NI1)/DG(IP)/DG(NI-IP+2)
      HP=HP+HQ*CP
  400 CONTINUE
      CJ=COEF(J)*(R2**NN)
      HJ=HJ+HP*CJ
  300 CONTINUE
      CI=COEF(I)
      HI=HI+HJ*CI
  200 CONTINUE
      HH(IT)=-HI/2.000/ZZ/ZZ
      HH(ITT)=HH(IT)
  100 CONTINUE

      RETURN
      END

      SUBROUTINE  HHP(COEF,CXP,NXC,NC,NV,HHPX,HHPZ)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/IMPCT1/V,RHO,RN,T1,PI,NT,NNT,NNT1
      COMMON/IMPCT2/R1(150),TIME(150),Z1,Z2
      COMMON/CONST/S(25,25),T(25,25),DF(30),DG(30)
      DIMENSION COEF(NC),CXP(NC),NXC(NC)
      DIMENSION F(30),FF(30),G(30),GG(30)
      DIMENSION CV(3),HHPX(150),HHPZ(150)
      COMMON/PTNTL/DLMV(3),VC(3,3),AZ,BZ,Z(3)
      C=0.0D0
      ZZI=BZ
      IF(NV.EQ.2)  ZZI=AZ
      ZZ2=Z(NV)
      DLM=DLMV(NV)
      CV(1)=VC(1,NV)
      CV(2)=VC(2,NV)
      CV(3)=VC(3,NV)
      DO 100 IT=1,NNT1
      ITT=NT-(IT-1)
      R2=R1(IT)/2.000
      HI0=C
      HI1=C
      DO 200 I=1,NC
      HJ0=C
      HJ1=C
      NI=NXC(I)
      DO 300 J=1,NC
      HP0=C
      HP1=C
      NJ=NXC(J)-1
      NM=NXC(I)+NXC(J)
      NM3=NM+3
      A=(CXP(I)+CXP(J))*R2
      B=A
      AA=A+DLM*R2
      BB=B-DLM*R2
      GX=C
      GGX=C
      FFX=C
      IF((2.000*B).LE.1.20D+02)  GX=DEXP(-2.000*B)
      IF((2.000*BB).LE.1.20D+02)  GGX=DEXP(-2.000*BB)
      IF((DABS(BB-AA)).LE.1.00D+02)  FFX=DEXP(BB-AA)
      F(1)=1.0D0/A
      FF(1)=FFX/AA
      G(1)=(1.000-GX)/B
      GG(1)=(1.000-GGX)/BB
      GM=-1.000
      DO 10 K=2,NM3
      F(K)=DF(K)*F(K-1)/A+F(1)
      FF(K)=DF(K)*FF(K-1)/AA+FF(1)
      G(K)=(DF(K)*G(K-1)+GM-GX)/B
      GG(K)=(DF(K)*GG(K-1)+GM-GGX)/BB
   10 GM=-GM

      DO 400 IP=1,NI
      HQ0=C
      HQ1=C
      DO 500 IQ=1,NJ
```

```
      HR0=C
      HR1=C
      CR=1.000
      DO 600 IR=1,2
      HS0=C
      HS1=C
      CS=1.000
      DO 700 IS=1,2
      N=NXC(1)+NXC(J)+IR+IS-IP-IQ-2
      M=IP+IQ+(R+(S-3
      Q0=FF(N)*GG(M)
      Q1=CV(1)*(FF(N+1)*GG(M)-FF(N)*GG(M+1))
      Q2=CV(2)*(FF(N+2)*GG(M)-2.000*FF(N+1)*GG(M+1)+FF(N)*GG(M+2))
      Q3=CV(3)*(FF(N+3)*GG(M)-3.000*(FF(N+2)*GG(M+1)
     *   -FF(N+1)*GG(M+2))-FF(N)*GG(M+3))
      Q=ZZ1*F(N)*G(M)+ZZ2*(Q0+R2*(Q1+R2*(Q2+R2*Q3)))
      HS0=HS0+Q
      N=NXC(1)+NXC(J)+4-2*IR-IP-IQ
      M=(P+(Q+2*(S-3
      Q0=FF(N)*GG(M)
      Q1=CV(1)*(FF(N+1)*GG(M)-FF(N)*GG(M+1))
      Q2=CV(2)*(FF(N+2)*GG(M)-2.000*FF(N+1)*GG(M+1)+FF(N)*GG(M+2))
      Q3=CV(3)*(FF(N+3)*GG(M)-3.000*(FF(N+2)*GG(M+1)
     *   -FF(N+1)*GG(M+2))-FF(N)*GG(M+3))
      Q=ZZ1*F(N)*G(M)+ZZ2*(Q0+R2*(Q1+R2*(Q2+R2*Q3)))
      HS1=HS1+Q*CS
  700 CS=-CS
      HR0=HR0+HS0
      HR1=HR1+HS1*CR
  600 CR=-CR
      CQ=OG(NJ)/OG(IQ)/OG(NJ-IQ+1)
      HQ0=HQ0+HR0*CQ
      HQ1=HQ1+HR1*CQ
  500 CONTINUE
      CP=OG(N1)/OG(IP)/OG(N1-IP+1)
      HP0=HP0+HQ0*CP
      HP1=HP1+HQ1*CP
  400 CONTINUE
      CJ=COEF(J)*(R2**NM)
      HJ0=HJ0+HP0*CJ
      HJ1=HJ1+HP1*CJ
  300 CONTINUE
      C1=COEF(1)
      HI0=HI0+HJ0*C1
      HI1=HI1+HJ1*C1
  200 CONTINUE
      HZ=-3.000*HI0/2.000/Z2/Z2
      HX=-3.000*HI1/4.000/Z2/Z2

      CD=V*V*TIME(IT)*TIME(IT)/R1(IT)/R1(IT)
      SD=RHO*RHO/R1(IT)/R1(IT)
      HHPX(IT)=CD*HX+SD*HZ
      HHPZ(IT)=CD*HZ+SD*HX
      HHPX(ITT)=HHPX(IT)
      HHPZ(ITT)=HHPZ(IT)
  100 CONTINUE

      RETURN
      END
      SUBROUTINE DER(T,Y,YP)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 SS,A,B,ADOT,BDOT,ENT,CMI
      COMPLEX*16 S12,S21,H12,H21,SAB,SBA,HAB,HBA
      COMMON/MATRX/S12(150),H12(150),H21(150),H11(150),H22(150)
      COMMON/IMPCT/V,RHO,RM,TZ,P1,NT,NNT,NNT1
      COMMON/WFA/EALP,ZA
      COMMON/WFB/EBTA,ZB
      COMMON/IMPCT2/R(150),TIME(150)
      DIMENSION Z(150),ARG(5),VAL(5),Y(4),YP(4)

      CMI=DCMPLX(0.000,-1.000)
      DO 301 I=1,NT
  301 Z(I)=S12(I)
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DAINT(T,ARG,VAL,Z1,5,1.E-5,IER)
      DO 302 I=1,NT
  302 Z(I)=S12(I)*CMI
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DAINT(T,ARG,VAL,Z2,5,1.E-5,IER)
      SAB=DCMPLX(Z1,Z2)
      DO 303 I=1,NT
  303 Z(I)=H12(I)
      CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
      CALL DAINT(T,ARG,VAL,Z1,5,1.E-5,IER)
      DO 304 I=1,NT
```

```
304 Z(I)=H12(I)*CMI
    CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
    CALL DAL1(T,ARG,VAL,Z2,5,1.E-5,IER)
    HAB=DCMPLX(Z1,Z2)
    DO 305 I=1,NT
305 Z(I)=H21(I)
    CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
    CALL DAL1(T,ARG,VAL,Z1,5,1.E-5,IER)
    DO 306 I=1,NT
306 Z(I)=H21(I)*CMI
    CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
    CALL DAL1(T,ARG,VAL,Z2,5,1.E-5,IER)
    HBA=DCMPLX(Z1,Z2)
    DO 307 I=1,NT
307 Z(I)=H11(I)
    CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
    CALL DAL1(T,ARG,VAL,Z1,5,1.E-5,IER)
    HAA=Z1
    DO 308 I=1,NT
308 Z(I)=H22(I)
    CALL DATSM(T,TIME,Z,NT,1,ARG,VAL,5)
    CALL DAL1(T,ARG,VAL,Z2,5,1.E-5,IER)
    HBB=Z2
    SBA=DCONJG(SAB)
    ENT=DCMPLX(0.000,T*(EBTA-EALP))
    SS=(1.000-SAB*SBA)*(0.000,1.000)
    SS=1.000/SS
    A=DCMPLX(Y(1),Y(2))
    B=DCMPLX(Y(3),Y(4))
    ADOT=SS*(A*(HAA-SAB*HBA)+B*(HAB-SAB*HBB)*CDEXP(ENT))
    BDOT=SS*(B*(HBB-SBA*HAB)+A*(HBA-SBA*HAA)*CDEXP(-ENT))
    YP(1)=ADOT
    YP(2)=ADOT*CMI
    YP(3)=BDOT
    YP(4)=BDOT*CMI
    RETURN
    END
    SUBROUTINE CDUPEQ(N,NCODE)
    IMPLICIT REAL*8 (A-H,O-Z)
    COMPLEX*16 GG,SUM,A,X,ENT,U,CMI,PRMP
    COMPLEX*16 XAB(600),XBA(600)
    COMPLEX*16 CA(600),CB(600),DELTA(600)
    COMPLEX*16 S12,S21,H12,H21
    COMPLEX*16 SAB(600),SBA(600),HAB(600),HBA(600)
    COMPLEX*16 ALP(600),BTA(600),AA(600),BB(600),CA(600),CB(600)
    COMPLEX*16 Y1,Y2,Y3
    COMMON/MATRX/S12(150),H12(150),H21(150),H11(150),H22(150)
    COMMON/OPTN/LIM(7,MM,IPNCH,IREO,IPNT1,IPNT2,N1TP,NDAB,NCAB
    DIMENSION HAA(600),HBB(600)
    DIMENSION ARG(10),VAL(10),TIME(600),R(600),Z(150)
    COMMON/IMPCT1/V,RHO
    COMMON/IMPCT2/RA(150),T(150)
    COMMON/IMPCT3/EB(20),EIMPT(10),NRG,NEV,LEV,IPB
   *,PRB(2,10,20),BRPB(2,10,20),PRMP(2,10,20)
    COMMON/WFA/EALP,ZA
    COMMON/WFB/EBTA,ZB


    CMI=DCMPLX(0.000,-1.000)
    SET UP TIME VECTOR
    T1=T(1)
    H=-16.000*T1/(31.000*MM)
    K=0
    MM1=MM+1
 92 CONTINUE
    DO 91 I=1,MM1
 91 TIME(K*MM+I)=T1+(I-1)*H
    K=K+1
    IF(K.GT.4) GO TO 93
    T1=TIME(K*MM+1)
    H=H/2.000
    GO TO 92
 93 CONTINUE
    NNT=5*MM
    NNT1=NNT+1
    NT=2*NNT+1
    DO 98 I=1,NNT1
    II=NT-(I-1)
    TIME(II)=-TIME(I)
    R(II)=DSQRT(RHO**2+(V*TIME(II))**2)
    R(III)=R(1)
 98 CONTINUE
    EKEV=25*V*V
    NK=2*N-1
```

```
      DO 500 K=1,NNT1
      KK=NT-(K-1)
      DO 301 I=1,N
  301 Z(I)=S12(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DAL1 (TIME(K),ARG,VAL,Z1,5,1.E-5,IER)
      DO 302 I=1,N
  302 Z(I)=S12(I)*CM1
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DAL1 (TIME(K),ARG,VAL,Z2,5,1.E-5,IER)
      SAB(K)=DCMPLX(Z1,Z2)
      DO 303 I=1,N
  303 Z(I)=H12(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DAL1 (TIME(K),ARG,VAL,Z1,5,1.E-5,IER)
      DO 304 I=1,N
  304 Z(I)=H12(I)*CM1
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DAL1 (TIME(K),ARG,VAL,Z2,5,1.E-5,IER)
      HAB(K)=DCMPLX(Z1,Z2)
      DO 305 I=1,N
  305 Z(I)=H21(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DAL1 (TIME(K),ARG,VAL,Z1,5,1.E-5,IER)
      DO 306 I=1,N
  306 Z(I)=H21(I)*CM1
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DAL1 (TIME(K),ARG,VAL,Z2,5,1.E-5,IER)
      HBA(K)=DCMPLX(Z1,Z2)
      DO 307 I=1,N
  307 Z(I)=H11(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DAL1 (TIME(K),ARG,VAL,Z1,5,1.E-5,IER)
      HAA(K)=Z1
      DO 308 I=1,N
  308 Z(I)=H22(I)
      CALL DATSM(TIME(K),T,Z,N,1,ARG,VAL,5)
      CALL DAL1 (TIME(K),ARG,VAL,Z2,5,1.E-5,IER)
      HBB(K)=Z2
  500 CONTINUE

      DC=1.000
      IF(NCODE.EQ.2)  DC=-1.000
      DO 600 I=1,NNT1
      II=NT-(I-1)
      SBA(I)=DCONJG(SAB(I))
      SAB(II)=DCONJG(SAB(I))*DC
      SBA(II)=DCONJG(SBA(I))*DC
      HAB(II)=DCONJG(HAB(I))*DC
      HBA(II)=DCONJG(HBA(I))*DC
      HAA(II)=HAA(I)
      HBB(II)=HBB(I)

      SS=1.000-SAB(I)*SBA(I)
      AA(I)=(HAA(I)-SAB(I)*HBA(I))/SS
      BB(I)=(HBB(I)-SBA(I)*HAB(I))/SS
      IF(I.EQ.NNT1)  GO TO 600
      AA(II)=DCONJG(AA(I))
      BB(II)=DCONJG(BB(I))
  600 CONTINUE

      IF(IPNT1.EQ.0) GO TO 11
      PRINT 520
  520 FORMAT(' ',////50X,'INTERPOLATED MATRIX ELEMENTS'///)
      PRINT 521
  521 FORMAT('   TIME...R   SAB...HAB...HBA...HAA...HBB  '//)
      DO 507 I=1,NNT1,NITP
  507 PRINT 508,TIME(I),P(I),SAB(I),HAB(I),HBA(I),HAA(I),HBB(I)
  508 FORMAT(2(1X,F10.5),5X,4(1PD11.4,1X,1P011.4,2X))
   11 CONTINUE


      NT1=NT-1
      DELTA(1)=(0.000,0.000)
      ALP(1)=(0.000,0.000)
      BTA(1)=(0.000,0.000)
      DO 700 J=2,NT1
      X1=TIME(J-1)
      X2=TIME(J)
      X3=TIME(J+1)
      Y1=AA(J-1)
      Y2=AA(J)
      Y3=AA(J+1)
      ALP(J)=ALP(J-1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      Y1=BB(J-1)
```

```
      Y2=BB(J)
      Y3=BB(J+1)
      BTA(J)=BTA(J-1)+SUM(X1,X2,X3,Y1,Y2,Y3)
  700 DELTA(J)=ALP(J)-BTA(J)
      X3=TIME(NT)
      X2=TIME(NT1)
      X1=TIME(NT1-1)
      Y1=AA(NT)
      Y2=AA(NT1)
      Y3=AA(NT1-1)
      ALP(NT)=ALP(NT1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      YI=BB(NT)
      Y2=BB(NT1)
      Y3=BB(NT1-1)
      BTA(NT)=BTA(NT1)+SUM(X1,X2,X3,YI,Y2,Y3)
      DELTA(NT)=ALP(NT)-BTA(NT)

      DO 800 K=1,NT
      DELTA(K)=DELTA(K)-TIME(K)*(EALP-EBTA)
      SS=1.000-SAB(K)*SBA(K)
      Y3=CDEXP(DELTA(K)/CM())
      XAB(K)=(HAB(K)-SAB(K)*HBB(K))*CMI*Y3/SS
      XBA(K)=(HBA(K)-SBA(K)*HAA(K))*CM(/Y3/SS
      DA(K)=(1.000,0.000)
  800 DB(K)=(0.000,0.000)


      SS3=0.000
      DO 1000 KI=1,LIMIT
      DA(1)=(1.000,0.000)
      DB(I)=(0.000,0.000)
      DO 900 J=2,NT1
      X1=TIME(J-1)
      X2=TIME(J)
      X3=TIME(J+1)
      Y1=XBA(J-1)*DA(J-1)
      Y2=XBA(J)*DA(J)
      Y3=XBA(J+1)*DA(J+1)
  900 DB(J)=DB(J-1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      X3=TIME(NT)
      X2=TIME(NT1)
      X1=TIME(NT1-1)
      Y1=XBA(NT)*DA(NT)
      Y2=XBA(NT1)*DA(NT1)
      Y3=XBA(NT1-1)*DA(NT1-1)
      DB(NT)=DB(NT1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      DO 950 J=2,NT1
      X1=TIME(J-1)
      X2=TIME(J)
      X3=TIME(J+1)
      Y1=XAB(J-1)*DB(J-1)
      Y2=XAB(J)*DB(J)
      Y3=XAB(J+1)*DB(J+1)
  950 DA(J)=DA(J-1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      X3=TIME(NT)
      X2=TIME(NT1)
      X1=TIME(NT1-1)
      Y1=XAB(NT)*DB(NT)
      Y2=XAB(NT1)*DB(NT1)
      Y3=XAB(NT1-1)*DB(NT1-1)
      DA(NT)=DA(NT1)+SUM(X1,X2,X3,Y1,Y2,Y3)
      IF(IPNT2.EQ.0)  GO TO 13
      PRINT 21, K1
   21 FORMAT('1     ITERATION=',I3/)
      PRINT 125
  125 FORMAT(4X,4X,'TIME',25X,'DA',30X,'DB'/)
      DO 22 I=1,NT,NDAB
   22 PRINT 23, TIME(I),DA(I),DB(I)
   23 FORMAT(2X,F12.8,4X,4E16.7)
   13 CONTINUE

      SS1=0.000
      SS2=0.000
      DO 960 I=1,NT
      SS1=SS1+CDABS(DA(I)-DB(I))
  960 SS2=SS2+CDABS(DB(I))
      SS3=SS1/SS2
 1000 CONTINUE


 2000 CONTINUE
      PRINT 3
    3 FORMAT(///' R    UA..UB   XAB..XBA    CA..CB..PROB..UNITARITY'//)
      DO 50 I=1,NT,NDAB
      CA(I)=DA(I)*CDEXP((0.000,-1.000)*ALP(I))
```

```
      CB(I)=CB(I)*COEXP((0.000,-1.000)*BTA(I))
      PROB=CB(I)*CONJG(CB(I))
      ENN=TIME(I)*(EALP-EBTA)
      ENT=CMPLX(0.000,-ENN)
      UNIT=PROB+CA(I)*CONJG(CA(I))
      U=CONJG(CA(I))*CB(I)*SAB(I)*COEXP(ENT)
      UNIT=UNIT+U+CONJG(U)
      UNIT=1.000-UNIT
      SS=1.000-SAB(I)*SBA(I)
      X1=(HAA(I)-SAB(I)*HBA(I))/SS-EALP
      X2=(HBB(I)-SBA(I)*HAB(I))/SS-EBTA
      Y1=(HAB(I)-SAB(I)*HBB(I))/SS
      Y2=(HBA(I)-SBA(I)*HAA(I))/SS
   50 PRINT 60, R(I),X1,X2,Y1,Y2,CA(I),CB(I),PROB,UNIT
   60 FORMAT(1X,F9.3,12(1X,1PD9.2))


      BP=PROB*EB(IPB)
      PRINT 229,BP,EB(IPB),EIGPT(LEV)
  229 FORMAT(/'    PROB*RHO=',E12.5,'  AT RHO=',F8.4,'   EKEV=',F10.3/)
      PRB(NCODE,LEV,IPB)=PROB
      BRPB(NCODE,LEV,IPB)=BP
      PRMP(NCODE,LEV,IPB)=CB(NT)
      RETURN
      END
```

APPENDIX III

Appendix III includes the publication of which this work is an extension.

PHYSICAL REVIEW A VOLUME 17, NUMBER 5 MAY 1978

Two-state atomic expansion methods for electron capture from multielectron atoms by fast protons

C. D. Lin, S. C. Soong,* and L. N. Tunnell
Department of Physics, Kansas State University, Manhattan, Kansas 66506
(Received 19 May 1977; revised manuscript received 13 October 1977)

The two-state, two-center atomic expansion method of Bates for charge transfer is generalized to calculate cross sections of electron capture from inner shells of multielectron atoms by fast protons. In the limit of small capture probabilities, the connections of the present approach with various first-order Born theories are investigated. It is shown that these Born methods for electron capture of multielectron atoms can be obtained from the present approach by further approximations. The method is applied to obtain cross sections of electron capture from C, N, O, Ne, and Ar atoms by fast protons in the energy region where the projectile velocity is nearly equal to the K-shell-electron orbital velocity of these atoms. Results of the calculations are compared with experimental measurements.

I. INTRODUCTION

The transfer of an electron from target to projectile during ion-atom collisions is the subject of recent experimental and theoretical investigations. It is known that this process plays an important role in vacancy production in ion-atom collisions.[1]

For collisions in which the projectile velocity is much smaller than the characteristic orbital velocity of the active electron to be transferred, the molecular theory (MO) of Fano and Lichten[2] has been applied successfully to explain qualitatively the observed low-energy ion-atom collision phenomena.[3] Recent developments by Briggs and Macek,[4] and by Taulbjerg *et al.*[5] have put the MO theory in quantitative form for K-shell vacancy transfer in symmetric and asymmetric ion-atom collisions.

For fast collisions such that the projectile velocity is comparable to or greater than the characteristic orbital velocity of the active electron, the capture of bound electrons from the target atom is less well understood. Whereas the first Born approximation or its variations have been useful in describing excitation and ionization in fast collisions,[6] considerable contention still persists in the application of the first Born theory in rearrangement collisions, particularly for the electron-capture process.[6,7,8] Even in the simplest resonant charge-transfer process, $p+\mathrm{H}(1s) \rightarrow \mathrm{H}(1s)+p$, the various first Born theories predict substantially different capture cross sections. Attempts to generalize these first-order Born theories to multielectron ion-atom collisions create even further questions.

Historically, the $p+\mathrm{H}(1s) \rightarrow \mathrm{H}(1s)+p$ resonant charge exchange has been calculated in the Oppenheimer,[9] Brinkman, and Kramers (OBK)[10] approximation. In the OBK approximation, the nuclear-nuclear interaction was completely neglected in evaluating the first Born transition amplitude. This is justified in that the nuclear-nuclear interaction can only deflect the trajectory of the projectile and does not change substantially the total electron-capture cross sections. Later, similar first-order approximations were adopted by Bates and Dalgarno,[11] and by Jackson and Schiff[12] (JS), but with the internuclear potential also included in the first Born amplitude.[11] As argued by Bates and Dalgarno, the complete nuclear-nuclear interaction is included in the perturbation on the grounds that this would compensate to some extent for the nonorthogonality of the wave functions of the initial and final states, and would consequently lead to more realistic cross sections.[13] Interestingly, the cross sections calculated in this method are much smaller than those calculated by the OBK method and agree much better with experimental data.

Recently, both the OBK and JS methods have been generalized to calculate electron-capture cross sections in multielectron ion-atom collisions.[14-17] Like the prediction in the proton-hydrogen resonant capture, the OBK approximation always predicts cross sections much higher than experimental results. Diverse efforts have been attempted to correct this either by reducing the OBK prediction by a semiempirical factor,[18,19] by semiempirical method,[20] or by introducing different amounts of core-core interactions.[21]

The straightforward generalization of the JS method includes the interaction between the two bare nuclei in the perturbation.[12,15,17] This method apparently fails because the predicted capture cross sections are a few orders of magnitude too high. For example, cross sections for the

capture of K-shell electrons of Ar atoms by protons are predicted to he about 320 times larger thao experimental data.[16]

Much of the discrepancy mentioned in the above is due to the fact that no proper allowance had been made for the nonorthogonality of the initial- and final-state wave functions. Bates[22] was the first to note that if the nonorthogonality is properly treated, the difficulty formally associated with the choice of internuclear potential can be resolved.

Io this paper, we extend the method of Bates to electron capture in multielcctron ion-atom collisions within the independent-electroo approxlmation. This approximation treats only the electroo to be transferred as active; the others are treated as passlve and provide only screening during the collision process.

In Sec. II, the Bates method is revlewed. The connections of Bates method, in the limit of small capture probability, to the different flrst Born methods are discussed in Sec. III. In Sec. IV, this method is applied to the capture of K-shell electrons of C, N, O, Ne, and Ar atoms by fast protons. The validity of the present method is dlscussed in Sec. V.

II. ATOMIC EXPANSION METHOD

Developed by Bates in 1958, the atomic expansion method was designed to properly account for thc nonorthogonality of the initial- and final-state wave functions in the electron-capture process.

In the Bates method, the motion of the electrons and the nuclei in ion-atom collisions is separated by using the perturbed-stationary-state (pss) method[23]; the motion of the nuclei is treated classically. The attractive nuclear field experienced by the electrons during the collision depends upon the trajectories of the two nuclei. In this paper, we are dealing with high-velocity projectiles; thus straight-line trajectories will be adopted.

To study electron-capture problems in multielectron ion-atom collisions, many approximations can be made if only the capture of inner-shell electrons is to be treated. In principle, the Bates approach can be used to deal with multielectron wave functions. However, it has been shown that electron correlation and exchange effects are not very important for the electron-capture process in the proton-helium system.[24-26] We thus expect the independent-electron model to be adequate, particularly for capture from the inner shells of atoms.

In this approximation, the wave function of the active elcctron is goveroed by the time-dependent Schrödinger equation

$$\left(H_e - i\frac{\partial}{\partial t}\right)\Psi(\vec{r},t) = 0, \tag{1}$$

where

$$H_e = -\tfrac{1}{2}\nabla^2 - Z_A/r_A - Z_B/r_B \tag{2}$$

is the effective Hamiltonian of the actlve electron. In Eq. (2), Z_A and Z_B are the effective charges experienced by the electron; r_A aod r_B are the positions of thc electron with respect to the target A aod to the projectile B, respectively. Atomic units will be used.

Equations (1) and (2) are to be solved with proper boundary conditions at $t = -\infty$. The method adopted by Bates is to expand $\Psi(\vec{r},t)$ in terms of the traveling eigenstates of the target aod of the projectile. The following derivation can be found in the paper of Bates[23] or in the book by McDowell and Colemao.[6] We will summarize it below for later discussion.

Thc time-dependent wave function $\Psi(\vec{r},t)$ can be expanded generally as

$$\Psi(\vec{r},t) = \sum_n a_n(t)\,\phi_n(\vec{r}_A)\exp\left[-i\left(\tfrac{1}{2}\vec{v}\cdot\vec{r} + \tfrac{1}{8}v^2 t + \epsilon_n t\right)\right] + \sum_m b_m(t)\phi_m(\vec{r}_B) \times \exp\left[-i\left(-\tfrac{1}{2}\vec{v}\cdot\vec{r} + \tfrac{1}{8}v^2 t + \epsilon_m t\right)\right], \tag{3}$$

where $\phi_n(\vec{r}_A)$ $[\phi_m(\vec{r}_B)]$ is the stationary eigenfunction of the target (projectile) with eigenenergy ϵ_n (ϵ_m), $\vec{v}$ is the velocity of the projectile in the laboratory frame and $\vec{r}$ is the position vector of the electron with respect to the midpoint of the internuclear axis.[27] In Eq. (3), the velocity-dependent exponents are introduced to preserve translational invartance.

To describe electron capture, the simplest approximation to Eq. (3) is to retain only the two states which are relevant to the capture process, the initial state of the target and the final state of the projectile. To simplify the notation, we rewrite Eq. (3) (in a self-evident way) as

$$\Psi(\vec{r},t) = a(t)\,\phi_A\exp\left[-i\left(\tfrac{1}{2}\vec{v}\cdot\vec{r} + \tfrac{1}{8}v^2 t + \epsilon_A t\right)\right] + b(t)\,\phi_B\exp\left[-i\left(-\tfrac{1}{2}\vec{v}\cdot\vec{r} + \tfrac{1}{8}v^2 t + \epsilon_B t\right)\right]. \tag{4}$$

Substitution of Eq. (4) into Eq. (1) yields a set of coupled equations:

$$\begin{aligned} i(1-s^2)\dot{a} &= a(h_{AA} - s_{AB}h_{BA}) + b(h_{AB} - s_{AB}h_{BB})\,e^{i\omega t}, \\ i(1-s^2)\dot{b} &= b(h_{BB} - s_{BA}h_{AB}) + a(h_{BA} - s_{BA}h_{AA})\,e^{-i\omega t}, \end{aligned} \tag{5}$$

where $\omega = \epsilon_A - \epsilon_B$ and

$$
\begin{aligned}
s_{AB} &= \int \phi_A^* \phi_B e^{(i\vec{v}\cdot\vec{r})} d\tau, \\
s_{BA} &= \int \phi_B^* \phi_A e^{(-i\vec{v}\cdot\vec{r})} d\tau, \\
h_{AB} &= \int \phi_A^* (-Z_A/r_A) \phi_B e^{(i\vec{v}\cdot\vec{r})} d\tau, \\
h_{BA} &= \int \phi_B^* (-Z_B/r_B) \phi_A e^{(-i\vec{v}\cdot\vec{r})} d\tau, \\
h_{AA} &= \int \phi_A^* (-Z_B/r_B) \phi_A d\tau, \\
h_{BB} &= \int \phi_B^* (-Z_A/r_A) \phi_B d\tau,
\end{aligned}
\tag{6}
$$

and where the integration is over the electronic coordinates. The identities $s_{BA} = s_{AB}^*$ and $s^2 = s_{AB} s_{BA}$ are obvious.

Introducing the transformation

$$
\begin{aligned}
a(t) &= d_A(t) \exp\left(-i \int_{-\infty}^{t} \alpha(t')\, dt'\right), \\
b(t) &= d_B(t) \exp\left(-i \int_{-\infty}^{t} \beta(t')\, dt'\right).
\end{aligned}
\tag{7}
$$

Eqs. (5) are simplified to

$$
\begin{aligned}
i\dot{d}_A &= \frac{h_{AB} - s_{AB} h_{BB}}{1 - s^2} e^{i\omega t + i\delta} d_B, \\
i\dot{d}_B &= \frac{h_{BA} - s_{BA} h_{AA}}{1 - s^2} e^{-i\omega t - i\delta} d_A,
\end{aligned}
\tag{8}
$$

where

$$
\delta = \int_{-\infty}^{t} [\alpha(t') - \beta(t')]\, dt' \tag{9}
$$

and

$$
\begin{aligned}
\alpha(t) &= (h_{AA} - s_{AB} h_{BA})/(1 - s^2), \\
\beta(t) &= (h_{BB} - s_{BA} h_{AB})/(1 - s^2).
\end{aligned}
\tag{10}
$$

Equations (8) are to be solved with the boundary conditions $d_A(-\infty) = 1$, $d_B(-\infty) = 0$ for each impact parameter ρ and each energy. The total capture cross section per atom is obtained from

$$
Q = 2\pi N_A \int_0^{\infty} \rho\, d\rho\, p(\rho), \tag{11}
$$

where $p(\rho) = |b(+\infty)|^2$ is the capture probability and N_A is the number of equivalent electrons in the target shell from which the active electron is captured.

III. CONNECTIONS WITH OTHER BORN APPROXIMATIONS

For collisions in which the capture probabilities are small, the capture amplitude can be solved from Eqs. (8) by first-order approximation. If we set $d_A(t) = 1$, then $d_B(+\infty)$ is given by

$$
d_B(+\infty) = -i \int_{-\infty}^{\infty} \frac{h_{BA} - s_{BA} h_{AA}}{1 - s^2} e^{-i\omega t - i\delta}\, dt. \tag{12}
$$

In Eq. (12), the transition amplitude $d_B(+\infty)$ can be easily shown to be independent of any arbitrary internuclear potentials added to the definitions of the matrix elements h_{BA} and h_{AA}. This is due to the fact that the nonorthogonality of initial and final states has been properly accounted for in Eqs. (5) through the introduction of overlap integrals s_{AB} and s_{BA}.

The δ term in Eq. (12) represents the distortion of the electron wave function in the nuclear field of the projectile and the target in the two-state atomic expansion approximation. If this distortion is neglected, then Eq. (12) becomes

$$
d_B(+\infty) = -i \int_{-\infty}^{\infty} \frac{h_{BA} - s_{BA} h_{AA}}{1 - s^2} e^{i\omega t}\, dt. \tag{13}
$$

For high-velocity collisions, $s^2 \ll 1$, Eq. (13) can then be written explicitly as

$$
\begin{aligned}
d_B(+\infty) = -i \int_{-\infty}^{\infty} dt\, d\tau\, \phi_B \left[-\frac{Z_B}{r_B} - h_{AA}\right] \\
\times \phi_A \exp\{-i(\vec{v}\cdot\vec{r} + \omega t)\},
\end{aligned}
\tag{14}
$$

in a form similar to the first Born transition amplitude with $(Z_B/r_B) - h_{AA}$ as the interaction "potential." For capture from the K shell of target A to the K shell of projectile B, h_{AA} is

$$
h_{AA} = (Z_B/R)[-1 + (1 - Z_A R) e^{-2Z_A R}]. \tag{15}
$$

For the charge-exchange $p + \mathrm{H}(1s) - H(1s) + p$, $Z_A = Z_B = 1$, Eq. (14) becomes identical to the distorted-wave approximation for electron capture derived by Bassel and Gerjuoy.[28] Thus, Eq. (14) is the generalization of their method to arbitrary Z_A and Z_B. Incidentally, Eq. (14), or more rigorously, Eq. (13), can also be derived from the usual first Born theory if the final-state wave function is required to be orthogonal to the initial-state wave function. Thus, we show that in the limit of small capture probabilities, the two-state atomic expansion method of Bates, the distorted-wave approximation of Bassel and Gerjouy and the first Born theory are all equivalent if the orthogonalized final state is used in the first Born theory.

To explore the meaning of Eq. (14) in more detail, we plot, in Fig. 1, $-R\, h_{AA}/Z_B$ as a function of $Z_A R$, where R is the internuclear separation. The function h_{AA} approaches zero as $Z_A R \to 0$ and approaches $-Z_B/R$ as $Z_A R \to \infty$. If

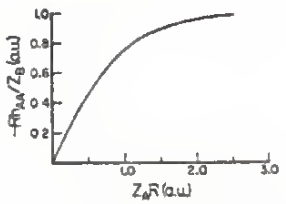


FIG. 1. Plot of $-Rh_{AA}/Z_B$ as a function of Z_AR.

h_{AA} is chosen to be zero in Eq. (14), we recover the usual OBK approximation. From Eq. (13), this is *equivalent* to neglecting the nonorthogonality of initial and final states as was done in the OBK approximation (by setting $s_{BA}\equiv 0$). On the other hand, if the large-R limit of h_{AA} is used in Eq. (14), the expression in the squared bracket becomes $[-Z_B/r_B+Z_B/R]$. In the $p+\mathrm{H}(1s)-\mathrm{H}(1s)+p$ capture problem, it becomes $[-1/r_B+1/R]$ and the second term resembles the internuclear interaction between the protons. This is equivalent to the method of JS in which the internuclear potential is included in the first Born transition amplitude. Therefore, we can interpret that the introduction of the internuclear interaction into the first Born transition amplitude has the effect of partially accounting for the nonorthogonality of the initial and final states in the $p+\mathrm{H}(1s)-\mathrm{H}(1s)+p$ reaction at intermediate and large R. However, this similarity *cannot* be generalized to ion-atom collisions of arbitrary Z_A and Z_B. The large R limit of h_{AA} is Z_B/R instead of the internuclear interaction Z_AZ_B/R. This partially explains why the straightforward generalization of the JS method to ion-atom collisions by including a full internuclear interaction results in unrealistic capture cross sections. Incidentally, the large-R limit of h_{AA} has also been introduced recently[15] in the Born amplitude, under the assumption of almost complete screening of the target nucleus charge by the passive electrons. This assumption is *not* valid for the capture of K-shell electrons. It is better to interpret Z_B/R as an approximation of the nonorthogonality contribution to the Born amplitude for electron capture and has no relation with the internuclear potential.

It is not difficult to understand why the JS or the Born method of Ref. 15 usually gives better absolute total capture cross section than the OBK approximation. In Fig. 1, h_{AA} is well approximated by its large-R limit $-Z_B/R$ for R near or greater than the K-shell radius. Thus, if the total electron capture comes primarily from large impact parameter ρ, such $\rho Z_A\gg 1$, then the JS or the Born method of Ref. 15 will give reasonable total capture cross sections [as compared with that obtained from Eq. (13)]. However, it must be realized that both methods will fail at small ρ or, correspondingly, at large scattering angles. Also, if the total capture cross section comes primarily from small impact parameters, then the total cross sections calculated from these two methods will be wrong.

It might then be speculated that the OBK method is a better approximation for collisions at small impact parameters. This is not quite true. For small impact parameters, the distortion of the active electron wave function by the projectile is very large and cannot be reasonably approximated by any first-order theory, even such as Eqs. (8) and (13).

IV. K-SHELL ELECTRON CAPTURE OF C, N, O, Ne, AND Ar ATOMS BY FAST PROTONS

The two-state atomic expansion method has previously been applied only to simple atomic systems. (See the review by Bransden.[3]) By comparing with experimental data or with more elaborate calculations, it is concluded that the simple two-state calculations predict reasonable capture cross sections when the projectile velocity is not very far away from the characteristic orbital velocity of the active electrons.

Theoretical calculations of electron-capture cross sections from multielectron atoms have been limited to the OBK or other Born methods.[15-20] The results of these calculations are often unreliable. We have applied the two-state atomic expansion method, under the independent-particle approximation as outlined in Sec. II, to calculate the electron-capture cross sections of the K shell of carbon, nitrogen, oxygen, neon, and argon atoms by fast protons.

The numerical method is straightforward. A screened hydrogenic 1s wave function with effective charge $Z_A=Z-\frac{1}{16}$, where Z is the nuclear charge of the target, is used for the target atom and a bare nuclear charge Z_B is used for the projectile. The matrix elements of Eq. (6) are evaluated by transforming the two-centered integrand to prolate spheroidal coordinates (λ,μ,ϕ).[29] Integrations over ϕ and μ can be carried analytically. The integration over λ is done using 24-point Gauss-Laguerre quadrature, although the 32-point formula has been used also to check the accuracy of the integration. The capture amplitude $b(+\infty)$, or equivalently $d(+\infty)$, is obtained by solving the coupled Eqs. (8), either by direct numerical integration or by an iterative method. The latter method is more suitable for calculating small amplitudes. In particular, the first it-

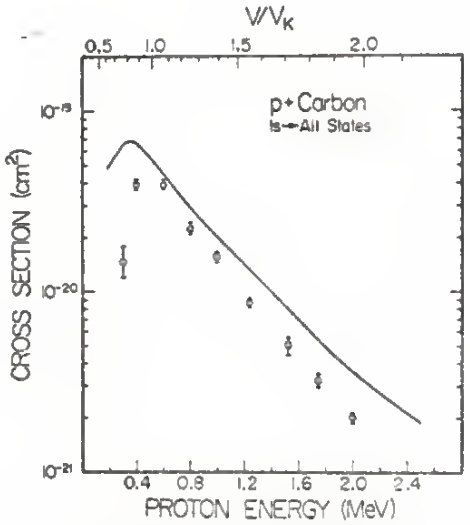


FIG. 2. Electron-capture cross sections from the K shells of carbon atoms by fast protons. The values are the total capture cross sections per target atom, including capture to the exited state of hydrogen atoms. The solid curve is the result of the present calculation. Experimental data are from Rödbro *et al.*, Ref. 33. Also shown are the values of V/V_K, the ratio of the projectile velocity V to the characteristic K-shell orbital velocity of the target atom, defined by $V_K=\sqrt{2I_K}$, where I_K is the K-shell ionization energy.

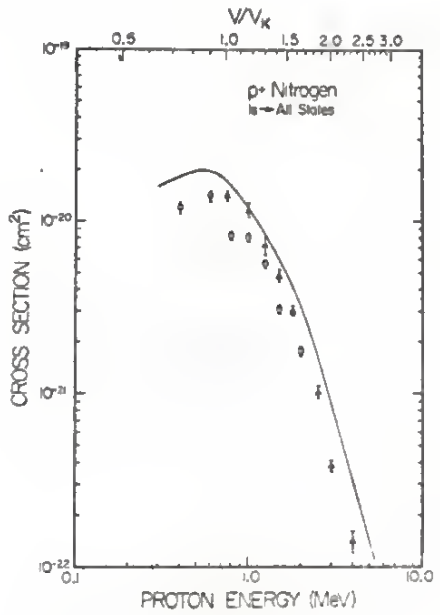


FIG. 3. Same as in Fig. 2, except for nitrogen atoms. Experimental data: Φ, from Rödbro *et al.*, Ref. 33; $\overline{\Delta}$, from Cocke *et al.*, Ref. 32.

erative solution for $d_\beta(+\infty)$ is then given by Eq. (12). Depending upon the systems, usually two or three iterations are enough for desirable accuracy. In solving Eqs. (8), we use experimental K-shell ionization energy for ϵ_A and $-Z_B^2/2$ for ϵ_B. By choosing ϵ_A and Z_A separately, the unitarity condition is not imposed in the calculation. This choice of ϵ_A is desirable because the capture probability, as given by its first-order solution Eq. (12), is dominated by the oscillatory function $e^{-i\omega t}$ in the integrand, as well as the damped oscillation in the matrix elements of $-Z_B/r_B-h_{AA}$. This explains why the OBK approximation (obtained by letting $h_{AA}=0$) usually predicts correct energy dependence for the total capture cross sections, even though the absolute values are often wrong.

The calculated total capture cross sections from the K shells of C, N, O, Ne, and Ar atoms by protons are displayed in Figs. 2–6. They are the total capture cross sections per target atom, including capture to the excited states of the projectile. The theoretical values shown in the figures are obtained from the calculated 1s-1s val-

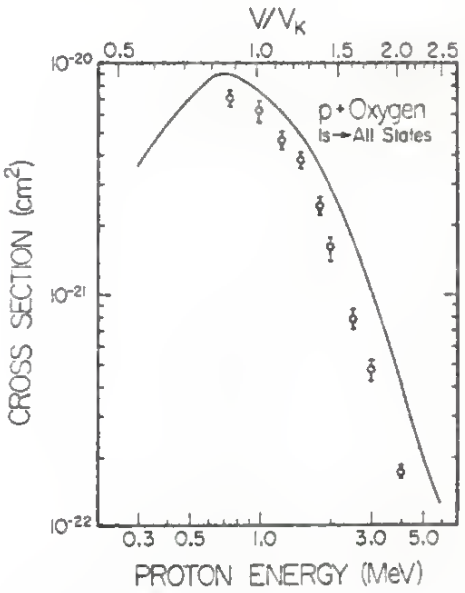


FIG. 4. Same as in Fig. 2 except for oxygen atoms. Experimental data from Cocke *et al.*, Ref. 32.

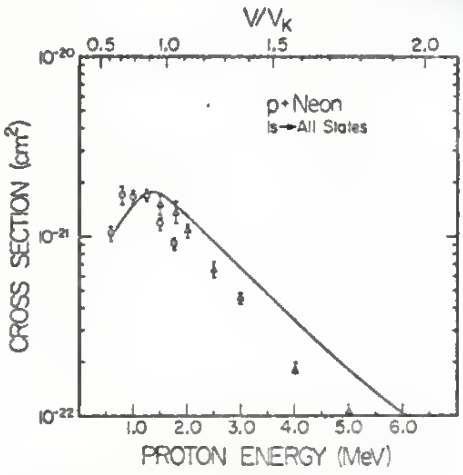


FIG. 5. Same as in Fig. 2 except for neon atoms. Experimental data: ⊕, from Rödbro *et al.*, Ref. 33; ⅄, Cocke *et al.*, Ref. 32.

ues by multiplying 1.2, corresponding to the high velocity $1/v^3$ scaling.[30] Experimental data shown on these figures are from Macdonald *et al.*,[31] Cocke *et al.*,[32] and from Rödbro *et al.*[33] For

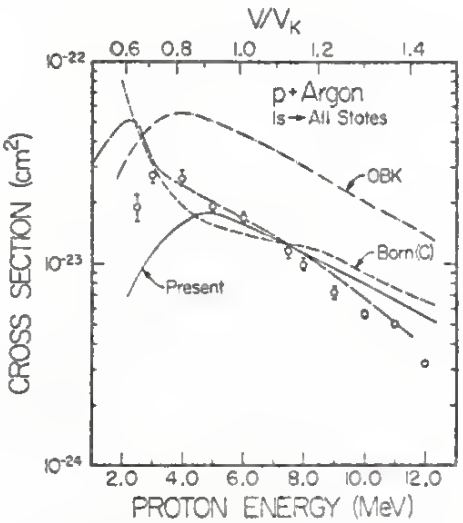


FIG. 6. Same as in Fig. 2 except for argon atoms. Other theoretical results: short-dashed lines, the Born (C) method of Ref. 15, dash-dotted lines, the OBK results of Ref. 15. Long-dashed lines, continuous distorted-wave (CDW) results of Ref. 34. Experimental data are from Macdonald *et al.*, Ref. 31.

C, N, and O, atoms, the experimental data are obtained from measuring capture in CH_4, N_2, and O_2 gases. The experimental K-shell capture cross sections are not expected to change much by any molecular binding effect.

It can be seen from Figs. 2–6 that the calculated values are generally in good accord with experimental data. In Fig. 6, the results of the OBK approximation, the Born method of Omidvar *et al.*,[15] and the continuum distortion-wave method of Belic and McCarroll[34] are also shown for comparison. The OBK predictions shown in Fig. 6 are about three times too large when compared with experimental data. The Born method of Omidvar *et al.*[15] predicts cross sections in reasonable agreement with data at higher energies but the predicted energy dependence differs from the experimental data. The continuum distorted-wave method of Belic and McCarroll[34] also predicts cross sections in excellent agreement with experimental data at the high-energy side,[35] but the energy dependence at the low-energy side is also incorrect.

V. DISCUSSION

From the results of Figs. 2–6, it is clear that the simple two-state expansion method is capable of predicting capture cross sections in *reasonable* agreement with experimental data. However, further improvement of the model is possible. In the following we discuss the limitation of the present method and possible further improvement.

A. Atomic model

In Eqs. (1) and (2), we use the active-electron approximation by disregarding the effects of passive electrons. It is possible to formulate a many-electron theory of electron capture based upon the Bates formulation. In fact, such a theory has been written explicitly by Msezane[36] recently for the two-electron systems. However, the complexity of such a theory for general N-electron problem will make such a formulation impractical in view of the numerical difficulties.

Improvement in the atomic model within the independent-electron approximation can be proceeded by using a more realistic potential $V(r_A)$ for the target atom. For example, the Green-Seilin-Zachor (GSZ) potential of Green *et al.*[37] can be introduced into the Hamiltonian (2).[38] These potentials predict the K-shell ionization energy accurately. We can thus use the eigenstates and eigenenergies generated from this potential in the expansion (4), thus preserving the unitarity relation in the coupled Eqs. (6) and (8). It is hoped that the choice of the more realistic potential will

improve the computed cross sections in the region where the cross section peaks. However, it is *not* expected that the improvement in the atomic model *alone* will make the theoretical calculations agree with experimental data over the entire energy range considered. The convergence of the truncated atomic expansion has to be investigated too.

B. Scattering model

To study the limitation of the two-state atomic expansion method, we examine the well-studied simple reaction $p+\mathrm{H}(1s)\rightarrow\mathrm{H}(1s)+p$. At the low-velocity limit, the potential curves of the quasimolecule H_2^+ are exactly known. These potential curves describe the distortion of the atomic electron wave function by the projectile in the adiabatic limit. By comparing the potential curves calculated by the two-state atomic expansion with the exact H_2^+ potential curves,[39] we can conclude that the two-state representation is adequate for $R \geq 1.0$, but not smaller R. Therefore, we can expect the two-state atomic expansion method adequate for describing the collision $p+\mathrm{H}(1s)\rightarrow\mathrm{H}(1s)+p$ at impact parameters $\rho \geq 1.0$, but not at smaller impact parameters. If the total capture cross section comes primarily from the impact parameters $\rho \geq 1.0$, then the total capture cross section obtained from the two-state atomic expansion will be adequate. This occurs at the intermediate energy region where the projectile velocity nearly matches the orbital velocity of the target electron. As the velocity of the projectile increases, the capture has to occur at smaller impact parameters for the projectile to pick up electrons close to the target nucleus, then the two-state atomic expansion becomes inadequate.[40]

Within the method of Bates, the charge exchange $p+\mathrm{H}(1s)\rightarrow\mathrm{H}(1s)+p$ has been studied by the multistate atomic expansion method[41] by using the Sturmian basis set[42] and by the pseudostate method.[43] In the multistate expansion method of Ref. 41, excited hydrogenic orbitals are used in the expansion of Eq. (3). It was found that the electron-transfer cross sections are not changed substantially by the inclusion of the excited states. However, this does not imply that the two-state calculation has converged in all the cases studied. It actually happens that the excited states included in the expansion are not important for this particular reaction. This can be easily understood from the discussion in the previous paragraph. It was shown there that the inadequacy of the two-state atomic expansion occurs at small $R \leq 1.0$ where the electronic motion cannot be represented by the excited-state wave functions of the target or the projectile because of the diffuse nature of these functions, but can only be represented by the continuum functions. The Sturmian basis set and the pseudostates are all chosen in the hope that the continuum states are thus partially accounted for. In Table I, we compare the two-state calculation of McCarroll[44] and the pseudostate calculation of Cheshire *et al.*[43] for the reaction $p+\mathrm{H}(1s)\rightarrow\mathrm{H}(1s)+p$. We can see the two-state calculations are quite adequate for $E_p \leq 100$ keV, but as E_p increases, the two-state calculations overestimate the capture cross sections by a factor of 2 as the contributions of capture from small impact parameters to the total cross section increase.

From Table I and the discussion above, it becomes clear that the two-state approximation is best in the energy region where $v_p \approx v_{el}$. The method becomes inadequate as the projectile energy increases, eventually reducing to the OBK approximation at extremely high energies. It is interesting to mention that this implies all the first Born approximations for electron transfers are inadequate, even at high energies. This is not inconsistent with the conclusion of Drisko[45] that the second Born term is more important than the first Born term in the extreme limit of high energies.

By examining the results of our calculations in Figs. 2–6, our values at the high-energy side are about a factor of 2 higher than experimental data. Thus one might speculate that the continuum states are also very important in our calculations. At this moment we tend to believe this is not the case. The discrepancy probably can be reduced

TABLE I. Comparison of the two-state calculations [a] and the pseudostate calculations [b] for the total capture cross sections for $p+\mathrm{H}(1s)\rightarrow\mathrm{H}(1s)+p$ reactions. The cross sections are given in cm^2 $A(-B)=A\times10^{-B}$

Energy (keV)	Two-state [a]	Pseudostate [b]
4	1.15(−15) [c]	1.13(−15)
10	7.79(−16) [c]	7.77(−16)
15	5.60(−16)	5.81(−16)
20	4.18(−16) [c]	4.14(−16)
25	2.76(−16)	2.93(−16)
40	1.51(−16) [c]	1.13(−16)
60	4.98(−17) [c]	4.20(−17)
100	1.012(−17)	8.69(−18)
300	1.711(−19)	8.51(−20)
1000	5.12(−22)	2.63(−22)

[a] Two-state calculations from McCarroll, Ref. 41.
[b] Pseudostate calculations from Cheshire *et al.*, Ref. 42.
[c] Interpolated from Ref. 41.

by including a few more atomic states of the target atom into expansion (3). It is noted that some excited orbitals of the target atoms have radii smaller or comparable to the radius of the 1s orbital of the hydrogen atom. The restriction of the two-state atomic expansion with basis functions differing substantially in the size of orbitals might have forced those amplitudes which would have otherwise ended up in the direct excitation channels into the electron-capture channel. The validity of this speculation has to be substantiated by actual calculations.

In summary, we applied the two-state atomic expansion method to compute the electron-capture cross sections of C, N, O, Ne, and Ar atoms. Comparisons of this method with other first-order Born methods are made to elucidate the region of validity of these methods. The limitation and possible further improvement of the present model is also discussed.

Note added in proof. The revised experimental electron capture cross sections for protons on carbon atoms at low energies, in units of 10^{-18} cm^2, are 0.81 ± 0.05 at 400 keV, 0.88 ± 0.08 at 300 keV, and 0.8 ± 0.08 at 250 keV of proton energies (J. R. Macdonald, private communication). These revised values are in good agreement with our calculations in Fig. 2.

ACKNOWLEDGMENTS

One of us (C.D.L.) acknowledges useful discussions with C. L. Cocke, J. M. Macdonald, and J. H. McGuire. This work is supported in part by Division of Basic Energy Sciences, U. S. Department of Energy. One of us (S.C.S.) is also supported partially by U. S. Army Research Office, Durham, North Carolina.

•Present address: Physics Dept., Hong Kong Baptist College, Hong Kong.

[1]For a recent review, see P. Richard, in *Atomic Inner-Shell Processes*, edited by B. Crasemann (Academic, New York, 1975), Vol. I.

[2]U. Fano and W. Lichten, Phys. Rev. Lett. 14, 627 (1965).

[3]Q. C. Kessel and B. Fastrup, *Case Studies in Atomic Physics*, edited by E. W. McDaniel and M. R. C. McDowell (Amsterdam, North-Holland, 1973), Vol. 3.

[4]J. S. Briggs and J. H. Macek, J. Phys. B 5, 579 (1972). See also the review by J. S. Briggs in Rep. Prog. Phys. 39, 217 (1976).

[5]K. Taulbjerg, J. S. Briggs, and J. Vaaben, J. Phys. B 9, 1351 (1976).

[6]M. R. C. McDowell and J. P. Coleman, *Introduction to the Theory of Ion-Atom Collisions* (North-Holland, Amsterdam, 1970).

[7]See the review by R. A. Mapleton, *Theory of Charge Exchange* (Wiley-Interscience, New York, 1972).

[8]B. H. Bransden, Rep. Prog. Phys. 35, 949 (1972).

[9]J. R. Oppenheimer, Phys. Rev. 31, 349 (1928).

[10]H. C. Brinkman and H. A. Kramers, Proc. Acad. Sci. (Amsterdam) 33, 973 (1930).

[11]D. R. Bates and A. Dalgarno, Proc. Phys. Soc. Lond. A 65, 919 (1952).

[12]J. D. Jackson and H. Schiff, Phys. Rev. 89, 359 (1953).

[13]In Ref. 12, the authors concluded that including the internuclear potential term will improve the convergence of the Born series. See also the remarks on p. 381 of Ref. 6.

[14]For earlier works, see Ref. 8.

[15]K. Omidvar, J. E. Golden, J. H. McGuire, and L. Weaver, Phys. Rev. A 13, 500 (1976).

[16]Y. B. Band, Phys. Rev. Lett. 37, 634 (1976).

[17]A. Halpern and J. Law, Phys. Rev. A 12, 1776 (1975).

[18]A. Halpern and J. Law, Phys. Rev. Lett. 31, 4 (1973).

[19]V. S. Nikolaev, Zh. Eksp. Teor. Fiz. 51, 1263 (1966) [Sov. Phys. JETP 24, 847 (1967)].

[20]G. Lapicki and W. Losonsky, Phys. Rev. A 15, 896 (1977).

[21]For example, the Born (G) method of Ref. 15.

[22]D. R. Bates, Proc. R. Soc. A 274, 294 (1958).

[23]H. S. W. Massey and R. A. Smith, Proc. R. Soc. A 142, 142 (1933).

[24]L. T. Sin Fai Lam, Proc. Phys. Soc. Lond. 92, 67 (1967).

[25]T. G. Winter and G. C. Lin, Phys. Rev. A 10, 2141 (1974).

[26]K. E. Banyard and B. J. Szuster, Phys. Rev. A 18, 129 (1977).

[27]However, it can be shown that the resulting coupled equations are independent of the choice of origin so long as the origin is on the internuclear axis. See Ref. 22.

[28]R. H. Bassel and E. Gerjuoy, Phys. Rev. 117, 749 (1960).

[29]See p. 212 of Ref. 6.

[30]The approximate $1/n^3$ rule is discussed on p. 379 of Ref. 6. See also Ref. 25.

[31]J. R. Macdonald, C. L. Cocke, and W. W. Eidson, Phys. Rev. Lett. 32, 648 (1974).

[32]G. L. Gocke, R. K. Gardner, B. Curnutte, T. Bratton, and T. K. Saylor, Phys. Rev. A 16, 2248 (1977).

[33]M. Rødbro, H. Pedersen, and J. R. Macdonald, in *Abstracts of the Tenth International Conference on the Physics of Electronic and Atomic Collisions, Paris, 1977*, edited by M. Barat and J. Reinhardt (Commissariat à l'Energie Atomique, Paris, 1977).

[34]Dz Belkic and R. McCarroll, J. Phys. B 10, 1933 (1977).

[35]It must be pointed out, however, the authors in Ref. 34 apparently use the binding energy of the target to be $-\frac{1}{2}Z_3^2$, with $Z_3 = 18$ for Ar. Their OBK results are in disagreement with the OBK results given in Refs. 15 and 31. Thus, it is not clear that the good agreement between calculations and data in the high-energy side given in Fig. 7 of Ref. 34 is exactly meaningful.

[36]A. Msezane, Phys. Rev. A 15, 666 (1977).
[37]A. E. S. Green, D. L. Sellin, and A. S. Zachor, Phys. Rev. 184, 1 (1969); A. E. S. Green, Adv. Quantum Chem. 7, 221 (1973).
[38]Calculation of MO potential curves upon the GSZ potential has been racently discussed by J. Eichler and U. Wille [Phys. Rev. A 11, 1973 (1975)].
[39]M. J. Antal, M. B. McElroy, and D. G. Anderson, J. Phys. B 8, 1513 (1975).
[40]A physically intuitive model would be to include in expansion (3) the eigenfunctions of the united atom He+ for collisions at small impact parameters. See Ref. 39 above and C. D. Lin (unpublished) (1977).
[41]L. Wilet and D. F. Gallaher, Phys. Rev. 147, 13 (1966).
[42]D. F. Gallaher and L. Wilet, Phys. Rev. 169, 139 (1968); R. Shakeshaft, Phys. Rev. A 14, 1626 (1976).
[43]I. M. Cheshire, D. F. Gallaher, and A. J. Taylor, J. Phys. B 3, 813 (1970).
[44]R. McCarroll, Proc. R. Soc. A 264, 547 (1961).
[45]R. M. Drisko, thesis (Carnegie Institute of Technology, 1955) (unpublished).

ELECTRON TRANSFER IN ION-ATOM COLLISIONS

by

LAURA NORMAN TUNNELL

B.S., East Texas State University, 1976

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Physics

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1979

ABSTRACT

The two state, two center atomic expansion method of Bates is applied within the independent electron approximation to charge transfer processes involving multielectron atoms. Usage of a realistic potential enabled a description of capture from both inner and outer shells of multielectron targets to be given. The theory is expected to be valid for projectile velocities near the characteristic orbital velocity of the active electron.

Comparison is made with an earlier work in which a screened hydrogenic model was employed to describe the transfer of electrons from the K shell of multielectron targets to the K shell of bare projectiles. It was concluded that the simple hydrogenic model could be used for high energy, asymmetric collision systems.

This is the first effort to describe capture from outer shells of multielectron targets in which a realistic atomic model is used. The method is applied to electron transfer from the outer shells of neon, argon, and krypton atoms to the K shell of hydrogen. Results of the calculations are compared with experimental data and further improvements in the theory are discussed.