A THEORETICAL ANALYSIS OF EXCITATION AND CHARGE EXCHANGE IN ION-ATOM COLLISIONS

by

TEH-HSIN KUNG WU

A dissertation submitted to the faculty of the University of North Carolina in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Physics and Astronomy.

Chapel Hill

1981

Eugen Merzbacher
Advisor

Reader

M.S. Davis
Reader

TEH-HSIN KUNG WU. A theoretical analysis of excitation and charge exchange in ion-atom collisions. (Under the direction of Eugen Merzbacher.)

ABSTRACT

The impact parameter dependence of the excitation and charge transfer process for the simple model of a collision between a bare nucleus and a one-electron nonrelativistic hydrogenic atom has been examined and calculated. Using semiclassical collision theory, Kepler motion on a hyperbolic trajectory is chosen to describe the relative nuclear motion. Applying a variational principle, coupled differential equations are derived for the amplitudes of the electronic states. Because of the Kepler motion, including recoil of the target, a general transformation theory connecting inertial and non-inertial frames of reference has been developed to obtain the appropriate Hamiltonian and wave functions. Fast Fourier transform and Romberg algorithms are applied to evaluate the matrix elements. The GEARS method has been used to solve the coupled differential equations. A preliminary calculation has been performed to evaluate the excitation and charge exchange probabilities.

Acknowledgements

I am indebted to the following people, without whose help, this work could not have been completed. Therefore, I would like to express my appreciation for their support:
Dr. Merzbacher, for his enthusiasm and assistance in physics, guidance in the ordinary life, and many patient enjoyable Saturday morning discussions;
Hai-ping Wu, for his complete understanding and support of my graduate work, and for sharing both schoolwork and housework with me;
Dr. Thompson, for his assistance on computing techniques.

I also like to express my appreciation to the Durham office of the Computer Sciences Corporation for letting me use the word-processor to type this dissertation.

Table of Contents

CHAPTER 1

INTRODUCTION

The theory of collisions of ions with atoms or other ions has been developed considerably in the last decade. Advances in computational technique have been largely responsible for this progress, but there has also been the influence of the challenge of improved experimental methods, notably in the area of ion sources and detection equipment. Furthermore, because of a great variety of accelerators, a large range of incident energy has been investigated and collisions between fully or partially stripped positive ions and atoms (or ions) have been achieved. The differential cross sections, which reveal more characteristics of the collision process than the total cross sections, have been measured as well as total cross sections.

During a general collision process between an incident ion (called 'projectile') and an atom or another ion (called 'target'), an active electron which is assumed to be attached to the target, and in the ground state, may be excited to a higher energy level of the target, or ionized as a free electron, or transferred to a bound state of the projectile. These three processes are called excitation, ionization, and charge exchange respectively.

There are three fundamental theories to characterize various inner-shell ion-atom collision processes:

(1) the plane wave Born approximation (MER58). This theory uses plane waves to describe the motion of the electron as well as the target and the projectile. This is essentially first order quantum mechanical approach.

(2) the semiclassical approximation (BAN59). In this

approach the relative motion of the projectile and target is described classically (namely the projectile and target are moving along a trajectory), but the motion of the electron is described quantum mechanically by a wave function.

(3) the binary encounter approximation (GAR70)(GAR73). In this approximation, the interaction of the incident nucleus and the (inner-shell) active electron is a binary encounter between two free particles described classically. The potential of the target nucleus only has an influence on the initial and final electron momentum distributions.

All these three theoretical approaches have their limitations for certain applications. For instance, the plane wave Born approximation is adequate for most ion-atom collisions at high incident energy, but not good for the very close collisions. The semiclassical approximation is valid for close collisions at intermediate incident energy, in which case the orbital velocity of the active electron is compatible with the incident projectile velocity. The binary encounter approximation is remarkably successful in the low-velocity region if the projectile charge is much less than the charge of the target nucleus.

Where appropriate, atomic physicists have modified these theories to examine certain effects. For instance, Dirac wave functions have been used for inner-shell electrons of a heavy target nucleus to investigate the relativistic effect (CLA75)(AMU75)(DEC77). To correct for electron screening, a modified nuclear charge has been used in the electronic wave function (CHE68). Various trajectories have been assumed to examine the effect of Coulomb deflection in the relative motion of the projectle and target nuclei. Since we are interested in a close-collision process at intermediate incident energy region, and since the projectile and target nuclear charges are assumed to be comparable, we apply the semiclassical theory in this dissertation. We

evaluate the excitation and charge exchange probabilities. The latter is at the center of our interest and we shall discuss the relation of our work to parallel approaches which have been used.

In 1966, Wilets and Gallaher (WIL66) performed a coupled-state calculation of H^+-H scattering in which the protons are described as moving on straight line trajectories and the (spinless) electronic wave function is expanded in terms of a finite subset of traveling hydrogenic wave functions centered about each proton. Five states are included in their calculation. Later, similar work was done with Sturmian states (GAL68), and a pseudo-state expansion using up to 7 states (CHE70).

More recently, Salop and Olsen evaluated the charge transfer and impact-ionization cross section by a (purely classical) three-dimensional Monte Carlo approach (OLS77a) and a coupled-state impact-parameter approximation (SAL77) (OLS77b). Adiabatic molecular wave functions were used for the second method. In addition to these representative papers, the theory of inelastic ion-atom collisions has been the subject of many other investigations. (MOR78)(LIN78) (LIN80)(MEY79)(STO73)

In this dissertation, we will concentrate on the impact-parameter dependence of the excitation and charge transfer process for the simple model of a collision between a bare charged nucleus and a one-electron nonrelativistic hydrogenic atom. We assume a hyperbolic Kepler trajectory for the relative nuclear motion. The Kepler orbit is more complicated and more realistic than the various trajectories such as a straight line, or two broken straight lines forming by a scattering angle to simulate the Coulomb deflection effect (AND76). We solve the time-dependent Schrödinger equation by expanding the electronic wave functions in a subset of hydrogenic traveling wave functions. Because a hyperbolic trajectory is used for the relative nuclear motion, the translation factors used in these traveling wave functions

do not depend on time linearly. This distinguishes our work from that of Wilets and Gallaher and all others who used straight-line trajectories and nuclear motion at a constant velocity, making the matrix elements easy to evaluate. Because of the hyperbolic trajectory and the Kepler motion, including recoil of the target, we consider the appropriate changes in the electronic Hamiltonian which give the potential, rotational, and angular coupling terms. For the three-body system (the projectile nucleus, the target nucleus, and the electron), we use the full Hamiltonian with no approximation involved. Our goal is to investigate the Coulomb deflection ('curved trajectory' (WU76) and recoiling target) effect on the excitation and charge exchange probabilities. The only approximation in this calculation is the limitation to a small number of basis states and the resulting truncation of the coupled equations.

In Chapter 2, we first introduce the necessary formalism for the nuclear motion. Then using a variational principle, we establish the coupled differential equations for the electronic states. In Chapter 3, we develop a general transformation theory (WU79a) which had previously caused some confusion in the literature. We first start with the Heisenberg picture in an inertial frame of reference and derive the correct formalism in the Schrödinger picture in a non-inertial frame of reference. The Hamiltonian relation and the translation factor between these inertial and non-inertial frames of reference are also derived. In Chapter 4, we evaluate direct and exchange transition matrix elements (WU79b) and solve the coupled differential equations numerically. Chapter 5 contains some conclusions.

In the dissertation, we are using atomic units. By that we mean the following:

(a) Mass is measured in units of the electron mass, so that $m_e=1$.
(b) Length is meassured in units of Bohr radii, so that $a_o=1$.

(c) Charge is measured in units of the electron charge, so that e=1.

(d) $\hbar = 1$

From these conventional definitions, we also observe that the velocity is measured in units of 2.18×10^8 cm/sec, energy is measured in units of 27.2 ev and time is measured in units of 2.29×10^{-19} sec.

CHAPTER 2

TIME-DEPENDENT SEMICLASSICAL APPROACH TO INELASTIC ION-ATOM COLLISIONS

2.1 Introduction

In the last decade, many different approaches have been developed to solve the inelastic ion-atom collision problem. Most of these fall into three catagories: (1) fully quantum mechanical treatment of both nuclear and active electron motions; (2) fully classical treatment of both nuclear and active electron motions; (3) classical treatment of the nuclear motion and quantum mechanical treatment of active electron motion. (In an independent-particle description, an active electron is an electron which is ionized, excited or transferred during the collision process). The third case is called impact-parameter or semiclassical atomic collision theory. In this dissertation, we will adopt the semiclassical approach.

Our approach starts with a system in which an ion called "projectile" impinges on an atom called "target", causing an atomic electron to undergo a transition from its initial state. We treat the projectile and target nucleus as classical particles. The interaction between these two nuclei is the Coulomb force. According to classical mechanics, the relative Kepler motion of the projectile and target nucleus with Coulomb interaction follows a hyperbolic trajectory. The atomic electron, which moves in the potential due to the projectile and target nucleus, is described quantum mechanically and its effect on the nuclear

motion is neglected. In this dissertation, we will not consider many-electron atoms in detail, but our methods can be readily generalized to many-electron atoms. Without committing an appreciable error, we also ignore the electron spin.

Atomic collision processes are divided into many different regimes, according to such variables as impact parameter and velocity. Our interest is directed in particular to the following conditions:

(1) Large scattering angle, i.e. small impact parameter. If we would consider small angle scattering, the hyperbolic trajectory is not appreciably different from a straight-line trajectory, which has been widely used in the semiclassical approximation. We want to investigate how a strongly curved hyperbolic trajectory will affect the whole process. Generally, we consider that scattering angles greater than 10 degrees are "large" angles.

(2) Intermediate velocity range, $1/2 < V_I/V_{el} < 2$. As presented by Briggs (BRI 78), the approximations made in our work are most appropriate for this velocity range. At higher incident velocity V_I, a Born approximation treatment becomes appropriate instead of a close-coupling approach. Too low incident velocity will be a condition for near-adiabatic motion which favors a molecular treatment of the three-body system.

We will direct our effort toward the solution of the time-dependent, nonrelativistic spinless Schrödinger equation. Our main interest is to calculate the probabilities for excitation and charge exchange of the atomic electron. In section 2.2, we will discuss nuclear motion first. In section 2.3, we develop the fundamental formalism for the electronic system and derive the coupled equations of motion of the electron.

2.2 Description of the Nuclear Motion

The goal of this section is to provide the basic formalism and notation for the classical motion of the two nuclei in the context of the semiclassical theory that is being developed here. In particular, we want to derive relations which allow us to determine the time of arrival of the projectile at a point of the trajectory as a function of its location defined by its polar coordinates.

The interaction between the projectile ion A and target nucleus B is the repulsive Coulomb force. In Figure 2-1A, viewed in the laboratory system, the target nucleus B recoils from its initial position as a result of the scattering. In Figure 2-1B, viewed in the C. M. system, two particles A and B are initially moving toward each other; then after the scattering they are moving away from each other (GOL62). Both these frames of reference are inertial. In addition, we will use a non-inertial frame of reference in which the target nucleus B is stationary through the whole scattering process such that the two-body central force problem can be reduced to an effective one-body problem. From now on, we will call this special frame the target frame-of-reference.

Let the masses of projectile ion A and target B be M_A and M_B, and

$$M = M_A + M_B$$

$$\mu = \frac{M_A M_B}{M}$$

where μ is the reduced mass. The charges of projectile A and target B are Z_1 and Z_2. The scattering between the projectile and the target is equivalent to a single particle of mass μ scattering from a fixed central potential $(Z_1 Z_2/R)$ where the internuclear distance R is also the distance of the single particle from the center of force. The incident

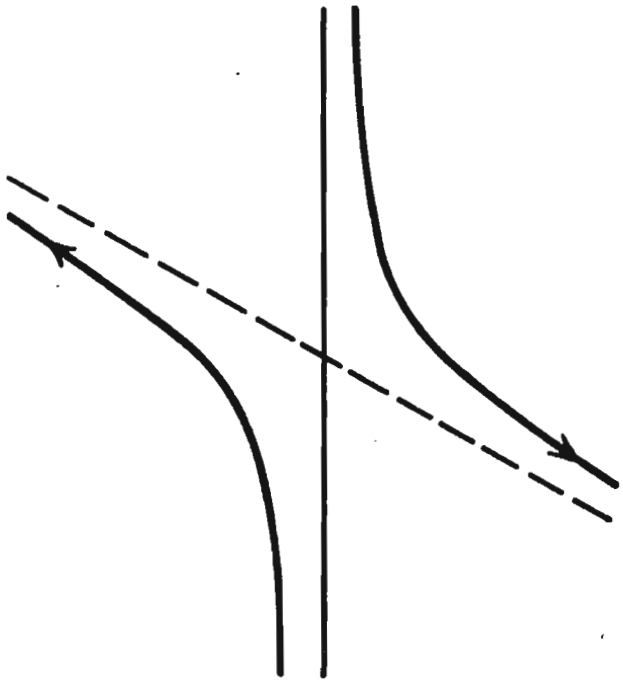


Figure 2-1 B

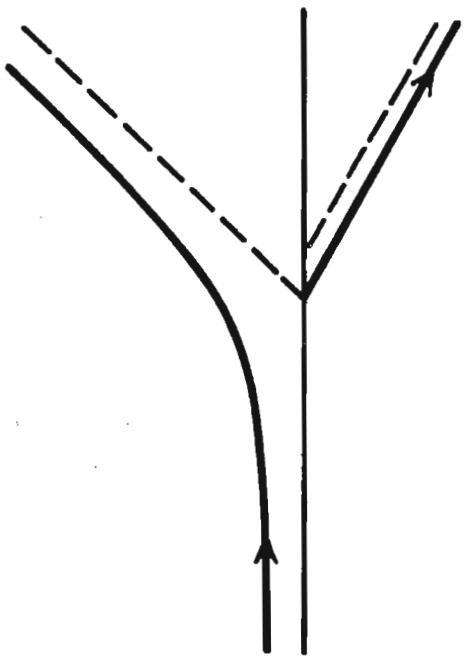


Figure 2-1 A

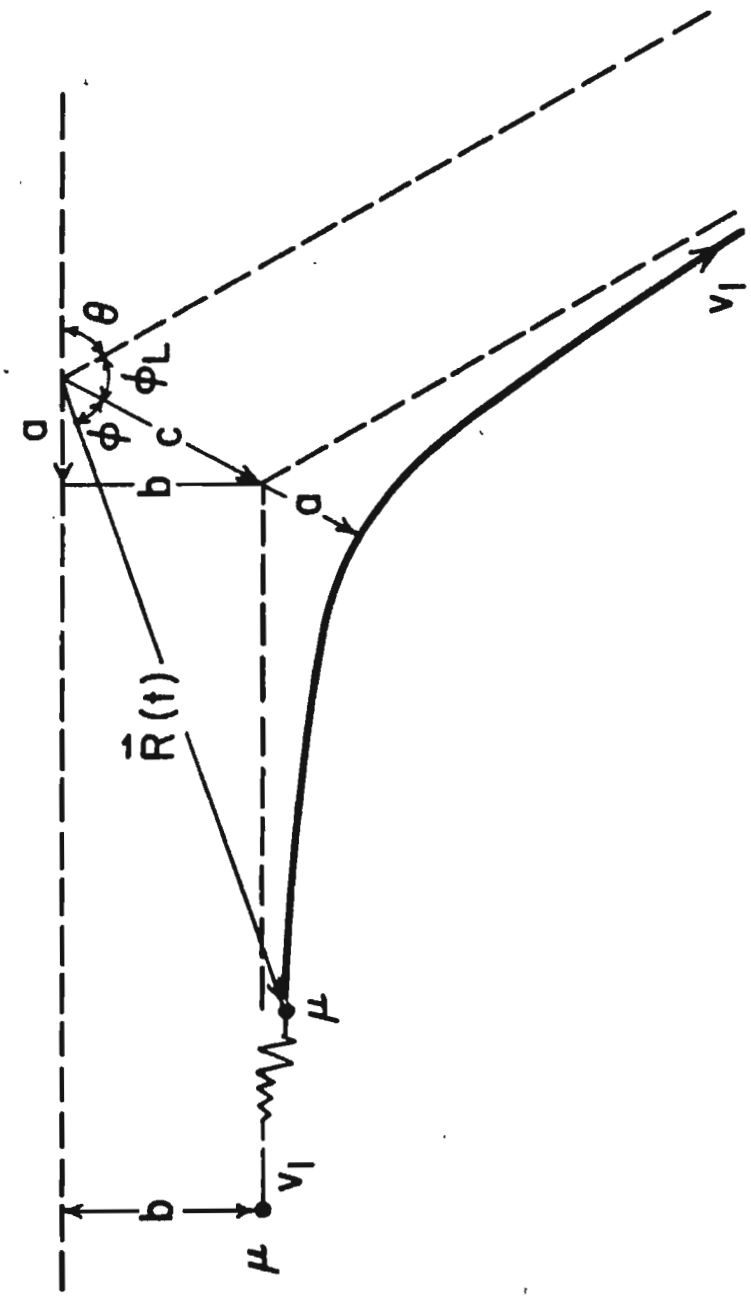


Figure 2-2

velocity is the initial laboratory velocity v_1, and the impact parameter is b. The orbit of the particle with mass μ is shown in Figure 2-2 (KON69). The orbit is a branch of a hyperbola, which has its focus at the center of target B and which has the equation

$$R(\phi) = \frac{b^2}{-a+c\cos\phi} \qquad (2.1)$$

The parameters a and c are defined in the following paragraphs.

Mathematically, "a" is the semimajor axis of the hyperbola. (b, the impact parameter, is also the semiminor axis). In collision theory, "a" is defined

$$a = \frac{z_1 z_2}{\mu v_1^2} \qquad (2.2.a)$$

which is half of the 'collision diameter'. The collision diameter equals the distance of closest approach for a 'head-on' (b=0) collision.

"c" is the length

$$c = (a^2 + b^2)^{1/2} \qquad (2.2.b)$$

In terms of these lengths, the distance of closest approach may be written

$$\gamma = a + c \qquad (2.2.c)$$

The angle ϕ is the polar angle referring to the line joining the center of force to the turning point ("perihelion"). The orbit has asymptotes ($R \rightarrow \infty$) at angle $\phi = \phi_L$ for which

$$\cos\phi_L = a/c \qquad (2.2.d)$$

and the angle ϕ has a range

$$-\phi_L < \phi < \phi_L$$

At $\phi=0$, $R(0)=b^2/(-a+c)=a+c=\gamma$, the distance of closest approach.

As shown in Figure 2-2, the scattering angle θ is the angle between the incident and final directions of the scattered particle, and is

$$\theta = |\pi - 2\phi_L| < \pi \qquad (2.2.e)$$

Now that we have discussed the geometric properties of the orbit, we proceed to compile some of the dynamic aspects of the problem.

Since the Coulomb interaction involves only the radial distance, the potential has spherical symmetry, and the total angular momentum in the C. M. frame

$$\vec{L} = \mu b v_1 \hat{y} = \ell \hat{y} \qquad (2.3.a)$$

is conserved. $\hat{y}$ is the unit vector perpendicular to the plane of motion. ℓ is the constant magnitude of the angular momentum. In terms of radial distance and angular velocity, ℓ is

$$\ell = \mu R^2 \dot{\phi} \qquad (2.3.b)$$

or

$$\ell = \mu R^2 \frac{d\phi}{dt}$$

Integrating the above equation and applying (2.1), we get

$$t = \frac{\mu}{\ell} \int \frac{b^4}{(-a+c\cdot\cos\phi)^2} d\phi + \text{const.}$$

To describe the collision, we choose the following initial conditions:

$$\begin{aligned} t &= -\infty, \quad R \longrightarrow \infty, \quad \phi \longrightarrow -\phi_L \\ t &= 0, \quad R = \gamma \ , \ \phi = 0 \\ t &= \infty, \quad R \longrightarrow \infty, \quad \phi \longrightarrow \phi_L \end{aligned} \tag{2.4}$$

so that time $t(\phi)$ as a function of the angle ϕ is

$$t(\phi) = \int_0^{\phi} dt = \frac{\mu b^4}{\ell} \int_0^{\phi} \frac{d\phi}{(c\cdot\cos\phi - a)^2}$$

or

$$t(\phi) = \frac{1}{v_1}\left(\frac{R}{b} c\cdot\sin\phi + a\cdot\ln\left(\frac{b\cdot\tan\phi/2 + c - a}{b\cdot\tan\phi/2 + a - c}\right)\right) \tag{2.5}$$

where we have used equation (2.3.a). This relation will be used frequently in the following.

Finally, we require an expression for the projectile velocity in terms of the position coordinates. Since Coulomb forces are conservative, the total energy of the system, E, in the center-of-mass frame is a constant of the motion:

$$E = \tfrac{1}{2}\,\mu(\dot{R}^2 + R^2\dot{\phi}^2) + V_N(R) \tag{2.6}$$

where $V_N(R)$ is the potential energy

$$V_N(R) = \frac{Z_1 Z_2}{R} \tag{2.7}$$

Applying (2.3.b) and solving (2.6) for $\dot{R}$, we obtain

$$\dot{R} = \pm \left\{\frac{2}{\mu}\left(E - V_N(R) - \frac{\ell^2}{2\,\mu R^2}\right)\right\}^{\frac{1}{2}} \tag{2.8}$$

where "+" is for $t>0$ and "-" is for $t<0$.

Let us choose Cartesian coordinates for the plane of motion: (x,y,z) coordinates denote the center-of-mass frame, and (x_B,y_B,z_B) (or (x_A,y_A,z_A)) coordinates refer to the target frame (or projectile frame). The internuclear axis is along the $z_A(z_B)$ axis, and the y (y_A and y_B) axis is out of the plane of motion as shown in Figure 2-3. In the target frame of reference, the internuclear distance vector is

$$\vec{R} = - R \hat{z}_B \qquad (2.9.a)$$

The velocity $\dot{\vec{R}}$ is

$$\dot{\vec{R}} = (-\dot{R})\hat{z}_B + (-R\dot{\phi})\hat{x}_B = (-\dot{R})\hat{z}_B + (-bv_1/R)\hat{x}_B \qquad (2.9.b)$$

and the acceleration $\ddot{\vec{R}}$, according to Newton's law,

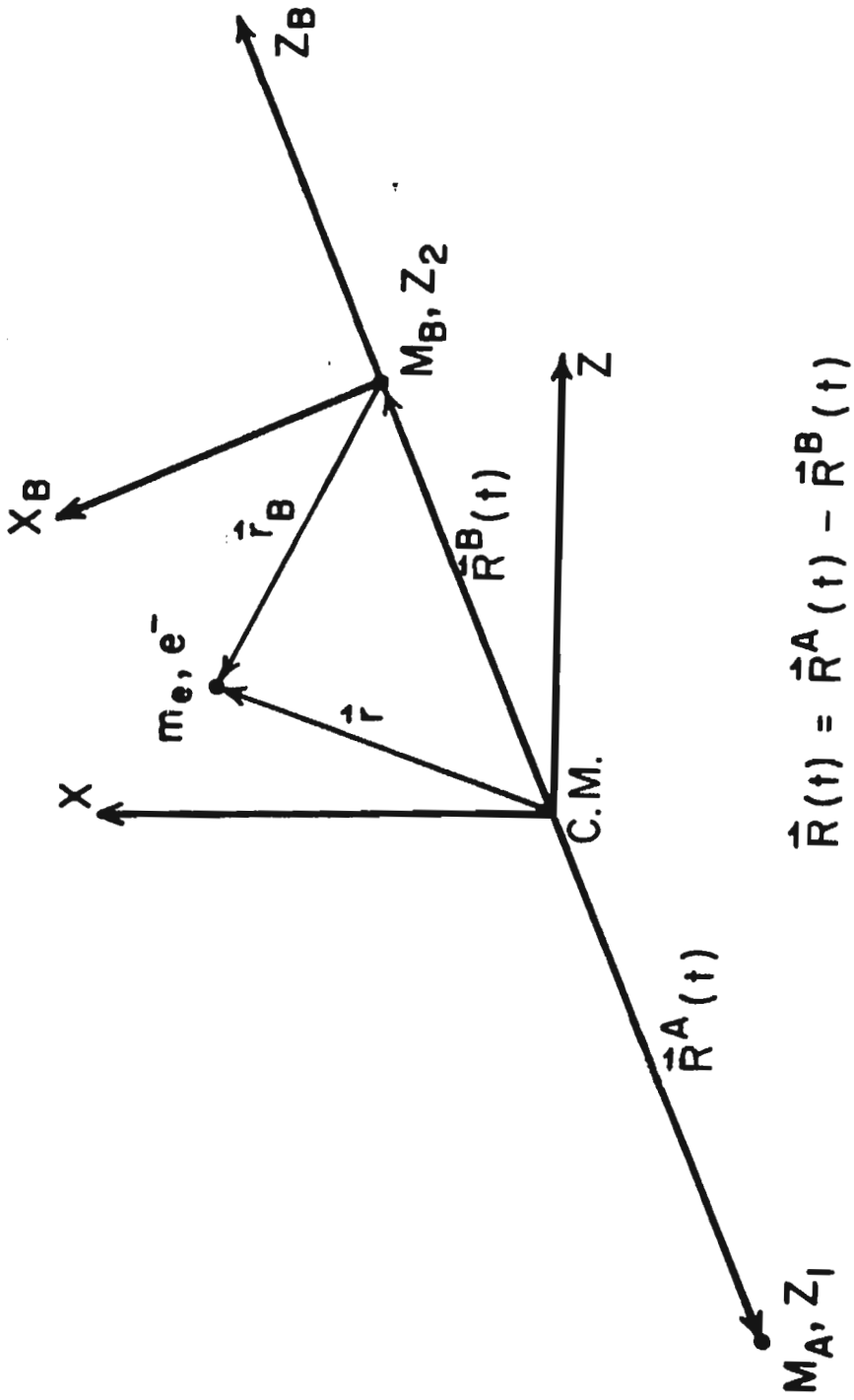
$$\ddot{\vec{R}} = - \frac{Z_1 Z_2}{\mu R^2} \hat{z}_B \qquad (2.9.c)$$

The position vectors of the projectile and the target from the center of mass are

$$\begin{aligned} \vec{R}^A &= \frac{M_B}{M} \vec{R} = - \frac{M_B}{M} R \hat{z}_B \\ \vec{R}^B &= - \frac{M_A}{M} \vec{R} = \frac{M_A}{M} R \hat{z}_B \end{aligned} \qquad (2.10.a)$$

The accelerations are

$$\begin{aligned} \ddot{\vec{R}}^A &= - \frac{Z_1 Z_2}{M_A R^2} \hat{z}_B \\ \ddot{\vec{R}}^B &= \frac{Z_1 Z_2}{M_B R^2} \hat{z}_B \end{aligned} \qquad (2.10.b)$$



$$\vec{R}(t) = \vec{R}^A(t) - \vec{R}^B(t)$$

Figure 2-3

Note that all time derivatives of vectors are taken with respect to the center-of-mass frame, but their components are here expressed in terms of the target frame (i.e. vectors are projected in the target frame).

2.3 Coupled Equations of Motion of the Electronic States

Now we will discuss the quantum mechanical treatment of the electronic system when a projectile ion is scattered from a target atom. Our ultimate purpose is to calculate the time-dependent state vector $|\Psi(t)\rangle$ of the active atomic electron. The electronic motion is considered in the time-dependent field determined by the Kepler motion of the projectile ion relative to the target nucleus. In the center-of-mass frame of reference, the time-dependent electronic Schrödinger equation of motion is

$$H_o \; |\Psi(t)\rangle = i \frac{\partial}{\partial t} \; |\Psi(t)\rangle \qquad (2.11)$$

where H_o is the time-dependent Schrödinger Hamiltonian of the electronic system.

The Hamiltonian is

$$H_o = \tfrac{1}{2}p^2 - \frac{Z_1}{|\vec{r}-\vec{R}^A(t)|} - \frac{Z_2}{|\vec{r}-\vec{R}^B(t)|} \qquad (2.12)$$

where the $\vec{R}^A(t)$ and $\vec{R}^B(t)$ are the position vectors of the projectile A and target B. The Hamiltonian H_o is time dependent through these two vectors. Here we ignore the atomic electron-electron or other electron-nucleus interactions. We only consider the interactions between the projectile, the target and the active atomic electron.

Equation (2.11) is subject to well-defined boundary

and initial conditions. Since all calculations of transition probabilities, cross sections, and scattering amplitudes must necessarily be approximate, it is desirable to formulate the quantum mechanical problem in a manner that allows maximum flexibility. We will apply a variational approach to this impact-parameter problem. (McD70) (SIL60)(STO73)(GEL69)(SHA74).

We define the action integral:

$$S = \int_{t_1}^{t_2} L \, dt \ ,$$

where L is a functional of the independent bra and ket trial states $\langle \Psi_1^t(t)|$ and $|\Psi_2^t(t)\rangle$, by

$$L = \langle \Psi_1^t(t)|H_o - i \frac{\partial}{\partial t}|\Psi_2^t(t)\rangle$$

Therefore, the action integral may be written in the form

$$S(\Psi_1^t,\Psi_2^t) = \int_{t_1}^{t_2} \langle \Psi_1^t(t)|H_o - i \frac{\partial}{\partial t}|\Psi_2^t(t)\rangle dt$$

If $|\Psi_1\rangle$ and $|\Psi_2\rangle$ are solutions of equation (2.11), then

$$S(\Psi_1,\Psi_2) = 0$$

To first order in $|\delta\Psi_1\rangle = |\Psi_1^t(t)\rangle - |\Psi_1\rangle$ and $|\delta\Psi_2\rangle = |\Psi_2^t(t)\rangle - |\Psi_2\rangle$, the variations of the action is

$$\delta S = \int_{t_1}^{t_2} \langle \delta\Psi_1|H_o - i\frac{\partial}{\partial t}|\Psi_2\rangle dt + \int_{t_1}^{t_2} \langle \Psi_1|H_o - i\frac{\partial}{\partial t}|\delta\Psi_2\rangle dt$$

$$+ \int_{t_1}^{t_2} \langle \delta\Psi_1|H_o - i\frac{\partial}{\partial t}|\delta\Psi_2\rangle \, dt$$

$$\delta S = \int_{t_1}^{t_2} \langle \delta\Psi_1 | H_o - i\frac{\partial^{\rightarrow}}{\partial t} | \Psi_2 \rangle dt + \int_{t_1}^{t_2} \langle \Psi_1 | H_o + i\frac{\partial^{\leftarrow}}{\partial t} | \delta\Psi_2 \rangle dt$$

$$- i\langle \Psi_1 | \delta\Psi_2 \rangle \Big|_{t_1}^{t_2} + \int_{t_1}^{t_2} \langle \delta\Psi_1 | H_o - i\frac{\partial^{\rightarrow}}{\partial t} | \delta\Psi_2 \rangle dt$$

Also, the variation of the action integral is equal to the value of the action S corresponds to the varied states:

$$\delta S = S(\Psi_1^t, \Psi_2^t)$$

$$= -i \langle \Psi_1 | \delta\Psi_2 \rangle \Big|_{t_1}^{t_2} + \int_{t_1}^{t_2} dt \langle \delta\Psi_1 | H_o - i \frac{\partial}{\partial t} | \delta\Psi_2 \rangle$$

Showing that, to the first order, the value of δS depends only on the endpoints and not on the behavior of the states between t_1 and t_2.

Since the equation of motion is of first order in time, the variational principle cannot be formulated in terms of stationary values for the action. To do so would imply that the variation $|\delta\Psi_2\rangle$ vanishes at both endpoints, but this is generally inconsistent with the equation of motion. (STO73)

In applying the variational principle to the determination of an optimum trial function, which approximates the actual solution closely, we note that to first order

$$\delta S = \int_{t_1}^{t_2} \langle \delta\Psi_1 | H_o - i\frac{\partial^{\rightarrow}}{\partial t} | \Psi_2^t(t) \rangle dt + \int_{t_1}^{t_2} \langle \Psi_1^t(t) | H_o + i\frac{\partial^{\leftarrow}}{\partial t} | \delta\Psi_2 \rangle dt$$

$$- i\langle \Psi_1 | \delta\Psi_2 \rangle \Big|_{t_1}^{t_2} + \int_{t_1}^{t_2} \langle \delta\Psi_1 | H_o - i\frac{\partial^{\rightarrow}}{\partial t} | \delta\Psi_2 \rangle dt$$

The requirement that

$$\delta S + i \langle \Psi_1 | \delta \Psi_2 \rangle \Big|_{t_1}^{t_2} = 0$$

to first order is therefore equivalent to the condition

$$\int_{t_1}^{t_2} \langle \delta \Psi_1 | H_o - i \frac{\partial^{\rightarrow}}{\partial t} | \Psi_2^t(t) \rangle \, dt = 0 \qquad (2.13.a)$$

$$\int_{t_1}^{t_2} \langle \Psi_1^t(t) | H_o + i \frac{\partial^{\leftarrow}}{\partial t} | \delta \Psi_2 \rangle \, dt = 0 \qquad (2.13.b)$$

The action principle is applied by substituting trial state vectors which are superpositions of a finite number of some simple physical states with variable coefficients into the variational conditions (2.13) to derive the coupled equations. We will expand $|\Psi_1^t(t)\rangle$ and $|\Psi_2^t(t)\rangle$ in terms of the same truncated set of basis vectors.

Trial state vectors can be constructed in many different ways (BRI78). Since we consider charge transfer processes as well as excitation explicitly, we shall include basis set functions that can represent adequately the bound states of the electron around the projectile. The trial state vectors are assumed to be superpositions of the basis sets on two centers, the target nucleus Z_2 and the projectile ion Z_1:

$$\begin{aligned} |\Psi_1^t(t)\rangle &= \sum_n a_n'(t) |\Phi_n(A,t)\rangle + \sum_n b_n'(t) |\Phi_n(B,t)\rangle \\ |\Psi_2^t(t)\rangle &= \sum_n a_n(t) |\Phi_n(A,t)\rangle + \sum_n b_n(t) |\Phi_n(B,t)\rangle \end{aligned} \qquad (2.14)$$

where

$|\Phi_n(A,t)\rangle$ = the normalized bound state of an electron around the projectile

$|\Phi_n(B,t)\rangle$ = the normalized bound state of an electron around the target nucleus

The variation of the trial function $|\Psi_1^t(t)\rangle$ is

$$|\delta\Psi_1\rangle = \sum_n \delta a_n'(t)|\Phi_n(A,t)\rangle + \sum_n \delta b_n'(t)|\Phi_n(B,t)\rangle$$

Substituting the above equations into (2.13.a), we get

$$\begin{aligned}
&\int dt \sum_m \sum_n \Big(\delta a_m'^*(t)\Big(a_n(t)\langle\Phi_m(A,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(A,t)\rangle \\
&\qquad + b_n(t)\langle\Phi_m(A,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(B,t)\rangle\Big) \\
&\quad + \delta b_m'^*(t)\Big(a_n(t)\langle\Phi_m(B,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(A,t)\rangle \\
&\qquad + b_n(t)\langle\Phi_m(B,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(B,t)\rangle\Big) \\
&-i\delta a_m'^*(t)\Big(\dot{a}_n(t)\langle\Phi_m(A,t)|\Phi_n(A,t)\rangle + \dot{b}_n(t)\langle\Phi_m(A,t)|\Phi_n(B,t)\rangle\Big) \\
&-i\delta b_m'^*(t)\Big(\dot{a}_n(t)\langle\Phi_m(B,t)|\Phi_n(A,t)\rangle + \dot{b}_n(t)\langle\Phi_m(B,t)|\Phi_n(B,t)\rangle\Big)\Big) \\
&= 0
\end{aligned} \tag{2.15}$$

Since the atomic basis on the same center are orthonormal sets, we have

$$\begin{aligned}
\langle\Phi_m(A,t)|\Phi_n(A,t)\rangle &= \delta_{mn} \\
\langle\Phi_m(B,t)|\Phi_n(B,t)\rangle &= \delta_{mn}
\end{aligned} \tag{2.16}$$

If we treat all $a_m'(t)$, $a_m'^*(t)$, $b_m'(t)$, $b_m'^*(t)$ as independent variables, then (2.15) requires that all the coefficients of $\delta a_m'^*(t)$ and $\delta b_m'^*(t)$ be zero. Applying (2.16), we obtain

the coupled equations

$$\sum_n a_n(t)H_{mn}(A) + \sum_n b_n(t)K_{mn}(A,B) = i\,\dot{a}_m(t) + i\sum_n \dot{b}_n(t)S_{mn}$$

$$\sum_n a_n(t)\bar{K}_{mn}(B,A) + \sum_n b_n(t)\bar{H}_{mn}(B) = i\sum_n \dot{a}_n(t)S^{\dagger}_{mn} + i\,\dot{b}_m(t) \qquad (2.17)$$

where the matrix elements are defined

$$H_{mn}(A) = \langle\Phi_m(A,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(A,t)\rangle$$

$$\bar{H}_{mn}(B) = \langle\Phi_m(B,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(B,t)\rangle$$

$$K_{mn}(A,B) = \langle\Phi_m(A,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(B,t)\rangle \qquad (2.18)$$

$$\bar{K}_{mn}(B,A) = \langle\Phi_m(B,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(A,t)\rangle$$

$$S_{mn} = \langle\Phi_m(A,t)|\Phi_n(B,t)\rangle$$

$$S^{\dagger}_{mn} = \langle\Phi_m(B,t)|\Phi_n(A,t)\rangle$$

If we vary the trial state $|\Psi^t_2(t)\rangle$ and substitute it into (2.13.b), we will get coupled equations which are just the complex conjugates of equation (2.17). So we consider that equations (2.17) and the matrix elements (2.18) are the coupled equations of motion, which approximate equation (2.11) for the active electron in the center-of-mass frame of reference.

In order to solve the coupled equations, the matrix elements shall be evaluated first. The matrices $H_{mn}(A)$ (and $\bar{H}_{mn}(B)$), $K_{mn}(A,B)$ (and $\bar{K}_{mn}(B,A)$), and S_{mn} in (2.18) are so-called 'direct coupling', 'exchange coupling', and

'overlap' matrices. In the next chapter, we will discuss a general transformation theory which will make these matrix elements easy to evaluate. In Chapter 4, the coupled equations will then be solved subject to the physically appropriate initial conditions:

$$b_n(-\infty) = \begin{cases} 1 & \text{for } n=1 \\ 0 & \text{for } n=2,3,4,\ldots \end{cases}$$

$$a_n(-\infty) = 0 \qquad \text{for } n=1,2,3,4,\ldots \tag{2.19}$$

so that the target atom is in the ground state before the collision starts.

CHAPTER 3

TRANSFORMATION THEORY

3.1 Introduction

In a semiclassical theory of the collision between an incident ion A and a target atom B, the motion of the projectile A relative to the target B is described by a hyperbolic Kepler orbit. During this process, an electron originally attached to target atom B may be excited, ionized or transferred to projectile A. From the previous chapter, the active electron's equations of motion in the center of mass system of nuclei A and B are derived as coupled differential equations. In order to solve the coupled differential equations, the coefficients need to be evaluated first. In the center-of-mass frame of reference, the Hamiltonian of the electron is straightforward. However, the wave functions of the electron are complicated and not easy to obtain. We can avoid this difficulty by using an accelerated frame of reference centered at either A or B, where the much less complicated atomic wave functions can be used to evaluate the matrix elements and to solve the equations of motion. Furthermore, the time-dependent interactions in the center of mass system

$$V(r,t) = - \frac{Z_1}{|\vec{r} - \vec{R}^A(t)|} - \frac{Z_2}{|\vec{r} - \vec{R}^B(t)|}$$

can be written as

$$V(r,t) = - \frac{Z_1}{|\vec{r}_B - \vec{R}(t)|} - \frac{Z_2}{r_B}$$

in the accelerated frame of reference centered at B (see Figure 2-3). Then the only remaining time dependence of the interacton is simply the internuclear distance R(t) between A and B. Therefore, the potential energy is easy to handle, but on the other hand in these non-inertial frames of reference, which are centered at either A or B, the electronic Hamiltonian is not simply the total energy and cannot be derived by the usual simple correspondence from the Hamiltonian function of classical mechanics. A transformation theory developed particularly for this problem will provide the correct relations among operators, wave functions and Hamiltonians, transformed from an inertial frame of reference to a non-inertial one.

Because of the close correspondence between the classical theory and quantum dynamics, the Heisenberg picture approach, with operators changing in time and state vectors staying fixed, is better suited than the Schrödinger picture to formulate a quantum theory of the classical analogue of the desired coordinate transformations. After the correct formalism has been derived in the Heisenberg picture for transformation from an inertial frame of reference to a non-inertial one, the theory will be re-written in the Schrödinger picture, which has a more intuitive form for scattering processes.

It is important to note that the transformation to a non-inertial frame is not the usual Galilean transformation. In contrast to a uniform velocity as in the Galilean case, the new frame of reference centered at either the projectile A or the target B is both linearly accelerating and rotating. However, the Galilean transformation is a special case of our general transformation theory. Section 3.2 and 3.3 will contain the general time-dependent unitary transformation theory, which is a generalization of the usual time-

independent quantum mechanical transformation theory. The application to the linear acceleration case and wave functions are examined in detail in section 3.4 and 3.5. The rotational transformation is discussed briefly in Section 3.6.

3.2 Heisenberg Picture

Consider a physical system which is viewed by observers in an inertial frame of reference F and in a non-inertial frame of reference F^{σ}. These two frames of reference relate to each other through some time-dependent parameters in a general way. Since the nature of the physical system is characterized by state vectors and the state vectors in the Heisenberg picture are time-independent, we will discuss state vectors first.

Let $|\psi>$ and $|\psi^{\sigma}>$ be the Heisenberg state vectors of the physical system in F and F^{σ} respectively. We choose

$$|\psi> = |\psi^{\sigma}> \tag{3.1}$$

as our assumption.

We also choose a unitary operator for the transformation of observables from F to F^{σ} in the Heisenberg picture. This unitary operator, provides a canonical transformation, preserves hermiticity, and keeps eigenvalues invariant in the F^{σ} frame. Let $U_t(t)$ be this unitary operator which in general may be explicitly time-dependent. The subscript "t" denotes the time-dependent feature of the Heisenberg operators. The operator $U_t(t)$ satisfies

$$U_t^{\dagger}(t)U_t(t) = U_t(t)U_t^{\dagger}(t) = 1 \tag{3.2}$$

and hence

$$\frac{dU_t(t)}{dt}\, U_t^{\dagger}(t) + U_t(t)\, \frac{dU_t^{\dagger}(t)}{dt} = 0 \qquad (3.3)$$

A_t and A_t^{σ} are defined as Heisenberg operators of the physical system in F and F^{σ}. With every Heisenberg operator A_t in F, we associate a Heisenberg operator A_t^{σ} in F^{σ} according to the rule:

$$A_t^{\sigma} = U_t(t)\, A_t\, U_t^{\dagger}(t) \qquad (3.4)$$

The equation of motion of operator A_t is

$$i\, \frac{dA_t}{dt} = [A_t, H_t] + i\, \left(\frac{\partial A_t}{\partial t}\right) \qquad (3.5)$$

where H_t is the Hamiltonian of the physical system in F. Equation (3.4) tells us how A_t^{σ} develops in time. It will be useful to determine a new Hamiltonian $\hat{H}_t^{\sigma}$, so that the equation of motion for A_t^{σ} becomes:

$$i\, \frac{dA_t^{\sigma}}{dt} = [A_t^{\sigma}, \hat{H}_t^{\sigma}] + i\, \left(\frac{\partial A_t^{\sigma}}{\partial t}\right) \qquad (3.6)$$

The last equation is a conjecture, and we shall have to prove its validity and derive the form of $\hat{H}_t^{\sigma}$.

According to quantum dynamics, the Hamiltonian H_t in the inertial frame of reference F is an observable whose expectation value $\langle H_t \rangle$ measures the total energy of the physical system. However, the Hamiltonian $\hat{H}_t^{\sigma}$ in the non-inertial frame of reference F^{σ} does generally not represent the total energy of the physical system. Generally, the Hamiltonians H_t and $\hat{H}_t^{\sigma}$ do not transform merely as operators according to (3.4). In other words, generally, H_t^{σ} differs

from $\hat{H}_t^\sigma$.

In order to find out how the Hamiltonian in one reference frame is related to the Hamiltonian in another reference frame, we use (3.4) to calculate the time derivative of A_t^σ:

$$i\frac{dA_t^\sigma}{dt} = i\frac{d}{dt}\left(U_t(t)A_tU_t^\dagger(t)\right)$$

$$= i\frac{dU_t(t)}{dt}A_tU_t^\dagger(t)+iU_t(t)\frac{dA_t}{dt}U_t^\dagger(t)+iU_t(t)A_t\frac{dU_t^\dagger(t)}{dt}$$

From (3.3), (3.4), and (3.5), we obtain

$$i\frac{dA_t^\sigma}{dt}$$

$$=i\frac{dU_t(t)}{dt}U_t^\dagger(t)A_t^\sigma+iA_t^\sigma U_t(t)\frac{dU_t^\dagger(t)}{dt}+U_t(t)[A_t,H_t]U_t^\dagger(t)+iU_t(t)\left(\frac{\partial A_t}{\partial t}\right)U_t^\dagger(t)$$

$$=i\left[A_t^\sigma,U_t(t)\frac{dU_t^\dagger(t)}{dt}\right] + [A_t^\sigma,U_t(t)H_tU_t^\dagger(t)] + iU_t(t)\left(\frac{\partial A_t}{\partial t}\right)U_t^\dagger(t)$$

$$=\left[A_t^\sigma,U_t(t)H_tU_t^\dagger(t) + iU_t(t)\frac{dU_t^\dagger(t)}{dt}\right] + i\left(\frac{\partial A_t^\sigma}{\partial t}\right)$$

where we have used (3.4) to define

$$\left(\frac{\partial A_t^\sigma}{\partial t}\right) = U_t(t)\frac{\partial A_t}{\partial t}U_t^\dagger(t)$$

We thus see that an equation of the form (3.6) is satisfied no matter which non-inertial frame of reference we are in,

if we define

$$\hat{H}_t^{\sigma} = U_t(t)H_tU_t^{\dagger}(t) + iU_t(t)\frac{dU_t^{\dagger}(t)}{dt} \qquad (3.7.a)$$

Equation (3.7.a) which transforms Hamiltonian H_t into $\hat{H}_t^{\sigma}$ is different from the operator transformation (3.4) by an extra term $iU_t(t)\frac{dU_t^{\dagger}(t)}{dt}$. The unitary operator $U_t(t)$ is expressed in terms of the observables in F and thus (3.5) is also true for $U_t(t)$. We may now write (3.7.a) in the form

$$\hat{H}_t^{\sigma} = U_t(t)H_tU_t^{\dagger}(t) + U_t(t)\left([U_t^{\dagger}(t),H_t]+i\frac{\partial U_t^{\dagger}(t)}{\partial t} \right)$$

$$= U_t(t)H_tU_t^{\dagger}(t) + H_t - U_t(t)H_tU_t^{\dagger}(t) + iU_t(t)\frac{\partial U_t^{\dagger}(t)}{\partial t}$$

Therefore,

$$\hat{H}_t^{\sigma} = H_t + i\,U_t(t)\frac{\partial U_t^{\dagger}(t)}{\partial t} \qquad (3.7.b)$$

One simple consequence of equation (3.7.b) is that if $(iU_t(t)\frac{\partial U_t^{\dagger}(t)}{\partial t})$ does not explicitly depend on time, $\langle\hat{H}_t^{\sigma}\rangle$ and $\langle H_t\rangle$ differ only by a time-independent constant. The Galilean transformation is an example.

3.3 Schrödinger Picture

We have derived the relationship between the Heisenberg state vectors, operators and Hamiltonians for the frames of reference F and F^{σ} in the previous section. Now we are

going to establish the equivalent formulations in the Schrödinger picture. In going from the Heisenberg to the Schrödinger picture, one requires that probabilities and eigenvalues and thus expectation values remain unchanged.

In the Schrödinger picture, the state of the physical system in frame F at time t is determined by the state vector $|\Psi(t)\rangle$. From quantum mechanics, this state vector can be obtained by a unitary time development operator $T(t,t_o)$

$$|\Psi(t)\rangle = T(t,t_o)|\Psi(t_o)\rangle \tag{3.8}$$

as long as the state vector $|\Psi(t_o)\rangle$ is known at an earlier time t_o. Here we choose the Schrödinger state vector at t_o to be the same as Heisenberg state vector $|\psi\rangle$

$$|\Psi(t_o)\rangle = |\psi\rangle \tag{3.9}$$

It follows from (3.8) and (3.9) that $T(t,t_o)$ connects the Heisenberg and the Schrödinger state vectors in the inertial frame of reference F

$$|\Psi(t)\rangle = T(t,t_o)|\psi\rangle \tag{3.10}$$

Similar equations can be written in the non-inertial frame of reference F^σ

$$|\Psi^\sigma(t)\rangle = T^\sigma(t,t_o)|\psi^\sigma\rangle = T^\sigma(t,t_o)|\psi\rangle \tag{3.11}$$

From now on $T(t)$ and $T^\sigma(t)$ are used as terms of convenience to represent $T(t,t_o)$ and $T^\sigma(t,t_o)$.

Operators A_o and A_o^σ referred to are the Schrödinger forms of the Heisenberg picture operators A_t and A_t^σ in F and F^σ. They are also transformed by the time development

operators according to

$$A_o = T(t)A_t T^\dagger(t) \tag{3.12}$$

and

$$A_o^\sigma = T^\sigma(t)A_t^\sigma T^{\sigma\dagger}(t) \tag{3.13}$$

where the subscript 'o' denotes the time-independent feature of the Schrödinger picture.

Differentiating (3.12) and applying (3.5), we obtain

$$\frac{dT(t)}{dt}A_t T^\dagger(t) + T(t)\frac{dA_t}{dt}T^\dagger(t) + T(t)A_t\frac{dT^\dagger(t)}{dt}$$

$$=\frac{dT(t)}{dt}T^\dagger(t)A_o - iT(t)[A_t, H_t]T^\dagger(t) + T(t)\frac{\partial A_t}{\partial t}T^\dagger(t) + A_o T(t)\frac{dT^\dagger(t)}{dt}$$

$$= 0$$

Hence, using (3.12) once more,

$$[A_o, \frac{dT(t)}{dt}T^\dagger(t) + iT(t)H_t T^\dagger(t)] = 0$$

Since this is true for any operator A_o, we get the equation of motion for the time development operator as follows:

$$i\frac{dT(t)}{dt} = T(t)H_t = H_o T(t) \tag{3.14}$$

where $H_o = TH_tT^{-1}$. The same operations can be applied to

the operator $T^{\sigma}(t)$. The equation of motion for $T^{\sigma}(t)$ is thus derived as

$$i \frac{dT^{\sigma}(t)}{dt} = T^{\sigma}(t)\hat{H}_t^{\sigma} \tag{3.15}$$

Now differentiating (3.10) with respect to t, we obtain

$$i \frac{d|\Psi(t)>}{dt} = i \frac{dT(t)}{dt} |\psi>$$

Then using (3.14) and (3.10), we get

$$i \frac{d|\Psi(t)>}{dt} = T(t)H_t|\psi> = T(t)H_tT^{\dagger}(t)|\Psi(t)>$$

such that the Schrödinger equation of motion of state vector $|\Psi(t)>$ is written as (MER70)

$$i \frac{d|\Psi(t)>}{dt} = H_o|\Psi(t)> \tag{3.16}$$

The same argument can be made in the non-inertial frame of reference F^{σ}. The Schrödinger equation of motion of state vector $|\Psi^{\sigma}(t)>$ is

$$i \frac{d|\Psi^{\sigma}(t)>}{dt} = \hat{H}_o^{\sigma}|\Psi^{\sigma}(t)> \tag{3.17}$$

where $\hat{H}_o^{\sigma}$, the Hamiltonian, is defined

$$\hat{H}_o^\sigma = T^\sigma(t)\hat{H}_t^\sigma T^{\sigma\dagger}(t) \qquad (3.18)$$

Let us summarize the transformations we have established so far. Table 3-1 gives a schematic relation.

frame of reference	Heisenberg		Schrödinger
inertial F	A_t , H_t	$=\!=\!\Rightarrow$ $T(t)$	A_o , H_o
	$\Vert$		$\Vert$
	$U_t(t)$		?
	$\Downarrow$		$\Downarrow$
non-inertial F^σ	A_t^σ , $\hat{H}_t^\sigma$	$=\!=\!\Rightarrow$ $T^\sigma(t)$	A_o^σ , $\hat{H}_o^\sigma$

Table 3-1

The transformation between the two frames of reference in the Schrödinger picture is our main interest. If we know the relationship between H_o and $\hat{H}_o^\sigma$, $|\psi(t)\rangle$ and $|\psi^\sigma(t)\rangle$, we are ready for the collision problem discussed in Section 3.1.

From (3.10) and (3.11), we may write

$$|\psi^\sigma(t)\rangle = T^\sigma(t)|\psi\rangle = T^\sigma(t)T^\dagger(t)|\psi(t)\rangle \qquad (3.19.a)$$

A new operator S(t) can be defined as

$$S(t) = T^\sigma(t)T^\dagger(t) \qquad (3.20)$$

such that it transforms the Schrödinger state vector $|\psi(t)\rangle$ into $|\psi^\sigma(t)\rangle$ according to

$$|\psi^\sigma(t)\rangle = S(t)|\psi(t)\rangle \qquad (3.19.b)$$

Here S(t) is also a unitary operator. This S(t) operator connects the Schrödinger state vectors in F and F^{σ} frames of reference together. Differentiating both sides of (3.20) with respect to time, we obtain

$$i\frac{dS(t)}{dt} = i\frac{dT^{\sigma}(t)}{dt}T^{\dagger}(t) + i\,T^{\sigma}(t)\frac{dT^{\dagger}(t)}{dt}$$

Substituting the equations of motion of $T^{\dagger}(t)$ and $T^{\sigma}(t)$, (3.14) and (3.15), into the above equation, we get

$$\begin{aligned} i\frac{dS(t)}{dt} &= T^{\sigma}(t)\hat{H}_t^{\sigma}T^{\dagger}(t) - T^{\sigma}(t)H_tT^{\dagger}(t) \\ &= T^{\sigma}(t)\,(\hat{H}_t^{\sigma} - H_t)\,T^{\dagger}(t) \end{aligned}$$

In Section 3.2, we have derived the relation of H_t and $\hat{H}_t^{\sigma}$. So from (3.7.b), we write

$$\begin{aligned} i\frac{dS(t)}{dt} &= i\,T^{\sigma}(t)\,(U_t(t)\frac{\partial U_t^{\dagger}(t)}{\partial t})\,T^{\dagger}(t) \\ &= i\,S(t)T(t)\,(U_t(t)\frac{\partial U_t^{\dagger}(t)}{\partial t})\,T^{\dagger}(t) \end{aligned}$$

which yields the equation of motion for the operator S(t):

$$\frac{dS(t)}{dt} = S(t)\,(U_o(t)\frac{\partial U_o^{\dagger}(t)}{\partial t}) \qquad (3.21)$$

where we use (3.12) to define

$$U_o(t) = T(t)U_t(t)T^{\dagger}(t)$$

and

$$\frac{\partial U_o^\dagger(t)}{\partial t} = T(t) \frac{\partial U_t^\dagger(t)}{\partial t} T^\dagger(t)$$

It is now apparent that if $U_o(t)$, the unitary operator in the Schrödinger picture which corresponds to the unitary operator $U_t(t)$ in the Heisenberg picture, is known, $S(t)$ can be solved and the relation between $|\Psi(t)>$ and $|\Psi^\sigma(t)>$ can be derived. The operator $S(t)$ also can be used to relate the Hamiltonians, H_o and $\hat{H}_o^\sigma$ in the Schrödinger picture. Applying (3.7.b) and (3.12) to equation (3.18), we obtain

$$\hat{H}_o^\sigma = T^\sigma(t) \left(H_t + i\, U_t(t) \frac{\partial U_t^\dagger(t)}{\partial t}\right) T^{\sigma\dagger}(t)$$

$$= T^\sigma(t)T^\dagger(t)H_o T(t)T^{\sigma\dagger}(t) + i\, T^\sigma(t)U_t(t)\frac{\partial U_t^\dagger(t)}{\partial t}T^{\sigma\dagger}(t)$$

$$= S(t)H_o S^\dagger(t) + iS(t)T(t)\left(U_t(t)\frac{\partial U_t^\dagger(t)}{\partial t}\right)T(t)S^\dagger(t)$$

which may be written

$$\hat{H}_o^\sigma = S(t) \left(H_o + i\, U_o(t) \frac{\partial U_o^\dagger(t)}{\partial t} \right) S^\dagger(t) \qquad (3.22)$$

Equations (3.19.b), (3.21) and (3.22) complete our general time-dependent transformation theory. Using these equations, we can relate operators, Hamiltonians, and state vectors in F and F^σ frames of reference, as long as $U_o(t)$ and $U_t(t)$ are constructed. In the following sections, we will discuss the application of this transformation theory to accelerated frames of reference and show its suitability for this purpose.

3.4 Linear Accelerating Frames of Reference

In the previous sections, we have derived a general transformation theory which can transform a state in an inertial frame of reference F to the corresponding state in a non-inertial frame of reference F^σ. Now we will apply this transformation theory to a specific non-inertial frame of reference F^a. We assume that this non-inertial frame F^a moves without rotation with a time-dependent linear, but not necessarily constant, acceleration $\vec{a}(t)$. The translational motion of F^a is characterized by the displacement $\vec{d}(t)$ of the coordinate origin, as shown in Figure 3-1. (Subscript or superscript 'a' attached to the notations indicate that F^a is a linearly accelerating frame of reference relative to an inertial frame of reference F.) We shall look into the dynamics of a single particle first. Our discussion will be quite explicit and detailed on purpose, because there is much confusion in the literature in these matters.

Classically, the coordinates, $\vec{r}=(x,y,z)$ and $\vec{r}^a=(x^a,y^a,z^a)$, and kinetic momenta, $\vec{p}=m\dot{\vec{r}}=(p_x,p_y,p_z)$ and $\vec{p}^a=(p_x^a,p_y^a,p_z^a)$, of the single particle in F and F^a have the following relations:

$$\vec{r}^a = \vec{r} - \vec{d}(t) \tag{3.23}$$

$$\vec{p}^a = \vec{p} - m\,\dot{\vec{d}}(t) \tag{3.24}$$

where m is the mass of the particle and $\dot{\vec{d}}(t)$ is the translation velocity of F^a relative to F. Owing to the close correspondence between classical dynamics and quantum mechanics, similar equations can be written in the Heisenberg picture in terms of operators:

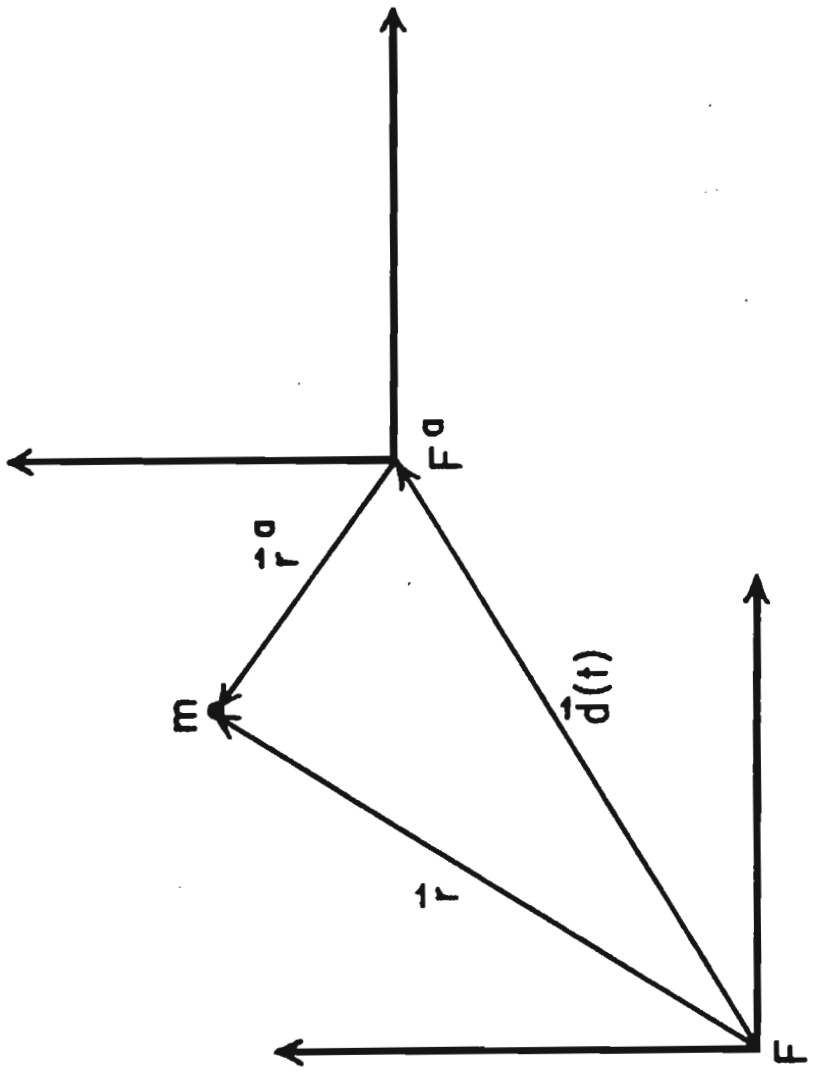


Figure 3-1

$$\vec{X}^a_t = \vec{X}_t - \vec{d}(t) \qquad (3.25.a)$$

$$\vec{P}^a_t = \vec{P}_t - m\,\dot{\vec{d}}(t) \qquad (3.26.a)$$

where $\vec{X}^a_t$ and $\vec{X}_t$ are the position operators and $\vec{P}^a_t$ and $\vec{P}_t$ are the momentum operators in the F^a and F frames of reference respectively. The transformation between the two frames of reference is effected by the unitary operator:

$$U_{ta}(t) = \exp(i(m\dot{\vec{d}}(t)\cdot\vec{X}_t - \vec{d}(t)\cdot\vec{P}_t)) \qquad (3.27)$$

which preserves the commutation relations. We shall prove the appropriateness of this choice by showing that the Heisenberg position and momentum operators are indeed transformed according to (3.4):

$$\vec{X}^a_t = U_{ta}(t)\ \vec{X}_t\ U^\dagger_{ta}(t) \qquad (3.25.b)$$

$$\vec{P}^a_t = U_{ta}(t)\ \vec{P}_t\ U^\dagger_{ta}(t) \qquad (3.26.b)$$

To this purpose, we rewrite the unitary operator $U_{ta}(t)$ in the following form:

$$U_{ta}(t) = e^{im\dot{\vec{d}}(t)\cdot\vec{X}_t}\ e^{-i\vec{d}(t)\cdot\vec{P}_t}\ e^{-im\dot{\vec{d}}(t)\cdot\vec{d}(t)/2}$$

where we have used the identity (MER70)

$$e^A\ e^B = e^{A+B+[A,B]/2} \qquad (3.28)$$

for operators A and B which commute with [A,B]. Let $U_{ta}(t)$ and its adjoint operate on $\vec{X}_t$:

$$U_{ta}(t)\vec{X}_t U_{ta}^{\dagger}(t) = e^{im\dot{\vec{d}}(t)\cdot\vec{X}_t} e^{-i\vec{d}(t)\cdot\vec{P}_t} X_t \; e^{i\vec{d}(t)\cdot\vec{P}_t} e^{-im\dot{\vec{d}}(t)\cdot\vec{X}_t}$$

$$= e^{im\dot{\vec{d}}(t)\cdot\vec{X}_t} \; (\vec{X}_t - \vec{d}(t)) \; e^{-im\dot{\vec{d}}(t)\cdot\vec{X}_t}$$

$$= \vec{X}_t - \vec{d}(t) = \vec{X}_t^a$$

Here we have applied another identity,

$$e^A B e^{-A} = B + [A,B] + \frac{1}{2!}[A,[A,B]] + \frac{1}{3!}[A,[A,[A,B]]] + \ldots \quad (3.29)$$

and the familiar commutation relations

$$[X_t, P_{X_t}] = [Y_t, P_{Y_t}] = [Z_t, P_{Z_t}] = i$$

$$[X_t, P_{Y_t}] = [X_t, P_{Z_t}] = [Y_t, P_{X_t}] = [Y_t, P_{Z_t}]$$

$$= [Z_t, P_{X_t}] = [Z_t, P_{Y_t}] = 0$$

Equation (3.26.b) can be similarly verified. Note that the unitary operator $U_{ta}(t)$ is an operator with time-dependent parameters $\vec{d}(t)$ and $\dot{\vec{d}}(t)$ which imply the linearly accelerated relative motion of F and F^a. We can say that $U_{ta}(t)$ is a "bridge" to connect these two frames of reference.

To obtain the corresponding unitary operator $U_{oa}(t)$ in the Schrödinger picture, the time-development operator $T(t)$ is introduced and applied to the unitary operator $U_{ta}(t)$ to give

$$U_{oa}(t) = T(t)U_{ta}(t)T^{\dagger}(t)$$

Therefore,

$$U_{oa}(t) = \exp(i(m\dot{\vec{d}}(t)\cdot\vec{X}_o - \vec{d}(t)\cdot\vec{P}_o)) \qquad (3.30)$$

where we use the operator transformation relations

$$\begin{aligned}\vec{X}_o &= T(t)\ \vec{X}_t\ T^\dagger(t)\\ \vec{P}_o &= T(t)\ \vec{P}_t\ T^\dagger(t)\end{aligned} \qquad (3.31)$$

according to equation (3.12). Of course, the position and momentum operators preserve their commutation relations in the Schrödinger picture.

In order to obtain an equation for the operator S(t), we now construct equation (3.21). We rewrite equation (3.30) as follows

$$U_{oa}(t) = e^{im\dot{\vec{d}}(t)\cdot\vec{X}_0}\ e^{-i\vec{d}(t)\cdot\vec{P}_0}\ e^{-im\dot{\vec{d}}(t)\cdot\vec{d}(t)/2} \qquad (3.30.a)$$

and then take the adjoint,

$$U^\dagger_{oa}(t) = e^{im\dot{\vec{d}}(t)\cdot\vec{d}(t)/2}\ e^{i\vec{d}(t)\cdot\vec{P}_0}\ e^{-im\dot{\vec{d}}(t)\cdot\vec{X}_0} \qquad (3.30.b)$$

where we utilize the identity (3.28). Differentiating $U^\dagger_{oa}(t)$ with respect to time and multiplying the result from the left by $U_{oa}(t)$ yields

$$U_{oa}(t)\frac{\partial U^\dagger_{oa}(t)}{\partial t} = i\frac{1}{2}m\vec{d}(t)\cdot\ddot{\vec{d}}(t) + i\frac{1}{2}m(\dot{\vec{d}}(t))^2 +$$

$$e^{im\dot{\vec{d}}(t)\cdot\vec{X}_0}(i\dot{\vec{d}}(t)\cdot\vec{P}_o\ )e^{-im\dot{\vec{d}}(t)\cdot\vec{X}_0} - im\ddot{\vec{d}}(t)\cdot\vec{X}_o$$

From the identity (3.29) and the commutation relations between the Schrödinger position and momentum operators, we may write

$$U_{oa}(t)\frac{\partial U_{oa}^{\dagger}(t)}{\partial t} = i\frac{1}{2}m\ddot{\vec{d}}(t)\cdot\vec{d}(t)-i\frac{1}{2}m(\dot{d}(t))^2+i\dot{\vec{d}}(t)\cdot\vec{P}_o-im\ddot{\vec{d}}(t)\cdot\vec{X}_o \quad (3.32)$$

Substituting (3.32) into (3.21) we get the differential equation for operator $S_a(t)$

$$\frac{dS_a(t)}{dt} = S_a(t)\ (iG(T) + i\dot{\vec{d}}(t)\cdot\vec{P}_o - im\ddot{\vec{d}}(t)\cdot\vec{X}_o) \quad (3.33)$$

where we define

$$G(t) = m\ (\ddot{\vec{d}}(t)\cdot\vec{d}(t)-(\dot{d}(t))^2)/2$$

(3.33) is the equation which we must solve to obtain the transformation between the two frames of reference.

In order to solve (3.33), we first integrate (3.33) and apply the initial conditions. Then we derive the solution of $S_a(t)$ at time t

$$S_a(t) = \exp\left(\int_0^t dt'(iG(t')+i\dot{\vec{d}}(t')\cdot\vec{P}_o-im\ddot{\vec{d}}(t')\cdot\vec{X}_o\right)$$

$$= \exp(iF'(t)+i(\vec{d}(t)-\vec{d}(0))\cdot\vec{P}_o-im(\dot{\vec{d}}(t)-\dot{\vec{d}}(0))\cdot\vec{X}_o)$$

$$= e^{iF(t)}\ e^{i(\vec{d}(t)-\vec{d}(0))\cdot\vec{P}_0}\ e^{-im(\dot{\vec{d}}(t)-\dot{\vec{d}}(0))\cdot\vec{X}_0} \quad (3.34)$$

where F(t) is defined as

$$F(t) = \int_0^t dt' G(t') + m(\vec{d}(t)-\vec{d}(0))\cdot(\dot{\vec{d}}(t)-\dot{\vec{d}}(0))/2$$

and identity (3.28) is used again here.

We now seek to demonstrate how the operators, $U_{oa}(t)$ and $S_a(t)$, transform the quantities in the Schrödinger picture from the F frame to the F^a frame of reference. To show this we consider Table 3-1 in a clockwise direction. First, we relate the Schrödinger position and momentum operators, $\vec{X}_o^a$ and $\vec{P}_o^a$, in F^a back to the corresponding Schrödinger operators in the F frame of reference. According to equations (3.13), (3.25.a) and (3.12), we obtain

$$\vec{X}_o^a = T^a(t)\ (\vec{X}_t - \vec{d}(t))\ T^{a\dagger}(t)$$

$$= S_a(t)\vec{X}_o S_a^\dagger(t) - \vec{d}(t) \qquad (3.35.a)$$

for the position operator and

$$\vec{P}_o^a = T^a(t)\ (\vec{P}_t - m\dot{\vec{d}}(t))\ T^{a\dagger}(t)$$

$$= S_a(t)\vec{P}_o S_a^\dagger(t) - m\dot{\vec{d}}(t) \qquad (3.36.a)$$

for the momentum operator, where $S_a(t)$ is defined as (3.20) and $T^a(t)$ is used as a unitary operator. Then substituting (3.34) in the above equations, we may write

$$\vec{X}_o^a = e^{i(\vec{d}(t)-\vec{d}(0))\cdot\vec{P}_0}\ \vec{X}_o\ e^{-i(\vec{d}(t)-\vec{d}(0))\cdot\vec{P}_0} - \vec{d}(t)$$

$$= \vec{X}_o + i(\vec{d}(t)-\vec{d}(0))\cdot[\vec{P}_o,\vec{X}_o] - \vec{d}(t)$$

$$\vec{X}_o^a = \vec{X}_o - \vec{d}(0) \qquad (3.35.b)$$

and

$$\vec{P}_o^a = e^{-im(\dot{\vec{d}}(t)-\dot{\vec{d}}(0))\cdot\vec{X}_0}\,\vec{P}_o\,e^{im(\dot{\vec{d}}(t)-\dot{\vec{d}}(0))\cdot\vec{X}_0} - m\dot{\vec{d}}(t)$$

$$= \vec{P}_o - im(\dot{\vec{d}}(t)-\dot{\vec{d}}(0))\cdot[\vec{X}_o,\vec{P}_o] - m\dot{\vec{d}}(t)$$

$$\vec{P}_o^a = \vec{P}_o - m\dot{\vec{d}}(0) \qquad (3.36.b)$$

Here we again use the identity (3.29) and $[\vec{X}_o,\vec{P}_o]=i$. These relations look almost trivial but nevertheless deserve some comments.

As shown in equation (3.35.b) and (3.36.b), our general transformation theory leaves the Schrödinger position and momentum operators time-independent. $\vec{X}_o^a$ and $\vec{P}_o^a$ are determined at time zero and are not changing as time proceeds, as is characteristics of Schrödinger operators. But $\vec{X}_o^a$ and $\vec{P}_o^a$ are of course observables, and their expectation values are the values of position and momentum in the classical limit and changing with time. (Momentum is here defined as a product of mass with velocity). Where does the time-dependence in the expectation values come from? In the Schrödinger picture, the time-dependence is embedded in the state vectors and we will show in a later section how the state vectors or wavé functions are changing with time.

Second, we consider how the Hamiltonian is transformed in the accelerating reference frame. Substituting (3.32) into (3.22), we may write

$$\hat{H}_o^a = S_a(t)H_oS_a^\dagger(t) - \frac{1}{2}m\ddot{\vec{d}}(t)\cdot\vec{d}(t) + \frac{1}{2}m\dot{\vec{d}}(t)\cdot\dot{\vec{d}}(t)$$

$$-\dot{\vec{d}}(t)\cdot(S_a(t)\vec{P}_oS_a^\dagger(t)) + m\ddot{\vec{d}}(t)\cdot(S_a(t)\vec{X}_oS_a^\dagger(t))$$

Let us assume that H_o is a function of $\vec{X}_o$, $\vec{P}_o$ and t and consists of kinetic energy and potential energy as usual,

$$H_o = H(\vec{X}_o,\vec{P}_o,t) = (\vec{P}_o)^2/2m + V(\vec{X}_o,t) \qquad (3.37)$$

Since $S_a(t)$ transforms $\vec{X}_o$ and $\vec{P}_o$ into $\vec{X}_o^a$ and $\vec{P}_o^a$ according to (3.35.b) and (3.36.b), we now rewrite $\hat{H}_o^a$ as follows:

$$\hat{H}_o^a = H(\vec{X}_o^a+\vec{d}(t),\vec{P}_o^a+m\dot{\vec{d}}(t),t) - m\ddot{\vec{d}}(t)\cdot\vec{d}(t)/2 + m\dot{\vec{d}}(t)\cdot\dot{\vec{d}}(t)/2$$

$$- \dot{\vec{d}}(t)\cdot(\vec{P}_o^a+m\dot{\vec{d}}(t)) + m\ddot{\vec{d}}(t)\cdot(\vec{X}_o^a+\vec{d}(t))$$

$$= (\vec{P}_o^a+m\dot{\vec{d}}(t))\cdot(\vec{P}_o^a+m\dot{\vec{d}}(t))/2m + V(\vec{X}_o^a+\vec{d}(t),t) - \dot{\vec{d}}(t)\cdot\vec{P}_o^a$$

$$+ m\ddot{\vec{d}}(t)\cdot\vec{X}_o^a + m\ddot{\vec{d}}(t)\cdot\vec{d}(t)/2 - m\dot{\vec{d}}(t)\cdot\dot{\vec{d}}(t)/2$$

so that

$$\hat{H}_o^a = (\vec{P}_o^a)^2/2m + V(\vec{X}_o^a,\vec{d}(t),t) + m\ddot{\vec{d}}(t)\cdot\vec{X}_o^a - m\ddot{\vec{d}}(t)\cdot\vec{d}(t)/2 \qquad (3.38)$$

It is interesting to see that the new Hamiltonian $\hat{H}_o^a$ in the F^a frame of reference which describes the same physical system as in F frame of reference, has two additional terms besides the kinetic and potential energy terms. These last two terms in (3.38) are explicitly products of force with distance. For a non-inertial frame of reference F^a, there must be an external force which accelerates the particle. If the frame of reference F^a is not accelerating but rather moving with constant velocity with respect to the F frame of reference, the last two terms in (3.38) vanish and the Hamiltonian satisfies the following relation:

$$\hat{H}_o^a = U_o(t)H_oU_o^\dagger(t) = U_o(t)H(\vec{X}_o,\vec{P}_o,t)U_o^\dagger(t) = H(\vec{X}_o^a,\vec{P}_o^a,t)$$

This special case is classified as a Galilean transformation.

Another important aspect to emphasize is that the Hamiltonian in equation (3.38) also satisfies Hamilton's equations of motion as it should in classical dynamics, even though the dynamical variables now refer to an accelerated frame of reference. It asserts the correctness of our general transformation theory. There is another transformation theory published in the literature (SCH77) in which the Hamiltonian does not satisfy Hamilton's equations of motion and the Schrödinger operators are time varying. These formulations of the theory are less satisfactory from our point of view.

Equation (3.19.b) gives the relationship between state vectors in frames of reference F and F^a. We will demonstrate how the wave functions are related with each other in the next section.

3.5 Wave Functions and Translation Factors

In scattering theory, we are interested in excitation and ionization probabilities which are calculated from wave functions. Most of scattering theory deals with wave packets which describe the propagation of a physical system in space and time. Here we represent the state vector in a coordinate representation, which is spanned by the eigenvectors of the position operator. Before we investigate how the wave funtions in F and F^a are related, we introduce a new notation for bra/ket vectors. We shall write

$$\vec{X}_o \; |\vec{X}_o;\vec{r}\rangle = \vec{r} \; |\vec{X}_o;\vec{r}\rangle \qquad (3.39)$$

where $\vec{X}_o$ is the position operator, and $\vec{r}=(x,y,z)$ is the eigenvalue of the eigenvector $|\vec{X}_o;\vec{r}\rangle$. $\vec{r}$ is a vector whose components are real numbers and is called the coordinate. (In Dirac's old terminology, $\vec{X}_o$ is a q-number, and $\vec{r}$ is a c-number.) The notation, which may appear redundant, is chosen in order to distinguish two or more eigenvectors of different operators but with the same eigenvalue. In this case the usual specification of an eigenvector by its corresponding eigenvalue is not adequate and sufficient. Let us first discuss some of the properties of this coordinate eigenvector.

Similar to equation (3.39), we also write

$$\vec{X}_o^a \; |\vec{X}_o^a;\vec{r}^a\rangle = \vec{r}^a \; |\vec{X}_o^a;\vec{r}^a\rangle \tag{3.40}$$

$\vec{r}^a$ represents the coordinate in the frame of reference F^a. From the discussion of the transformation theory in the previous sections and equations (3.13), (3.4), and (3.12), we may write

$$\begin{aligned}\vec{X}_o^a|\vec{X}_o^a;\vec{r}^a\rangle &= T^a(t)U_{ta}(t)T^\dagger(t)\vec{X}_o T(t)U_{ta}^\dagger(t)T^{a\dagger}(t)|\vec{X}_o^a;\vec{r}^a\rangle \\ &= T^a(t)U_{ta}(t)T^\dagger(t)\vec{X}_o \; |\vec{X}_o;\vec{r}^a\rangle \\ &= T^a(t)U_{ta}(t)T^\dagger(t)\vec{r}^a|\vec{X}_o;\vec{r}^a\rangle\end{aligned}$$

Therefore

$$\vec{X}_o^a|\vec{X}_o^a;\vec{r}^a\rangle = \vec{r}^a \; (T^a(t)U_{ta}(t)T^\dagger(t)|\vec{X}_o;\vec{r}^a\rangle) \tag{3.41}$$

where we have used the transformation between operators $\vec{X}_o$ and $\vec{X}_o^a$. Comparing the above equation with (3.40), we

can see that $|\vec{X}_o^a;\vec{r}^a\rangle$ and $(T^a(t)U_{ta}(t)T^\dagger(t)|\vec{X}_o;\vec{r}^a\rangle)$ have the same eigenvalue, $\vec{r}^a$, and we obtain the relation between vectors $|\vec{X}_o^a;\vec{r}^a\rangle$ and $|\vec{X}_o;\vec{r}^a\rangle$:

$$|\vec{X}_o^a;\vec{r}^a\rangle = S_a(t)U_{oa}(t)\ |\vec{X}_o;\vec{r}^a\rangle \tag{3.42}$$

where we have used

$$U_{oa}(t) = T(t)U_{ta}(t)T^\dagger(t)$$

and equation (3.20).

To link the wave function with the state vector explicitly, we use the eigenvectors of the position operator $\vec{X}_o$ as a basis and write

$$\Psi(\vec{r};t) = \langle\vec{X}_o;\vec{r}|\Psi(t)\rangle \tag{3.43}$$

Similarly, in frame of reference F^a,

$$\Psi^a(\vec{r}^a;t) = \langle\vec{X}_o^a;\vec{r}^a|\Psi^a(t)\rangle \tag{3.44}$$

Substituting (3.19.b), (3.42) and (3.30.b) into (3.44), we get

$$\begin{aligned}\Psi^a(\vec{r}^a;t) &= \langle\vec{X}_o;\vec{r}^a|U_{oa}^\dagger(t)S_a^\dagger(t)S_a(t)|\Psi(t)\rangle \\ &= \langle\vec{X}_o;\vec{r}^a|U_{oa}^\dagger(t)|\Psi(t)\rangle \\ &= \langle\vec{X}_o;\vec{r}^a|e^{im\dot{\vec{d}}(t)\cdot\vec{d}(t)/2}\ e^{i\vec{d}(t)\cdot\vec{P}_0}\ e^{-im\dot{\vec{d}}(t)\cdot\vec{X}_0}|\Psi(t)\rangle\end{aligned}$$

because $S_a(t)$ is a unitary operator. Since $\exp(-i\vec{d}(t)\cdot\vec{P}_o)$ is a translation operator, whose effect is to shift the

coordinate by a constant distance $\vec{d}(t)$, i.e.,

$$e^{-i\vec{d}(t)\cdot\vec{P}_0}\ |\vec{X}_o;\vec{r}\rangle = |\vec{X}_o;\vec{r}+\vec{d}(t)\rangle \tag{3.45}$$

we may write

$$\psi^a(\vec{r}^a,t) = e^{im\dot{\vec{d}}(t)\cdot\vec{d}(t)/2}\ \langle\vec{X}_o;\vec{r}^a+\vec{d}(t)|\ e^{-im\dot{\vec{d}}(t)\cdot\vec{X}_0}|\psi(t)\rangle$$

$$= e^{im\dot{\vec{d}}(t)\cdot\vec{d}(t)/2}\ e^{-im\dot{\vec{d}}(t)\cdot\vec{r}}\ \langle\vec{X}_o;\vec{r}|\psi(t)\rangle$$

where we have used equation (3.23) and

$$e^{im\dot{\vec{d}}(t)\cdot\vec{X}_0}\ |\vec{X}_o;\vec{r}\rangle = e^{im\dot{\vec{d}}(t)\cdot\vec{r}}\ |\vec{X}_o;\vec{r}\rangle$$

we may therefore write the relation between wave functions $\psi^a(\vec{r}^a,t)$ and $\psi(\vec{r},t)$:

$$\psi^a(\vec{r}^a;t) = e^{im\dot{\vec{d}}(t)\cdot\vec{d}(t)/2}\ e^{-im\dot{\vec{d}}(t)\cdot\vec{r}}\ \psi(r;t) \tag{3.46.a}$$

or conversely since $\vec{r}=\vec{r}^a+\vec{d}(t)$,

$$\psi(\vec{r};t) = e^{im\dot{\vec{d}}(t)\cdot\vec{d}(t)/2}\ e^{im\dot{\vec{d}}(t)\cdot\vec{r}^a}\psi^a(\vec{r}^a;t) \tag{3.46.b}$$

The phase factors in (3.46.a) and (3.46.b) are termed translation factors and have been the subject of much discussion in the literature of atomic collision physics. They provide the relation between the wave propagation in two different frames of reference F and F^a which are in relative motion. If there is no relative velocity between F and F^a, i.e. $\dot{\vec{d}}(t)=0$, wave packets propagate the same in both frames. If there is a constant relative velocity,

i.e. $\dot{\vec{d}}(t)=\vec{v}$ or $\vec{d}(t)=\vec{v}t$, the translation factor becomes $\exp(im(v^2t/2-\vec{v}\cdot\vec{r}))$, corresponding to a Galilean transformation.

3.6 Rotating Frames

In this section, we will consider another frame of reference F^R which rotates around a fixed axis (y-axis) with angular velocity $\dot{\theta}(t)$ relative to frame of reference F as shown in Figure 3-2. In classical mechanics, the coordinates $\vec{r}=(x,y,z)$ and $\vec{r}^R=(x^R,y^R,z^R)$ in the frames of reference F and F^R respectively are related as follows:

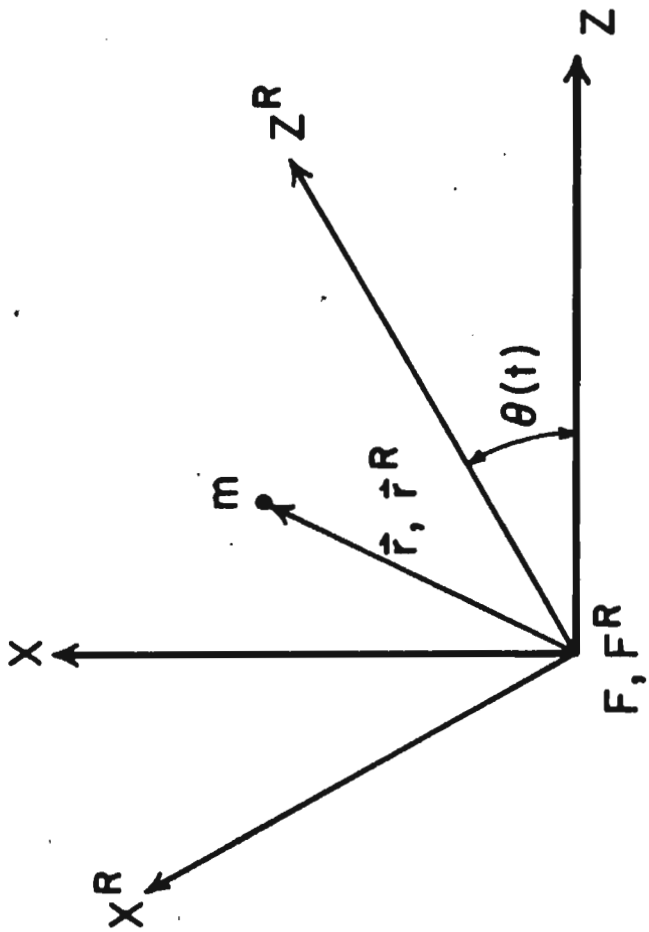
$$\begin{aligned} x^R &= -z\ \sin\theta(t) + x\ \cos\theta(t) \\ y^R &= y \\ z^R &= z\ \cos\theta(t) + x\ \sin\theta(t) \end{aligned} \tag{3.47}$$

where $\theta(t)$ is the angle of rotation.

In quantum mechanics, the relationship between the corresponding Heisenberg position operators $\vec{X}_t=(X_t,Y_t,Z_t)$ and $\vec{X}^R_t=(X^R_t,Y^R_t,Z^R_t)$ in frames F and F^R may be written as

$$\begin{aligned} X^R_t &= -Z_t\ \sin\theta(t) + X_t\ \cos\theta(t) \\ Y^R_t &= Y_t \\ X^R_t &= Z_t\ \cos\theta(t) + X_t\ \sin\theta(t) \end{aligned} \tag{3.48}$$

The relation between the momentum operators are



Y axis is out of the paper

Figure 3-2

$$
\begin{aligned}
P_{X_t}^R &= -P_{Z_t} \sin\theta(t) + P_{X_t} \cos\theta(t) \\
P_{Y_t}^R &= P_{Y_t} \\
P_{Z_t}^R &= P_{Z_t} \cos\theta(t) + P_{X_t} \sin\theta(t)
\end{aligned}
\tag{3.49}
$$

Here we choose a particular canonical transformation such that the position and momentum operators in the F^R frame are the position and momentum values projected from the frame of reference F. Therefore, the transformations (3.48) and (3.49) are merely geometric and leave the kinematic relationships unchanged. In particular, the momentum is defined in terms of the motion with respect to the frame F.

In contrast, in the case of transformation to a linear accelerating frame, we made a kinematic transformation and introduced momenta with respect to F^a as well as F. An analogous canonical transformation between F and F^R in terms of rotating Cartesian coordinates is not possible, since the commutation relations cannot be preserved. Using polar coordinates, it is possible to construct such a canonical transformation and derive a Hamiltonian, but the formula are so complicated that it seems no practical use for the present purpose.

Corresponding to the transformations (3.48) and (3.49), a unitary rotation operator can be constructed in the form

$$U_{tR}(t) = \exp\left(-i\, \hat{j}\cdot\vec{L}_t\theta(t)\right) \tag{3.50.a}$$

where $\hat{j}$ is the unit vector defining the axis of rotation and $\vec{L}_t$ is the angular momentum operator. We may rewrite (3.50.a) as

$$U_{tR}(t) = \exp\left(-i\, L_{Y_t}\theta(t)\right) \tag{3.50.b}$$

It is easy to show that this unitary operator accomplishes the transformation (3.48) and (3.49).

If we apply the time-development operator to $U_{tR}(t)$, we obtain the rotation operator in the Schrödinger picture

$$U_{oR}(t) = \exp\ (-iL_{Y_0}\theta(t)) \tag{3.51}$$

Following the same procedure as we discussed in section 3.3, we find that the derivative of $U^{\dagger}_{oR}(t)$ with respect to time is

$$\frac{\partial U^{\dagger}_{oR}(t)}{\partial t} = i\ L_{Y_0}\dot{\theta}(t)\ \exp(iL_{Y_0}\theta(t))$$

so that

$$U_{oR}(t)\frac{\partial U^{\dagger}_{oR}(t)}{\partial\ t} = i\ L_{Y_0}\omega(t) \tag{3.52}$$

where $\omega(t) = \dot{\theta}(t)$ is defined as the angular velocity which is generally not constant. According to (3.21), the equation of motion of the operator $S_R(t)$ is

$$\frac{dS_R(t)}{dt} = S_R(t)\ (iL_{Y_0}\omega(t))$$

from which we can solve for $S_R(t)$:

$$S_R(t) = \exp\ (\ i\ L_{Y_0}(\theta(t)-\theta(0)))$$

If we suppose that the two frames of reference coincide with each other at time zero, then $\theta(0)=0$, and $S_R(t)$ is

$$S_R(t) = \exp\,(i\, L_{Y_0}\theta(t)) \tag{3.53}$$

Now, applying the time-development operators to the x-components of position operators $\vec{x}^R_t$ and $\vec{x}_t$, we can further deduce the relation between the x-components of operators $\vec{x}^R_o$ and $\vec{x}_o$

$$X^R_o = T^R(t)X^R_t T^{R\dagger}(t)$$

$$= T^R(t)(-Z_t\sin\theta(t) + X_t\cos\theta(t))T^{R\dagger}(t)$$

$$= -(T^R(t)T^\dagger(t)Z_o T(t)T^{R\dagger}(t))\sin\theta(t) +$$

$$(T^R(t)T^\dagger(t)Z_o T(t)T^{R\dagger}(t))\cos\theta(t)$$

$$= S_R(t)Z_o S^\dagger_R(t)(-\sin\theta(t)) + S_R(t)X_o S^\dagger_R(t)(\cos\theta(t))$$

As demonstrated before, the results of the operator $S_R(t)$ operating on Z_o and X_o are

$$S_R(t)Z_o S^\dagger_R(t) = Z_o\cos\theta(t) - X_o\sin\theta(t)$$

$$S_R(t)X_o S^\dagger_R(t) = X_o\cos\theta(t) + Z_o\sin\theta(t)$$

Therefore we conclude that

$$X^R_o = X_o$$

This equation again exhibits the time-independence of the Schrödinger operators.

Now we can examine how the Schrödinger Hamiltonians in frames F and F^R differ from each other. Let us assume that the Hamiltonian H_o is a function of $\vec{x}_o$, $\vec{p}_o$ and t, as

expressed in equation (3.37). According to (3.22) and (3.52), the Hamiltonian $\hat{H}_o^R$ in the frame of reference F^R can be written as

$$\hat{H}_o^R(\vec{X}_o^R,\vec{P}_o^R,t) = H(\vec{X}_o^R,\vec{P}_o^R,t) - L_{Y_0R}\omega(t) \qquad (3.54)$$

If the frame of reference F^R does not rotate, that is $\omega(t)=0$, then the two Hamiltonian H_o and $\hat{H}_o^R$ are the same.

Finally we are going to show how the wave functions in frames of reference F and F^R are related to each other. We shall use the new notation introduced in Section 3.5. In the coordinate representation, the wave functions can be written in the forms

$$\psi(\vec{r},t) = \langle\vec{X}_o;\vec{r}|\psi(t)\rangle$$

$$\psi^R(\vec{r}^R;t) = \langle\vec{X}_o^R;\vec{r}^R|\psi^R(t)\rangle$$

From the previous discussion, we may derive

$$\psi^R(\vec{r}^R;t) = \langle\vec{X}_o^R;\vec{r}^R|S_R(t)|\psi(t)\rangle$$

$$= \langle\vec{X}_o;\vec{r}^R|U_{oR}^\dagger(t)|\psi(t)\rangle$$

$$= \langle\vec{X}_o;\vec{r}|\psi(t)\rangle$$

or

$$\psi^R(\vec{r}^R;t) = \psi(\vec{r};t) \qquad (3.55)$$

Here we have used equation (3.19.b),

$$|\vec{X}_o^R;\vec{r}^R\rangle = S_R(t)U_{oR}(t)\ |\vec{X}_o;\vec{r}^R\rangle$$

which is similar to equation (3.42), and the fact that $U_{OR}(t)$ $(=\exp(-iL_{Y_0}\theta(t)))$ will rotate the coordinates $\vec{r}^R = (x^R, y^R, z^R)$ clockwise by an angle $\theta(t)$.

Equation (3.55) has no rotation factor between wave functions. Also, as seen equation (3.54), the two Hamiltonians differ only by $L_{Y_O^R}\omega(t)$, whereas in the case of translations there are two correction terms. This result is explained by the purely geometric nature of the canonical transformation.

CHAPTER 4

NUMERICAL METHODS AND COMPUTATIONAL RESULTS

4.1 Determination of Matrix Elements

Armed with the theory developed in the last two chapters, we now set out to perform the actual calculation. Before solving the coupled equations, the matrix elements (2.18) must be evaluated first. The matrix elements are expressed in the C. M. frame of reference. As we discussed in Chapter 3, if we use different frames of reference, i.e., the target or the projectile frames of reference, and make the proper transformation, we may use the hydrogenic wave functions to evaluate the matrix elements.

To evaluate the direct coupling matrix elements $H_{mn}(A)$, we will use the projectile frame of reference. Applying equation (3.19.b) to transform the state vectors from the C.M. frame to the projectile frame, we get

$$H_{mn}(A) = \langle \Phi_m(A,t) | \; H_o - i\frac{\partial}{\partial t} \; | \Phi_n(A,t) \rangle$$

$$= \langle \Phi_m^A(A,t) | \; S_A(t)(H_o - i\frac{\partial}{\partial t})S_A^\dagger(t) | \Phi_n^A(A,t) \rangle$$

where $|\Phi_m^A(A,t)\rangle$ and $|\Phi_n^A(A,t)\rangle$ are the Schrödinger state vectors in the projectile frame of reference. From equations (3.21) and (3.22), we obtain

$H_{mn}(A)$

$$=\langle\Phi_m^A(A,t)|\hat{H}_O^A-iS_A(t)U_{OA}(t)\frac{\partial U_{OA}^\dagger(t)}{\partial t}S_A^\dagger(t)-iS_A(t)\frac{\partial S_A^\dagger(t)}{\partial t}-i\frac{\partial}{\partial t}|\Phi_n^A(A,t)\rangle$$

$$=\langle\Phi_m^A(A,t)|\hat{H}_O^A-i(\frac{dS_A(t)}{dt}S_A^\dagger(t)+S_A(t)\frac{dS_A^\dagger(t)}{dt})-i\frac{\partial}{\partial t}|\Phi_n^A(A,t)\rangle$$

So

$$H_{mn}(A) = \langle\Phi_m^A(A,t)|\ \hat{H}_O^A - i\ \frac{\partial}{\partial t}\ |\Phi_n^A(A,t)\rangle \qquad (4.1)$$

Here we have used the unitary property:

$$S_A(t)S_A^\dagger(t) = 1$$

and

$$\frac{dS_A(t)}{dt}S_A^\dagger(t) + S_A(t)\frac{dS_A^\dagger(t)}{dt} = 0$$

The Schrödinger Hamiltonian $\hat{H}_O^A$ in equation (4.1) is in the projectile frame of reference. Since the projectile frame is linear accelerating and rotating with respect to the C. M. frame of reference in the coordinate representation, the Hamiltonian becomes according to (3.38) and (3.54)

$$\hat{H}_O^A= \tfrac{1}{2}(\nabla^A)^2- \frac{Z_1}{r_A} - \frac{Z_2}{|\vec{r}_A+\vec{R}(t)|}+\ddot{\vec{R}}^A(t)\cdot\vec{X}_O^A+ \tfrac{1}{2}\ddot{\vec{R}}^A(t)\cdot\vec{R}^A(t)-L_{\hat{y}_A}\omega(t) \qquad (4.2)$$

Here there are three correction terms: two corresponding to linear acceleration and one to rotation.

Similarly, if we use the target frame to evaluate the

direct coupling matrix elements $\bar{H}_{mn}(B)$, we write

$$\bar{H}_{mn}(B) = \langle\Phi_m(B,t)|H_o - i\frac{\partial}{\partial t}|\Phi_n(B,t)\rangle$$

$$= \langle\Phi_m^B(B,t)|\hat{H}_o^B - i\frac{\partial}{\partial t}|\Phi_n^B(B,t)\rangle \qquad (4.3)$$

where $\hat{H}_o^B$ is defined as

$$\hat{H}_o^B = \tfrac{1}{2}(\nabla^B)^2 - \frac{Z_1}{|\vec{r}_B - \vec{R}(t)|} - \frac{Z_2}{r_B} + \ddot{\vec{R}}^B(t)\cdot\vec{X}_o^B + \tfrac{1}{2}\ddot{\vec{R}}^B(t)\cdot\vec{R}^B(t) - L_{\hat{y}_B}\omega(t) \qquad (4.4)$$

and $|\Phi_m^B(B,t)\rangle$ and $|\Phi_n^B(B,t)\rangle$ are Schrödinger state vectors in the target frame of reference.

Finally, in the coupled equations, the exchange coupling matrix elements and the overlap matrix elements are defined as

$$K_{mn}(A,B) = \langle\Phi_m^B(A,t)|\ \hat{H}_o^B - i\frac{\partial}{\partial t}\ |\Phi_n^B(B,t)\rangle \qquad (4.5)$$

$$\bar{K}_{mn}(B,A) = \langle\Phi_m^A(B,t)|\ \hat{H}_o^A - i\frac{\partial}{\partial t}\ |\Phi_n^A(A,t)\rangle \qquad (4.6)$$

$$S_{mn} = \langle\Phi_m^A(A,t)|\Phi_n^A(B,t)\rangle \qquad (4.7)$$

$$S_{mn}^{\dagger} = \langle\Phi_m^B(B,t)|\Phi_n^B(A,t)\rangle \qquad (4.8)$$

The next consideration to arise is the total number of states to be used in the expansion. The more states we include, the more accurate the calculation will be, but the number of coupled equations that need to be solved

increases correspondingly. We decide to consider states with principal quantum number n=1 and n=2, which are a total of five states. We express the state vectors in the projectile and target frames of reference, both of which share a common y-axis perpendicular to the plane of projectile motion. Because of the reflection symmetry about the plane of motion, the Hamiltonians $\hat{H}_o^A$ and $\hat{H}_o^B$ both are even operators under the reflection $y \to -y$, and the five states break up into two groups. The four of these states, 1S, 2S, $2P_z$, and $2P_x$ are even with respect to transformation from y to -y. Only the state $2P_y$ is odd with respect to y reflection, and because of selection rules, does not interact with the other four states. Therefore, we include only the 1S, 2S, $2P_z$, and $2P_x$ hydrogenic states centered both on the target nucleus and the projectile ion. We write the general form of the wave function as

$$\langle \vec{r}|n\rangle = \Phi_n(\vec{r})\exp(-iE_nt) \qquad (4.9.a)$$

where E_n is the energy level of a hydrogenic atom with nuclear charge Z.

$$E_n = Z^2/(2n^2) \qquad (4.9.b)$$

Then the four hydrogenic wave functions are

$$\Phi_{1S}(\vec{r}) = R_{10}Y_0^0(\theta,\phi) = \frac{1}{\sqrt{\pi}}\, Z^{3/2}\exp(-Zr)$$

$$\Phi_{2S}(\vec{r}) = R_{20}(r)Y_0^0(\theta,\phi) = \frac{1}{4\sqrt{2\pi}}\, Z^{3/2}(2-Zr)\exp(-Zr/2) \qquad (4.10)$$

$$\Phi_{2P_z}(\vec{r}) = R_{21}(r)Y_1^0(\theta,\phi) = \frac{1}{4\sqrt{2\pi}}\, Z^{3/2}(rZ)\cdot\cos\theta\,\exp(-Zr/2)$$

$$\phi_{2P_x}(\vec{r}) = \tfrac{1}{2} R_{21}(r)(Y_1^{-1}(\theta,\phi) - Y_1^1(\theta,\phi))$$

$$= \frac{1}{4\sqrt{2\pi}} Z^{3/2} (rZ)\cdot\sin\theta\cos\phi \exp(-Zr/2)$$

where $R_{n\ell}(r)$ is the radial wave function and Y_ℓ^m is a spherical harmonic (MOT65). Z is the charge of the nucleus, and (r,θ,ϕ) are the polar coordinates of the electron. Therefore the equations (2.17) become 16 coupled equations. In our case, there is a complete lack of orthogonality within the expansion, because at finite time t eigenfunctions centered on different nuclei are generally not orthogonal. Only at $t=+\infty$ and $t=-\infty$, do the hydrogenic wave functions centered on different nuclei approach orthogonality. The hydrogenic wave functions centered on the same nuclei remain orthogonal through the whole time interval.

Explicit expressions for the direct coupling, exchange coupling, and overlap matrix elements are derived in Appendix A and B respectively.

4.2 Numerical Method

The coupled differential equations (2.17) can be written as

$$T\,\vec{a} = V\,\dot{\vec{a}} \tag{4.11}$$

where $\vec{a}$ is a 16-elements vector defined as

$$\vec{a} = \begin{pmatrix} a_{1Sr} \\ a_{2Sr} \\ a_{2P_zr} \\ a_{2P_xr} \\ a_{1Si} \\ a_{2Si} \\ a_{2P_zi} \\ a_{2P_xi} \\ b_{1Sr} \\ b_{2Sr} \\ b_{2P_zr} \\ b_{2P_xr} \\ b_{1Si} \\ b_{2Si} \\ b_{2P_zi} \\ b_{2P_xi} \end{pmatrix} \tag{4.12}$$

and T and V are two 16 by 16 matrices:

$$T = \begin{pmatrix} H_r(A) & -H_i(A) & K_r(A,B) & -K_i(A,B) \\ H_i(A) & H_r(A) & K_i(A,B) & K_r(A,B) \\ \overline{K}_r(B,A) & -\overline{K}_i(B,A) & \overline{H}_r(B) & -\overline{H}_i(B) \\ \overline{K}_i(B,A) & \overline{K}_r(B,A) & \overline{H}_i(B) & \overline{H}_r(B) \end{pmatrix} \tag{4.13.a}$$

$$V = \begin{pmatrix} 0 & -I & -S_i & -S_r \\ I & 0 & S_r & -S_i \\ -S_i^\dagger & -S_r^\dagger & 0 & -I \\ S_r^\dagger & -S_i^\dagger & I & 0 \end{pmatrix} \qquad (4.13.b)$$

The subscripts 'r' and 'i' in the above matrix elements denote the real and imaginary parts. The matrix elements $H(A)$, $\bar{H}(B)$, $K(A,B)$, $\bar{K}(B,A)$, S and $S^\dagger$ are themselves 4 by 4 matrices which are defined in equation (2.18).

We may rewrite equation (4.11) as

$$\frac{d\,\vec{a}}{d\,t} = V^{-1}\,T\,\vec{a} = W\,\vec{a} \qquad (4.11.b)$$

This is the coupled differential equations which we want to solve.

The matrix elements of T and V in equation (4.13) are combinations of volume integrals over the electron coordinates. As shown in Table B-1, these volume integrals are double integrations over 'r' and 'θ' with the following format:

$$D_{j\pm}(\eta_1,\eta_2,k,\ell,m,n) = \int_0^\infty dr\; r^k e^{-\eta_1 r} \int_{-1}^{1} dx\; e^{-\eta_2 y_\pm(r,x)}\; e^{i\dot{R}(t)rx}$$

$$J_j(\lambda r(1-x^2)^{\frac{1}{2}})(y_\pm(r,x))^\ell x^m\,(1-x^2)^{n/2}$$

where

$$x = \cos\theta$$

$$\lambda = v_1 b/R(t)$$

$$y_\pm(r,x) = (r^2+R^2(t)\pm 2R(t)r\cdot\cos\theta)^{\frac{1}{2}}$$

We evaluate these double integrals numerically. There are 21 different complex angular integrals (integrals over $\cos\theta$) and a total of 57 complex double integrals. We compute real and imaginary parts separately. For the angular integrals, we use the Clenshaw-Curtis quadrature (GEN72) which applies the Chebyshev series expansion. A fast Fourier transform algorithm is used to calculate the cosine transform coefficients from which the angular integrals can be derived. For the radial integrations, we use the modified Romberg quadrature method (FAI69). The range of integrations from 0 to ∞ for the radial coordinate 'r' is divided into subintervals:

$$D_{j\pm}(\eta_1,\eta_2,k,\ell,m,n) = \int_0^{d_r} dr\ r^k e^{-\eta_1 r}\ F_{j\pm}(r,\eta_2,\ell,m,n)$$

$$+ \int_{d_r}^{2d_r} dr\ r^k\ e^{-\eta_1 r} F_{j\pm}(r,\eta_2,\ell,m,n) + \ldots \qquad (4.14)$$

The integration procedure is terminated whenever the relative value of the contributions from a subinterval with respect to the total sum of the integral is less than the requred tolerance. 'ELECAP' is the program which evaluates these double integrals. For an internuclear distance R(t), it takes an average of 10 minutes CPU time to evaluate all the exchange matrix elements (K_{mn}, $\bar{K}_{mn}$, S_{mn} and $S^{\dagger}_{mn}$) by using the IBM 360/370 optimization compiler (H-compiler).

After we compute all the matrix elements, we solve the coupled equations (4.11) by dividing the trajectory into three regions. In region I and III, the internuclear distance R(t) is the independent variable. Equation (4.11.b) becomes

$$\frac{d\vec{a}}{dR} = (\dot{R}(t))^{-1}\ W\ \vec{a} \qquad (4.15.a)$$

In region II, the angle ϕ is the independent variable. We may write equation (4.11.b) as

$$\frac{d\vec{a}}{d\phi} = (\dot{\phi}(t))^{-1} W \vec{a} = \frac{R^2(t)}{bv_1} W \vec{a} \qquad (4.15.b)$$

Here we have used equation (2.3).

We used GEARS method or the variable-order ADAMS predictor-corrector method (GEA71), provided by an IMSL package, to integrate the coupled differential equations. For He^{++} projectiles scattered from hydrogen atoms, we started our time integration at R(t)=6 a.u. at a negative time (t<0) and ended at R(t)=6.0 a.u. at a positive time (t>0). The integration variables R(t) and ϕ(t) were exchanged at R(t)=1.0×10^{-2} a.u.. It took about 400 steps to integrate the coupled differential equations. Because it is very expensive to evaluate the double integrations, we evaluated only 22 data points for different R(t) values ranging from 10 a.u. to 2.074×10^{-3} (the distance of closest approach). Then we use a cubic spline fit method to interpolate the matrix elements while solving the differential equations. The direct transition matrix elements, as shown in Table A-1, have explicit series expansions. So they are explicitly evaluated during the time integrations. The whole time integration process took about 6 minutes CPU time for IBM 360/370 optimization compiler (H-compiler).

A complete program listing is in Appendix D which includes three parts: ELECAP, TRANMAX, and DCOUPL.

4.3 Computational Results and Numerical Errors

Although the main purpose of this dissertation was to establish a method for evaluating the excitation and charge transfer probabilities with as few approximations as possible, one set of data was calculated as an illustration of the method for this dissertation. From the discussion of the previous section, it is clear that the numerical computation procedure is complex, lengthy, and expensive, but from the preliminary sample calculation we may understand the problem better and find out where difficulties lie. Further work with more realistic assumptions may be performed using the lessons of this dissertation.

Most previous calculations have assumed that the ion-atom collision can be adequately described by uniform motion on a straight-line trajectory. On the other hand, as discussed in the beginning of Chapter 2, the collision process of interest in this dissertation is characterized by two conditions: (1) the scattering angle is greater than 10 degrees and (2) the slowing down factor is about 10% [slowing down factor: $\Delta v=(v_1-v)/v_1$, where v_1 is the incident projectile velocity and v is the relative velocity at the distance of closest approach). From (2.2.e), the first condition gives

$$\pi - 2 \tan^{-1}(b/a) = \theta > 10^{o} \qquad (4.16)$$

The second condition gives

$$v/v_1 = b/(a+c) < 90\% \qquad (4.17)$$

Combining (4.16) and (4.17), we conclude that if the ratio of the impact parameter and the distance of closest approach is in the range:

$$0.9 < b/a < 9$$

the two conditions will be satisfied.

For the sample calculation, the following initial conditions were chosen:

$$E = 100 \text{ Kev/amu}$$

$$b = 5a = 1.04 \times 10^{-2} \text{ a.u.}$$

$$Z_1 = 2$$

$$Z_2 = 1$$

The initial values for the amplitudes were specified as in equation (2.19). Therefore this calculation examines the collision process for He^{++} ions incident on hydrogen atoms at moderate velocities and relatively small but non-zero impact parameters.

Figures 4-1 to 4-4 represent the typical behavior of the matrix elements which always damp off at large internuclear distance R. The accuracy of the double integration was verified by checking the overlap matrix elements to determine if they satisfy the relation: $S^{*}_{nm} = S^{\dagger}_{mn}$. For internuclear distance less than 1 a.u., the matrix elements satisfy this relation accurately. But at large internuclear distance, the imaginary parts of the matrix elements tend to deviate from the expected equality. These errors are attributable to errors in the double integration which occur for large internuclear distance R when the matrix elements are numerically small. These errors may be reduced by changing the integration intervals while integrating over the electronic radial coordinate at large internuclear distance.

Figures 4-5 and 4-6 show the normalized excitation probabilities ($|b_{2S}|^2$, $|b_{2P_z}|^2$, and $|b_{2P_x}|^2$) and the normalized charge exchange probabilities ($|a_{1S}|^2$, $|a_{2S}|^2$, $|a_{2P_z}|^2$, and $|a_{2P_x}|^2$) versus the internuclear distance and time.

The internuclear distance is plotted on a logorithmic scale, so that there is a discontinuity at the distance of closest approach which corresponds to t=0. The general behavior of each probabilty is reasonable and shows that for this particular impact parameter the charge transfer to the 1S state is the largest and the most important of these probabilities.

It should be noted, however, that this result does not imply that the charge transfer to the 1S state is most probable when one sums over all impact parameters. On the qualitative argument that charge transfer is most probable to states with the least change in binding energy, one expects the total cross sections for transfer to the 2S state in He^{++} to be dominant.(SHG74)(SHG76)(RAP73) It should be interesting to make the necessary calculations for a range of impact parameters to be able to estimate total cross sections.

As the collision proceeds from $t=-\infty$, the total inelastic transition probability for removal of the electron from the 1S orbit of the target hydrogen atoms increases rapidly toward 30-40%. Eventually, at the end of the collision as $t\rightarrow +\infty$, it approaches 50%, but the calculation was not accurate enough to show satisfactory convergence.

As proved in Appendix C, in spite of the truncation of the coupled equations of motion and the use of basis states which are not linearly independent, the approximate wave functions obtained in this calculation satisfy in principle the condition of conservation of probability. However, because of numerical inaccuracy, the sample calculation gives an error as large as 60% at the end of the collision as $t= +\infty$. This error is accumulated from three sources:

(1) the error from the matrix elements at large internuclear distance R, as discussed above,
(2) the error caused by interpolating the matrix elements, and

(3) the error accumulated while integrating the coupled differential equations.

In Chapter 5, some methods will be discussed to correct or reduce these errors.

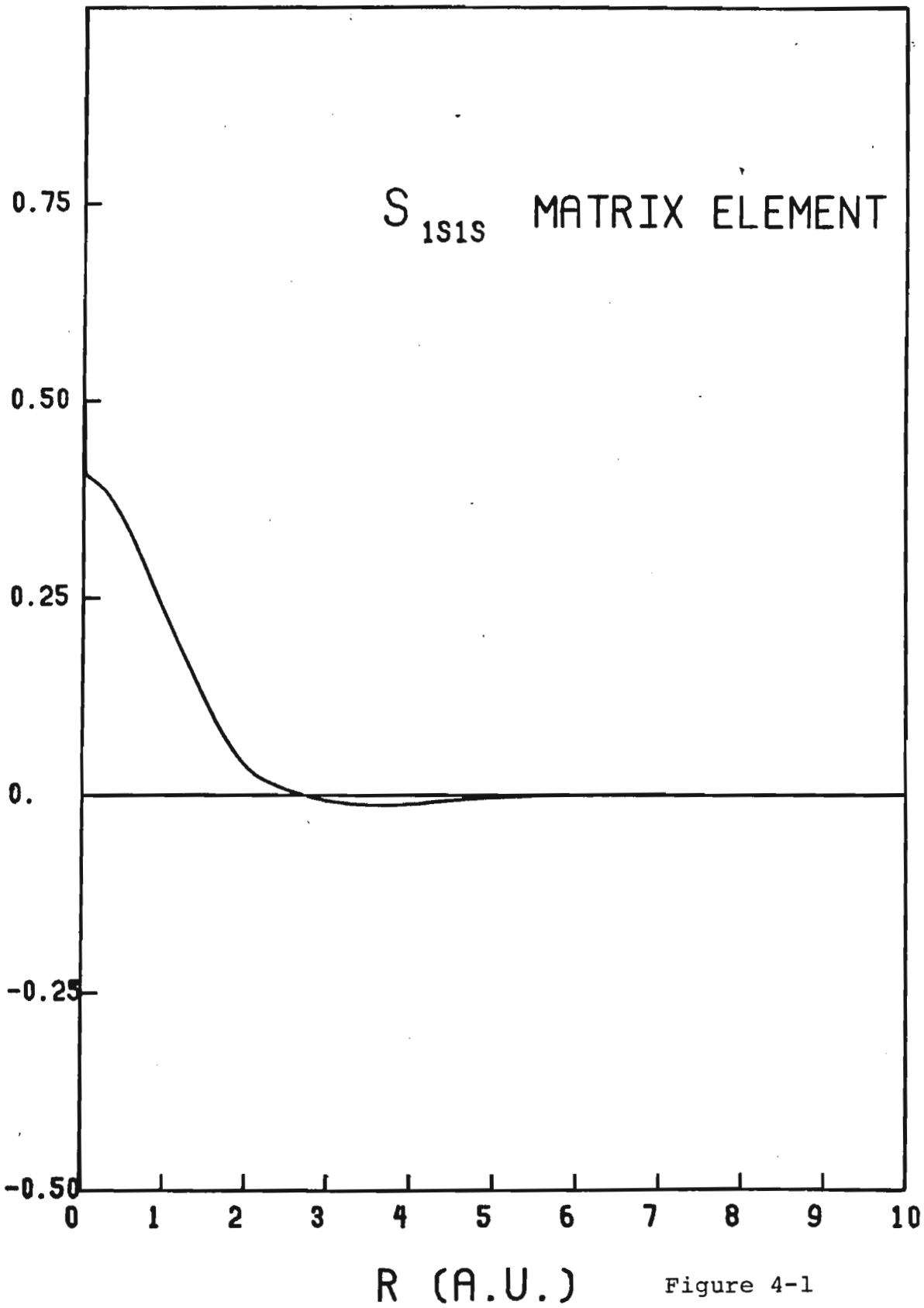


Figure 4-1

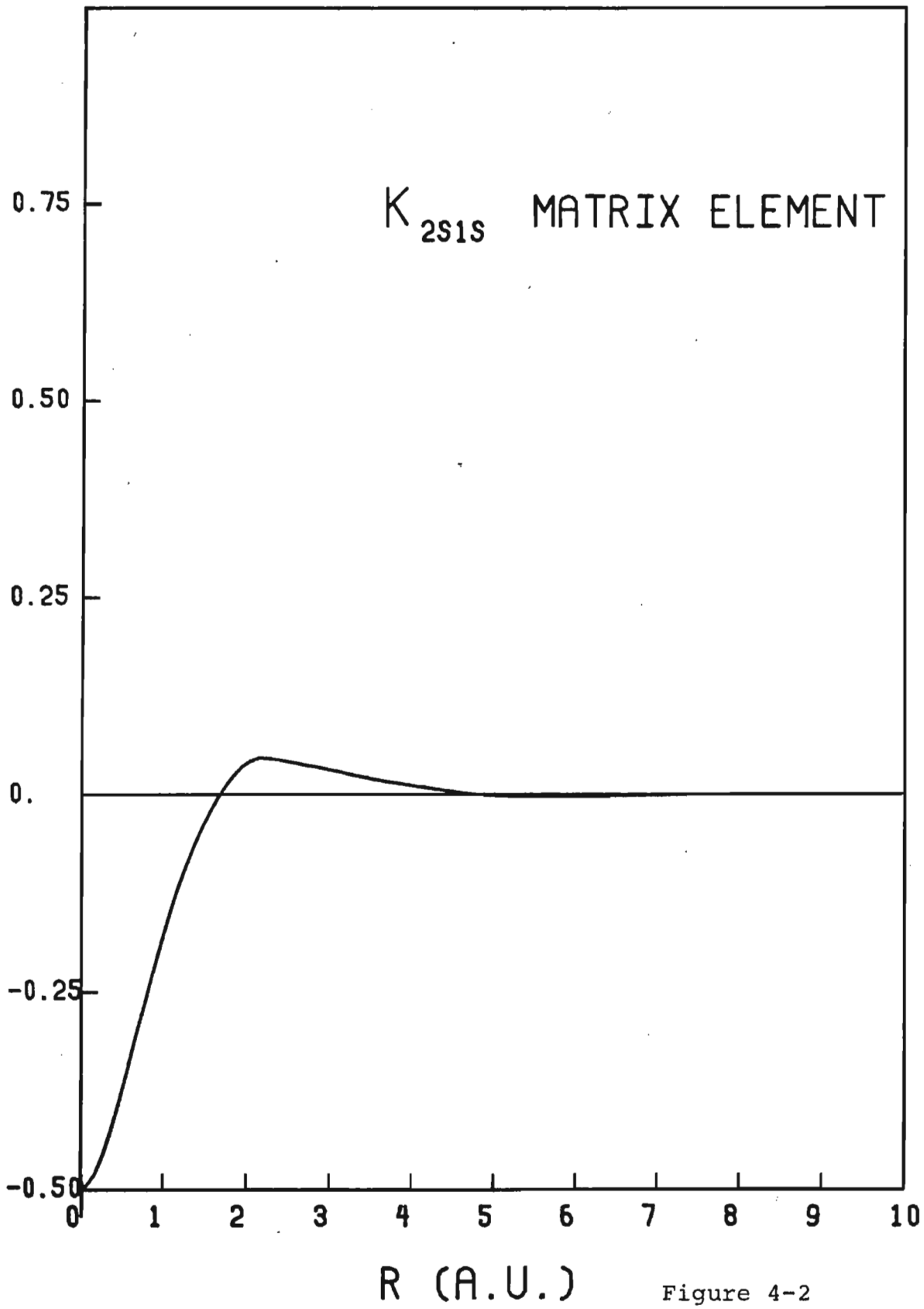


Figure 4-2

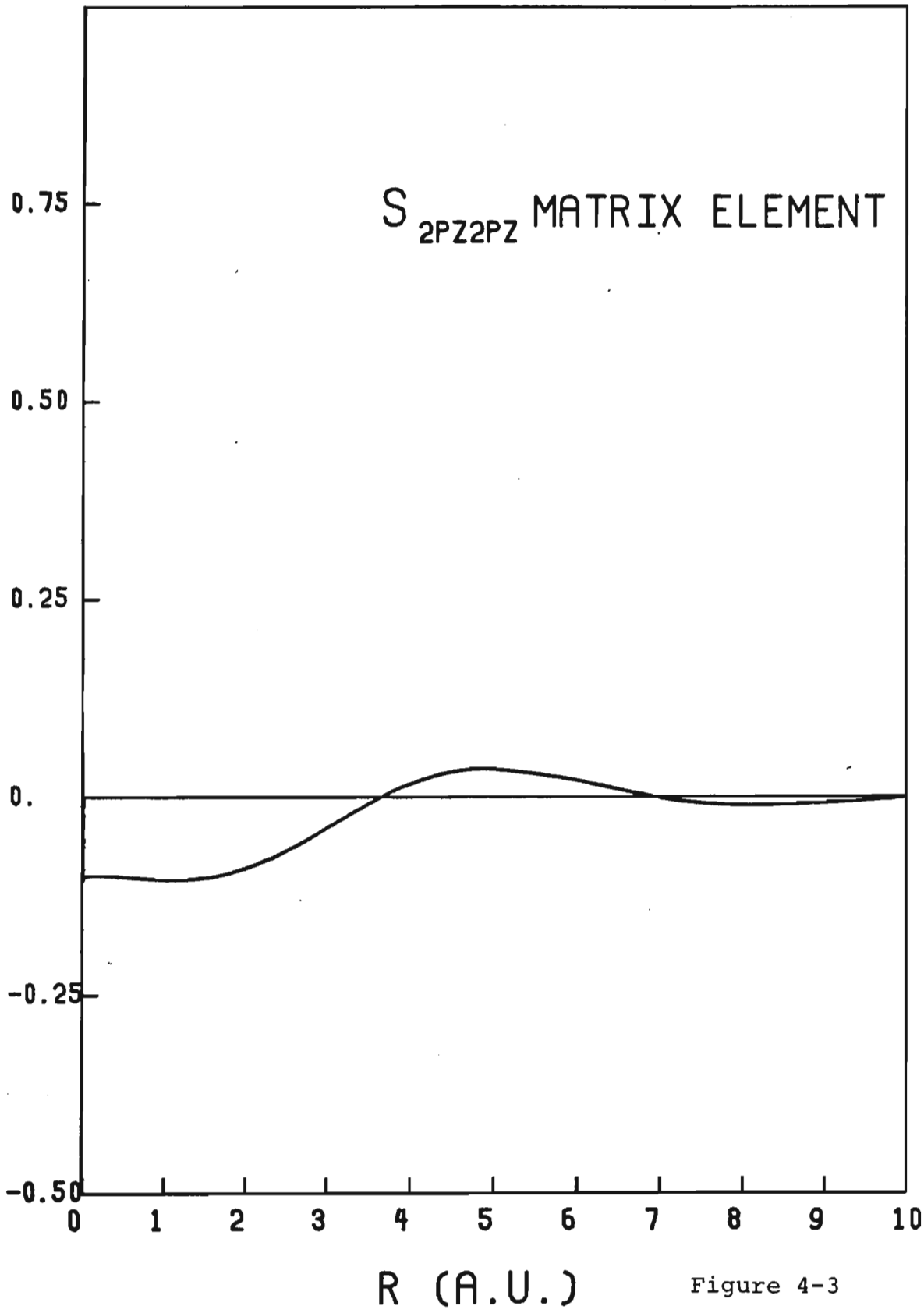

S_{2PZ2PZ} MATRIX ELEMENT
0.75
0.50
0.25
0.
-0.25
-0.50
0
1
2
3
4
5
6
7
8
9
10
R (A.U.)

Figure 4-3

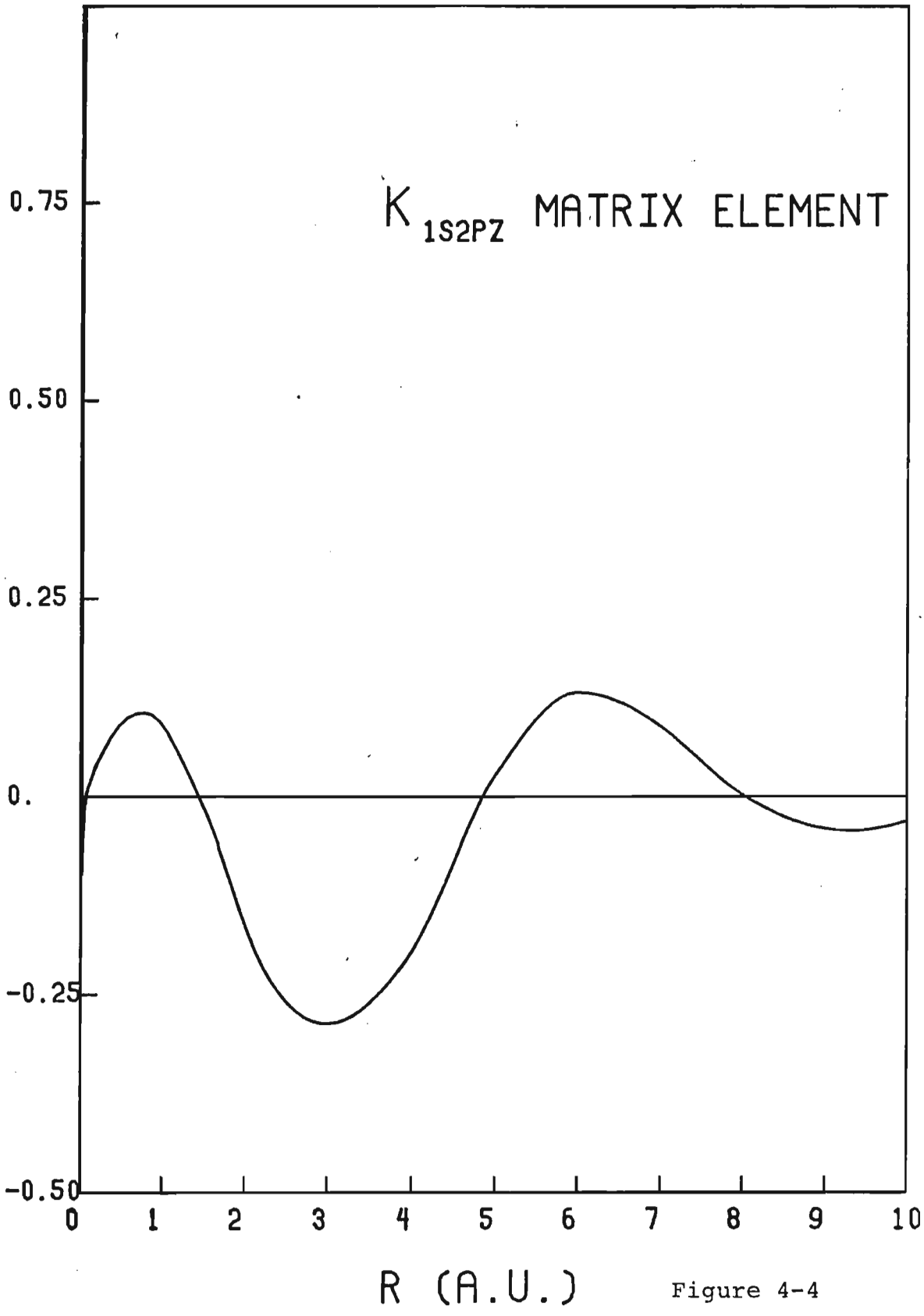
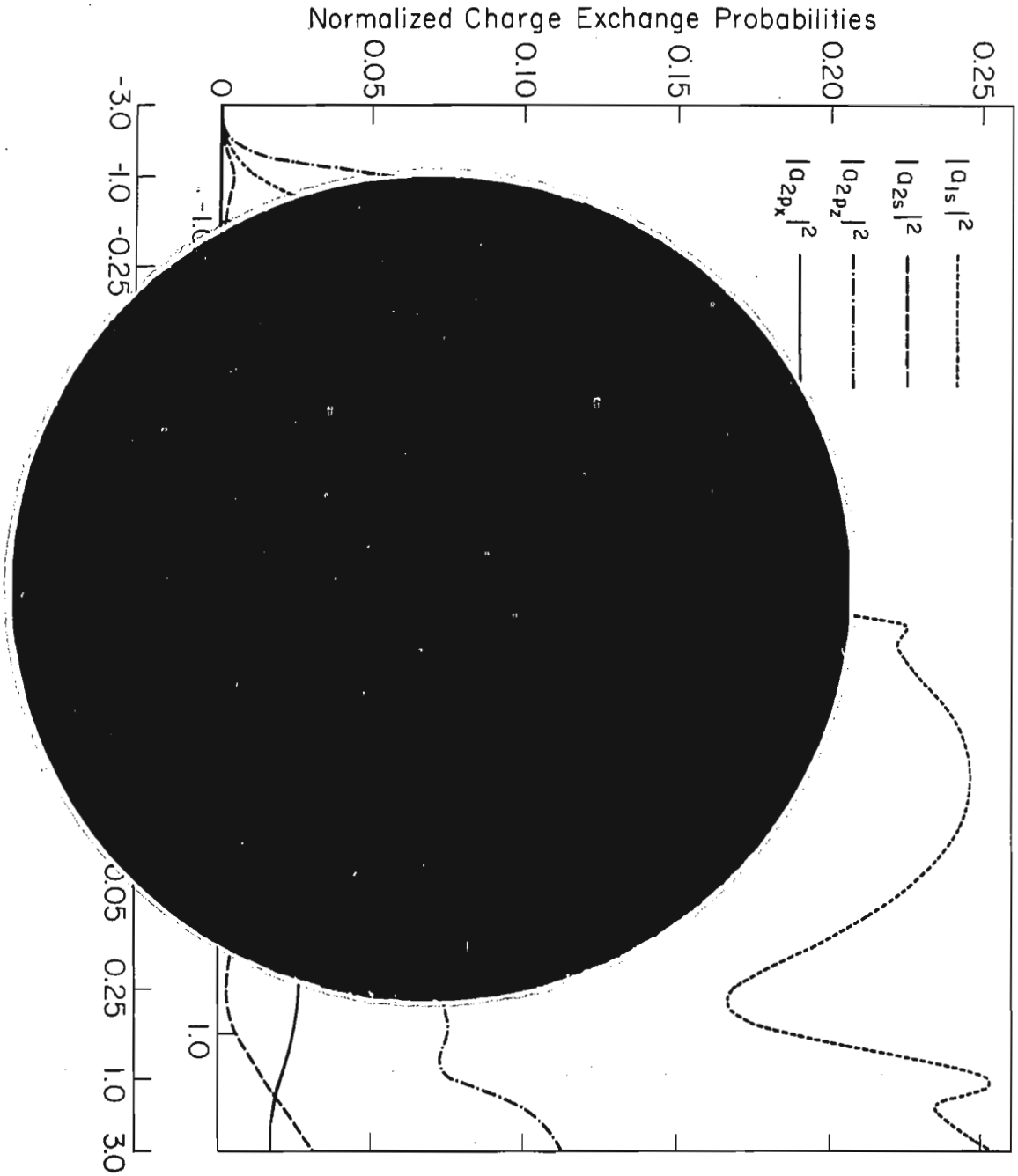

K_{1S2PZ} MATRIX ELEMENT
0.75
0.50
0.25
0.
-0.25
-0.50
0
1
2
3
4
5
6
7
8
9
10
R (A.U.)

Figure 4-4


Normalized Charge Exchange Probabilities
0.25
0.20
0.15
0.10
0.05
0
$|a_{1s}|^2$
$|a_{2s}|^2$
$|a_{2p_z}|^2$
$|a_{2p_x}|^2$
-3.0
-1.0
-0.25
0.05
0.25
1.0
3.0

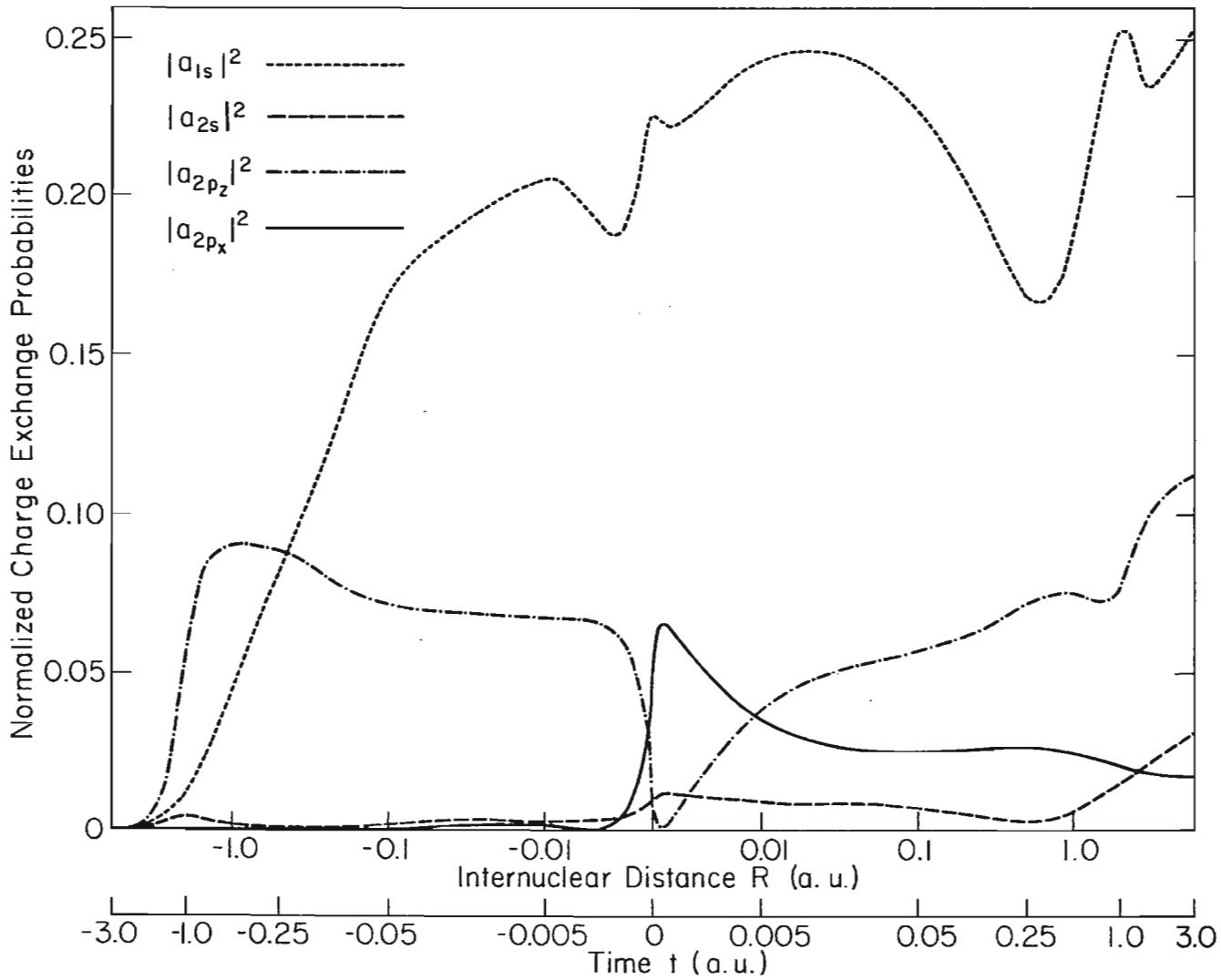

0.25
0.20
0.15
0.10
0.05
0
$|a_{1s}|^2$
$|a_{2s}|^2$
$|a_{2p_z}|^2$
$|a_{2p_x}|^2$
Normalized Charge Exchange Probabilities
-1.0
-0.1
-0.01
0.01
0.1
1.0
Internuclear Distance R (a. u.)
-3.0
-1.0
-0.25
-0.05
-0.005
0
0.005
0.05
0.25
1.0
3.0
Time t (a.u.)

Figure 4-6

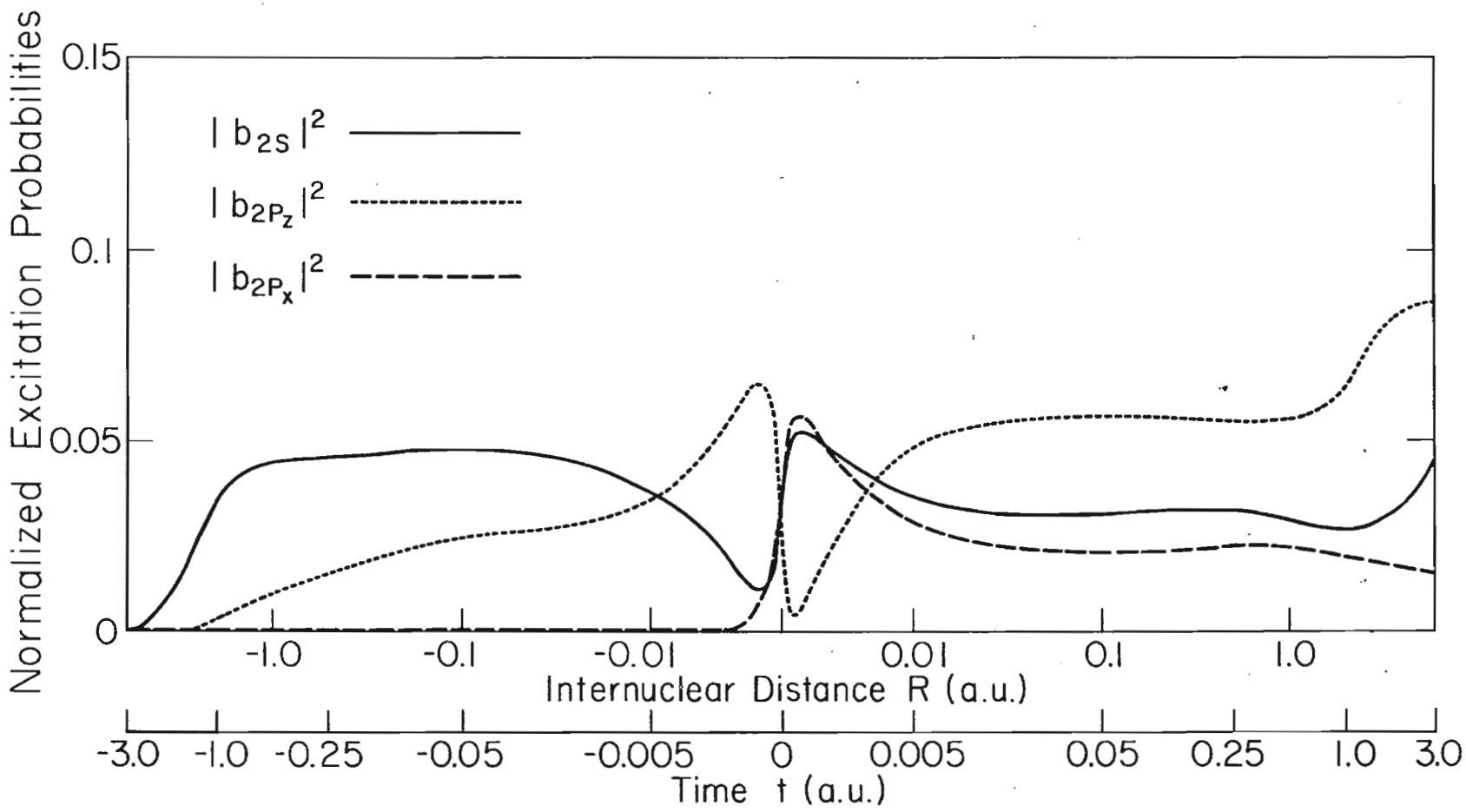

Normalized Excitation Probabilities
0.15
0.1
0.05
0
$|b_{2s}|^2$
$|b_{2P_z}|^2$
$|b_{2P_x}|^2$
-1.0
-0.1
-0.01
0.01
0.1
1.0
Internuclear Distance R (a.u.)
-3.0
-1.0
-0.25
-0.05
-0.005
0
0.005
0.05
0.25
1.0
3.0
Time t (a.u.)

Figure 4-5

CHAPTER 5

CONCLUSION

The goal of this dissertation was to establish a method to solve the problem of inelastic inner-shell ion-atom collision with only truncation errors, by using a finite basis set, and to evaluate the excitation and charge exchange probabilities. Only one sample calculation was performed to demonstrate the procedure. This preliminary calculation will give us enough information to examine the collision procedure, reveal numerical difficulties, and seek better methods to improve the results. In this concluding chapter, we briefly discuss some possible future improvements, using the general approach developed here.

As discussed in Chapter 4, the sample calculation accumulates a considerable amount of error. There are three ways to improve the results:

(1) A smaller integration interval d_r in equation (4.14) should be used for large internuclear distance R. But this change will require more steps to integrate, and will be more expensive.

(2) A different interpolation method should be used for different sets of matrix elements because the relationship between the matrix elements and the internuclear distance is different for each set of matrix elements. For the sample calculation, an interpolation routine handles four cases. A more complicated routine may be written and tested for more accurate interpolation. Of course, the modification will increase the complexity.

(3) A high-accuracy method (i.e. extrapolation method in IMSL packages) to solve the coupled differential equations

may be used to reduce the accumulated error. In the sample calculation, the GEARS method is of low-accuracy, but able to solve both nonstiff and stiff (i.e. rapidly and not-rapidly converging) coupled differential equations. Since the behavior of the coupled equations (2.17) is unknown, the GEARS method was chosen to produce a quick and preliminary evaluation for these excitation and charge exchange amplitudes. The sample calculation only took about 400 steps to integrate the coupled differential equations. It shows that this procedure is a non-stiff case. A high-accuracy, but costly, method may be used to eliminate the errors. (BUL66) (GRA65) Investigation of the above improvements will consume about one-half man-year. To solve these numerical difficulties requires an understanding of numerical analysis and different 'experiments' to examine and compare the results to produce a more satisfactory result than the sample calculation. More discussion on this topic is beyond the scope of this dissertation.

As far as the physics is concerned, the basic method developed in this dissertation is a general one. Further work may be accomplished in the follwing directions:

(1) A different basis set from the hydrogenic states may be chosen for the wave function expansion such that the matrix elements may have simple expressions. For much lower energy collisions, a molecular basis set may be used(PIA74). If a different basis set is used, the analytic expressions for all the matrix elements will change.

(2) An effective charge may be used for the hydrogenic wave functions and for the nuclear potentials for partially stripped ion-ion collisions. For such a choice, all the expressions for the matrix elements derived in this dissertation can be used, and the only change is in using effective charges for Z_1 and Z_2.

Only one value of the impact parameters was considered in this dissertation. Since most of the experimental data measure the total cross sections for the charge exchange process, probabilities for a range of impact parameters may be calculated as an extension of the present work. The total cross sections can be then derived by integrating over the impact parameters and compared with the experimental results. More experimental work in measuring impact-parameter dependent charge transfer probabilities is likely be performed in the future, affording opportunities for direct comparisons between the experimental data and the detailed calculation initiated in this dissertation.

References

AND76 J. U. Andersen, L. Kocbach, E. Laegsgaard, M. Lund, and C. D. Moak, J. Phys. B, 9, 3247, (1976)

AMU75 P. A. Amundsen and L. Kocbach, J. phys. B, 8, L122, (1975)

BAN59 J. Bang and J. M. Hansteen, Kgl. Dan. Vidensk. Selsk. Mat. Fys. Medd. 31, No. 13, (1959)

BRI78 J. S. Briggs and K. Taulbjerg, 'Topics in Current Physics: Structure and Collisions of Ions and Atoms' (I. A. Sellin ed.) Spring-Verlag Berlin Heidelberg, 1978, p. 105

BUL66 R. Bulirsch and J. Stoer, 'Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods' Numerische Mathematik, 8(1), 1-13, (1966)

CHE68 I. M. Cheshire, J. Phys. B, 1, 428, (1968)

CHE70 I. M. Cheshire, D. F. Gallaher, and A. J. Taylor, J. Phys. B, 3, 813, (1970)

CLA75 D. L. Clark, G. W. Greenlees, J. H. Broadhurst, M. E. Cage and T. K. Li, J. of Phys. B, 8, L378, (1975)

DEC77 G. Deconninck and M. Longree, Phys. Rev. A, 16, 1390, (1977)

FAI69 G. Fairweather, COMM, ACM 12 (June, 1969), 324

GAL68 D. F. Gallaher and L. Wilets, Phys. Rev. 169, 139, (1968)

GAR70 J. D. Garcia, Phys. Rev. A, 1, 1402(1970)

GAR73 J. D. Garcia, D. J. Fortner, and T. M. Kavanaugh, Rev. Mod. Phys. 45, 111 (1973)

GEA71 C. W. Gear, 'Numerical Initial Value Problems in Ordinary Differential Equations', Prentice-Hall, Englewood Cliffs, New Jersey, 1971

GEL69 S. Geltman, 'Topics in Atomic Collision Theory', Academic Press, 1969

GEN72 W. M. Gentleman, COMM. ACM 15 (May 1972), 337

GOL62 H. Goldstein, 'Classical Mechanics', Addison-Wesley, 1962

GRA65 W. B. Gragg, 'An Extraploation Algorithm for Ordinary Initial-value Problems', J. SIAM-Numerical Analysis, Series B, 2, 384-403, (1965)

GRE65 T. A. Green, Proc. Phys. Soc. 86, 1017, (1965)

KON69 E. J. Konopinski,'Classical Despcriptions of Motion', W. H. Freeman and Company, (1969)

LIN78 C. D. Lin, S. C. Soong, and L. N. Tunnell, Phys. Rev. A, 17, 1646 (1978)

LIN80 C. D. Lin and L. N. Tunnel, Phys. Rev. A, 22, 76, (1980)

McD70 M. R. C. McDowell and J. P. Coleman, 'Introduction to the Theory of Ion-Atom Collisions', North-Holland, Amsterdam (1970)

MER58 E. Merzbacher and H. W. Lewis in 'Handbuch der Physik' (S. Flügge ed.) Vol. 34, p. 166, Springer-Verlag, Berlin (1958)

MER70 E. Merzbacher, 'Quantum Mechanics', 2nd ed., Wiley, New York, (1970)

MEY79 W. E. Meyerhof and R. Anholt, J. Phys. B, 12, 3919, (1979)

MOR78 H. G. Morrison and U. Öpik, J. Phys. B, 11, 473, (1978)

MOT65 N. F. Mott and H. S. Massey, 'The Theory of Atomic Collisions' Oxford Univ. Press, London(1965)

OLS77a R. E. Olsen and A. Salop, Phys. Rev A, 16, 531,(1977)

OLS77b R. E. Olsen and A. Salop, Phys. Rev. A, 16, 1867(1977)

PIA74 R. D. Piaccentini and A. Salin, J. Phys. B 7, 1666, (1974)

RAP73 D. Rapp, J. Chem. Phys. 58, 2043 (1973)

SAL77 A. Salop and R. E. Olsen, Phys. Rev. A, 16, 1811, (1977)

SCH77 G. Bruno Schmid, Phys. Rev. A, 15, 1459(1977)
G. Bruno Schmid Am. J. Phys., 45, 652(1977)

SHA74 R. Shakeshaft and L. Spruch, Phys. Rev. A, 10, 92, (1974)

SHG74 M. B. Shah and H. B. Gilbody, J. of Phys. B, 7, 630 (1974)

SHG76 M. B. Shah and H. B. Gilbody, J. of Phys. B, 9, 1933 (1976)

SIL60 N. C. Sil, Proc. Phys. Soc. (London), 75, 194

STO73 D. Storm, Phys. Rev. A, 8, 1765,(1979)

WIL66 L. Wilets and D. F. Gallaher, Phys. Rev. 147, 13, (1966)

WU76 T. K. Wu and E. Merzbacher, Bull. Am. Phys. Soc. Ser. II, 21, 549, (1976)

WU79a T. K. Wu and E. Merzbacher, Bull. Am. Phys. Soc. 24, 101 (1979)

WU79b T. K. Wu and E. Merzbacher, Bull. Am. Phys. Soc. 24, 581 (1979)

APPENDIX A

The Direct Coupling Matrix Elements

We will drive the direct coupling matrix elements $H_{mn}(A)$ and $\bar{H}_{mn}(B)$ here. We rewrite equation (4.1) in an integral form so that we may evaluate the integrals:

$$H_{mn}(A) = \int d^3\tau_A \Phi_m^{A*}(A,\vec{r}_A)\left(\frac{Z_2}{r_B} + \ddot{\vec{R}}^A(t)\cdot\vec{r}_A + \frac{1}{2}\ddot{\vec{R}}^A(t)\cdot\vec{R}^A(t) - \omega(t)L_{y_A}\right)\Phi_n^A(A,\vec{r}_A)e^{i(E_m^A - E_n^A)t} \qquad (A\text{-}1)$$

where we have used the coordinate representation for the state vectors and expanded in the projectile frame of reference:

$$\langle\vec{r}_A|\Phi_n^A(A,t)\rangle = \Phi_n^A(A,\vec{r}_A)e^{-iE_n^A t}$$

where $\vec{r}_A$ is the position vector of the active electron from the projectile A. E_n^A are the energy levels of the projectile atom. The argument 'A' of $\Phi_n^A(A,\vec{r}_A)$ denotes that the wave function is centered at the projectile. The superscript 'A' denotes the projectile frame of reference. $\Phi_m^A(A,\vec{r}_A)$ and $\Phi_m^B(B,\vec{r}_B)$ represent the projectile and target hydrogenic atomic wave functions respectively.

Applying equations (2.10) and (2.3), we rearrange the integral (A.1)

$$H_{mn}(A) = \left(-Z_2 DP^A(m,n) - \frac{Z_1 Z_2}{M_A R^2}DD^A(m,n) + \frac{M_B Z_1 Z_2}{M_A M R}\delta_{mn} - \frac{v_1 b}{R^2}DR^A(m,n)\right) \times e^{i(E_m^A - E_n^A)t} \qquad (A\text{-}2)$$

where we introduce three new notations which are defined as follows:

$$DP^A(m,n) = \int d^3\tau_A \ \phi_m^{A*}(A,\vec{r}_A)\frac{1}{r_B}\ \phi_n^A(A,r_A) \qquad (A\text{-}3)$$

$$DD^A(m,n) = \int d^3\tau_A \ \phi_m^{A*}(A,\vec{r}_A) r_A \cos\theta_A \ \phi_n^A(A,\vec{r}_A) \qquad (A\text{-}4)$$

$$DR^A(m,n) = \int d^3\tau_A \ \phi_m^{A*}(A,\vec{r}_A) L_{\hat{y}_A} \phi_n^A(A,\vec{r}_A) \qquad (A\text{-}5)$$

and δ_{mn}

$$\delta_{mn} = \begin{cases} 0 & \text{if } m \neq n \\ 1 & \text{if } m = n \end{cases}$$

(A-3), (A-4) and (A-5) are the so-called "potential"(or "radial"), "dipole", and "rotational" coupling terms.

Similar equations may be derived for $\bar{H}_{mn}(B)$

$$\bar{H}_{mn}(B) = \left(-Z_1 DP^B(m,n) - \frac{Z_1 Z_2}{M_B R^2} DD^B(m,n) + \frac{M_A Z_1 Z_2}{2M_B MR}\delta_{mn} - \frac{v_1 b}{R^2} DR^B(m,n)\right) \times e^{i(E_m^B - E_n^B)t} \qquad (A\text{-}6)$$

where DP^B, DD^B, and DR^B are defined as

$$DP^B(m,n) = \int d^3\tau_B \phi_m^{B*}(B,\vec{r}_B)\frac{1}{r_A}\phi_n^B(B,\vec{r}_B) \qquad (A\text{-}7)$$

$$DD^B(m,n) = \int d^3\tau_B \phi_m^{B*}(B,\vec{r}_B) r_B \cdot \cos\theta_B \phi_n^B(B,\vec{r}_B) \qquad (A\text{-}8)$$

$$DR^B(m,n) = \int d^3\tau_B \phi_m^{B*}(B,\vec{r}_B) L_{\hat{y}_B} \phi_n^B(B,\vec{r}_B) \qquad (A\text{-}9)$$

These equations are the same format as (A-3), (A-4), and (A-5), except that the subscripts and superscripts A, B are interchanged. Table A-1 gives the formula of the relevant integrals.

When we evaluate the integrals, we first use $Z=Z_1$, $M=M_A$, and $\vec{r}=(r_A,\theta_A,\phi_A)$ for the basis set of the projectile atom. Then we substitute $Z=Z_2$, $M=M_B$, and $\vec{r}=(r_B,\theta_B,\phi_B)$. Because of the relative orientation of the two frames of reference, there are some integral values which need a change of sign. We will discuss the consequence of these sign adjustments in following sections:

(a) Potential coupling integrals:

Equation (A-3) may be written in the explicit integral forms:

$$DP^A(m,n)=\int dr_A r_A^2 R_{n_1\ell_1}(r_A)R_{n_2\ell_2}(r_A)\int d(\cos\theta_A)P_{\ell_1}^{|m_1|}(\cos\theta_A)P_{\ell_2}^{|m_2|}(\cos\theta_A)$$

$$\frac{1}{(r_A^2+R^2(t)-2r_AR(t)\cos\theta_A)^{\frac{1}{2}}}\int d\phi_A\exp(i(m_2-m_1)\phi_A) \qquad \text{(A-10)}$$

and (A-7) is

$$DP^B(m,n)=\int dr_B r_B^2 R_{n_1\ell_1}(r_B)R_{n_2\ell_2}(r_B)\int d(\cos\theta_B)P_{\ell_1}^{|m_1|}(\cos\theta_B)P_{\ell_2}^{|m_2|}(\cos\theta_B)$$

$$\frac{1}{(r_B^2+R^2(t)+2r_BR(t)\cos\theta_B)^{\frac{1}{2}}}\int d\phi_B\exp(i(m_2-m_1)\phi_B) \qquad \text{(A-11)}$$

where we have used the trigometric cosine law for r_A, r_B, and $R(t)$. If $DP^B(m,n)$ is evaluated by substituting Z_2 and M_B for Z_1 and M_A into equation (A-10), there is a sign

factor $(-1)^{\ell_1+|m_1|+\ell_2+|m_2|}$ in the front because the associate Lengendre polynomial P_ℓ^m has following property:

$$P_\ell^m(-x) = (-1)^{\ell+m} P_\ell^m(x) \qquad \text{(A-12)}$$

Table A-2 gives a summary of sign changes for interchanging A and B.

(m,n)	n_1,ℓ_1,m_1	n_2,ℓ_2,m_2	change (Y:yes) sign (N:no)
(1S,1S)	1, 0, 0	1, 0, 0	N
(1S,2S)	1, 0, 0	2, 0, 0	N
(1S,$2P_z$)	1, 0, 0	2, 1, 0	Y
(1S,$2P_x$)	1, 0, 0	2, 1, ±1	N
(2S,2S)	2, 0, 0	2, 0, 0	N
(2S,$2P_z$)	2, 0, 0	2, 1, 0	Y
(2S,$2P_x$)	2, 0, 0	2, 1, ±1	N
($2P_z$,$2P_z$)	2, 1, 0	2, 1, 0	N
($2P_z$,$2P_x$)	2, 1, 0	2, 1, ±1	Y
($2P_x$,$2P_x$)	2, 1, ±1	2, 1, ±1	N

Table A-2 $\quad DP^A(m,n) \longrightarrow DP^B(m,n)$

(b) Dipole coupling integrals:

Equations (A-4) and (A-8) are

$$DD^A(m,n) = \int dr_A r_A^3 R_{n_1\ell_1}(r_A) R_{n_2\ell_2}(r_A) \int d(\cos\theta_A) \cos\theta_A P_{\ell_1}^{|m_1|}(\cos\theta_A)$$

$$P_{\ell_2}^{|m_2|}(\cos\theta_A) \int d\phi_A \exp(i(m_2-m_1)\phi_A) \qquad \text{(A-13)}$$

and

$$DD^B(m,n)= \int dr_B r_B^3 R_{n_1\ell_1}(r_B) R_{n_2\ell_2}(r_B) \int d(\cos\theta_B) \cos\theta_B P_{\ell_1}^{|m_1|}(\cos\theta_B)$$

$$P_{\ell_2}^{|m_2|}(\cos\theta_B) \int d\phi_B \exp(i(m_2-m_1)\phi_B) \qquad \text{(A-14)}$$

Comparing (A-13) and (A-14), there is no sign change for evaluating $DD^B(m,n)$ by substituting Z_2 and M_B into equation (A-13).

(c) Rotational coupling integrals:

We write (A-5) and (A-9) as

$$DR^A(m,n)=\int dr_A d\theta_A d\phi_A \; r_A^2 \sin\theta_A \Phi_n^A(A,\vec{r}_A) \frac{L_+ - L_-}{2i} \Phi_n^A(A,\vec{r}_A) \qquad \text{(A-15)}$$

and

$$DR^B(m,n)=\int dr_B d\theta_B d\phi_B r_B^2 \sin\theta_B \Phi_n^B(B,\vec{r}_B) \frac{L_+ - L_-}{2i} \Phi_n^B(B,\vec{r}_B) \qquad \text{(A-16)}$$

If we substitute Z_2 and M_B for Z_1 and M_A in equation (A-15), we get $DR^B(m,n)$ with no sign change.

Table A-1

$$G(n,\alpha) = \int_0^\infty dr \, \frac{r^n}{R(t)} \, (r+R(t)-|r-R(t)|) \, e^{-\alpha r}$$

$$= \frac{2(n!)}{\alpha^{n+1}}\left[\frac{n+1}{\alpha R(t)} - e^{-\alpha R(t)}\left(\frac{n+1}{\alpha R(t)} + \sum_{m=0}^{n-1} \frac{(n-m)}{m!(m+1)}(\alpha R(t))^m\right)\right]$$

$$B(n,\alpha) = \int_0^\infty dr \, \frac{r^n}{R(t)} \, (r+R(t)+|r-R(t)|)e^{-\alpha r}$$

$$= \frac{2(n!)}{\alpha^{n+1}}\left[1 + \frac{e^{-\alpha R(t)}}{\alpha R(t)} \sum_{m=0}^{n} \frac{(n-m+1)}{m!}(\alpha R(t))^m\right]$$

$$DP^A(1S,1S) = 2Z_1^3 \, G(1,2Z_1)$$

$$DP^A(1S,2S) = \frac{Z_1^3}{2\sqrt{2}}[2G(1,3Z_1/2)-Z_1G(2,3Z_1/2)]$$

$$DP^A(1S,2P_z) = \frac{Z_1^4}{6\sqrt{2}}\left[\frac{1}{R(t)}G\left(3,\frac{3Z_1}{2}\right)+R(t)G\left(1,\frac{3Z_1}{2}\right)-B\left(2,\frac{3Z_1}{2}\right)\right]$$

$$DP^A(2S,2P_z) = \frac{Z_1^4}{48}\left[-\frac{Z_1}{R(t)}G(4,Z_1)+\frac{2}{R(t)}G(3,Z_1)-Z_1R(t)G(2,Z_1)\right.$$

$$\left.+2R(t)G(1,Z_1)-2B(2,Z_1)+Z_1B(3,Z_1)\right]$$

$$DP(2S,2S) = \frac{Z_1^3}{16}[4G(1,Z_1)-4Z_1G(2,Z_1)+Z_1^2G(3,Z_1)]$$

Table A-1 (con't)

$$DP^A(2P_z,2P_z) = \frac{Z_1^5}{240}\left[\frac{2}{R^2(t)}G(5,Z_1)+2R^2(t)G(1,Z_1)+7G(3,Z_1)\right.$$

$$\left. - \frac{2}{R(t)}B(4,Z_1) - 2R(t)B(2,Z_1)\right]$$

$$DP^A(2P_x,2P_x) = \frac{Z_1^5}{240}\left[4G(3,Z_1)- \frac{1}{R^2(t)}\, G(5,Z_1) - R^2(t)G(1,Z_1)\right.$$

$$\left. + \frac{1}{R(t)}B(4,Z_1) + R(t)B(2,Z_1)\right]$$

$DP^A(2S,1S)=DP^A(1S,2S)$ $\qquad$ $DP^A(2P_z,2S)=DP^A(2S,2P_z)$

$DP^A(2P_z,1S)=DP^A(1S,2P_z)$

All the other direct potential coupling terms are zero.

$$DD^A(1S,2P_z) = DD^A(2P_z,1S) = \frac{128\sqrt{2}}{243Z_1}$$

$$DD^A(2S,2P_z) = DD^A(2P_z,2S) = -\frac{3}{Z_1}$$

All the other direct dipole coupling terms are zero.

$DR^A(2P_z,2P_x) = -i$ $\qquad$ $DR^A(2P_x,2P_z) = i$

All the other direct radial coupling terms are zero.

APPENDIX B

The Exchange Coupling Matrix Elements

The evaluations of the exchange coupling matrix elements are more complicated than the ones of the direct coupling matrix elements discussed in Appendix A. Before we starts to investigate the $K_{mn}(A,B)$ and $\bar{K}_{mn}(B,A)$ matrices, we must prepare ourselves by fixing the notation. If we use target frame of reference, the target coordinate representations of the state vectors are:

$$\begin{aligned} \langle \vec{r}_B | \phi_m^B(A,t) \rangle &= \phi_m^B(A,\vec{r}_B) \exp(-iE_m^A t) \\ \langle \vec{r}_B | \phi_n^B(B,t) \rangle &= \phi_n^B(B,\vec{r}_B) \exp(-iE_n^B t) \end{aligned} \tag{B-1}$$

where $\vec{r}_B$ is the position vector of the active electron from the target B. The subscript and superscript definitions please refer to Appendix A. Here $\phi_n^B(B,\vec{r}_B)$ does represent the hydrogenic basis of the target atom which have the wave functions expressed in (4.9). $\phi_m^B(A,\vec{r}_B)$ needs phase factors to transform to the hydrogenic basis of the projectile atom. Using the theory developed in Chapter 3 and equation (3.46), we relate the two wave functions as

$$\phi_m^B(A,\vec{r}_B) = \exp(i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2) \exp(i\dot{\vec{R}}(t)\cdot\vec{r}_A)\phi_m^A(A,\vec{r}_A)$$

or (B-2)

$$\phi_m^A(A,\vec{r}_A) = \exp(i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2) \exp(-i\dot{\vec{R}}(t)\cdot\vec{r}_B)\phi_m^B(A,\vec{r}_B)$$

since $\vec{r}_B=\vec{r}_A+\vec{R}(t)$. Now we are ready to examine the exchange coupling matrix.

Applying equations (4.4), (2.10), and (2.3) to (4.5), we get

$$K_{mn}(A,B) = (-Z_1 IP(A,B)_{mn} - \frac{Z_1 Z_2}{M_B R^2(t)} ID(A,B)_{mn} + \frac{M_A Z_1 Z_2}{2M_B MR(t)} S_{mn} - \frac{v_1 b}{R^2(t)} IR(A,B)_{mn}) \exp(i(E_m^A - E_n^B)t) \tag{B-3}$$

where we again introduce four functions which are defined as following integrals:

$$IP(A,B)_{mn} = \langle \Phi_m^B(A,t) | \frac{1}{r_A} | \Phi_n^B(B,t) \rangle \tag{B-4}$$

$$= \int d^3\tau_A \Phi_m^{A*}(A,\vec{r}_A) e^{-i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2} \, e^{-i\dot{\vec{R}}(t)\cdot\vec{r}_A} \frac{1}{r_A} \Phi_n^B(B,\vec{r}_B)$$

$$ID(A,B)_{mn} = \langle \Phi_m^B(A,t) | r_B \cdot \cos\theta_B | \Phi_n^B(B,t) \rangle \tag{B-5}$$

$$= \int d^3\tau_B \Phi_m^{A*}(A,\vec{r}_A) e^{i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2} \, e^{-i\dot{\vec{R}}(t)\cdot\vec{r}_B} r_B \cos\theta_B \Phi_n^B(B,\vec{r}_B)$$

$$S_{mn} = \int d^3\tau_B \Phi_m^{A*}(A,\vec{r}_A) e^{i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2} \, e^{-i\dot{\vec{R}}(t)\cdot\vec{r}_B} \, \Phi_n^B(B,\vec{r}_B) \tag{B-6}$$

$$IR(A,B)_{mn} = \langle \Phi_m^B(A,t) | \hat{L_{y_B}} | \Phi_n^B(B,t) \rangle \tag{B-7}$$

$$= \int d^3\tau_B \Phi_m^{A*}(A,\vec{r}_A) e^{i\dot{\vec{R}}(t)\vec{R}(t)/2} \, e^{-i\dot{\vec{R}}(t)\vec{r}_B} \frac{L_+ - L_-}{2i} \Phi_n^B(B,\vec{r}_B)$$

(B-4), (B-5), and (B-7) are the "potential" (or "radial"), "dipole", and "rotational" coupling terms for the charge

exchange. (B-6) is the overlap integral.

Similar equations may be derived for $\bar{K}_{mn}(B,A)$:

$$\bar{K}_{mn}(B,A) = (-Z_2 IP(B,A)_{mn} - \frac{Z_1 Z_2}{M_A R^2(t)} ID(B,A)_{mn} + \frac{M_B Z_1 Z_2}{2M_A MR(t)} S^{\dagger}_{mn} - \frac{v_1 b}{R^2(t)} IR(B,A)_{mn}) \exp(i(E^B_m - E^A_n)t) \tag{B-8}$$

where we introduce four functions which are defined as following integrals:

$$IP(B,A)_{mn} = \langle \Phi^A_m(B,t) | \frac{1}{r_B} | \Phi^A_n(A,t) \rangle \tag{B-9}$$

$$= \int d^3\tau_B \Phi^{B*}_m(B,\vec{r}_B) e^{-i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2} e^{i\dot{\vec{R}}(t)\cdot\vec{r}_B} \frac{1}{r_B} \Phi^A_n(A,\vec{r}_A)$$

$$ID(B,A)_{mn} = \langle \Phi^A_m(B,t) | r_A \cdot \cos\theta_A | \Phi^A_n(A,t) \rangle \tag{B-10}$$

$$= \int d^3\tau_A \Phi^{B*}_m(B,\vec{r}_B) e^{i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2} e^{i\dot{\vec{R}}(t)\cdot\vec{r}_A} r_A \cos\theta_A \Phi^A_n(A,\vec{r}_A)$$

$$S^{\dagger}_{mn} = \int d^3\tau_A \Phi^{B*}_m(B,\vec{r}_B) e^{i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2} e^{i\dot{\vec{R}}(t)\cdot\vec{r}_A} \Phi^A_n(A,\vec{r}_A) \tag{B-11}$$

$$IR(B,A)_{mn} = \langle \Phi^A_m(B,t) | L_{\hat{y}_A} | \Phi^A_n(A,t) \rangle \tag{B-12}$$

$$= \int d^3\tau_A \Phi^{B*}_m(B,\vec{r}_B) e^{i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2} e^{i\dot{\vec{R}}(t)\cdot\vec{r}_A} \frac{L_+ - L_-}{2i} \Phi^A_n(A,\vec{r}_A)$$

These equations have the same format as (B-4) to (B-7), except that the subscripts and superscripts A, B are interchanged. Table B-1 gives the formula of the relevant integrals.

We evaluate these matrix elements, $IP(A,B)_{mn}$, $ID(A,B)_{mn}$, $S_{mn}(A)$, and $IR(A,B)_{mn}$, by first substituting Z_1, Z_2 and M_A, M_B into the expressions in Table B-1. Then we substitute exchanged Z_1 and Z_2, M_A and M_B values into the expressions in Table B-1 and change the integral variable $\cos\theta$ into $-\cos\theta$ to evaluate the $IP(B,A)_{mn}$, $ID(B,A)_{mn}$, $S^{\dagger}_{mn}(B)$, and $IR(B,A)_{mn}$ matrix elements. There is a sign adjustment from this operation. For instance, the potential coupling integrals (B-4) and (B-9) may be written

$$IP(A,B)mn = 2\pi\, e^{-i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2}\int dr_A r_A R_{n_1\ell_1}(r_A)\int d(\cos\theta_A) P_{\ell_1}^{|m_1|}(\cos\theta_A)$$
$$e^{i\dot{R}(t)r_A\cos\theta_A}(i)^{m_2-m_1} J_{m_2-m_1}(bv_1 r_A\sin\theta_A/R(t))$$
$$P_{\ell_2}^{|m_2|}(\cos\theta_B)R_{n_2\ell_2}((r_A^2+R^2(t)-2r_A R(t)\cos\theta_A)^{\frac{1}{2}}) \quad (B\text{-}13)$$

and

$$IP(B,A)mn = 2\pi\, e^{-i\dot{\vec{R}}(t)\cdot\vec{R}(t)/2}\int dr_B r_B R_{n_1\ell_1}(r_B)\int d(\cos\theta_B) P_{\ell_1}^{|m_1|}(\cos\theta_B)$$
$$e^{-i\dot{R}(t)r_B\cos\theta_B}(i)^{m_2-m_1}J_{m_2-m_1}(-v_1 b r_B\sin\theta_B/R(t))$$
$$P_{\ell_2}^{|m_2|}(\cos\theta_A)R_{n_2\ell_2}((r_B^2+R^2(t)+2r_B R(t)\cos\theta_B)^{\frac{1}{2}}) \quad (B\text{-}14)$$

If we interchange A and B and transform

$$r_A \longrightarrow r_B$$
$$\cos\theta_A \longrightarrow -\cos\theta_B$$
$$\sin\theta_A \longrightarrow \sin\theta_B$$
$$\phi_A \longrightarrow \phi_B,$$

and from the Bessel function's property

$$J_n(-x) = (-1)^n J_n(x)$$

(B-14) will have the same expression as (B-13) except a sign factor $(-1)^{m_1 - m_2}$ in the front. Therefore, whenever an odd order of Bessel function appears in the integrand, a sign should be adjusted.

Table B-1

$$F_{0\pm}(r,\eta_2,\ell,m,n) = \int dx\ e^{-\eta_2 y_\pm(r,x)}\ e^{ir\dot{R}(t)x}\ J_0(r\lambda(1-x^2)^{\frac{1}{2}})\ (y_\pm(r,x))^{\ell}\ x^m\ (1-x^2)^{n/2}$$

$$F_{1\pm}(r,\eta_2,\ell,m,n) = \int dx\ e^{-\eta_2 y_\pm(r,x)}\ e^{ir\dot{R}(t)x}\ J_1(r\lambda(1-x^2)^{\frac{1}{2}})\ (y_\pm(r,x))^{\ell}\ x^m\ (1-x^2)^{n/2}$$

$$F_{2\pm}(r,\eta_2,\ell,m,n) = \int dx\ e^{-\eta_2 y_\pm(r,x)}\ e^{ir\dot{R}(t)x}\ J_2(r\lambda(1-x^2)^{\frac{1}{2}})\ (y_\pm(r,x))^{\ell}\ x^m\ (1-x^2)^{n/2}$$

$$D_{0\pm}(\eta_1,\eta_2,k,\ell,m,n) = \int dr\ r^k\ e^{-\eta_1 r}\ F_{0\pm}(r,\eta_2,\ell,m,n)$$

$$D_{1\pm}(\eta_1,\eta_2,k,\ell,m,n) = \int dr\ r^k\ e^{-\eta_1 r}\ F_{1\pm}(r,\eta_2,\ell,m,n)$$

$$D_{2\pm}(\eta_1,\eta_2,k,\ell,m,n) = \int dr\ r^k\ e^{-\eta_1 r}\ F_{2\pm}(r,\eta_2,\ell,m,n)$$

where $\lambda = v_1 b/R(t)$

$$y_\pm(r,x) = (r^2 + x^2 \pm 2rR(t)x)^{\frac{1}{2}}$$

and all above equations are functions of R(t), the internuclear distance.

Table B-1 (con't)

$$IP(A,B)_{1S1S} = 2(Z_1Z_2)^{3/2}\, D_{O-}(Z_1,Z_2,1,0,0,0)$$

$$IP(A,B)_{1S2S} = \frac{(Z_1Z_2)^{3/2}}{2\sqrt{2}}[2D_{O-}(Z_1,Z_2/2,1,0,0,0)-Z_2D_{O-}(Z_1,Z_2/2,1,1,0,0)]$$

$$IP(A,B)_{1S2P_z} = \frac{(Z_1Z_2)^{3/2}Z_2}{2\sqrt{2}}[D_{O-}(Z_1,Z_2/2,2,0,1,0)-R(t)D_{O-}(Z_1,Z_2/2,1,0,0,0)]$$

$$IP(A,B)_{1S2P_x} = \frac{(Z_1Z_2)^{3/2}Z_2}{2\sqrt{2}}\, i\, [D_{1-}(Z_1,Z_2/2,2,0,0,1)]$$

$$IP(A,B)_{2S1S} = \frac{(Z_1Z_2)^{3/2}}{2\sqrt{2}}[2D_{O-}(Z_1/2,Z_2,1,0,0,0)-Z_1D_{O-}(Z_1/2,Z_2,2,0,0,0)]$$

$$IP(A,B)_{2S2S} = \frac{(Z_1Z_2)^{3/2}}{16}[4D_{O-}(Z_1/2,Z_2/2,1,0,0,0)-2Z_1D_{O-}(Z_1/2,Z_2/2,2,0,0,0)$$
$$-2Z_2D_{O-}(Z_1/2,Z_2/2,1,1,0,0)+Z_1Z_2D_{O-}(Z_1/2,Z_2/2,2,1,0,0)]$$

$$IP(A,B)_{2S2P_z} = \frac{(Z_1Z_2)^{3/2}Z_2}{16}[2D_{0-}(Z_1/2,Z_2/2,2,0,1,0)-Z_1D_{O-}(Z_1/2,Z_2/2,3,0,1,0)$$
$$-2R(t)D_{O-}(Z_1/2,Z_2/2,1,0,0,0)+Z_1R(t)D_{O-}(Z_1/2,Z_2/2,2,0,0,0)]$$

$$IP(A,B)_{2S2P_x} = \frac{Z_2(Z_1Z_2)^{3/2}}{16}\, i[2D_{1-}(Z_1/2,Z_2/2,2,0,0,1)-Z_1D_{1-}(Z_1/2,Z_2/2,3,0,0,1)]$$

Table B-1 (con't)

$$IP(A,B)_{2P_z 1S} = \frac{(Z_1 Z_2)^{3/2} Z_1}{2\sqrt{2}} [D_{O-}(Z_1/2, Z_2, 2, 0, 1, 0)]$$

$$IP(A,B)_{2P_z 2S} = \frac{(Z_1 Z_2)^{3/2} Z_1}{16} [2D_{O-}(Z_1/2, Z_2/2, 2, 0, 1, 0) - Z_2 D_{O-}(Z_1/2, Z_2/2, 2, 1, 1, 0)]$$

$$IP(A,B)_{2P_z 2P_z} = \frac{(Z_1 Z_2)^{5/2}}{16} [D_{O-}(Z_1/2, Z_2/2, 3, 0, 2, 0) - R(t) D_{O-}(Z_1/2, Z_2/2, 2, 0, 1, 0)]$$

$$IP(A,B)_{2P_z 2P_x} = \frac{(Z_1 Z_2)^{5/2}}{16} \; i \; [D_{1-}(Z_1/2, Z_2/2, 3, 0, 1, 1)]$$

$$IP(A,B)_{2P_x 1S} = \frac{(Z_1 Z_2)^{3/2} Z_1}{2\sqrt{2}} \; i \; D_{1-}(Z_1/2, Z_2, 2, 0, 0, 1)$$

$$IP(A,B)_{2P_x 2S} = \frac{(Z_1 Z_2)^{3/2} Z_1}{16} i [2D_{1-}(Z_1/2, Z_2/2, 2, 0, 0, 1) - Z_2 D_{1-}(Z_1/2, Z_2/2, 2, 1, 0, 1)]$$

$$IP(A,B)_{2P_x 2P_z} = \frac{(Z_1 Z_2)^{5/2}}{16} \; i [D_{1-}(Z_1/2, Z_2/2, 3, 0, 1, 1) - R(t) D_{1-}(Z_1/2, Z_2/2, 2, 0, 0, 1)]$$

$$IP(A,B)_{2P_x 2P_x} = \frac{(Z_1 Z_2)^{5/2}}{32} [D_{O-}(Z_1/2, Z_2/2, 3, 0, 0, 2) - D_{2-}(Z_1/2, Z_2/2, 3, 0, 0, 2)]$$

Table B-1 (con't)

$$ID(A,B)_{1S1S} = 2(Z_1Z_2)^{3/2} D_{O+}(Z_2,Z_1,3,0,1,0)$$

$$ID(A,B)_{1S2S} = \frac{(Z_1Z_2)^{3/2}}{2\sqrt{2}}[2D_{O+}(Z_2/2,Z_1,3,0,1,0)-Z_2D_{O+}(Z_2/2,Z_1,4,0,1,0)]$$

$$ID(A,B)_{1S2P_z} = \frac{(Z_1Z_2)^{3/2}Z_2}{2\sqrt{2}}[D_{O+}(Z_2/2,Z_1,4,0,2,0)]$$

$$ID(A,B)_{1S2P_x} = \frac{(Z_1Z_2)^{3/2}Z_2}{2\sqrt{2}}\, i\, [D_{1+}(Z_2/2,Z_1,4,0,1,1)]$$

$$ID(A,B)_{2S1S} = \frac{(Z_1Z_2)^{3/2}}{2\sqrt{2}}[2D_{O+}(Z_2,Z_1/2,3,0,1,0)-Z_1D_{O+}(Z_2,Z_1/2,3,1,1,0)]$$

$$ID(A,B)_{2S2S} = \frac{(Z_1Z_2)^{3/2}}{16}[4D_{O+}(Z_2/2,Z_1/2,3,0,1,0)-2Z_1D_{O+}(Z_2/2,Z_1/2,3,1,1,0)$$
$$-2Z_2D_{O+}(Z_2/2,Z_1/2,4,0,1,0)+Z_1Z_2D_{O+}(Z_2/2,Z_1/2,4,1,1,0)]$$

$$ID(A,B)_{2S2P_z} = \frac{(Z_1Z_2)^{3/2}Z_2}{16}[2D_{0+}(Z_2/2,Z_1/2,4,0,2,0)-Z_1D_{O+}(Z_2/2,Z_1/2,4,1,2,0)$$

$$ID(A,B)_{2S2P_x} = \frac{(Z_1Z_2)^{3/2}}{16}\, i[2D_{1+}(Z_2/2,Z_1/2,4,0,1,1)-Z_1D_{1+}(Z_2/2,Z_1/2,4,1,1,1)]$$

Table B-1 (con't)

$$ID(A,B)_{2P_z 1S} = \frac{(Z_1Z_2)^{3/2}Z_1}{2\sqrt{2}}[D_{O+}(Z_2,Z_1/2,4,0,2,0)+R(t)D_{O+}(Z_2,Z_1/2,3,0,1,0)]$$

$$ID(A,B)_{2P_z 2S} = \frac{(Z_1Z_2)^{3/2}Z_1}{16}[2D_{O+}(Z_2/2,Z_1/2,4,0,2,0)-Z_2D_{O+}(Z_1/2,Z_2/2,5,0,2,0)$$
$$+R(t)(2D_{O+}(Z_1/2,Z_2/2,3,0,1,0)-Z_2D_{O+}(Z_2/2,Z_1/2,4,0,1,0)]$$

$$ID(A,B)_{2P_z 2P_z} = \frac{(Z_1Z_2)^{5/2}}{16}[D_{O+}(Z_2/2,Z_1/2,5,0,3,0)+R(t)D_{O+}(Z_2/2,Z_1/2,4,0,2,0)]$$

$$ID(A,B)_{2P_z 2P_x} = \frac{(Z_1Z_2)^{5/2}}{16}\, i[D_{1+}(Z_2/2,Z_1/2,5,0,2,1)+R(t)D_{1+}(Z_2/2,Z_1/2,4,0,1,1)]$$

$$ID(A,B)_{2P_x 1S} = \frac{(Z_1Z_2)^{3/2}Z_1}{2\sqrt{2}}\, i\, D_{1+}(Z_2,Z_1/2,4,0,0,1)$$

$$ID(A,B)_{2P_x 2S} = \frac{(Z_1Z_2)^{3/2}Z_1}{16}i[2D_{1+}(Z_2/2,Z_1/2,4,0,1,1)-Z_2D_{1+}(Z_2/2,Z_1/2,5,0,1,1)]$$

$$ID(A,B)_{2P_x 2P_z} = \frac{(Z_1Z_2)^{5/2}}{16}\, i\, D_{1+}(Z_2/2,Z_1/2,5,0,2,1)$$

$$ID(A,B)_{2P_x 2P_x} = \frac{(Z_1Z_2)^{5/2}}{32}[D_{O+}(Z_2/2,Z_1/2,5,0,1,2)-D_{2+}(Z_2/2,Z_1/2,5,0,1,2)]$$

Table B-1 (con't)

$$S_{1S1S} = 2(Z_1Z_2)^{3/2}\, D_{O+}(Z_2,Z_1,2,0,0,0)$$

$$S_{1S2S} = \frac{(Z_1Z_2)^{3/2}}{2\sqrt{2}}[2D_{O+}(Z_2/2,Z_1,2,0,0,0)-Z_2D_{O+}(Z_2/2,Z_1,3,0\ 0,0)]$$

$$S_{1S2P_z} = \frac{(Z_1Z_2)^{3/2}Z_2}{2\sqrt{2}}[D_{O+}(Z_2/2,Z_1,3,0,1,0)]$$

$$S_{1S2P_x} = \frac{(Z_1Z_2)^{3/2}Z_2}{2\sqrt{2}}\, i\ [D_{1+}(Z_2/2,Z_1,3,0,0,1)]$$

$$S_{2S1S} = \frac{(Z_1Z_2)^{3/2}}{2\sqrt{2}}[2D_{O+}(Z_2,Z_1/2,2,0,0,0)-Z_1D_{O+}(Z_2,Z_1/2,2,1,0\ 0)]$$

$$S_{2S2S} = \frac{(Z_1Z_2)^{3/2}}{16}[4D_{O+}(Z_2/2,Z_1/2,2,0,0,0)-2Z_1D_{O+}(Z_2/2,Z_1/2,2,1,0\ 0)$$

$$-2Z_2D_{O+}(Z_2/2,Z_1/2,3,0,0,0)+Z_1Z_2D_{O+}(Z_2/2,Z_1/2,3,1,0,0)]$$

$$S_{2S2P_z} = \frac{(Z_1Z_2)^{3/2}Z_2}{16}[2D_{0+}(Z_2/2,Z_1/2,3,0,1,0)-Z_1D_{O+}(Z_2/2,Z_1/2,3,1,1,0)$$

$$S_{2S2P_x} = \frac{(Z_1Z_2)^{3/2}}{16}\, i[2D_{1+}(Z_2/2,Z_1/2,3,0,0,1)-Z_1D_{1+}(Z_2/2,Z_1/2,3,1,0,1)]$$

Table B-1 (con't)

$$S_{2P_z 1S} = \frac{(Z_1 Z_2)^{3/2} Z_1}{2\sqrt{2}} [D_{0+}(Z_2, Z_1/2, 3, 0, 1, 0) + R(t) D_{0+}(Z_2, Z_1/2, 2, 0, 0, 0)]$$

$$S_{2P_z 2S} = \frac{(Z_1 Z_2)^{3/2} Z_1}{16} [2D_{0+}(Z_2/2, Z_1/2, 3, 0, 1, 0) - Z_2 D_{0+}(Z_1/2, Z_2/2, 4, 0, 1, 0)$$

$$+ R(t)(2D_{0+}(Z_1/2, Z_2/2, 2, 0, 0, 0) - Z_2 D_{0+}(Z_2/2, Z_1/2, 3, 0, 0, 0)]$$

$$S_{2P_z 2P_z} = \frac{(Z_1 Z_2)^{5/2}}{16} [D_{0+}(Z_2/2, Z_1/2, 4, 0, 2, 0) + R(t) D_{0+}(Z_2/2, Z_1/2, 3, 0, 1, 0)]$$

$$S_{2P_z 2P_x} = \frac{(Z_1 Z_2)^{5/2}}{16} i [D_{1+}(Z_2/2, Z_1/2, 4, 0, 1, 1) + R(t) D_{1+}(Z_2/2, Z_1/2, 3, 0, 0, 1)]$$

$$S_{2P_x 1S} = \frac{(Z_1 Z_2)^{3/2} Z_1}{2\sqrt{2}} i\, D_{1+}(Z_2, Z_1/2, 3, 0, 0, 1)$$

$$S_{2P_x 2S} = \frac{(Z_1 Z_2)^{3/2} Z_1}{16} i [2D_{1+}(Z_2/2, Z_1/2, 3, 0, 0, 1) - Z_2 D_{1+}(Z_2/2, Z_1/2, 4, 0, 0, 1)]$$

$$S_{2P_x 2P_z} = \frac{(Z_1 Z_2)^{5/2}}{16} i\, D_{1+}(Z_2/2, Z_1/2, 4, 0, 1, 1)$$

$$S_{2P_x 2P_x} = \frac{(Z_1 Z_2)^{5/2}}{32} [D_{0+}(Z_2/2, Z_1/2, 4, 0, 0, 2) - D_{2+}(Z_2/2, Z_1/2, 4, 0, 0, 2)]$$

Table B-1 (con't)

$$IR(A,B)_{1S2P_z} = -\frac{(Z_1Z_2)^{3/2}Z_2}{2\sqrt{2}}[D_{1+}(Z_2/2,Z_1,3,0,0,1)]$$

$$IR(A,B)_{1S2P_x} = \frac{(Z_1Z_2)^{3/2}Z_2}{2\sqrt{2}}(-i)\ [D_{0+}(Z_2/2,Z_1,3,0,1,0)]$$

$$IR(A,B)_{2S2P_z} = -\frac{(Z_1Z_2)^{3/2}Z_2}{16}[2D_{1+}(Z_2/2,Z_1/2,3,0,0,1)-Z_1D_{1+}(Z_2/2,Z_1/2,3,1,0,1)]$$

$$IR(A,B)_{2S2P_x} = -\frac{(Z_1Z_2)^{3/2}}{16}\ i[2D_{0+}(Z_2/2,Z_1/2,3,0,1,0)-Z_1D_{0+}(Z_2/2,Z_1/2,3,1,1,0)]$$

$$IR(A,B)_{2P_z2P_z} = -\frac{(Z_1Z_2)^{5/2}}{16}[D_{1+}(Z_2/2,Z_1/2,4,0,1,1)+R(t)D_{0+}(Z_2/2,Z_1/2,3,0,0,1)]$$

$$IR(A,B)_{2P_z2P_x} = -\frac{(Z_1Z_2)^{5/2}}{16}\ i[D_{0+}(Z_2/2,Z_1/2,4,0,2,0)+R(t)D_{0+}(Z_2/2,Z_1/2,3,0,1,0)]$$

$$IR(A,B)_{2P_x2P_z} = \frac{(Z_1Z_2)^{5/2}}{32}i\ [D_{0+}(Z_2/2,Z_1/2,4,0,0,2-D_{2+}(Z_2/2,Z_1/2,4,0,0,2)]$$

$$IR(A,B)_{2P_x2P_x} = \frac{(Z_1Z_2)^{5/2}}{16}[D_{1+}(Z_2/2,Z_1/2,4,0,1,1)]$$

All other IR terms are zero.

APPENDIX C

CONSERVATION OF PROBABILITY

The purpose of this appendix is to prove that the probability is conserved for the trial function (2.14)(GRE65). Summation convention is used here.

From (2.17), the first equation and its complex conjugates may be written

$$a_n a_m^* H_{mn}(A) + a_m^* b_n K_{mn}(A,B) = i\dot{a}_m a_m^* + i\dot{b}_n a_m^* S_{mn} \qquad (C\text{-}1)$$

$$a_m a_n^* H_{mn}^*(A) + b_n^* a_m K_{mn}^*(A,B) = -i a_m \dot{a}_m^* - i\dot{b}_n^* a_m S_{mn}^* \qquad (C\text{-}2)$$

and the second equation and its complex conjugates are

$$a_n b_m^* \bar{K}_{mm}(B,A) + b_n b_m^* \bar{H}_{mn}(B) = i\dot{a}_n b_m^* S_{mn}^\dagger + i\dot{b}_m b_m^* \qquad (C\text{-}3)$$

$$a_n^* b_m \bar{K}_{mn}^*(B,A) + b_n^* b_m \bar{H}_{mn}^*(B) = -i\dot{a}_n^* b_m S_{mn}^{\dagger *} - i\dot{b}_m^* b_m \qquad (C\text{-}4)$$

Changing dummy indices in (C-2) and (C-4), we get

$$a_n a_m^* H_{nm}^*(A) + b_m^* a_n K_{nm}^*(A,B) = -i a_n \dot{a}_n^* - i\dot{b}_m^* a_n S_{nm}^* \qquad (C\text{-}5)$$

$$a_m^* b_n \bar{K}_{nm}^*(B,A) + b_m^* b_n \bar{H}_{nm}^*(B) = -i\dot{a}_m^* b_n S_{nm}^{\dagger *} - i\dot{b}_n^* b_n \qquad (C\text{-}6)$$

Combining (C-1), (C-5), (C-3), and (C-6) together and using the equation

$$S^{*}_{nm} = S^{\dagger}_{mn}$$

we obtain

$$\begin{aligned} & i(a^{*}_{m}\dot{a}_{m}+a_{n}\dot{a}^{*}_{n}+\dot{b}_{m}b^{*}_{m}+b_{n}\dot{b}^{*}_{n})+\dot{S}_{mn} \quad (\dot{b}_{n}a^{*}_{m}+\dot{a}^{*}_{m}b_{n})+i\dot{S}^{\dagger}_{mn} \quad (\dot{a}_{n}b^{*}_{m}+\dot{b}^{*}_{m}a_{n}) \\ = \; & a_{n}a^{*}_{m}(H_{mn}(A)-H^{*}_{nm}(A)) + b_{n}b^{*}_{m}(\bar{H}_{mn}(B)-\bar{H}^{*}_{nm}(B)) + \\ & b^{*}_{m}a_{n}(\bar{K}_{mn}(B,A)-K^{*}_{nm}(A,B)) + a^{*}_{m}b_{n}(K_{mn}(A,B)-\bar{K}^{*}_{nm}(B,A)) \end{aligned} \tag{C-7}$$

From the definition of the matrix elements (2.18), the following equations are valid:

$$H_{mn}(A) = H^{*}_{nm}(A) = H^{\dagger}_{mn}(A)$$

$$\bar{H}_{mn}(B) = \bar{H}^{*}_{nm}(B) = \bar{H}^{\dagger}_{mn}(B)$$

$$K_{mn}(A,B) - \bar{K}^{*}_{nm}(B,A) = -i \frac{\partial}{\partial t} S_{mn}$$

$$\bar{K}_{mn}(B,A) - K^{*}_{nm}(A,B) = -i \frac{\partial}{\partial t} S^{\dagger}_{mn}$$

Hence, equation (C-7) becomes

$$\frac{d}{dt}(a^{*}_{m}a_{m} + b^{*}_{m}b_{m} + S_{mn} \quad a^{*}_{m}b_{n} + S^{\dagger}_{mn} \quad a_{n}b^{*}_{m}) = 0$$

Q.E.D.

APPENDIX D

PROGRAM LISTING

The complete program consists of three parts:

Part I: ELECAP - Evaluate the double integrals. (p. 104)

Part II: TRANMAX - Combines the integral results to produce the matrix elements. (p. 133)

Part III: DCOUPL - Solves the coupled differential equations. (p. 140)

The JCL and the input data which produce the results shown in Figures 4-1 to 4-6 are given on page 169.

```
:VEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
            C     PROGRAM: ELECAP
            C  *
            C  *  AN INCIDENT ION Z1 BOMBARDS A TARGET ATOM Z2 WITH        *
            C  *  IMPACT PARAMETER B AND INCIDENT ENERGY E.  FROM THESE    *
            C  *  INITIAL CONDITIONS, THE INCIDENT VELOCITY V1 AND A       *
            C  *  KEPLER TRAJECTORY OF THE PROJECTILE CAN BE DETERMINED.   *
            C  *  ALL CALCULATIONS ARE IN A.U.                             *
            C
            C  *
            C  *  ALL THE CONSTANTS HERE ARE FROM A SMALL TABLE COMPILED   *
            C  *  BY E.R.COHEN, SCIENCE CENTER,ROCKWELL INTERNATIONAL      *
            C  *  FROM PHYSICS TODAY, SEPTEMBER 1974.                      *
            C
            C
            C PURPOSE---
            C     EVALUATE THE DOUBLE INTEGRATIONS FOR MATRIX ELEMENT
            C     OF HE++ ON H.  SEE TABLE B-1.
            C     USING ROMBERG INTEGRATION OVER RO AND CLENSHAW-
            C     CURTIS QUADRATURE OVER ANGLE.
            C
            C INPUT---
            C     NFILE:=1, READ FROM CARDS OR TERMINALS.
            C           =12, READ FROM DISK FILE.
            C     NOUT: =5, POINTS TO DISK FILE 5.
            C     NBACK:=6, POINTS TO DISK FILE 6.
            C     Z1,Z2: PROJECTILE AND TARGET CHARGE
            C     MZ1,MZ2:
            C           MASS OF PROJECTILE, TARGET IN PROTON (APPROXIMATELY AMU)
            C           BOTH OF THEM ARE INTEGERS.
            C     NFUN: TOTAL NUMBER OF ANGULAR FUNCTIONS ARE EVALUATED.
            C     NFUNO: TOTAL NUMBER OF RADIAL FUNCTIONS ARE EVALUATED.
            C     LIMIT: MAXIMUM NUMBER OF SUBINTERVALS FOR ANGULAR INTEGRATION.
            C     NFLAG: MAXIMUM NUMBER OF ITERATIONS FOR DOUBLE INTEGRATION.
            C     NDIFRO: (NFUN);EACH ELEMENT IS THE NUMBER WHICH INDICATES
            C              HOW MANY DIFFERENT POWERS OF  RO  CORRESPOND TO THE
            C              SAME ANGULAR INTEGRATION.
            C     RANG:  TWO-DIMENSION (16,2) ARRAY.
            C           EACH ROW CORRESPONDS TO DIFFERENT COMBINATIONS OF
            C           BESSEL FUNCTIONS AND TRIGONOMETRIC FUNCTIONS.
            C           FIRST COLUMN CORRESPONDS TO ALL (EXP(-EDA2)**2) TERMS.
            C           SECOND COLUMN CORRESPONDS TO ALL (EXP(-EDA2)) TERMS.
            C     EL:    PROJECTILE ENERGY IN KEV/AMU IN LAB SYSTEMS.
            C     B,BA:  IMPACT-PARAMETER IN CM AND A.U.
            C     LAM1,LAM2:
            C           VALUES ARE (1.0 AND 0.5) OR (0.5 OR 1.0)
            C           DIFFERENT COMBINATIONS CORRESPOND TO DIFFERENT
            C           SETS FO CALCULATIONS.
            C     TOLERR: REQUIRED TOLERANCE.
            C     SMALL:=0, FOR SMALL RANGE OF R, THETA IS ACTUALLY USED TO
            C           CALCULATE R.
            C           =1, FOR LARGE RANGE OF R, R IS USED DIRECTLY.
            C     THETA: FROM THIS VALUE, THE INTERNUCLEAR DISTANCE  R  CAN
            C           BE DETERMINED FOR SMALL RANGE OF  R  .
            C     R:    INTERNUCLEAR DISTANCE IN A.U.
```

```
C            INPUT  R  MUST BE GREATER THAN THE CLOSEST APPROACH.
C OUTPUT---
C     RDOT: RADIAL RELATIVE VELOCITY OF THE PROJECTILE AND TARGET.
C     RC:   SCALED ELECTRON COORDINATE.
C     BCA,BOB:
C           UPPER OR LOWER BOUNDS OF THE  BO  INTERVAL FOR EACH
C           SUBINTEGRATION.
C     NUM:  NUMBER OF FUNCTIONS LEFT FOR THE DOUBLE INTEGRATION.
C     MU:   REDUCED MASS OF MZ1 AND MZ2 IN A.U.
C     V1:   INCIDENT VELOCITY IN A.U.
C     V0:   ATOMIC UNIT OF VELOCITY IN CM/SEC
C     V:    TANGENTIAL VELOCITY AT THE CLOSEST APPROACH.
C     BO:   HALF OF THE COLLISION DIAMETER IN A.U.
C     E0:   ONE ATOMIC ENERGY UNIT IN EV.
C     SCANGL:SCATTERING ANGLE IN DEGREE
C     R:    INTERNUCLEAR DISTANCE IN A.U.
C     LAMDA:A FUNCTION OF  R  IF THE INCIDENT ENERGY AND
C           THE IMPACT PARAMETER ARE GIVEN.
C           AT THE CLOSEST APPROACH,  LAMDA  EQUALS  V  .
C     FLAG:   AN INDICATOR, (NFUNO) ARRAY.
C             (=2) MEANS TERMINATING THE INTEGRATION
C              OVER  BO  FOR THE CORRESPONDING FUNCTION.
C     DBLINT: (NFUNO) THE RESULTS OF THE DOUBLE INTEGRATION.
C     UPPER:  (NFUNO), EACH ELEMENT CORRESPONDS THE UPPER
C             BOUNDS OF  BO  INTEGRATIONS.
C
C
C  *  VALO(NFUNO),ESTERO(NFUNO),INDEX(NFUNO),POINTO(NFUNO)
C  *  ARE EXPLAINED IN SUBROUTINE  ROMINT  .
C  *  THOSE ARE RESULTS OF  BO  INTEGRATION.
C
C  *  SUBPROGRAM USED:
C  *   ROMINT
C
C
ISN 0002        REAL*8 VALO(114),ESTERO(114),DBLINT(114),UPPER(114)
ISN 0003        REAL*8 B,BA,BA2,BO,C,RDOT,
               1       E,E0,EDA1,EDA2,EDA3,ROA,ROB,DRO,
               2       R,RO,SCANGL,TOLERR,THETA,TOLERO,
               3       V,V1,V0,Z12,PI,LAM1,LAM2,LAM(5)
C
C  *  MRATIO:RATIO OF PROTON TO ELECTRON MASS.
C
ISN 0004        REAL*8 LAMDA,MU,SMALL,MZ1A,MRATIO/  1.83615152D+3/
C
ISN 0005        INTEGER INDEXO(114),POINTO(114),FLAG(114),NDIFRC(42,2)
ISN 0006        INTEGER Z1,Z2,NANG(16,2)
C
ISN 0007        DATA E0/.27211608D+2/,
               1     PI/3.14159265358979D0/
C
ISN 0008        COMMON /CONST1/EDA1,EDA2,EDA3
ISN 0009        COMMON /CONST2/R,LAMDA
ISN 0010        COMMON /INDIT1/NFUN,NFUNO,LIMIT
ISN 0011        COMMON /INDIT2/FLAG,NDIFRO,NANG
ISN 0012        COMMON /DATOL/TOLERR
```

```
ISN 0013            COMMON /INTBOM/INDEX0,POINT0
ISN 0014            COMMON /ESTBOM/VAL0,ESTER0
                    C
                    C  *  INPUT
ISN 0015              READ (1,1000)NFILE,NBACK,NOUT,NROM
ISN 0016              WRITE(3,1000)NFILE,NBACK,NOUT,NROM
                    C
ISN 0017            READ(1,1010)Z1,Z2,MZ1,MZ2,NFUN,NFUN0,LIMIT,NFLAG,
                   1      (NDIFRO(I,1),NDIFRO(I,2),I=1,NFUN),(NANG(J,1),NANG(J,2),
                   2       J=1,16)
                    C
                    C  *  B IS THE RATIO OF IMPACT PARAMETER AND B0.
                    C  *  THETA  DETERMINES THE  R  VALUE,
                    C  *  I.E.  THETA  =0, THE CLOSEST APPROACH.
                    C
ISN 0018            10    CONTINUE
ISN 0019              READ(1,1020,END=2000)EL,B,LAM1,LAM2,TOLERR,SMALL,E,THETA
                    C
                    C  *  THE REQUIRED TOLERANCE OF THE SECOND (RO) INTEGRATION
                    C  *  IS 5 TIMES LARGER THAN THAT OF THE FIRST (ANGULAR)
                    C  *  INTEGRATION.
ISN 0020              TOLER0=5*TOLERR
                    C
                    C  *  CHANGE EVERYTHING INTO A.U.
                    C  *  E  IS THE REDUCED MASS' ENERGY IN A.U. IN RELATIVE COORDINATE
                    C  *  OF THE PROJECTILE AND TARGET.
                    C
ISN 0021              E=(1.0D+3)*EL/E0
ISN 0022              MU=DFLOAT(MZ1*MZ2)/DFLOAT(MZ1+MZ2)
ISN 0023              E=MU*E
                    C
ISN 0024              RU=MU*MRATIO
ISN 0025              Z12=Z1*Z2
ISN 0026              B0=Z12/(2*E)
ISN 0027              BA=B*B0
ISN 0028              V1=DSQRT(E*2/MU)
ISN 0029              BA2=BA*BA
ISN 0030              C=DSQRT(BA2+B0*B0)
ISN 0031              V=V1*BA/(B0+C)
ISN 0032              SCANGL=180.D0*(1.D0-DATAN(BA/B0)*2/PI)
                    C
                    C     FOR SMALL RANGE OF R, THETA IS THE INPUT
                    C     OTHERWISE, R IS THE INPUT
                    C
ISN 0033              IF (SMALL.EQ.0.D0)R=BA2/(-B0+C*DCOS(THETA))
                    C
ISN 0035              LAMDA=V1*BA/R
ISN 0036              RDOT=DSQRT(2*(E-Z12/R)/MU-LAMDA**2)
                    C
                    C
                    C  *  AT THE CLOSEST APPROACH, VELOCITY ALONG THE
                    C  *  INTERNUCLEAR AXIS IS ZERO.
                    C
ISN 0037              IF (SMALL.EQ.0.D0.AND.THETA.EQ.0.D0)RDOT=0.D0
                    C
ISN 0039              EDA1=LAM1
```

```
ISN 0040            EDA2=LAM2
ISN 0041            EDA3=RDOT
ISN 0042            DRO=2*2.D0/(EDA1+EDA2)
ISN 0043            WRITE(3,1040)E,V1,R,RDOT,LAMDA,EDA1,EDA2,DRO
                C
                C   *  CHECK TO SEE IF THE REST OF INPUT IS READ FROM DISK OR CARDS.
                C
ISN 0044            IF (NFILE.EQ.1)GO TO 60
ISN 0046            READ(NBACK,1030)ROA,NUM,(DBLINT(I),FLAG(I),I=1,NFUNO)
ISN 0047            GO TO 80
                C
                C
                C   *  INITIALIZE  NUM. THE TWO ARRAY  FLAG  AND  DBLINT  .
                C
ISN 0048        60  NUM=NFUNO
ISN 0049            DO 70 I=1,NFUNO
ISN 0050               FLAG(I)=0
ISN 0051               DBLINT(I)=0.D0
ISN 0052        70  CONTINUE
                C
ISN 0053            RCA=0.D0
ISN 0054        80  ROB=ROA+DRO
ISN 0055            WRITE(3,1020)ROA,ROB
                C
                C   *  INITIALIZE  INDEXO  BEFORE CALLING SUBROUTINE  ROMINT .
                C   *  INDEXO  IS AN INDICATOR SHOWING WHICH FUNCTION NEEDS
                C   *  INTEGRATION.
                C
ISN 0056            DO 100 I=1,NFUNO
ISN 0057               IF (FLAG(I)-NFLAG)90,95,90
ISN 0058        90     INDEXO(I)=0
ISN 0059               GO TO 100
ISN 0060        95     INDEXO(I)=1
ISN 0061        100   CONTINUE
                C
ISN 0062            CALL ROMINT(ROA,ROB,NFUNO,TOLERO,NROM)
                C
ISN 0063            DO 120 I=1,NFUNO
ISN 0064               IF (FLAG(I).EQ.NFLAG)GO TO 120
ISN 0066               DBLINT(I)=DBLINT(I)+VALO(I)
ISN 0067               IF (DABS(VALO(I)).GT.DABS(DBLINT(I)*TOLERR))GO TC 110
ISN 0069               FLAG(I)=FLAG(I)+1
ISN 0070               IF (FLAG(I)-NFLAG)120,105,120
ISN 0071        105     NUM=NUM-1
ISN 0072               UPPER(I)=ROB
ISN 0073               IF (NUM.EQ.0)GO TO 130
ISN 0075               GO TO 120
ISN 0076        110    FLAG(I)=0
ISN 0077        120 CONTINUE
                C
ISN 0078            REWIND NBACK
ISN 0079            WRITE(NBACK,1030)ROB,NUM,(DBLINT(I),FLAG(I),I=1,NFUNO)
ISN 0080            ENDFILE NBACK
ISN 0081            ROA=ROB
ISN 0082            GO TO 80
                C
```

```
ISN 0083     130   CONTINUE
                   C
                   C
ISN 0084           WRITE(NOUT,1010)Z1,Z2,MZ1,MZ2
ISN 0085           WRITE(NOUT,1040)MRATIO,EL,V1,BA,SCANGL,R,RDOT,LAMDA,
                  1       (DBLINT(I),I=1,NFUNO)
                   C
ISN 0086           GO TO 10
                   C
ISN 0087     1000  FORMAT(5I5)
ISN 0088     1010  FORMAT(2I5)
ISN 0089     1020  FORMAT(2D15.8)
ISN 0090     1030  FORMAT(D25.16,I5)
ISN 0091     1040  FORMAT(2D25.16)
                   C
ISN 0092     2000  STOP
ISN 0093           END
```

```
LEVEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE ROMINT(A,B,NFUNO,TOLERO,NROM)
                C  *
                C  *       ***  MODIFIED ROMBERG QUADRATURE  ***
                C  *
                C  *  THIS SUBROUTINE IS APPLYING MODIFIED ROMBERG QUADRATURE
                C  *  TO INTEGRATE  NFUNO  DIFFERENT FUNCTIONS FROM A TO B
                C  *  (ALL INTEGRALS MUST HAVE THE SAME UPPER AND LOWER BOUNDS)
                C  *  TO A REQUIRED RELATIVE TOLERANCE.
                C  *
                C  *  THE ALGORITHM IN THE REFERENCE HAS BEEN REWRITTEN IN FORTRAN
                C  *  AND MODIFIED FOR MULTI-FUNCTION INTEGRATIONS.
                C  *  THIS PROGRAM WILL RETURN VALO(NFUNO),ESTERO(NFUNO),
                C  *  POINTO(NFUNO) AND INDEXO(NFUNO) THROUGH THE
                C  *  COMMON BLOCKS.
                C  *  ONE CAN PASS THESE VARIABLES INTO THIS SUBROUTINE
                C  *  THROUGH THE INPUT ARGUMENTS.
                C  *  SLIGHTLY CHANGE.
                C  *  THE MAXIMUM NUMBER OF EXTRAPOLATIONS IS 15.
                C  *
                C  *  REFERENCE: G. FAIRWEATHER, COMM. ACM 12(JUNE, 1969),
                C  *             P. 324-325.
                C  *
                C  *  SUBPROGRAMS USED:
                C  *   DVALUE
                C  *
                C  *  A:       LOWER BOUND.
                C  *  B:       UPPER BOUND.
                C  *  NFUNO:   NUMBER OF DIFFERENT FUNCTIONS.
                C  *  N:       NUMBER OF FUNCTION EVALUATIONS.
                C  *  TOLERO:  THE DESIRED ACCURACY, RELATIVE TOLERANCE.
                C  *  ESTERO:  A (NFUNO) ARRAY OF ESTIMATED RELATIVE ERRORS.
                C  *  VALO:    A (NFUNO) ARRAY OF INTEGRATED RESULTS.
                C  *  POINTO:  AN (NFUNO) ARRAY TO RECORD NUMBER OF FUNCTION
                C  *           EVALUATIONS.
                C  *  INDEXO:  A (NFUNO) ARRAY INDICATOR.
                C  *           INITIALLY PASSED IN BY COMMON BLOCK.
                C  *           0: THE CORRESPONDENT FUNCTION NEEDS TO INTEGRATE.
                C  *           1: THIS FUNCTION DOES NOT NEED TO INTEGRATE.
                C  *           IF THE ESTIMATED INTEGRAL REACHES THE REQUIRED
                C  *           TOLERANCE, CHANGE  0  TO  1  .
                C  *  T:       A (NFUNO) ARRAY OF THE IMPROVED TRAPEZOID VALUES.
                C  *  FM:      A (NFUNO,15) TWO-DIMENSION ARRAY OF THE ROMBERG
                C  *           TABLE.
                C  *  F:       A TEMPORARY (NFUNO) ARRAY
                C  *  NUM:     INITIALLY SET TO THE NUMBER OF FUNCTIONS
                C  *           NEED INTEGRATION.
                C  *           DECREASES BY 1 WHEN ONE INTEGRAL REACHES
                C  *           THE TOLERANCE.
                C  *           EQUALS 0 WHEN ALL THE INTEGRALS ARE ALL DONE.
                C  *  S,S1,SO:  (NFUNO) TEMPORARY ARRAYS OF SUMS.
                C  *  DVALUE:   A SUBROUTINE RETURNS THE FUNCTION EVALUATIONS.
                C  *           ARGUMENTS(X,F): INPUT X AND FUNCTION VALUES  F  .
                C  *
                C
```

```
                C
ISN 0003            REAL*8 A,B,SH,H,X,R,BB,TOLER0
                C
ISN 0004            REAL*8 F(114),T(114),RM(114,15),VAL0(114),
                   1       ESTER0(114),S(114),S1(114),S0(114)
                C
ISN 0005            REAL*8 OLD(114),OLDERR(114)
                C
ISN 0006            INTEGER INDEX0(114),POINT0(114)
                C
ISN 0007            COMMON /ESTROM/VAL0,ESTER0
ISN 0008            COMMON /INTROM/INDEX0,POINT0
                C
                C
ISN 0009            SH=B-A
                C
                C   *  IF TESTING THE PROGRAM, SET ITEST0=1
                C
ISN 0010            ITEST0=1
                C
                C   *  INITIAL TRAPEZOID VALUE
                C
ISN 0011            CALL DVALUE(A,F)
ISN 0012            CALL DVALUE(B,T)
ISN 0013            H=SH*.5D0
                C
                C   *  BY THE FOLLOWING LOOP, OBTAIN THE NUMBER OF FUNCTIONS
                C   *  NEED TO INTEGRATE.
ISN 0014            NUM=0
ISN 0015            DO 20 I=1,NFUN0
ISN 0016               IF (INDEX0(I).EQ.1)GO TO 20
ISN 0018               T(I)=H*(F(I)+T(I))
ISN 0019               NUM=NUM+1
ISN 0020      20    CONTINUE
                C
                C   *  INITIAL RECTANGLE VALUE.
                C
ISN 0021            X=(A+B)*.5D0
ISN 0022            CALL DVALUE(X,F)
ISN 0023            DO 30 I=1,NFUN0
ISN 0024               IF (INDEX0(I).EQ.1)GO TO 30
ISN 0026               RM(I,1)=SH*F(I)
ISN 0027               OLD(I)=T(I)+RM(I,1)
ISN 0028               IF (OLD(I))24,26,24
ISN 0029      26       OLDERR(I)=1.D0
ISN 0030               GO TO 30
ISN 0031      24       OLDERR(I)=DABS(T(I)-RM(I,1))
ISN 0032      30    CONTINUE
                C
                C
ISN 0033            N=2
ISN 0034            R=4
                C
ISN 0035            DO 11 K=1,NROM
ISN 0036               BB=(R*.5D0-1.D0)/(R-1.D0)
                C
```

```
C  *  IMPROVED TRAPEZOID VALUE
C
ISN 0037          DO 40 I=1,NFUNO
ISN 0038             IF (INDEX0(I).EQ.1)GO TO 40
ISN 0040             S(I)=0
ISN 0041             T(I)=BM(I,1)+BB*(T(I)-BM(I,1))
ISN 0042    40    CONTINUE
C
C  *  DOUBLE NUMBER OF SUBDIVISIONS OF (A,B).
ISN 0043          N=2*N
ISN 0044          H=SH/DFLOAT(N)
C
C  *  CALCULATE THE RECTANGLE VALUE
ISN 0045          IF (N-32)1,1,2
ISN 0046    1     NO=N
ISN 0047          GO TO 4
ISN 0048    2     NO=32
C
ISN 0049    3     IF (N-512)4,4,5
ISN 0050    4     N1=N
ISN 0051          GO TO 6
ISN 0052    5     N1=512
C
C
ISN 0053    6     DO 9 K2=1,N,512
ISN 0054             DO 50 I=1,NFUNO
ISN 0055    50          S1(I)=0
C
ISN 0056             KK=K2+N1-1
C
ISN 0057             DO 8 K1=K2,KK,32
ISN 0058                DO 60 I=1,NFUNO
ISN 0059    60             S0(I)=0
ISN 0060                KKK=K1+NO-1
C
ISN 0061                DO 7 K0=K1,KKK,2
ISN 0062                   X=A+DFLOAT(K0)*H
ISN 0063                   CALL DVALUE(X,F)
ISN 0064                   DO 70 I=1,NFUNO
ISN 0065                      IF (INDEX0(I).EQ.1)GO TO 70
ISN 0067                      S0(I)=S0(I)+F(I)
ISN 0068    70                CONTINUE
ISN 0069    7              CONTINUE
C
ISN 0070                DO 80 I=1,NFUNO
ISN 0071                   IF (INDEX0(I).EQ.1)GO TO 80
ISN 0073                   S1(I)=S0(I)+S1(I)
ISN 0074    80             CONTINUE
ISN 0075    8           CONTINUE
C
ISN 0076             DO 90 I=1,NFUNO
ISN 0077                IF (INDEX0(I).EQ.1)GO TO 90
ISN 0079                S(I)=S(I)+S1(I)
ISN 0080    90          CONTINUE
ISN 0081    9        CONTINUE
C
```

```
ISN 0082                DO 100 I=1,NFUNO
ISN 0083                   IF (INDEX0(I).EQ.1)GO TO 100
ISN 0085                   RM(I,K+1)=2.D0*H*S(I)
ISN 0086        100     CONTINUE
                C
                C  *  END CALCULATION OF RECTANGLE VALUE.
                C
ISN 0087                R=4
                C
                C  *  FORM ROMBERG TABLE FROM RECTANGLE VALUE.
ISN 0088                DO 10 J=1,K
ISN 0089                   L=K+1-J
ISN 0090                   DO 110 I=1,NFUNO
ISN 0091                      IF (INDEX0(I).EQ.1)GO TO 110
ISN 0093                      RM(I,L)=RM(I,L+1)+(RM(I,L+1)-RM(I,L))/(R-1.D0)
ISN 0094        110           CONTINUE
                C
ISN 0095                   R=4.D0*R
ISN 0096        10      CONTINUE
                C
                C     CHECK IF THE INTEGRATION REACHES THE TOLERANCE.
                C     OLD(I) AND OLDERR(I) RECORD THE PREVIOUS INTEGRATED
                C     VALUES AND ABSOLUTE ERRORS.
                C     VALO AND ESTERO RECORD THE CURRENT INTEGRATED
                C     VALUES AND ABSOLUTE ERRORS.
                C
ISN 0097                DO 120 I=1,NFUNO
ISN 0098                   IF (INDEX0(I).EQ.1)GO TO 120
ISN 0100                   VALO(I)=T(I)+RM(I,1)
ISN 0101                   IF (VALO(I))150,170,150
ISN 0102        150        ESTERO(I)=DABS(T(I)-RM(I,1))
ISN 0103                   IF (ESTERO(I).GT.DABS(TOLERO*VALO(I)))GO TO 160
ISN 0105                   ESTERO(I)=DABS(ESTERO(I)/VALO(I))
ISN 0106                   VALO(I)=VALO(I)*.5D0
ISN 0107                   GO TO 170
                C
                C     IF THE CURRENT RELATIVE ERROR STARTS INCREASING,
                C     I.E. GREATER THAN THE PREVIOUS ONE, TAKE THE
                C     PREVIOUS INTEGRATED VALUE AND ERROR AND
                C     STOP GOING INTO FURTHER STEPS.
                C     ALSO PRINT OUT THIS SPECIAL CASE.
                C     OTHERWISE, DIVIDE MORE SUBINTERVALS AND
                C     INTEGRATE AGAIN.
                C
ISN 0108        160        IF (K.LE.3)GO TO 180
ISN 0110                   IF (ESTERO(I)-OLDERR(I))180,190,190
ISN 0111        190        ESTERO(I)=DABS(OLDERR(I)/OLD(I))
ISN 0112                   VALO(I)=OLD(I)*0.5D0
ISN 0113                   WRITE (3,1030)I,N,VALO(I),ESTERO(I)
ISN 0114        170        NUM=NUM-1
ISN 0115                   INDEX0(I)=1
ISN 0116                   POINT0(I)=N+1
ISN 0117                   GO TO 120
ISN 0118        180        OLDERR(I)=ESTERO(I)
ISN 0119                   OLD(I)=VALO(I)
ISN 0120        120     CONTINUE
```

```
                C
                C  *  IF TESTING THIS PROGRAM, SET ITEST0=1.
                C  *  THEN IT WILL PRINT OUT (''E OUTPUT.
ISN 0121                IF (ITEST0.EQ.0)GC TO 135
ISN 0123                IF (K.LT.3)GO TO 135
ISN 0125                WRITE (3,1000)K
ISN 0126                DO 130 I=1,NFUN0
ISN 0127                   IF (INDEX0(I).EQ.1)GO TO 130
ISN 0129                   WRITE (3,1010)I,N,VAL0(I),ESTER0(I)
ISN 0130        130     CONTINUE
                C
ISN 0131        135        IF (NUM.EQ.0)GO TO 12
ISN 0133        11      CONTINUE
ISN 0134                WRITE (3,1020)K
                C
ISN 0135        12      RETURN
                C
ISN 0136        1000    FORMAT('0',I5)
ISN 0137        1010    FORMAT(5X,2I5,2D20.8)
ISN 0138        1020    FORMAT(I5,25H    REFINEMENT NOT ALLOWED)
ISN 0139        1030    FORMAT(16H    SPECIAL CASE ,2I5,2D25.16)
                C
ISN 0140                END
```

```
:VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE DVALUE(X,D)
                    C
                    C  *  CALCULATE INTEGRANT F(NFUNO) WITH INPUT  X (=RO).
                    C  *  ANGINT  IS A SUBROUTINE DOING THE ANGULAR-INTEGRATION,
                    C  *  F(X).
                    C  *
                    C  *  ARRAYS ARE:
                    C  *  D(NFUNO), FLAG(NFUNO),INDEXO(NFUNO),POINTO(NFUNO),
                    C  *  NEWINT(NFUN),ESTERR(NFUN),INDEX(NFUN),POINTS(NFUN),
                    C  *  NDIFRO(NFUN,2)
                    C  *
                    C  *  SUBROUTINE USED:
                    C  *   ANGINT
                    C
                    C
ISN 0003            REAL*8 EDA1,EDA2,EDA3,RO,X,Y
ISN 0004            REAL*8 D(114),NEWINT(42),ESTERR(42),ROEXP(5,2)
ISN 0005            INTEGER Z1,Z2,INDEX(42),POINTS(42),FLAG(114),NDIFRO(42,2),
                   1        INDEX1(42),NANG(16,2),INDEXO(114),POINTO(114)
                    C
ISN 0006            COMMON /CONST1/EDA1,EDA2,EDA3
ISN 0007            COMMON /INDIT1/NFUN,NFUNO,LIMIT
ISN 0008            COMMON /INDIT2/FLAG,NDIFRO,NANG
ISN 0009            COMMON /VARIBL/RO
ISN 0010            COMMON /INTARY/INDEX,POINTS
ISN 0011            COMMON /ESTINT/NEWINT,ESTERR
ISN 0012            COMMON /INTROM/INDEXO,POINTO
                    C
                    C
ISN 0013            RO=X
                    C
                    C  *  IF RO=0, ALL THE FUNCTION VALUES ARE 0.
                    C  *  NO NEED TO EVALUATE FURTHER.
                    C
ISN 0014            IF (RO) 9,4,9
ISN 0015       4    DO 8 I=1,NFUNO
ISN 0016       8       D(I)=0.D0
                    C
ISN 0017            GO TO 100
                    C
                    C  *  INITIALIZE  ROEXP(5,2)
                    C  *  ROEXP(5,2) IS AN ARRAY WHICH HAS VALUES OF
                    C  *  EXP(-EDA1*RO)*RO**N  AND  EXP(-EDA1*2*RO)*RO**N  .
                    C  *  N  IS THE INDEX OF ROW.
                    C
ISN 0018       9    Y=DEXP(-EDA1*RO)
ISN 0019            ROEXP(1,1)=Y*RO
ISN 0020            ROEXP(1,2)=ROEXP(1,1)*Y
                    C
ISN 0021            DO 10 K=2,5
ISN 0022               ROEXP(K,1)=ROEXP(K-1,1)*RO
ISN 0023               ROEXP(K,2)=ROEXP(K,1)*Y
ISN 0024      10    CONTINUE
                    C
```

```
C  *  INITIALIZE  INDEX(NFUN)
C  *  INDEX(NFUN)  IS AN INDICATOR AND EXPLAINED IN SUBROUTINE
C  *  ANGINT .
C
ISN 0025        NFUN2=NFUN/2
ISN 0026        NFUN02=NFUN0/2
C
ISN 0027        KK=0
ISN 0028        DO 20 J=1,NFUN,2
ISN 0029           K1=KK
ISN 0030           K=NDIFRO(J,1)+NDIFRO(J+1,1)
ISN 0031           J2=(J+1)/2
C  *  REAL PART OF INTEGRAL F(X).
ISN 0032           INDEX(J2)=1
C  *  IMAGINARY PART OF INTEGRAL F(X).
ISN 0033           JTEMP=J2+NFUN2
ISN 0034           INDEX(JTEMP)=1
ISN 0035           DO 14 JJ=1,K
ISN 0036              K1=K1+1
ISN 0037              IF (INDEX0(K1).EQ.1)GO TO 12
ISN 0039              INDEX(J2)=0
ISN 0040     12       KTEMP=K1+NFUN02
ISN 0041              IF (INDEX0(KTEMP).EQ.1)GO TO 14
ISN 0043              INDEX(JTEMP)=0
ISN 0044     14       CONTINUE
ISN 0045              KK=KK+K
ISN 0046     20    CONTINUE
C
C     INDEX1 IS A TEMPORARY INDICATOR ARRAY
C
ISN 0047        DO 22 I=1,NFUN
ISN 0048     22    INDEX1(I)=INDEX(I)
C
ISN 0049        CALL ANGINT(-1.0D0,+1.D0,NFUN,LIMIT)
C  *  ICOP 90 CONSTRUCTS THE NEEDED FUNCTION VALUES BY USING
C  *   NDIFRO  AND  INDEX0  .
C
ISN 0050        II=0
ISN 0051        KK=NFUN02
ISN 0052        DO 90 J=1,NFUN2
ISN 0053           IF (INDEX1(J).EQ.1)GO TO 50
ISN 0055           DO 40 II=1,2
ISN 0056              L1=NDIFRO(2*J+II-2,1)
ISN 0057              IF (L1) 25,40,25
ISN 0058     25       L2=NDIFRO(2*J+II-2,2)-1
ISN 0059              DO 30 JJ=1,L1
ISN 0060                 LL=LL+1
ISN 0061                 IF (INDEX0(LL).EQ.1)GO TO 30
ISN 0063                 D(LL)=NEWINT(J)*ROEXP(L2+JJ,II)
ISN 0064     30       CONTINUE
ISN 0065     40    CONTINUE
ISN 0066           GO TO 55
ISN 0067     50    LL=LL+NDIFRO(2*J-1,1)+NDIFRO(2*J,1)
ISN 0068     55    JK=J+NFUN2
ISN 0069           IF (INDEX1(JK).EQ.1)GO TO 80
ISN 0071           DO 70 II=1,2
```

```
ISN 0072                L1=NDIFRO(2*J+II-2,1)
ISN 0073                IF (L1) 57,70,57
ISN 0074      57        L2=NDIFRO(2*J+II-2,2)-1
ISN 0075                DO 60 JJ=1,L1
ISN 0076                   KK=KK+1
ISN 0077                   IF (INDEXO(KK).EQ.1)GO TO 60
ISN 0079                   D(KK)=NEWINT(JK)*ROEXP(L2+JJ,II)
ISN 0080      60        CONTINUE
ISN 0081      70     CONTINUE
ISN 0082             GO TO 90
ISN 0083      80     KK=KK+NDIFRO(2*J-1,1)+NDIFRO(2*J,1)
ISN 0084      90  CONTINUE
                  C
ISN 0085      100 RETURN
ISN 0086          END
```

```
LVEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE ANGINT(A,B,NFUN,LIMIT)
                C
                C  *          ***  CLENSHAW-CURTIS QUADRATURE  ***          *
                C  *
                C  *  EVALUATE THE ANGULAR INTEGRATION BY CHEBYSHEV SERIES  *
                C  *  EXPANSION, CLENSHAW-CURTIS QUADRATURE.                *
                C  *  BY FAST FOURIER TRANSFORM ALGORITHM, THE COSINE       *
                C  *  TRANSFORM COEFFICIENTS CAN BE CALCULATED.             *
                C
                C  *
                C  *  REFERENCE: GENTLEMAN, W. M., COMM. ACM 15 (MAY 1972), *
                C  *             P337-355.                                  *
                C  *                                                        *
                C  *  THE ORIGINAL FORTRAN PROGRAM IN THE ABOVE REFERENCE   *
                C  *  DOES NOT WORK IN IBM 360/370 COMPUTER SYSTEM.         *
                C  *  SO THIS SUBROUTINE HAS BEEN MODIFIED TO BE USED IN    *
                C  *  IBM 360/370 SYSTEM AND ALSO TO BE ABLE TO INTEGRATE   *
                C  *   NFUN  AT THE SAME TIME FROM A TO B TO THE REQUESTED  *
                C  *  RELATIVE ACCURACY WHILE USING NO MORE THAN  LIMIT     *
                C  *  FUNCTION EVALUATIONS.                                 *
                C  *  ALL THE INTEGRALS MUST HAVE THE SAME UPPER AND LOWER  *
                C  *  BOUNDS.                                               *
                C  *                                                        *
                C  *  IN ORDER TO SAVE SOME STORAGE, PUT  NEWINT  ,  ESTERR
                C  *  AND  INDEX  INTO COMMON BLOCKS WITH THE OUTSIDE PROGRAM.
                C  *  AN ALTERNATIVE METHOD IS PASSING IN  NEWINT  ,  ESTERR
                C  *  AS ARGUMENTS, I.E.
                C  *      SUBROUTINE ANGINT(A,B,NFUN,LIMIT,NEWINT,ESTERR)
                C  *
                C  *  THIS SUBROUTINE WILL RETURN NEWINT(NFUN)
                C  *  ESTERR(NFUN) , POINTS(NFUN)  AND INDEX(NFUN)
                C  *  THROUGH THE COMMON BLOCKS.
                C  *
                C  *  SUBPROGRAMS USED:
                C  *   FVALUE , R3PASS
                C  *
                C  *  A,B:    LOWER AND UPPER BOUNDS OF THE INTEGRAL.
                C  *  LIMIT:  MAXIMUM NUMBER OF INTERVALS CAN BE USED TO
                C  *          EVALUATE THE INTEGRAL. (>19)
                C  *  NFUN:   TOTAL NUMBER OF FUNCTIONS TO BE INTEGRATED.
                C  *  CSXFRM: (NFUN*LIMIT) ARRAY OF COSINE TRANSFORM COEFFICIENTS.
                C  *  NEWINT: A (NFUN) ARRAY OF INTEGRATED RESULTS FOR  NFUN  .
                C  *  ESTERR: ARRAY OF ESTIMATED RELATIVE ERRORS OF 'NFUN' FUNCTIONS.
                C  *  INDEX:  A (NFUN) ARRAY IS INITIALLY PASSED IN BY COMMON BLOCK.
                C  *          EACH ELEMENT IS  0  (DO THE INTEGRAL) OR
                C  *          1  (DO NOT CARRY OUT THE INTEGRATION).
                C  *          IF THE INTEGRAL REACHES THE TOLERANCE, SET TO 1.
                C  *          THEN NO MORE CALCULATIONS FOR THIS INTEGRAL.
                C  *  POINTS: EACH ELEMENT  A (NFUN) ARRAY IS THE TOTAL NUMBER
                C  *          OF INTERVALS HAVE BEING USED.
                C  *  NUM:    INITIALLY EQUALS THE NUMBER OF FUNCTIONS NEED
                C  *          INTEGRATIONS.
                C  *          DECREASES BY 1 WHEN ONE INTEGRAL HAS REACHED
                C  *          TOLERANCE.
```

```
C  *          IF NUM EQUALS ZERO, ALL INTEGRALS HAVE REACHED
C  *          THE TOLERANCE.
C  *  FVALUE: FVALUE(X,CT1) IS A SUBROUTINE TO EVALUATE  NFUN
C  *          DIFFERENT FUNCTIONS WITH ARGUMENT X AND RETURNS
C  *          A NFUN-DIM VECTOR CT1 AS ALL FUNCTION VALUES.
C  *
C
C
ISN 0003        REAL*8 A,B,TOLERR
C               REAL*8 SCLINT
ISN 0004        REAL*8 PI,RT3,CENTRE,WIDTH,SHIFT,FUND,ANGLE,C,S,TEMP
C
ISN 0005        REAL*8 CSXFRM(42,163),ESTERR(42),OLDINT(42),
               1       NEWINT(42),T1(42),T2(42),T3(42),T4(42),
               2       T5(42),T6(42),T7(42),T8(42),
               3       T9(42),T10(42),T11(42),T12(42)
C
ISN 0006        REAL*8 CT1(42),CT2(42),CT3(42),CT4(42),
               1       CT5(42),CT6(42),CT7(42)
C
ISN 0007        INTEGER L(8),INDEX(42),POINTS(42)
C
ISN 0008        EQUIVALENCE(CSXFRM(1,1),CT1(1)),(CSXFRM(1,2),CT2(1)),
               1           (CSXFRM(1,3),CT3(1)),(CSXFRM(1,4),CT4(1)),
               2           (CSXFRM(1,5),CT5(1)),(CSXFRM(1,6),CT6(1)),
               3           (CSXFRM(1,7),CT7(1))
C
ISN 0009        EQUIVALENCE (L(1),L1),(L(2),L2),(L(3),L3),(L(4),L4),
               1            (L(5),L5),(L(6),L6),(L(7),L7),(L(8),L8),
               2            (J8,JREV)
C
ISN 0010        COMMON /DATOL/TOLERR
ISN 0011        COMMON /INTARY/INDEX,POINTS
ISN 0012        COMMON /BEARY/CSXFRM
ISN 0013        COMMON /ESTINT/NEWINT,ESTERR
C
ISN 0014        DATA PI,RT3/   3.141592653589793D0,   1.732050807568D0 /
C
C  *  MAXIMUM INTERVALS ONE CAN DO IN THIS PROGRAM ARE
C  *  2*3**(MMAX+1).
C  *  FOR MMAX=8, THE  LIMIT  IS 39366.
C
ISN 0015        DATA MMAX/   8/
C
C  *  IF TEST THE PROGRAM AND WANT TO CHECK THE INPUT VALUES,
C  *  SET ITEST=1
C
ISN 0016        ITEST=0
ISN 0017        IF (ITEST.EQ.0)GO TO 8
ISN 0019        WRITE(3,1110)A,B,NFUN,LIMIT
ISN 0020  1110  FORMAT('0',F5.1,5X,F5.1,5X,2I5)
ISN 0021  8     CONTINUE
C
C  **   INITIALIZATION
C
ISN 0022        CENTRE=(A+B)*.5D0
```

```
ISN 0023            WIDTH=(B-A)*.5D0
                C
ISN 0024            NUM=0
ISN 0025            DO 5 I=1,NFUN
ISN 0026               IF (INDEX(I).EQ.1)GO TO 5
ISN 0028               NUM=NUM+1
ISN 0029         5  CONTINUE
                C
ISN 0030            DO 10 J=1,MMAX
ISN 0031               L(J)=1
ISN 0032        10  CONTINUE
                C
                C  ** COSINE TRANSFORM WITH N=6
                C
ISN 0033            N=6
                C
                C  *  CT1 TO CT7 ARE TEMPORARY VECTORS WITH DIMENSION  NFUN  .
                C
ISN 0034            CALL FVALUE(A,CT1)
ISN 0035            CALL FVALUE(B,CT7)
ISN 0036            SHIFT=WIDTH*RT3*.5D0
ISN 0037            TEMP=CENTRE-SHIFT
ISN 0038            CALL FVALUE(TEMP,CT2)
ISN 0039            TEMP=CENTRE+SHIFT
ISN 0040            CALL FVALUE(TEMP,CT6)
ISN 0041            SHIFT= WIDTH*.5D0
ISN 0042            TEMP=CENTRE-SHIFT
ISN 0043            CALL FVALUE(TEMP,CT3)
ISN 0044            TEMP=CENTRE+SHIFT
ISN 0045            CALL FVALUE(TEMP,CT5)
ISN 0046            CALL FVALUE(CENTRE,CT4)
                C
ISN 0047            DO 20 I=1,NFUN
ISN 0048               IF (INDEX(I).EQ.1)GO TO 18
ISN 0050               POINTS(I)=N
ISN 0051               GO TO 20
ISN 0052        18     POINTS(I)=0
ISN 0053        20  CONTINUE
                C
                C  ** EVALUATE THE FACTORED N=6 COSINE TRANSFORM
                C  ** FOR  NFUN  FUNCTIONS
                C
ISN 0054            DO 30 I=1,NFUN
ISN 0055               IF (INDEX(I).EQ.1) GO TO 30
ISN 0057               T1(I)=CSXFRM(I,1)+CSXFRM(I,7)
ISN 0058               T2(I)=CSXFRM(I,1)-CSXFRM(I,7)
ISN 0059               T3(I)=2.D0*CSXFRM(I,4)
ISN 0060               T4(I)=CSXFRM(I,2)+CSXFRM(I,6)
ISN 0061               T5(I)=(CSXFRM(I,2)-CSXFRM(I,6))*RT3
ISN 0062               T6(I)=CSXFRM(I,3)+CSXFRM(I,5)
ISN 0063               T7(I)=CSXFRM(I,3)-CSXFRM(I,5)
ISN 0064               T8(I)=T1(I)+2.D0*T6(I)
ISN 0065               T9(I)=2.D0*T4(I)+T3(I)
ISN 0066               T10(I)=T2(I)+T7(I)
ISN 0067               T11(I)=T1(I)-T6(I)
ISN 0068               T12(I)=T4(I)-T3(I)
```

```
ISN 0069              CSXFRM(I,1)=T8(I)+T9(I)
ISN 0070              CSXFRM(I,2)=T10(I)+T5(I)
ISN 0071              CSXFRM(I,3)=T11(I)+T12(I)
ISN 0072              CSXFRM(I,4)=T2(I)-2.D0*T7(I)
ISN 0073              CSXFRM(I,5)=T11(I)-T12(I)
ISN 0074              CSXFRM(I,6)=T10(I)-T5(I)
ISN 0075              CSXFRM(I,7)=T8(I)-T9(I)
ISN 0076        30    CONTINUE
                C
                C  *  NUMBER OF POINTS HAS BEEN USED.
                C
ISN 0077              NUSED=7
                C
                C  ** GO TO INTEGRAL COMPUTATION
                C
ISN 0078              GO TO 210
                C
                C  *  COMPUTE REFINED APPROXIMATION
                C  *  EVALUATE FUNCTIONS FOR N=18,54,...,2*3**9.
                C  *  SAMPLE FUNCTION AT INTERMEDIATE POINTS IN DIGIT
                C  *  REVERSED ORDER.
                C  *  AS THE SEQUENCE IS GENERATED, COMPUTE THE FIRST
                C  *  (RADIX FOUR TRANSFORM) PASS OF THE FAST FOURIER
                C  *  TRANSFORM (FFT).
                C
ISN 0079        100   DO 110 J=2,MMAX
ISN 0080                 L(J-1)=L(J)
ISN 0081        110   CONTINUE
                C
ISN 0082              L(MMAX)=3*L(MMAX-1)
ISN 0083              J=NUSED
ISN 0084              FUND=PI/DFLOAT(3*N)
ISN 0085              DO 128 J1=1,L1,1
ISN 0086                 DO 127 J2=J1,L2,L1
ISN 0087                    DO 126 J3=J2,L3,L2
ISN 0088                       DO 125 J4=J3,L4,L3
ISN 0089                          DO 124 J5=J4,L5,L4
ISN 0090                             DO 123 J6=J5,L6,L5
ISN 0091                                DO 122 J7=J6,L7,L6
ISN 0092                                   DO 121 J8=J7,L8,L7
ISN 0093                                      ANGLE=FUND*DFLOAT(3*JREV-2)
ISN 0094                                      SHIFT=WIDTH*DCOS(ANGLE)
ISN 0095                                      TEMP=CENTRE-SHIFT
ISN 0096                                      CALL FVALUE(TEMP,T1)
ISN 0097                                      TEMP=CENTRE+SHIFT
ISN 0098                                      CALL FVALUE(TEMP,T3)
ISN 0099                                      SHIFT=WIDTH*DSIN(ANGLE)
ISN 0100                                      TEMP=CENTRE+SHIFT
ISN 0101                                      CALL FVALUE(TEMP,T2)
ISN 0102                                      TEMP=CENTRE-SHIFT
ISN 0103                                      CALL FVALUE(TEMP,T4)
ISN 0104                                      DO 120 I=1,NFUN
ISN 0105                                         IF (INDEX(I).EQ.1)GO TO 120
ISN 0107                                         T5(I)=T1(I)+T3(I)
ISN 0108                                         T6(I)=T2(I)+T4(I)
ISN 0109                                         CSXFRM(I,J+1)=T5(I)+T6(I)
```

```
ISN 0110                                        CSXFRM(I,J+2)=T1(I)-T3(I)
ISN 0111                                        CSXFRM(I,J+3)=T5(I)-T6(I)
ISN 0112                                        CSXFRM(I,J+4)=T2(I)-T4(I)
ISN 0113     120                              CONTINUE
ISN 0114                                      J=J+4
ISN 0115     121                           CONTINUE
ISN 0116     122                        CONTINUE
ISN 0117     123                     CONTINUE
ISN 0118     124                  CONTINUE
ISN 0119     125               CONTINUE
ISN 0120     126            CONTINUE
ISN 0121     127         CONTINUE
ISN 0122     128      CONTINUE
             C
             C   ** DO RADIX3 PASSES OF FAST FOURIER TRANSFORM
             C
ISN 0123           N2=2*N
ISN 0124           NSTEP=4
ISN 0125     150   J1=NUSED+NSTEP
ISN 0126           J2=J1+NSTEP
ISN 0127           CALL R3PASS (N2,NSTEP,NFUN,NUSED,J1,J2)
ISN 0128           NSTEP=3*NSTEP
ISN 0129           IF (NSTEP .LT. N) GO TO 150
             C
             C   ** COMBINE RESULTS
             C
             C   ** FIRST DO J=0 AND J=N
             C
ISN 0131           DO 40 I=1,NFUN
ISN 0132              IF (INDEX(I).EQ.1)GO TO 40
ISN 0134              T1(I)=CSXFRM(I,1)
ISN 0135              T2(I)=CSXFRM(I,NUSED+1)
ISN 0136              CSXFRM(I,1)=T1(I)+2.D0*T2(I)
ISN 0137              CSXFRM(I,NUSED+1)=T1(I)-T2(I)
ISN 0138              T1(I)=CSXFRM(I,N+1)
ISN 0139              T2(I)=CSXFRM(I,N2+2)
ISN 0140              CSXFRM(I,N+1)=T1(I)+T2(I)
ISN 0141              CSXFRM(I,N2+2)=T1(I)-2.D0*T2(I)
ISN 0142     40    CONTINUE
             C
             C   ** NOW DO REMAINING VALUES OF J
             C
ISN 0143           N3=3*N
ISN 0144           NLESS1=N-1
ISN 0145           DO 180 J=1,NLESS1
ISN 0146              J1=N+J
ISN 0147              J2=N3-J
ISN 0148              ANGLE=FUND*DFLOAT(J)
ISN 0149              C=DCOS(ANGLE)
ISN 0150              S=DSIN(ANGLE)
ISN 0151              DO 185 I=1,NFUN
ISN 0152                 IF (INDEX(I).EQ.1)GO TO 185
ISN 0154                 T1(I)=C*CSXFRM(I,J1+2)-S*CSXFRM(I,J2+2)
ISN 0155                 T2(I)=(S*CSXFRM(I,J1+2)+C*CSXFRM(I,J2+2))*RT3
ISN 0156                 CSXFRM(I,J1+2)=CSXFRM(I,J+1)-T1(I)-T2(I)
ISN 0157                 CSXFRM(I,J2+2)=CSXFRM(I,J+1)-T1(I)+T2(I)
```

```
ISN 0158                  CSXFRM(I,J+1)=CSXFRM(I,J+1)+2.D0*T1(I)
ISN 0159      185      CONTINUE
ISN 0160      180   CONTINUE
              C
              C  *  NOW UNSCRAMBLE THE DIGIT REVERSED ORDER TO GET
              C  *  THE CORRECT DIGIT ORDER.
              C  *  SPECIAL FEATURE OF FFT.
              C
ISN 0161           DO 50 I=1,NFUN
ISN 0162              IF (INDEX(I).EQ.1)GO TO 50
ISN 0164              T1(I)=CSXFRM(I,N2+1)
ISN 0165              T2(I)=CSXFRM(I,N2+2)
ISN 0166      50   CONTINUE
              C
ISN 0167           DO 190 J=1,NLESS1
ISN 0168              J1=NUSED+J
ISN 0169              J2=N2+J
ISN 0170              DO 195 I=1,NFUN
ISN 0171                 IF (INDEX(I).EQ.1)GO TO 195
ISN 0173                 CSXFRM(I,J2)=CSXFRM(I,J1)
ISN 0174                 CSXFRM(I,J1)=CSXFRM(I,J2+2)
ISN 0175      195      CONTINUE
ISN 0176      190   CONTINUE
              C
ISN 0177           DO 60 I=1,NFUN
ISN 0178              IF (INDEX(I).EQ.1)GO TO 60
ISN 0180              CSXFRM(I,N3)=T1(I)
ISN 0181              CSXFRM(I,N3+1)=T2(I)
ISN 0182      60   CONTINUE
              C
ISN 0183           N=N3
ISN 0184           NUSED=N+1
              C
              C  ** INTEGRAL EVALUATION
              C
              C  ** EVALUATE NEW ESTIMATE OF INTEGRAL
              C
ISN 0185      210  NLESS3=N-3
ISN 0186           DO 70 I=1,NFUN
ISN 0187              IF (INDEX(I).EQ.1)GO TO 70
ISN 0189              NEWINT(I)=.5D0*CSXFRM(I,NUSED)/DFLOAT(1-N**2)
ISN 0190              NEWINT(I)=NEWINT(I)+.5D0*CSXFRM(I,1)
ISN 0191      70   CONTINUE
              C
ISN 0192           DO 220 J=1,NLESS3,2
ISN 0193              JREV=N-J
ISN 0194              DO 225 I=1,NFUN
ISN 0195              IF (INDEX(I).EQ.1)GO TO 225
ISN 0197              NEWINT(I)=NEWINT(I)+CSXFRM(I,JREV)/DFLOAT(JREV*(2-JREV))
ISN 0198      225      CONTINUE
ISN 0199      220   CONTINUE
              C
              C  *  SINCE N=6 IS THE MINIMUM NUMBER OF INTERVALS
              C  *  FOR INTEGRATION, SKIP THE ERROR ESTIMATION.
              C
ISN 0200           IF (N.EQ.6)GO TO 230
```

```
                C
                C  ** TEST IF ESTIMATED ERROR ADEQUATE FOR N>=18.
                C
ISN 0202              DO 80 I=1,NFUN
ISN 0203                 IF (INDEX(I).EQ.1)GO TC 80
ISN 0205                 ESTERR(I)=DABS(OLDINT(I)*3.D0-NEWINT(I))
ISN 0206                 POINTS(I)=N
                C
                C  *  IF TESTING, PUT THE FOLLOWING STATEMENTS IN.
                C        SCLINT=WIDTH*NEWINT(I)/DFLOAT(N/2)
                C        WRITE (3,930)I,POINTS(I),SCLINT,ESTERR(I)
                C930     FORMAT(3H I=,I5,3H N=,I5,23H INTEGRAL ESTIMATED AS ,
                C    1      D15.8,7H ERROR ,D15.8)
                C
ISN 0207                 IF (DABS(NEWINT(I)*TOLERR) .LT. ESTERR(I)) GC TO 75
ISN 0209                 NUM=NUM-1
ISN 0210                 INDEX(I)=1
ISN 0211      75         IF (NEWINT(I))77,80,77
ISN 0212      77         ESTERR(I)=ESTERR(I)/DABS(NEWINT(I))
ISN 0213      80      CONTINUE
                C
                C  *  ALL INTEGRATIONS REACH THE REQUIRED TOLERANCE.
                C
ISN 0214              IF (NUM.EQ.0)GO TO 400
                C
ISN 0216      230     DO 90 I=1,NFUN
ISN 0217                 IF (INDEX(I).EQ.1)GO TO 90
ISN 0219                 OLDINT(I)=NEWINT(I)
ISN 0220      90      CONTINUE
                C
                C  ** IF ESTIMATED ERROR TOO LARGE, REFINE SAMPLING IF PERMITTED
                C  *  I.E.  N (NUMBER OF INTERVALS USED) <  LIMIT  .
                C
ISN 0221              IF (3*N+1 .LE. LIMIT) GO TO 100
                C
                C  ** IF REFINEMENT NCT PERMITTED, OR IF ESTIMATED ERROR
                C  ** SATISFACTORY, RESCALE ANSWERS AND RETURN
                C
ISN 0223      400     DC 300 I=1,NFUN
ISN 0224              IF (POINTS(I).EQ.0)GO TO 300
ISN 0226              NEWINT(I)=WIDTH*NEWINT(I)/DFLOAT(POINTS(I)/2)
                C     WRITE (3,920)POINTS(I),NEWINT(I),ESTERR(I)
                C     IF (INDEX(I).EQ.0)WRITE(3,910)
ISN 0227      300     CONTINUE
                C
                C910  FORMAT('+',50X,25H REFINEMENT NOT PERMITTED)
                C920  FORMAT(/I5,5X,D15.8,5X,D15.8)
                C
ISN 0228              RETURN
ISN 0229              END
```

```
VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,IC,XREF
                C
                C
ISN 0002                SUBROUTINE R3PASS (N2,N,NFUN,L0,L1,L2)
                C
                C  *  RADIX 3 PASS FOR FFT OF REAL SEQUENCE OF LENGTH  N2  .
                C
                C  *  REFERENCES:
                C  *    GENTLEMAN, W.M., COMM. ACM 15(MAY 1972) 343-346
                C  *    GENTLEMAN, W.M. AND SANDE, G.,PROC. AFIPS 1966
                C  *          FJCC, VOL. 29,SPARTAN BOOKS,N.Y., 563-578
                C
                C
ISN 0003                REAL*8 C(42,163)
ISN 0004                REAL*8 TWOPI,HAFRT3,RSUM,RDIFF,RSUM2,ISUM,IDIFF,IDIFF2
ISN 0005                REAL*8 FUND,ANGLE,C1,S1,C2,S2,R0,R1,R2,I0,I1,I2
ISN 0006                INTEGER HALFM,M3,K,K0,K1,J,J0,J1,N2,M
                C
                C  *  CHANGE THE ARGUMENT OF  IND  ARRAY TO   NFUN  .
                C
ISN 0007                INTEGER IND(42),POINTS(42)
                C
ISN 0008                DATA TWOPI, HAFRT3/   6.283185307179586D0, .86602540378 4D0  /
ISN 0009                COMMON /INTARY/IND,POINTS
ISN 0010                COMMON /REARY/C
                C
                C
ISN 0011                HALFM=(M-1)/2
ISN 0012                M3=M*3
ISN 0013                FUND=TWOPI/DFLOAT(M3)
                C
                C  ** TWIDDLE FACTOR UNITY
                C
ISN 0014                DO 10 K=1,N2,M3
ISN 0015                   DO 10 I=1,NFUN
ISN 0016                      IF (IND(I).EQ.1)GO TO 10
ISN 0018                      RSUM=(C(I,L1+K)+C(I,L2+K))
ISN 0019                      C(I,L2+K)=(C(I,L1+K)-C(I,L2+K))*HAFRT3
ISN 0020                      C(I,L1+K)=C(I,L0+K)-RSUM*.5D0
ISN 0021                      C(I,L0+K)=C(I,L0+K)+RSUM
ISN 0022                10 CONTINUE
                C
                C
ISN 0023                J=M/2+1
ISN 0024                DO 20 K=J,N2,M3
ISN 0025                   DO 20 I=1,NFUN
ISN 0026                      IF (IND(I).EQ.1)GO TO 20
ISN 0028                      RSUM=(C(I,L1+K)+C(I,L2+K))*HAFRT3
ISN 0029                      RDIFF=(C(I,L1+K)-C(I,L2+K))
ISN 0030                      C(I,L1+K)=C(I,L0+K)-RDIFF
ISN 0031                      C(I,L2+K)=RSUM
ISN 0032                      C(I,L0+K)=C(I,L0+K)+RDIFF*.5D0
ISN 0033                20 CONTINUE
                C
                C
```

```
ISN 0034            DO 40 J=1,HALFM
ISN 0035               J0=J+1
ISN 0036               J1=M-J+1
                C   ** COMPUTE THE TWIDDLE FACTOR
ISN 0037               ANGLE=FUND*DFLOAT(J)
ISN 0038               C1=DCOS(ANGLE)
ISN 0039               S1=DSIN(ANGLE)
ISN 0040               C2=C1*C1-S1*S1
ISN 0041               S2=2.D0*C1*S1
                C   ** CHOOSE THE REPLICATION
ISN 0042               DO 30 K0=J0,N2,M3
ISN 0043                  K1=K0-J0+J1
                C   ** OBTAIN TWIDDLED VALUES
ISN 0044                  DO 30 I=1,NFUN
ISN 0045                     IF (IND(I).EQ.1)GO TO 30
ISN 0047                     R0=C(I,L0+K0)
ISN 0048                     I0=C(I,L0+K1)
ISN 0049                     R1=C1*C(I,L1+K0)-S1*C(I,L1+K1)
ISN 0050                     I1=S1*C(I,L1+K0)+C1*C(I,L1+K1)
ISN 0051                     R2=C2*C(I,L2+K0)-S2*C(I,L2+K1)
ISN 0052                     I2=S2*C(I,L2+K0)+C2*C(I,L2+K1)
                C   ** COMPUTE THE TRANSFORM AND RETURN IN PLACE
ISN 0053                     RSUM=R1+R2
ISN 0054                     RDIFF=(R1-R2)*HAFRT3
ISN 0055                     RSUM2=R0-.5D0*RSUM
ISN 0056                     ISUM=I1+I2
ISN 0057                     IDIFF=(I1-I2)*HAFRT3
ISN 0058                     IDIFF2=I0-.5D0*ISUM
ISN 0059                     C(I,L0+K0)=R0+RSUM
ISN 0060                     C(I,L0+K1)=RSUM2+IDIFF
ISN 0061                     C(I,L1+K0)=RSUM2-IDIFF
ISN 0062                     C(I,L1+K1)=RDIFF+IDIFF2
ISN 0063                     C(I,L2+K0)=RDIFF-IDIFF2
ISN 0064                     C(I,L2+K1)=I0+ISUM
ISN 0065            30    CONTINUE
ISN 0066            40 CONTINUE
ISN 0067               RETURN
ISN 0068               END
```

```
!VEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE FVALUE(C,F)
                C
                C  *  This program evaluates the function for the angular
                C  *  integrals, C  corresponds the variable X(i.e. cosTHETA)
                C  *  in the formula.  RO is constant in this program.
                C  *  R: the internuclear distance, an input constant through
                C  *     the whole program
                C  *  There are 21 complex angular integrands corresponding to
                C  *  different argument combinations.
                C  *
                C  *  N  EQUALS  NFUN  .
                C
ISN 0003            REAL*8 Y0,Y1,Y1C,Y1S,Y,
                   1       EDA1,EDA2,EDA3,R,LAMDA,RO,ROVB2,ROVB,
                   2       C,S,S2,F(42),FF(16),YEXP(2),
                   3       BESJ0,BESJ1,BESJ2,BESEL0,BESEL1
                C
ISN 0004            INTEGER NANG(16,2),FLAG(114),NDIFRO(42,2),
                   1        INDEX(42),POINTS(42)
                C
ISN 0005            COMMON /CONST1/EDA1,EDA2,EDA3
ISN 0006            COMMON /CONST2/R,LAMDA
ISN 0007            COMMON /INTARY/INDEX,POINTS
ISN 0008            COMMON /INDIT1/NFUN,NFUN0,LIMIT
ISN 0009            COMMON /INDIT2/FLAG,NDIFRO,NANG
ISN 0010            COMMON /VARIBL/RO
                C
                C
ISN 0011            NFUN2=NFUN/2
ISN 0012            ROVB2=RO*RO+R*R
ISN 0013            ROVB=RO*R*2.D0
ISN 0014            Y=DSQRT(ROVB2-ROVB*C)
ISN 0015            IF (EDA1.LT.EDA2)Y=DSQRT(ROVB2+ROVB*C)
ISN 0017            S2=1-C*C
ISN 0018            S=DSQRT(S2)
ISN 0019            Y0=RO*S*LAMDA
ISN 0020            Y1=EDA3*RO*C
ISN 0021            Y1C=DCOS(Y1)
ISN 0022            Y1S=DSIN(Y1)
                C
                C
                C  *  CHECK TO SEE IF BESSEL FUNCTION NEEDS TO BE CALLED.
                C
ISN 0023            DO 5   I=18,21
ISN 0024               IF (INDEX(I).EQ.1)GO TO 3
ISN 0026               BESJ1=BESEL1(Y0)
ISN 0027               BESJ0=BESEL0(Y0)
ISN 0028               GO TO 7
                C
ISN 0029        3       II=I+NFUN2
ISN 0030               IF (INDEX(II).EQ.1)GO TO 5
ISN 0032               BESJ1=BESEL1(Y0)
ISN 0033               BESJ0=BESEL0(Y0)
ISN 0034               GO TO 7
```

```
ISN 0035      5      CONTINUE
ISN 0036            GO TO 30
ISN 0037      7      IF (Y0)10,20,10
ISN 0038      10    BESJ2=2.D0*BESJ1/Y0-BESJ0
ISN 0039            GO TO 90
              C
ISN 0040      20    BESJ2=0.D0
ISN 0041            GO TO 90
ISN 0042      30    BESJ0=0.D0
ISN 0043            BESJ2=0.D0
ISN 0044            DO 50 I=1,10
ISN 0045               IF (INDEX(I).EQ.1)GO TO 40
ISN 0047               BESJ0=BESEL0(Y0)
ISN 0048               GO TO 60
              C
ISN 0049      40       II=I+NFUN2
ISN 0050               IF (INDEX(II).EQ.1)GO TO 50
ISN 0052               BESJ0=BESEL0(Y0)
ISN 0053               GO TO 60
ISN 0054      50    CONTINUE
              C
ISN 0055      60    BESJ1=0.D0
ISN 0056            DO 80 I=11,17
ISN 0057               IF (INDEX(I).EQ.1)GO TO 70
ISN 0059               BESJ1=BESEL1(Y0)
ISN 0060               GO TO 90
              C
ISN 0061      70       II=I+NFUN2
ISN 0062               IF (INDEX(II).EQ.1)GO TO 80
ISN 0064               BESJ1=BESEL1(Y0)
ISN 0065               GO TO 90
ISN 0066      80    CONTINUE
              C
ISN 0067      90    CONTINUE
              C
              C  *  YEXP(2) IS A TWO-COLUMN ARRAY.
              C  *  YEXP(1): EXP(-EDA2*2*Y)
              C  *  YEXP(2): EXP(-EDA2*Y)
              C
ISN 0068            YEXP(2)=DEXP(-EDA2*Y)
ISN 0069            YEXP(1)=YEXP(2)**2
              C
              C     INITIAL CALCULATION FOR NECESSARY FUNCTION EVALUATIONS
              C
ISN 0070            FF(1)=BESJ0
ISN 0071            FF(2)=Y*BESJ0
ISN 0072            FF(3)=C*BESJ0
ISN 0073            FF(4)=FF(2)*C
ISN 0074            FF(5)=FF(3)*C
ISN 0075            FF(6)=FF(5)*Y
ISN 0076            FF(7)=FF(5)*C
              C
ISN 0077            FF(8)=S*BESJ1
ISN 0078            FF(9)=FF(8)*C
ISN 0079            FF(10)=Y*FF(8)
ISN 0080            FF(11)=FF(10)*C
```

```
ISN 0081            FF(12)=FF(9)*C
                C
ISN 0082            FF(13)=S2*(BESJ0-BESJ2)
ISN 0083            FF(14)=FF(13)*C
ISN 0084            FF(15)=S2*(BESJ0+BESJ2)
ISN 0085            FF(16)=C*FF(15)
                C
                C   REAL AND IMAGINARY PARTS OF FUNCTIONS LISTED IN TABLE B-1
                C
ISN 0086            K=0
ISN 0087            DO 200 I=1,16
ISN 0088               DO 100 J=1,2
ISN 0089                  L1=NANG(I,J)
ISN 0090                  IF (L1.EQ.0)GO TO 100
ISN 0092                  K=K+1
ISN 0093                  F(K)=FF(I)*YEXP(J)
ISN 0094                  KK=K+NFUN2
ISN 0095                  F(KK)=F(K)*Y1S
ISN 0096                  F(K)=F(K)*Y1C
ISN 0097       100     CONTINUE
ISN 0098       200  CONTINUE
                C
ISN 0099            RETURN
ISN 0100            END
```

```
:VEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            DOUBLE PRECISION FUNCTION BESEL1(XX)
                C
                C  *  EVALUATE BESSEL FUNCTION J1 TO A HIGHER ACCURACY.  *
                C  *  USE CHEBYSHEV SERIES EXPANSION.                     *
                C  *
                C  *  REFERENCE: NATIONAL PHYSICAL LABORATORY MATHEMATICAL*
                C  *  TABLES, VOL. 5, CHEBYSHEV SERIES FOR MATHEMATICAL   *
                C  *  FUNCTIONS, BY C. W. CLENSHAW. (1962)                *
                C
ISN 0003            INTEGER SIGN
                C
ISN 0004            REAL*8 AS(15),AP(10),AQ(10)
ISN 0005            REAL*8 X,Y,B2,E,Q2,P2,XPI2,PI4,B,B1,P,P1,Q,Q1,XX
                C
ISN 0006            DATA PI4/2.356194490192345D0/
                C
ISN 0007            DATA AS(15),AS(14),AS(13),AS(12),AS(11),AS(10),AS(9),
                   1     AS(8),AS(7),AS(6),AS(5),AS(4),AS(3),AS(2),AS(1)
                   2      /1.296717541210530D0,-1.191801160541217D0,
                   3       1.287994098857678D0,-.661443934134543D0,
                   4       .177709117239728D0,-.029175524806154D0,
                   5       .003240270182684D0,-.000260444389348D0,
                   6       .158870192400-4,-.761758780D-6,.294970700-7,
                   7       -.942421D-9,.25281D-10,-.578D-12,.11D-13/
                C
ISN 0008            DATA AP(10),AP(9),AP(8),AP(7),AP(6),AP(5),AP(4),AP(3),
                   1     AP(2),AP(1)/-.001806081720027D0,-.000898989833086D0,
                   2     -.3987284300D-5,-.61776340D-7,-.1871891D-8,
                   3     .88169D-10,-.5705D-11,-.470D-12,-.47D-13,-.5D-14/
                C
ISN 0009            DATA AQ(10),AQ(9),AQ(8),AQ(7),AQ(6),AQ(5),AQ(4),AQ(3),
                   1     AQ(2),AQ(1)/-.093555574139071D0,-.96277235492D-4,
                   2     .913861526D-6,-.20959781D-7,.822919D-9,-.46864D-10,
                   3     .3515D-11,-.326D-12,.36D-13,-.4D-14/
                C
ISN 0010            X=XX
ISN 0011            SIGN=0
ISN 0012            IF (X) 20,30,40
                C  *  FLAG IS SET FOR NEGATIVE X
ISN 0013          20  SIGN=1
ISN 0014            X=DABS(X)
ISN 0015          40 IF (X-8.D0) 50,60,70
                C
                C  *  FOR X<8
ISN 0016          50 Y=X*X/16.D0-2.D0
ISN 0017            B1=0.D0
ISN 0018            B=0.D0
ISN 0019            DO 80 I=1,15
ISN 0020               B2=B1
ISN 0021               B1=B
ISN 0022               B=Y*B1-B2+AS(I)
ISN 0023          80 CONTINUE
ISN 0024            BESEL1=X*(B-B2)/16.D0
ISN 0025            GO TO 100
```

```
                C
                C  *   FOR X>8
ISN 0026        70     XPI2=.7978845608028650
ISN 0027               XPI2=XPI2/DSQRT(X)
ISN 0028               E=X-PI4
ISN 0029               X=8.D0/X
ISN 0030               Y=4.D0*X*X-2.D0
ISN 0031               P1=0.D0
ISN 0032               P=0.D0
ISN 0033               Q1=0.D0
ISN 0034               Q=0.D0
ISN 0035               DO 90 I=1,10
ISN 0036                  P2=P1
ISN 0037                  P1=P
ISN 0038                  Q2=Q1
ISN 0039                  Q1=Q
ISN 0040                  P=Y*P1-P2+AP(I)
ISN 0041                  Q=Y*Q1-Q2+AQ(I)
ISN 0042           90  CONTINUE
ISN 0043               BESEL1=XPI2*((P-P2)*DCOS(E)-X*(Q-Q2)*DSIN(E))/2.D0
ISN 0044               GO TO 100
                C
                C  *   X=0
ISN 0045           30  BESEL1=0.D0
ISN 0046               GO TO 100
                C
                C  *   X=8
ISN 0047           60  BESEL1=.23463634685391 5D0
                C  *   OUTPUT
ISN 0048          100  IF (SIGN.EQ.1) BESEL1=-BESEL1
                C
ISN 0050               RETURN
ISN 0051               END
```

```
:VEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
              C
ISN 0002            DOUBLE PRECISION FUNCTION BESEL0(XX)
              C
              C  *  EVALUATE BESSEL FUNCTION J0 TO A HIGHER ACCURACY.   *
              C  *  USE CHEBYSHEV SERIES EXPANSION.                     *
              C  *
              C  *  REFERENCE: NATIONAL PHYSICAL LABORATORY MATHEMATICAL*
              C  *  TABLES, VOL. 5, CHEBYSHEV SERIES FOR MATHEMATICAL   *
              C  *  FUNCTIONS, BY C. W. CLENSHAW. (1962)                *
              C
ISN 0003            REAL*8 AS(15),AP(10),AQ(10)
ISN 0004            REAL*8 X,Y,B2,B,Q2,P2,XPI2,PI4,B,B1,P,P1,Q,Q1,XX
              C
ISN 0005            DATA PI4/.785398163397448D0/
              C
ISN 0006            DATA AS(15),AS(14),AS(13),AS(12),AS(11),AS(10),AS(9),
                    1      AS(8),AS(7),AS(6),AS(5),AS(4),AS(3),AS(2),AS(1)
                    2      /-.3154559429497800D0,-.008723442352852D0,
                    3       .2651786132033370D0,-.3700949938726500D0,
                    4       .1580671023320970D0,-.0348937694114090D0,
                    5       .0048191800694680D0,-.0004606261662060D0,
                    6       .0000324603288210D0,-.1761946908D-5,
                    7       .76081636D-7,-.2679254D-8,.78487D-10,
                    8      -.1944D-11,.41D-13/
              C
ISN 0007            DATA AP(10),AP(9),AP(8),AP(7),AP(6),AP(5),AP(4),AP(3),
                    1    AP(2),AP(1)/1.9989206986950370D0,-.5365220468130-3,
                    2     .3075184788D-5,-.51705945D-7,.1630646D-8,
                    3     -.78641D-10,.5168D-11,-.430D-12,.43D-13,-.5D-14/
              C
ISN 0008            DATA AQ(10),AQ(9),AQ(8),AQ(7),AQ(6),AQ(5),AQ(4),AQ(3),
                    1     AQ(2),AQ(1)/-.0311117092106740D0,.6838519942 6D-4,
                    2       -.741449841D-6,.17972457D-7,-.727192D-9,.42201D-10,
                    3       -.3207D-11,.301D-12,-.33D-13,.4D-14/
              C
              C
ISN 0009            X=XX
ISN 0010            IF (X) 10,20,30
ISN 0011         10 X=DABS(X)
ISN 0012         30 IF (X-8.D0) 40,50,60
              C
              C  *  FOR X<8
ISN 0013         40 Y=X*X/16.D0-2.D0
ISN 0014            B1=0.D0
ISN 0015            B=0.D0
ISN 0016            DO 70 I=1,15
ISN 0017               B2=B1
ISN 0018               B1=B
ISN 0019               B=Y*B1-B2+AS(I)
ISN 0020         70 CONTINUE
ISN 0021            BESEL0=(B-B2)/2.D0
ISN 0022            GO TO 90
              C
              C  *  FOR X>8
```

```
ISN 0023      60     XPI2=.7978845608028650
ISN 0024             XPI2=XPI2/DSQRT(X)
ISN 0025             E=X-PI4
ISN 0026             X=8.D0/X
ISN 0027             Y=4.D0*X*X-2.D0
ISN 0028             P1=0.D0
ISN 0029             P=0.D0
ISN 0030             Q1=0.D0
ISN 0031             Q=0.D0
ISN 0032             DO 80 I=1,10
ISN 0033                P2=P1
ISN 0034                P1=P
ISN 0035                Q2=Q1
ISN 0036                Q1=Q
ISN 0037                P=Y*P1-P2+AP(I)
ISN 0038                Q=Y*Q1-Q2+AQ(I)
ISN 0039          80 CONTINUE
ISN 0040             BESEL0=XPI2*((P-P2)*DCOS(E)-X*(Q-Q2)*DSIN(E))/2.D0
ISN 0041             GO TO 90
                C
                C  *  X=0
ISN 0042          20 BESEL0=1.D0
ISN 0043             GO TO 90
                C
                C  *  X=8
ISN 0044          50 BESEL0=.171650807137554D0
                C
ISN 0045        90   RETURN
ISN 0046             END
```

```
:VEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
             C
             C     PROGRAM NAME: TRANMAX
             C
             C PURPOSE--
             C     COMBINE ALL THE DOUBLE INTEGRATIONS INTO FOUR 5*5 MATRICES.
             C     FROM THESE MATRICES, THE INTERCHANGE TRANSITION MATRIX CAN BE
             C     CONSTRUCTED.
             C
             C     FOR SOME OTHER MATRICES WITH DIFFERENT DIMENSIONS,
             C     ONE NEEDS A DIFFERENT PROGRAM.
             C
             C INPUT--
             C     NFILE:  UNIT NUMBER WHERE THE INPUT CAN BE READ IN.
             C     NOUT:   UNIT NUMBER WHERE THE OUTPUT CAN BE WRITTEN ON.
             C     NFUNO:  TOTAL NUMBER OF DIFFERENT DOUBLE INTEGRALS.
             C     ITEST:  =0, IF TESTING THE PROGRAM, PRINT OUT THE RESULTS.
             C             =1, OTHERWISE DO NOT PRINT OUT THE RESULTS.
             C     Z(2):   Z(1)=Z1 THE PROJECTILE NUCLEAR CHARGE.
             C             Z(2)=Z2 THE TARGET NUCLEAR CHARGE.
             C     MZ(2):  MZ(1)=MASS OF THE PROJECTILE.
             C             MZ(2)=MASS OF THE TARGET.
             C     MRATIO: RATIO OF PROTON MASS TO ELECTRON MASS.
             C     EL:     INCIDENT ENERGY IN KEV PER AMU IN LAB FRAME.
             C     V1:     INCIDENT RELATIVE VELOCITY IN A.U. OF THE
             C             PROJECTILE.
             C     BA:     IMPACT PARAMETER IN A.U._
             C     SCANGL: SCATTERING ANGLE IN DEGREE.
             C     R:      INTERNUCLEAR DISTANCE OF THE PROJECTILE AND
             C             TARGET ATOM IN A.U.
             C     RDOT:   RADIAL VELOCITY OF THE RELATIVE MOTION OF
             C             TWO NUCLEI.
             C     LAMDA:  =V1*BA/R.
             C             EQUALS THE TANGENTIAL VELOCITY AT THE
             C             CLOSEST APPROACH POINT.
             C     DBL:    A 3-DIMENSIONAL (2,2,NFUNO2) ARRAY.
             C             I=1, AND 2 POINT TO REAL AND IMAGINARY PARTS.
             C             J=1, AND 2 POINT TO (A,B) AND (B,A) COMBINATIONS.
             C             K=NFUNO2, DIFFERENT FUNCTION VALUES.
             C
             C OUTPUT--
             C     RCOUP,DCOUP,OVLAP,ACOUP,
             C     FOUR 3-DIMENSIONAL (2,2,25) ARRAY FOR RADIAL COUPLING,
             C     DIPOLE COUPING, OVERLAPPING AND ANGULAR (ROTATIONAL)
             C     COUPLING.
             C
             C     ETMK
             C     ONE 3-DIMENSIONAL (2,2,25) ARRAY FOR INTERCHANGE
             C     TRANSITION MATRIX.
             C
             C SUGGESTIONS AND HINTS --
             C     THE GENERAL ROW AND COLUMN DEFINITION OF (2,2,25) ARRAYS IS
             C
             C                        1               2
             C
```

```
C            1          REAL              REAL
C                       (A,B)             (B,A)
C
C            2       IMAGINARY         IMAGINARY
C                       (A,B)             (B,A)
C
C
C     THE 25 ELEMENTS WHICH CORRESPOND TO (5X5) MATRICES ARE
C     ARRANGED AS IN THE FOLLOWING
C
C              1S       2S       2PZ      2PX      2PY
C
C     1S        1        2        3        4        5
C     2S        6        7        8        9       10
C     2PZ      11       12       13       14       15
C     2PX      16       17       18       19       20
C     2PY      21       22       23       24       25
C
C     PROGRAMMER: TEH-HSIN K. WU
C     DATE:       JUNE, 1979
C       MODIFIED: MARCH, 1981
C
ISN 0002        REAL*8 MRATIO,EL,V1,BA,SCANGL,R,RDOT,LAMDA,B2,
               1       Z12,Z32,TWO2,C1,C2,C3,C4,C5,C6,C7,C8,
               2       DBL(2,2,57),RCOUP(2,2,25),DCOUP(2,2,25),
               3       RR,OVLAP(2,2,25),ACOUP(2,2,25),
               4       FASREL,FASIMG,ETHK(2,2,25),WT(2)
ISN 0003        INTEGER Z(2),MZ(2),S1,S2
C
ISN 0004        READ(1,1000)NFILE,NOUT,NFUNO,ITEST
ISN 0005        NFUNO2=NFUNO/2
ISN 0006        NDATA=0
C       INPUT
ISN 0007    5   CONTINUE
ISN 0008        DO 10 J=1,2
ISN 0009           READ (NFILE,1010,END=2000)Z(1),Z(2),MZ(1),MZ(2)
ISN 0010           READ(NFILE,1020)MRATIO,EL,V1,BA,SCANGL,B,RDOT,LAMDA,
               1       ((DBL(I,J,K),K=1,NFUNO2),I=1,2)
ISN 0011   10   CONTINUE
ISN 0012        NDATA=NDATA+1
C
ISN 0013        Z12=DFLOAT(Z(1)*Z(2))
ISN 0014        Z32=Z12*DSQRT(Z12)
ISN 0015        TWO2=2*DSQRT(2.D0)
ISN 0016        C1=2*Z32
ISN 0017        C2=Z32/TWO2
ISN 0018        C3=Z32/16.D0
ISN 0019        C5=Z12*C3
ISN 0020        C4=C5/2.D0
C       INITIALIZATION
ISN 0021        DO 18 I=1,2
ISN 0022           DO 16 J=1,2
ISN 0023              DO 14 K=1,25
ISN 0024                 RCOUP(I,J,K)=0.D0
ISN 0025                 DCOUP(I,J,K)=0.D0
ISN 0026                 OVLAP(I,J,K)=0.D0
```

```
ISN 0027                    ACOUP(I,J,K)=0.D0
ISN 0028      14          CONTINUE
ISN 0029      16        CONTINUE
ISN 0030      18      CONTINUE
                      C
                      C     THE COMBINATIONS ARE:
                      C     I=1,II=2  OR  I=2,II=1
                      C     J=1,JJ=2  OR  J=2,JJ=1
                      C     SO WHEN  I  INDICATES THE REAL PARTS(I.E. I=1),  II  POINTS
                      C     TO THE IMAGINARY PARTS(I.E. II=2), AND VICE VERSA.
                      C     SIMILARLY WHEN  J  POINTS TO  A  OR  (A,B),  JJ  POINTS
                      C     TO  B  OR  (B,A).
                      C     THESE INTERCHANGE PROPERTIES IS DUE TO THE SYMMETRY OF EXCHANGE
                      C     A  AND  B  .
                      C     THE FOLLOWING EQUATIONS ARE LISTED IN TABLE B-1.
                      C
ISN 0031              II=2
ISN 0032              DO 50 I=1,2
ISN 0033                 DO 40 J=1,2
ISN 0034                    IF (J-1)22,20,22
ISN 0035      20            JJ=2
ISN 0036                    GO TO 24
ISN 0037      22            JJ=1
ISN 0038      24            CONTINUE
ISN 0039                    BR=B*((-1)**JJ)
ISN 0040                    S2=1
ISN 0041                    S1=1
ISN 0042                    IF (I.EQ.1.AND.J.EQ.1)S1=-1
ISN 0044                    IF (I.EQ.2.AND.J.EQ.2)S2=-1
                      C     RADIAL COUPLING
ISN 0046                    RCOUP(I,J,1)=S2*C1*DBL(I,J,4)
ISN 0047                    RCOUP(I,J,2)=S2*C2*(2*DBL(I,J,9)-Z(JJ)*DBL(I,J,14))
ISN 0048                    RCOUP(I,J,3)=S2*C2*Z(JJ)*(DBL(I,J,23)-BR*DBL(I,J,9))
ISN 0049                    RCOUP(I,J,4)=S1*C2*Z(JJ)*DBL(II,J,41)
                      C
ISN 0050                    RCOUP(I,J,6)=S2*C2*(2*DBL(I,J,1)-Z(J)*DBL(I,J,2))
ISN 0051                    RCOUP(I,J,7)=S2*C3*(2*(2*DBL(I,J,6)-Z(J)*DBL(I,J,7)-
                     1                Z(JJ)*DBL(I,J,11))+Z12*DBL(I,J,12))
ISN 0052                    RCOUP(I,J,8)=S2*C3*Z(JJ)*(2*(DBL(I,J,20)-BR*DBL(I,J,6))
                     1               +Z(J)*(BR*DBL(I,J,7)-DBL(I,J,21)))
ISN 0053                    RCOUP(I,J,9)=C3*Z(JJ)*(2*DBL(II,J,38)-Z(J)*DBL(II,J,39))
                     1               *S1
                      C
ISN 0054                    RCOUP(I,J,11)=S2*C2*Z(J)*DBL(I,J,16)
ISN 0055                    RCOUP(I,J,12)=S2*C3*Z(J)*(2*DBL(I,J,20)-Z(JJ)*DBL(I,J,25))
ISN 0056                    RCOUP(I,J,13)=S2*C5*(DBL(I,J,30)-BR*DBL(I,J,20))
ISN 0057                    RCOUP(I,J,14)=S1*C5*DBL(II,J,44)
                      C
ISN 0058                    RCOUP(I,J,16)=S1*C2*Z(J)*DBL(II,J,36)
ISN 0059                    RCOUP(I,J,17)=C3*Z(J)*(2*DBL(II,J,38)-
                     1              Z(JJ)*DBL(II,J,48))*S1
ISN 0060                    RCOUP(I,J,18)=S1*C5*(DBL(II,J,44)-BR*DBL(II,J,38))
ISN 0061                    RCOUP(I,J,19)=S2*C4*DBL(I,J,52)
ISN 0062                    RCOUP(I,J,25)=S2*C4*DBL(I,J,55)
                      C
                      C     DIPOLE COUPLING
```

```
ISN 0063                DCOUP(I,J,1)=S2*C1*DBL(I,JJ,19)
ISN 0064                DCOUP(I,J,2)=S2*C2*(2*DBL(I,JJ,17)-Z(JJ)*DBL(I,JJ,18))
ISN 0065                DCOUP(I,J,3)=S2*C2*Z(JJ)*DBL(I,JJ,29)
ISN 0066                DCOUP(I,J,4)=S1*C2*Z(JJ)*DBL(II,JJ,43)
                C
ISN 0067                DCOUP(I,J,6)=S2*C2*(2*DBL(I,JJ,24)-Z(J)*DBL(I,JJ,28))
ISN 0068                DCOUP(I,J,7)=C3*(2*(2*DBL(I,JJ,21)-Z(J)*DBL(I,JJ,26)
                1             -Z(JJ)*DBL(I,JJ,22))+Z12*DBL(I,JJ,27))*S2
ISN 0069                DCOUP(I,J,8)=C3*Z(JJ)*(2*DBL(I,JJ,31)-Z(J)*DBL(I,JJ,34))
                1             *S2
ISN 0070                DCOUP(I,J,9)=C3*Z(JJ)*(2*DBL(II,JJ,45)-
                1             Z(J)*DBL(II,JJ,50))*S1
                C
ISN 0071                DCOUP(I,J,11)=S2*C2*Z(J)*(RR*DBL(I,JJ,24)+DBL(I,JJ,33))
ISN 0072                DCOUP(I,J,12)=C3*Z(J)*(2*(RR*DBL(I,JJ,21)+DBL(I,JJ,31))
                1             -Z(JJ)*(RR*DBL(I,JJ,22)+DBL(I,JJ,32)))*S2
ISN 0073                DCOUP(I,J,13)=S2*C5*(DBL(I,JJ,35)+RR*DBL(I,JJ,31))
ISN 0074                DCOUP(I,J,14)=C5*(RR*DBL(II,JJ,45)+DBL(II,JJ,51))
                1             *S1
                C
ISN 0075                DCOUP(I,J,16)=S1*C2*Z(J)*DBL(II,JJ,47)
ISN 0076                DCOUP(I,J,17)=C3*Z(J)*(2*DBL(II,JJ,45)-
                1             Z(JJ)*DBL(II,JJ,46))*S1
ISN 0077                DCOUP(I,J,18)=S1*C5*DBL(II,JJ,51)
ISN 0078                DCOUP(I,J,19)=S2*C4*DBL(I,JJ,54)
ISN 0079                DCOUP(I,J,25)=S2*C4*DBL(I,JJ,57)
            C  *  OVERLAPPING
ISN 0080                OVLAP(I,J,1)=S2*C1*DBL(I,JJ,5)
ISN 0081                OVLAP(I,J,2)=S2*C2*(2*DBL(I,JJ,2)-Z(JJ)*DBL(I,JJ,3))
ISN 0082                OVLAP(I,J,3)=S2*C2*Z(JJ)*DBL(I,JJ,17)
ISN 0083                OVLAP(I,J,4)=S1*C2*Z(JJ)*DBL(II,JJ,37)
ISN 0084                OVLAP(I,J,6)=C2*(2*DBL(I,JJ,10)-Z(J)*DBL(I,JJ,15))
                1             *S2
ISN 0085                OVLAP(I,J,7)=C3*(2*(2*DBL(I,JJ,7)-Z(J)*DBL(I,JJ,12)
                1             -Z(JJ)*DBL(I,JJ,8))+Z12*DBL(I,JJ,13))*S2
ISN 0086                OVLAP(I,J,8)=C3*(2*DBL(I,JJ,21)-Z(J)*DBL(I,JJ,26))
                1             *S2*Z(JJ)
ISN 0087                OVLAP(I,J,9)=C3*Z(JJ)*(2*DBL(II,JJ,39)
                1             -Z(J)*DBL(II,JJ,49))*S1
ISN 0088                OVLAP(I,J,11)=C2*Z(J)*(RR*DBL(I,JJ,10)+DBL(I,JJ,24))
                1             *S2
ISN 0089                OVLAP(I,J,12)=C3*Z(J)*(2*(RR*DBL(I,JJ,7)+DBL(I,JJ,21))
                1             -Z(JJ)*(RR*DBL(I,JJ,8)+DBL(I,JJ,22)))*S2
ISN 0090                OVLAP(I,J,13)=S2*C5*(RR*DBL(I,JJ,21)+DBL(I,JJ,31))
ISN 0091                OVLAP(I,J,14)=C5*(RR*DBL(II,JJ,39)+DBL(II,JJ,45))
                1             *S1
ISN 0092                OVLAP(I,J,16)=S1*C2*Z(J)*DBL(II,JJ,42)
ISN 0093                OVLAP(I,J,17)=S1*C3*Z(J)*(2*DBL(II,JJ,39)
                1             -Z(JJ)*DBL(II,JJ,40))
ISN 0094                OVLAP(I,J,18)=S1*C5*DBL(II,JJ,45)
ISN 0095                OVLAP(I,J,19)=S2*C4*DBL(I,JJ,53)
ISN 0096                OVLAP(I,J,25)=S2*C4*DBL(I,JJ,56)
                C
ISN 0097                ACOUP(I,J,3)=S2*(-C2)*Z(JJ)*DBL(I,JJ,37)
ISN 0098                ACOUP(I,J,8)=S2*(-C3)*Z(JJ)*(2*DBL(I,JJ,39)-
                1             Z(J)*DBL(I,JJ,49))
```

```
ISN 0099                ACOUP(I,J,13)=S2*(-C5)*(RR*DBL(I,JJ,39)+DBL(I,JJ,45))
ISN 0100                ACOUP(I,J,18)=S1*C4*DBL(II,JJ,53)
ISN 0101                ACOUP(I,J,4)=S1*(-C2)*Z(JJ)*DBL(II,JJ,17)
ISN 0102                ACOUP(I,J,9)=S1*(-C3)*Z(JJ)*(2*DBL(II,JJ,21)-
               1              Z(J)*DBL(II,JJ,26))
ISN 0103                ACOUP(I,J,14)=(-C5)*(RR*DBL(II,JJ,21)+DBL(II,JJ,31))
               1              *S1
ISN 0104                ACOUP(I,J,19)=S2*C5*DBL(I,JJ,45)
ISN 0105      40        CONTINUE
ISN 0106                II=1
ISN 0107      50        CONTINUE
                        C
                        C
                        C     THESE DOUBLE INTEGRATIONS (I.E. RADIAL,
                        C     DIPOLE,OVERLAPPING AND ANGULAR COUPLING)
                        C     HAVE AN ADDITIONAL PHASE, EXP(I*R*RDOT/2).
                        C     FASREL AND FASIMG ARE THE REAL AND THE
                        C     IMAGINARY PARTS OF THE PHASE FACTOR.
                        C
ISN 0108      100       CONTINUE
ISN 0109                FASIMG=R*RDOT/2.D0
ISN 0110                FASREL=DCOS(FASIMG)
ISN 0111                FASIMG=DSIN(FASIMG)
ISN 0112                WRITE(3,1020)FASREL,FASIMG
                        C
ISN 0113                II=2
ISN 0114                DO 130 I=1,2
ISN 0115                   DO 120 J=1,2
ISN 0116                      DO 110 K=1,25
ISN 0117                         RCOUP(I,J,K)=RCOUP(I,J,K)*FASREL
               1                      -((-1)**I)*ECOUP(II,J,K)*FASIMG
ISN 0118                         DCOUP(I,J,K)=DCOUP(I,J,K)*FASREL
               1                      +((-1)**I)*DCOUP(II,J,K)*FASIMG
ISN 0119                         OVLAP(I,J,K)=OVLAP(I,J,K)*FASREL
               1                      +((-1)**I)*OVLAP(II,J,K)*FASIMG
ISN 0120                         ACOUP(I,J,K)=ACOUP(I,J,K)*FASREL
               1                      +((-1)**I)*ACOUP(II,J,K)*FASIMG
ISN 0121      110               CONTINUE
ISN 0122      120            CONTINUE
ISN 0123                     II=1
ISN 0124      130     CONTINUE
                      C
                      C     CHANGE SIGN FOR SOME MATRIX ELEMENTS
                      C
ISN 0125              DO 131 I=1,2
ISN 0126                 RCOUP(I,2,4)=-RCOUP(I,2,4)
ISN 0127                 RCOUP(I,2,9)=-RCOUP(I,2,9)
ISN 0128                 RCOUP(I,2,14)=-RCOUP(I,2,14)
ISN 0129                 RCOUP(I,2,16)=-RCOUP(I,2,16)
ISN 0130                 RCOUP(I,2,17)=-RCOUP(I,2,17)
ISN 0131                 RCOUP(I,2,18)=-RCOUP(I,2,18)
                      C
ISN 0132                 DCOUP(I,2,4)=-DCOUP(I,2,4)
ISN 0133                 DCOUP(I,2,9)=-DCOUP(I,2,9)
ISN 0134                 DCOUP(I,2,14)=-DCOUP(I,2,14)
ISN 0135                 DCOUP(I,2,16)=-DCOUP(I,2,16)
```

```
ISN 0136                DCOUP(I,2,17)=-DCOUP(I,2,17)
ISN 0137                DCOUP(I,2,18)=-DCOUP(I,2,18)
                C
ISN 0138                CVLAP(I,2,4)=-OVLAP(I,2,4)
ISN 0139                OVLAP(I,2,9)=-OVLAP(I,2,9)
ISN 0140                CVLAP(I,2,14)=-OVLAP(I,2,14)
ISN 0141                CVLAP(I,2,16)=-OVLAP(I,2,16)
ISN 0142                CVLAP(I,2,17)=-OVLAP(I,2,17)
ISN 0143                CVLAP(I,2,18)=-OVLAP(I,2,18)
                C
ISN 0144                ACOUP(I,2,3)=-ACOUP(I,2,3)
ISN 0145                ACOUP(I,2,8)=-ACOUP(I,2,8)
ISN 0146                ACOUP(I,2,13)=-ACCUP(I,2,13)
ISN 0147                ACOUP(I,2,19)=-ACOUP(I,2,19)
ISN 0148          131   CONTINUE
                  C
                  C     EVALUATE THE INTERCHANGE MATRIX ELEMENTS.
                  C
ISN 0149                B2=B*B
ISN 0150                C6=Z12/B2
ISN 0151                WT(1)=WZ(1)*WRATIO
ISN 0152                WT(2)=WZ(2)*WRATIO
ISN 0153                C7=Z12/(B*2.D0*(WT(1)+WT(2)))
ISN 0154                C8=V1*BA/B2
                C
ISN 0155                IF (ITEST.EQ.1) GO TO 138
ISN 0157                WRITE(3,1130)C6,C7,C8,WT(1),WT(2),B2
ISN 0158                DO 136 I=1,2
ISN 0159                DO 134 J=1,2
ISN 0160                DO 132 K=1,25
ISN 0161                   WRITE(3,1030)BCOUP(I,J,K),DCOUP(I,J,K),
                       1                OVLAP(I,J,K),ACOUP(I,J,K)
ISN 0162          132   CONTINUE
ISN 0163          134   CONTINUE
ISN 0164          136   CONTINUE
                  C
ISN 0165          138   CONTINUE
ISN 0166                JJ=2
ISN 0167                DO 160 J=1,2
ISN 0168                   DO 150 I=1,2
ISN 0169                      DO 140 K=1,25
ISN 0170                         ETMK(I,J,K)=-Z(J)*BCOUP(I,J,K)+C6*DCOUP(I,J,K)/WT(JJ)
                       1                +C7*WT(J)*OVLAP(I,J,K)/WT(JJ)-C8*ACOUP(I,J,K)
ISN 0171          140         CONTINUE
ISN 0172          150      CONTINUE
ISN 0173                   JJ=1
ISN 0174          160   CONTINUE
                  C
                  C     PRINT OUT THE INTERCHANGE TRANSITION MATRIX ELEMENT
                  C     AND OVERLAPPING MATRIX ELEMENTS.
                  C
ISN 0175                WRITE(3,1040)
ISN 0176                WRITE(3,1010) NDATA,NOUT
ISN 0177                IF (NDATA.EQ.1) WRITE(NOUT,1020) EL,V1,BA,SCANGI
ISN 0179                WRITE(NOUT,1020) B,RDOT,(((ETMK(I,J,K),OVLAP(I,J,K),
                       1    K=1,25),I=1,2),J=1,2)
```

```
ISN 0180            GO TO 5
                C
ISN 0181        1000  FORMAT(4I5)
ISN 0182        1010  FORMAT(2I5)
ISN 0183        1020  FORMAT(2D25.16)
ISN 0184        1030  FORMAT(4D20.8)
ISN 0185        1040  FORMAT(/33H    INTERCHANGE TRANSITION MATRIX ,/)
ISN 0186        1130  FORMAT(/' COEFFICIENTS:'/1X,6D15.4)
                C
ISN 0187        2000  STOP
                C
ISN 0188            END
```

```
21.8 ( JUN 74 )                    OS/360  FORTRAN H

   COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                      SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
        C     PROGRAM: DCOUPL
        C
        C PURPOSE:
        C     Use the precalculated matrix elements to solve the 16-coupled
        C     differential equations as in equation (4.11)
        C
        C METHOD/STRATEGY:
        C     This program calls DGEAR, a IMSL routine to solve the coupled-
        C     differential equations.  The method used is the variable order
        C     ADAMS predictor-corrector method or GEARS method.
        C
        C INPUT:
        C     EL:      Incident energy in KEV/AMU in the lab frame
        C     Z1,Z2:   Projectile and target charge
        C     AM1,AM2: Projectile and target mass in proton (approximately
        C              AMU)
        C     NDATA:   number of data points of the matrix elements
        C     NINPUT:  unit number of the file where the matrix elements are stored
        C     NREAD:   Unit number for the backup file which stores the
        C              intermediate integrated values.
        C     V1:      Incident velocity in A.U.
        C     BA:      Impact parameter in A.U.
        C     SCANGL:  Scattering angle in degrees
        C     R(22):   Array of internuclear distance
        C     RDOT(22):Array of the time derivatives of the internuclear distance
        C     ETKREL, ETKIMG:
        C               (16,2,22) arrays of the real and imaginary parts of the
        C               exchange matrix elements
        C     OVLREL, OVLIMG:
        C              (16,2,22) arrays of the real and imaginary parts of the
        C              overlap integrals
        C     METH, MITER:
        C              parameters necessary for DGEAR subroutine, refer to
        C              IMSL package for detail information
        C     ND:      number of coupled differential equations
        C     INDT:    indicator for the postive or negative time
        C               =-1, negative time
        C               =1,  positive time
        C     KMAX:    maximum steps allowed in one run
        C     INITIL:  an indicator to determine where to read the intial data
        C              =0, read from cards
        C              =1, read from backup disk file (unit number = NREAD)
        C     RMAX:    Maximum R(T) where the time integration will stop
        C     DELR:    increment of R(T) for each integration step
        C     DELT:     increment of PHI angle in radians for each integration step
        C     TOL:      required tolerance
        C     H:        step size required by DGEAR subroutine, refer to IMSL
        C               package for detail information
        C     RDIV:     the internuclear distance at which the independent
        C               (integration) parameter R or PHI are exchanged.
        C     RI(22):   array contains the logorithmic of R(22)
        C     IBT:      an indicator
        C               =0, R, the internuclear distance, is the integraton parameter
        C               =1, PHI angle is the integration parameter
```

```
C     XSTR,XEND:input parameter for DGEAB subroutine
C
C OUTPUT:
C     AY:       (16) array contains the values of the a vector,
C               equation (4.12), this is the result for the coupled
C               differential equations
C     AYSQR:    (8) array contains the squares of AY elements
C     SUM:      the sum of the squires of AY elements
C     PRBREL:   real parts of the probability
C     PRBIMG:   imaginary parts of the probability
C
C SUBROUTINE CALLED:
C     DGEAR (IMSL package)
C     PROBAB
C
ISN 0002      REAL*8 EL,V1,BA,SCANGL,B(22),RDCT(22),ETKREL(16,2,22),
             1       ETKIMG(16,2,22),DUM,OVLREL(16,2,22),EL(22),
             2       OVLIMG(16,2,22),P1KREL(2,22),P1KIMG(2,22),MRATIO,
             3       P1OREL(2,22),P1OIMG(2,22)
ISN 0003      REAL*8 AY(16),DUMMY(48),RMAX,XSTR,XEND,H,
             1       DELR,DELT,RDIV,TOL,WK(273),
             2       E0,E,MU,B0,BA2,C,RSTR,PI,SUM,AYSQR(8),
             3       PRBREL,PRBIMG
C
ISN 0004      REAL SDUMMY(4)
C
ISN 0005      INTEGER ND,METH,MITER,INDEX,IWK(2),INDT,IRT,IDUMMY(38),
             1        NREAD,IDATE(2),ITIME(2),INITIL
C
ISN 0006      INTEGER Z1,Z2,MZ1,MZ2
C
ISN 0007      REAL*8 Z12,Z22,Z124,Z224,EAB1,EBA1,EAB2,EBA2,EAA,EBB
C
ISN 0008      COMMON /RLOG/EL
             1       /CONST1/Z1,Z2,MZ1,MZ2
             2       /CONST2/MRATIO,V1,BA,B0,E,C,BA2,MU
             3       /CONST3/NDATA
             4       /ETMTRX/ETKREL,ETKIMG,OVLREL,OVLIMG
             5       /RNUC/R
             6       /FLAG/INDT,IRT
             7       /GEAR/DUMMY,SDUMMY,IDUMMY
             8       /ENERGY/EAA,EBB,EAB1,EAB2,EBA1,EBA2
C
ISN 0009      EXTERNAL FCN,FCNJ
C
ISN 0010      DATA E0/0.27211608D+2/,
             1     PI/3.14159265358979D0/
C
ISN 0011      MRATIO=1.836 15152D+3
C
ISN 0012      READ (1,1030)Z1,Z2,MZ1,MZ2
C
ISN 0013      READ(1,1000)NDATA,NINPUT,NREAD
C
C     Read the exchange and overlap transition matrix elements
C     from disk file
```

```
                C
ISN 0014            READ(NINPUT,1020)EL,V1,BA,SCANGL
ISN 0015            DO 50 I=1,NDATA
ISN 0016               READ(NINPUT,1020)R(I),RDOT(I)
ISN 0017               DO 40 J=1,2
ISN 0018                  KKK=0
ISN 0019                  DO 20 K=1,4
ISN 0020                     DO 10 KK=1,4
ISN 0021                        KKK=KKK+1
ISN 0022                        READ(NINPUT,1020)ETKREL(KKK,J,I),CVLREL(KKK,J,I)
ISN 0023     10                 CONTINUE
ISN 0024                        READ(NINPUT,1010)DUM
ISN 0025     20                 CONTINUE
ISN 0026                        DO 25 M=1,4
ISN 0027                           READ(NINPUT,1010)DUM
ISN 0028     25                 CONTINUE
ISN 0029                        READ(NINPUT,1020)P1KREL(J,I),P1OREL(J,I)
ISN 0030                        KKK=0
ISN 0031                        DO 35 K=1,4
ISN 0032                           DO 30 KK=1,4
ISN 0033                              KKK=KKK+1
ISN 0034                              READ(NINPUT,1020)ETKIMG(KKK,J,I),OVLIMG(KKK,J,I)
ISN 0035     30                    CONTINUE
ISN 0036                           READ(NINPUT,1010)DUM
ISN 0037     35                 CONTINUE
ISN 0038                        DO 38 M=1,4
ISN 0039     38                    READ(NINPUT,1010)DUM
ISN 0040                        READ(NINPUT,1020)P1KIMG(J,I),P1OIMG(J,I)
ISN 0041     40             CONTINUE
ISN 0042     50         CONTINUE
                C
                C
                C     CHANGE THE INTERNUCLEAR DISTANCE INTO LOG SCALE.
ISN 0043            DO 100 I=1,NDATA
ISN 0044               RL(I)=DLOG(R(I))
ISN 0045     100    CONTINUE
                C     INITIALIZATION
                C
ISN 0046            Z12=DFLOAT(Z1*Z1)
ISN 0047            Z22=DFLOAT(Z2*Z2)
ISN 0048            Z124=Z12/4.D0
ISN 0049            Z224=Z22/4.D0
ISN 0050            EAB1=(Z12-Z224)*0.5D00
ISN 0051            EAB2=(Z124-Z22)*0.5D00
ISN 0052            EBA1=(Z22-Z124)*0.5D00
ISN 0053            EBA2=(Z224-Z12)*0.5D00
ISN 0054            EAA=Z12*3.D0/8.D0
ISN 0055            EBB=Z22*3.D0/8.D0
ISN 0056            E=(1.0D+3)*EL/E0
ISN 0057            MU=DFLOAT(MZ1*MZ2)/DFLOAT(MZ1+MZ2)
ISN 0058            E=MU*E
                C
ISN 0059            M0=MU*MRATIO
ISN 0060            B0=DFLOAT(Z1*Z2)/(2*E)
ISN 0061            BA2=BA*BA
ISN 0062            C=DSQRT(BA2+B0*B0)
```

```
ISN 0063            RSTB=(1.D0-2.D0*SCANGL/180.D0)*0.5D0*PI
ISN 0064            WRITE(3,1040)RSTB,C,BA2,B0
                    C
                    C     READ INPUTS TO SOLVE DIFFERENTIAL EQUATIONS
                    C
ISN 0065            READ(1,1030)METH,MITER,ND,INDT,KMAX,INITIL
ISN 0066            WRITE(3,1030)METH,MITER,ND,INDT,KMAX,INITIL
ISN 0067            READ(1,1040)RMAX,DELR,DELT,TOL,H,RDIV
ISN 0068            WRITE(3,1040)RMAX,DELR,DELT,TOL,H,RDIV
                    C
                    C     CHECK IF THE INITIAL VALUES ARE READ FROM DISK
                    C
ISN 0069            IF (INITIL.EQ.0)GO TO 110
ISN 0071            READ(NREAD)IRT,XSTR,AY
ISN 0072            GO TO 120
ISN 0073      110   READ (1,1045)IRT
ISN 0074            READ (1,1040)XSTR,(AY(II),II=1,16)
ISN 0075      120   WRITE(3,1050)IRT,XSTR,(AY(II),II=1,16)
                    C
                    C     FIRST CALL TO DGEAR
                    C
ISN 0076            INDEX=1
ISN 0077            XEND=XSTR+DELR
ISN 0078            IF (IRT.NE.0)XEND=XSTR+DELT
                    C
ISN 0080            DO 200 K=1,KMAX
                    C
ISN 0081                CALL DGEAR(ND,FCN,FCNJ,XSTR,H,AY,XEND,TOL,
                   1                METH,MITER,INDEX,IWK,WK,IER)
                    C
                    C       Evaluate (a star)*a +(b star)*b
                    C
ISN 0082                SUM=0.D0
ISN 0083                DO 135 I=1,4
ISN 0084                   J=I+4
ISN 0085                   II=I+8
ISN 0086                   JJ=II+4
ISN 0087                   AYSQR(I)=AY(I)**2+AY(J)**2
ISN 0088                   AYSQR(J)=AY(II)**2+AY(JJ)**2
ISN 0089                   SUM=SUM+AYSQR(I)+AYSQR(J)
ISN 0090      135       CONTINUE
ISN 0091                WRITE(3,1055)IRT,XSTR,(AY(II),AY(II+8),AYSQR(II),II=1,8)
ISN 0092                WRITE(3,1060)H,XEND,SUM
ISN 0093                INDT=-1
ISN 0094                IF (XSTR.GT.0.D0)INDT=1
                    C
                    C       Evaluate the overlapping probability
                    C
ISN 0096                CALL PROBAB(XSTR,AY,PRBREL,PRBIMG)
ISN 0097                PRBREL=SUM+PRBREL
ISN 0098                WRITE(3,1065)PRBREL,PRBIMG
                    C
                    C       CHECK IF THE INTEGRATION REACHES THE TRASACTION
                    C       POINT.
                    C
ISN 0099                IF (IRT.NE.0) GO TO 130
```

```
                C
                C       R IS THE INDEPENDENT VARIABLE OF THE COUPLED
                C       DIFFERENTIAL EQUATIONS
                C
ISN 0101                RSTR=DABS(XSTR)
ISN 0102                IF (RSTR.GT.RDIV) GO TO 150
ISN 0104                IRT=1
ISN 0105                WRITE (3,1100)
ISN 0106                XSTR=INDT*DARCOS((BA2/RSTR+B0)/C)
ISN 0107                H=DELT
ISN 0108                INDEX=1
ISN 0109                GO TO 150
                C
                C       PHI IS THE INDEPENDENT VARIABLE OF THE COUPLED
                C       DIFFERENTIAL EQUATIONS
                C
ISN 0110        130     RSTR=BA2/(-B0+C*DCOS(XSTR))
ISN 0111                IF (RSTR.LT.RDIV) GO TO 150
ISN 0113                IRT=0
ISN 0114                WRITE(3,1110)
ISN 0115                XSTR=RSTR
ISN 0116                H=DELR
ISN 0117                INDEX=1
                C
ISN 0118        150     REWIND NREAD
ISN 0119                WRITE (NREAD)IRT,XSTR,AY
ISN 0120                ENDFILE NREAD
ISN 0121                WRITE(3,1070)IRT,INDT,XSTR,RSTR
ISN 0122                WRITE(3,1120)DUMMY(8),IDUMMY(6),IDUMMY(7),IDUMMY(8)
ISN 0123                XEND=XSTR+H
                C
ISN 0124                IF (RSTR.GT.RMAX.AND.INDT.GT.0) GO TO 2000
ISN 0126                IF (IER.GT.128) GO TO 2000
                C
ISN 0128        200     CONTINUE
                C
                C
                C
ISN 0129        1000    FORMAT(3I5)
ISN 0130        1010    FORMAT(D25.16)
ISN 0131        1020    FORMAT(2D25.16)
ISN 0132        1030    FORMAT(2I3)
ISN 0133        1040    FORMAT(D10.3)
ISN 0134        1045    FORMAT(I5)
ISN 0135        1050    FORMAT(I2,D15.6/16(2X,D15.6/))
ISN 0136        1055    FORMAT(I2,D15.6/
                       1        6X,'EXCHANGE',13X,'DIRECT',14X,'SQUARE'/
                       2        8(2X,3(D15.6,5X)/))
ISN 0137        1060    FORMAT(/' H= ',D15.8,' XEND= ',D15.8,' SUM= ',D15.8)
ISN 0138        1065    FORMAT(/' PROBABILITY : REAL= ',D15.6,5X,'IMAGINARY= ',D15.6)
ISN 0139        1070    FORMAT(/' IRT=',I3,'  INDT=',I3,'  XSTR=',D15.8,
                       1     '  RSTR=',D15.8)
ISN 0140        1100    FORMAT(/' ** The independent variable has changed',
                       1        ' from R to PHI **')
ISN 0141        1110    FORMAT(/' ** The independent variable has changed',
                       1        ' from PHI to R **')
```

```
ISN 0142     1120  FORMAT(/' ** The step size H last used successfully =',D25.8/
                   1       ' ** The order last used successfully= ',I10/
                   2       ' ** The cumulative number of steps taken = ',I10/
                   3       ' ** The cumulative number of FCN evaluations = ',I10)
                  C
ISN 0143     2000  STOP
ISN 0144           END
```

```
:VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
              C
              C
ISN 0002            SUBROUTINE PROBAB(X,Y,PRBREL,PRBIMG)
              C
ISN 0003            REAL*8 X,Y(16),PRBREL,PRBIMG
ISN 0004            REAL*8 R,BSTRA1(16),BSTRA2(16),ASTRB1(16),ASTRB2(16),
                   1       YR(22),YI(22),FRMFAC
              C
ISN 0005            COMMON /RLOG/RL
ISN 0006            REAL*8 RL(22)
              C
ISN 0007            COMMON /CONST3/NDATA
ISN 0008            INTEGER NDATA
              C
ISN 0009            COMMON /ETMTRX/ETKREL,ETKIMG,OVLREL,OVLIMG
ISN 0010            REAL*8 ETKREL(16,2,22),ETKIMG(16,2,22),OVLREL(16,2,22),
                   1       OVLIMG(16,2,22)
              C
ISN 0011            COMMON /FLAG/INDT,IRT
ISN 0012            INTEGER INDT,IRT
              C
ISN 0013            COMMON /ENERGY/EAA,EBB,EAB1,EAB2,EBA1,EBA2
ISN 0014            REAL*8 EAA,EBB,EAB1,EAB2,EBA1,EBA2
              C
ISN 0015            COMMON /PHASE/RELAB1,IMGAB1,RELBA1,IMGBA1,
                   1              RELAB2,IMGAB2,RELBA2,IMGBA2,
                   2              SI,SR
ISN 0016            REAL*8 RELAB1,IMGAB1,RELBA1,IMGBA1,RELAB2,IMGAB2,RELBA2,
                   1       IMGBA2,SI(2,16),SR(2,16)
              C
              C
              C     INITIALIZATION
              C
ISN 0017            CALL TIMENG(X,R)
              C
              C     Interpolate Smn(A,B) and Smn(B,A), both imaginary and real
              C
ISN 0018            DO 30 J=1,2
ISN 0019               DO 20 II=1,16
ISN 0020                  DO 10 I=1,NDATA
ISN 0021                     YR(I)=OVLREL(II,J,I)
ISN 0022                     YI(I)=OVLIMG(II,J,I)
ISN 0023     10           CONTINUE
ISN 0024                  SR(J,II)=FRMFAC(NDATA,RL,YR,R,10,10.D0)
ISN 0025                  SI(J,II)=FRMFAC(NDATA,RL,YI,R,10,10.D0)
ISN 0026     20         CONTINUE
ISN 0027     30      CONTINUE
              C
              C     REDEFINE SR AND SI, CONSIDERING THE PHASE FACTORS.
              C
ISN 0028            CALL EPHASE(INDT)
              C
              C     Evaluate (b Star)*a
              C
```

```
ISN 0029            K=0
ISN 0030            DO 50 J=9,12
ISN 0031               DO 40 I=1,4
ISN 0032                  K=K+1
ISN 0033                  II=I+4
ISN 0034                  JJ=J+4
ISN 0035                  BSTRA1(K)=Y(I)*Y(J)+Y(II)*Y(JJ)
ISN 0036                  BSTRA2(K)=Y(II)*Y(J)-Y(JJ)*Y(I)
ISN 0037      40       CONTINUE
ISN 0038      50    CONTINUE
              C
              C     Evaluate (a Star)*b
              C
ISN 0039            K=0
ISN 0040            DO 70 I=1,4
ISN 0041               DO 60 J=9,12
ISN 0042                  K=K+1
ISN 0043                  II=I+4
ISN 0044                  JJ=J+4
ISN 0045                  ASTRB1(K)=Y(I)*Y(J)+Y(II)*Y(JJ)
ISN 0046                  ASTRB2(K)=Y(I)*Y(JJ)-Y(II)*Y(J)
ISN 0047      60       CONTINUE
ISN 0048      70    CONTINUE
              C
              C
              C     Evaluate real parts and imaginary parts
              C
ISN 0049            PRBREL=0.D0
ISN 0050            PRBIMG=0.D0
ISN 0051            DO 80 I=1,16
ISN 0052               PRBREL=PRBREL+ASTRB1(I)*SR(1,I)-ASTRB2(I)*SI(1,I)
                   1                +BSTRA1(I)*SR(2,I)-BSTRA2(I)*SI(2,I)
ISN 0053               PRBIMG=PRBIMG+ASTRB1(I)*SI(1,I)+ASTRB2(I)*SR(1,I)
                   1                +BSTRA2(I)*SR(2,I)+BSTRA1(I)*SI(2,I)
ISN 0054      80    CONTINUE
              C
              C
ISN 0055            RETURN
ISN 0056            END
```

```
!VEL 21.8 ( JUN 74 )                          CS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE FCN(N,X,Y,YPRIME)
                    C
                    C PURPOSE:
                    C     this program provides the first derivatives (YPRIME(1),...
                    C     YPRIME(N)) of the Y(I) with respect to X. Please refer to
                    C     Section 4.2 for details.
                    C
                    C CALLING SEQUENCE:
                    C
                    C   INPUT:
                    C     N:       # of coupled differential equations
                    C     X:       independent variable, at which the first derivative
                    C              should be evaluated.
                    C     Y:       a N vector, a dependent variable
                    C
                    C   OUTPUT:
                    C     YPRIME: a N vector, which is the first derivative of Y repect
                    C             to X.
                    C
                    C SUGGESTION/HINT:
                    C     The real and imaginary parts of the matrix elements are stored
                    C     in two different arrays, SR and SI.  Each array has dimension
                    C     (2,16), the row number indicates (A,B)(I=1) and (E,A)(I=2)
                    C     combinations.
                    C
                    C CALLED BY:  DGEAR
                    C
                    C SUBROUTINES AND FUNCTION(S) CALLED BY THIS PROGRAM:
                    C
                    C     TIMENG
                    C     FRMFAC (REAL*8 FUNCTION)
                    C     EPHASE
                    C     DIBTBN
                    C     LINV2F (IMSL ROUTINE)
                    C     VBULFF (IMSL ROUTINE)
                    C
                    C PROGRAMMER: TEH-HSIN K. HU
                    C DATE:       MAR. 1980
                    C
ISN 0003            REAL*8 X,Y(16),YPRIME(16),RL(22),YR(22),YI(22),R,
                   1        ETKREL(16,2,22),ETKIMG(16,2,22),OVLREL(16,2,22),
                   2        OVLIMG(16,2,22),OVELAP(16,16),INVOVE(16,16),
                   3        WKAREA(304),SI(2,16),SR(2,16),DTHREL(2,16),
                   4        DTHIMG(2,16),HAMP(16,16),TRNAMP(16,16),FRMFAC
ISN 0004            REAL*8 MBATIC,V1,BA,BO,E,C,BA2,MU,VEL
ISN 0005            REAL*8 EAA,EBB,EAB1,EAE2,EBA1,EBA2
ISN 0006            REAL*8 RELAB1,IMGAB1,RELBA1,IMGBA1,RELAB2,IMGAB2,
                   1       RELBA2,IMGBA2,RELEAA,IMGEAA,RELEEB,IMGEEB
ISN 0007            REAL*8 REL,IMG
ISN 0008            INTEGER N,NDATA,INDI,IBT,Z1,Z2,MZ1,MZ2
                    C
ISN 0009            COMMON /BLOG/BL
                   1       /CONST3/NDATA
                   2       /ETMTRX/ETKREL,ETKIMG,OVLREL,OVLIMG
```

```
                     3       /DTMTRX/DTHREL,DTHIMG
                     4       /FLAG/INDT,IRT
                     5       /CONST2/MRATIO,V1,EA,B0,E,C,BA2,MU
                     6       /CONST1/Z1,Z2,M21,MZ2
                     7       /ENERGY/EAA,EBB,EAB1,EAB2,EBA1,EBA2
                     8       /PHASE/RELAB1,IMGAB1,RELBA1,IMGBA1,
                     9              RELAB2,IMGAB2,RELBA2,IMGBA2,
                     A              SI,SR
ISN 0010              COMMON /DIRFAS/RELEAA,IMGEAA,EELEBB,IMGEBB
                    C
                    C
                    C     INITIALIZATION
ISN 0011              CALL TIMENG(X,R)
                    C
                    C     Interpolate Smn(A,B) and Smn(B,A), both imaginary and real
                    C
ISN 0012              DO 30 J=1,2
ISN 0013                 DO 20 II=1,16
ISN 0014                    DO 10 I=1,NDATA
ISN 0015                       YR(I)=OVLREL(II,J,I)
ISN 0016                       YI(I)=OVLIMG(II,J,I)
ISN 0017         10         CONTINUE
ISN 0018                    SR(J,II)=FRMFAC(NDATA,RL,YR,R,10,10.D0)
ISN 0019                    SI(J,II)=FRMFAC(NDATA,RL,YI,R,10,10.D0)
ISN 0020         20       CONTINUE
ISN 0021         30     CONTINUE
                    C
                    C     REDEFINE SR AND SI, CONSIDERING THE PHASE FACTORS.
                    C
ISN 0022              CALL EPHASE(INDT)
                    C
                    C     Initially set OVRLAP zero.
                    C
ISN 0023              DO 50 I=1,16
ISN 0024                 DO 40 J=1,16
ISN 0025                    OVRLAP(I,J)=0.D0
ISN 0026         40         CONTINUE
ISN 0027         50     CONTINUE
                    C
                    C     Construct the nonzero upper-right part of OVRLAP
                    C
ISN 0028              II=0
ISN 0029              DO 70 KK=1,4
ISN 0030                 LL=KK+4
ISN 0031                 DO 60 K=9,12
ISN 0032                    L=K+4
ISN 0033                    II=II+1
ISN 0034                    OVRLAP(KK,K)=-SI(1,II)
ISN 0035                    OVRLAP(LL,L)=-SI(1,II)
ISN 0036                    OVRLAP(LL,K)=SR(1,II)
ISN 0037                    OVRLAP(KK,L)=-SR(1,II)
ISN 0038         60         CONTINUE
ISN 0039         70     CONTINUE
                    C
                    C      Construct the lower-left part of OVRLAP
                    C
```

```
ISN 0040            II=0
ISN 0041            DO 90 KK=9,12
ISN 0042               LL=KK+4
ISN 0043               DO 80 K=1,4
ISN 0044                  L=K+4
ISN 0045                  II=II+1
ISN 0046                  OVRLAP(KK,K)=-SI(2,II)
ISN 0047                  OVRLAP(KK,L)=-SR(2,II)
ISN 0048                  OVRLAP(LL,K)=SR(2,II)
ISN 0049                  OVRLAP(LL,L)=-SI(2,II)
ISN 0050      80       CONTINUE
ISN 0051      90    CONTINUE
              C
ISN 0052            DO 100 K=1,4
ISN 0053               KK=K+4
ISN 0054               OVRLAP(K,KK)=-1.D0
ISN 0055               OVRLAP(KK,K)=1.D0
ISN 0056      100   CONTINUE
              C
ISN 0057            DO 110 K=9,12
ISN 0058               KK=K+4
ISN 0059               OVRLAP(K,KK)=-1.D0
ISN 0060               OVRLAP(KK,K)=1.D0
ISN 0061      110   CONTINUE
              C
              C     Find the inversion of OVRLAP
              C
ISN 0062            CALL LINV2F(OVRLAP,16,16,INVOVR,6,WKAREA,IER)
              C
              C     Interpolate Kmn(A,B) and Kmn(B,A)
              C
ISN 0063            DO 140 J=1,2
ISN 0064               DO 130 II=1,16
ISN 0065                  DO 120 I=1,NDATA
ISN 0066                     YR(I)=ETKREL(II,J,I)
ISN 0067                     YI(I)=ETKIMG(II,J,I)
ISN 0068      120            CONTINUE
ISN 0069                     SR(J,II)=FRMFAC(NDATA,RL,YR,R,10,10.D0)
ISN 0070                     SI(J,II)=FRMFAC(NDATA,RL,YI,R,10,10.D0)
ISN 0071      130         CONTINUE
ISN 0072      140   CONTINUE
              C
              C     Redefine SI and SR, considering the phase factors.
              C
ISN 0073            CALL EPHASE(INDT)
              C
              C     Calculate the direct transition matrix elements,
              C     Hmn(A) and Hmn(B), these matrices will pass in
              C     through the common block DTMTRX
              C
ISN 0074            CALL DIRTRN(R)
              C
              C     Redefine real and imaginary parts of the direct transition
              C     matrix elements.
              C
ISN 0075            DO 142 I=2,4
```

```
ISN 0076              REL=DTHREL(1,I)
ISN 0077              IMG=DTHIMG(1,I)
ISN 0078              DTHREL(1,I)=REL*RELEAA-IMG*IMGEAA
ISN 0079              DTHIMG(1,I)=REL*IMGEAA+IMG*RELEAA
ISN 0080              REL=DTHREL(2,I)
ISN 0081              IMG=DTHIMG(2,I)
ISN 0082              DTHREL(2,I)=REL*RELEBB-IMG*IMGEBB
ISN 0083              DTHIMG(2,I)=REL*IMGEBB+IMG*RELEBB
ISN 0084     142   CONTINUE
             C
ISN 0085           DO 144 I=5,13,4
ISN 0086              REL=DTHREL(1,I)
ISN 0087              IMG=DTHIMG(1,I)
ISN 0088              DTHREL(1,I)=REL*RELEAA-IMG*(-IMGEAA)
ISN 0089              DTHIMG(1,I)=REL*(-IMGEAA)+IMG*RELEAA
ISN 0090              REL=DTHREL(2,I)
ISN 0091              IMG=DTHIMG(2,I)
ISN 0092              DTHREL(2,I)=REL*RELEBB-IMG*(-IMGEBB)
ISN 0093              DTHIMG(2,I)=REL*(-IMGEBB)+IMG*RELEBB
ISN 0094     144   CONTINUE
             C
             C     Rearrange Hmn(A), Hmn(B), Kmn(A,B) and Kmn(B,A)
             C
ISN 0095           MM=0
ISN 0096           DO 160 K=1,4
ISN 0097              L=K+4
ISN 0098              I=L+4
ISN 0099              J=I+4
ISN 0100              DO 150 KK=1,4
ISN 0101                 LL=KK+4
ISN 0102                 II=LL+4
ISN 0103                 JJ=II+4
ISN 0104                 MM=MM+1
ISN 0105                 HAMP(K,KK)=DTHREL(1,MM)
ISN 0106                 HAMP(K,LL)=-DTHIMG(1,MM)
ISN 0107                 HAMP(K,II)=SR(1,MM)
ISN 0108                 HAMP(K,JJ)=-SI(1,MM)
ISN 0109                 HAMP(L,KK)=DTHIMG(1,MM)
ISN 0110                 HAMP(L,LL)=DTHREL(1,MM)
ISN 0111                 HAMP(L,II)=SI(1,MM)
ISN 0112                 HAMP(L,JJ)=SR(1,MM)
ISN 0113                 HAMP(I,KK)=SR(2,MM)
ISN 0114                 HAMP(I,LL)=-SI(2,MM)
ISN 0115                 HAMP(I,JJ)=-DTHIMG(2,MM)
ISN 0116                 HAMP(I,II)=DTHREL(2,MM)
ISN 0117                 HAMP(J,KK)=SI(2,MM)
ISN 0118                 HAMP(J,LL)=SR(2,MM)
ISN 0119                 HAMP(J,II)=DTHIMG(2,MM)
ISN 0120                 HAMP(J,JJ)= DTHREL(2,MM)
ISN 0121     150      CONTINUE
ISN 0122     160   CONTINUE
             C
             C     Multiply (INVOVR*HAMP)
             C
ISN 0123           CALL VMULFF(INVOVR,HAMP,16,16,16,16,16,TRNAMP,16,IER)
             C
```

```
ISN 0124            VEL=V1*BA/R
ISN 0125            IF (IRT.EQ.0) GO TO 163
                C
                C     THETA is the independent variable
                C
ISN 0127            VEL=VEL/R
ISN 0128            GO TO 165
                C
                C     R is the independent varaible
                C
ISN 0129      163   VEL=DSQRT(2.D0*(E-DFLOAT(Z1*Z2)/R)/MU-VEL**2)
ISN 0130            VEL=VEL*INDT
ISN 0131      165   CONTINUE
ISN 0132            WRITE(3,1030) VEL
ISN 0133      1030  FORMAT(' VEL = ',D25.16)
                C
                C     Calculate YPRIME
                C
ISN 0134            DO 180 I=1,N
ISN 0135               YPRIME(I)=0.D0
ISN 0136               DO 170 J=1,N
ISN 0137                  YPRIME(I)=YPRIME(I)+TRNAMP(I,J)*Y(J)
ISN 0138      170      CONTINUE
ISN 0139               YPRIME(I)=YPRIME(I)/VEL
ISN 0140      180   CONTINUE
                C
ISN 0141            RETURN
                C
ISN 0142            END
```

```
:VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE FCNJ (N,X,Y,PD)
ISN 0003            INTEGER N
ISN 0004            REAL*8 Y(N),PD(N,N),X
ISN 0005            RETURN
ISN 0006            END
```

```
:VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE EPHASE(INDT)
                    C
                    C PURPOSE:
                    C     When the exchange matrix elements were evaluated in TRANMAX, it
                    C     did not consider time factor.  Since t<0, RDOT<0 and t>0, RDOT>0,
                    C     an adjustment of sign according to the time range is required.
                    C     EPHASE first handles the above adjustment.  Then the energy
                    C     phase factor which is evaluated in TIMENG and passed in
                    C     through common block is multiplied to the exchange matrix
                    C     elements.
                    C
ISN 0003            REAL*8 RELAB1,IMGAB1,RELBA1,IMGBA1,RELAB2,IMGAB2,
                   1       RELBA2,IMGBA2,SI(2,16),SR(2,16)
ISN 0004            REAL*8 REL,IMG
ISN 0005            INTEGER INDT
                    C
ISN 0006            COMMON /PHASE/RELAB1,IMGAB1,RELBA1,IMGBA1,
                   1              RELAB2,IMGAB2,RELBA2,IMGBA2,
                   2              SI,SR
                    C
                    C     Consider the translation factor EXP(iR*RDOT/2)
                    C     where RDOT has different sign for time greater
                    C     than or less than zero.
                    C
ISN 0007            DO 10 I=1,16
ISN 0008               SI(1,I)=SI(1,I)*INDT
ISN 0009               SI(2,I)=SI(2,I)*INDT
ISN 0010      10    CONTINUE
                    C
                    C     Consider the energy phase factor.
                    C
ISN 0011            DO 20 I=2,4
ISN 0012               REL=SR(1,I)
ISN 0013               IMG=SI(1,I)
ISN 0014               SR(1,I)=REL*RELAB1-IMG*IMGAB1
ISN 0015               SI(1,I)=REL*IMGAB1+IMG*RELAB1
ISN 0016               REL=SR(2,I)
ISN 0017               IMG=SI(2,I)
ISN 0018               SR(2,I)=REL*RELBA1-IMG*IMGBA1
ISN 0019               SI(2,I)=REL*IMGBA1+IMG*RELBA1
ISN 0020      20    CONTINUE
                    C
ISN 0021            DO 30 I=5,13,4
ISN 0022               REL=SR(1,I)
ISN 0023               IMG=SI(1,I)
ISN 0024               SR(1,I)=REL*RELAB2-IMG*IMGAB2
ISN 0025               SI(1,I)=REL*IMGAB2+IMG*RELAB2
ISN 0026               REL=SR(2,I)
ISN 0027               IMG=SI(2,I)
ISN 0028               SR(2,I)=REL*RELBA2-IMG*IMGBA2
ISN 0029               SI(2,I)=REL*IMGBA2+IMG*RELBA2
ISN 0030      30    CONTINUE
                    C
ISN 0031            RETURN

ISN 0032            END
```

```
!VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE RELIMG(E,T,REL,IMG)
                C
                C   This program evaluate the real and imaginary parts
                C   (REL, IMG) of the value (E*T) where E and T are
                C   energy level and time respectively.
ISN 0003            REAL*8 E,T,REL,IMG,D
                C
ISN 0004            D=E*T
ISN 0005            REL=DCOS(D)
ISN 0006            IMG=DSIN(D)
ISN 0007            RETURN
ISN 0008            END
```

```
LEVEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,KEEP
ISN 0002            DOUBLE PRECISION FUNCTION FBMFAC(NX,X,Y,XE,NN,YM)
                    C
                    C  INPUT:
                    C     NX:   number of data points.
                    C     X(NX):  the logarithmic of the R(NX) , internuclear
                    C             distances.
                    C     Y(NX):  the real scale of the matrix elements corresponds
                    C             to the different R, (R(NX) is input through the
                    C             common block.)
                    C
                    C  OUTPUT:
                    C     Cubic spline fit to (X,Y)*.
                    C     FBMFAC is the interpolated value at XE
                    C
                    C  *  There are several ways to fit the data (X,Y).
                    C     1. for X(NX/3) <= XE <= X(NX), use cubic spline fit to
                    C        (X,Y).
                    C     2. for X(1) <= XE <= X(NX/3), the data near the left end
                    C        are very large compared with YM (input) and changing
                    C        rapidly, use cubic spline fit to (X,DSQRT(R)*LOG(Y)).
                    C        Otherwise, just use the cubic spline fit to (X,Y).
                    C
ISN 0003                 REAL*8 X(NX),Y(NX),R(22),XE,YM,XL,YS(10),YL,
                        1       YAM,YMIN,SHIFT,XX(10),YY(10)
ISN 0004                 INTEGER FLAG,NX3,IEV,INTRVD,JJ
ISN 0005                 COMMON /RNUC/R
                    C
                    C     NN number of points adjacent to XE will be fitted.
                    C     find XE's position.
ISN 0006                 XL=DLOG(XE)
ISN 0007                 NX3=NX/3
ISN 0008                 IEV=INTRVD(NX,X,XL)
                    C
ISN 0009                 IF (IEV-NX3) 20,10,10
                    C
                    C     XE falls to the right part of the data.
ISN 0010            10   CALL ABAFIX(IEV,NX,X,Y,NN,XX,YY,JJ)
ISN 0011                 CALL CSPLN(XX,YY,NN,XL,FBMFAC)
ISN 0012                 RETURN
                    C
                    C     XE falls to the left 1/3 part of the data.
ISN 0013            20   IF (Y(1)) 40,30,40
ISN 0014            30   IF (DABS(Y(2)).LE.YM)GO TO 10
ISN 0016                 IF(DABS(Y(2)/Y(3)).LE.2.D0) GO TO 10
ISN 0018                 FLAG=0
ISN 0019                 GO TO 50
ISN 0020            40   IF (DABS(Y(1)).LE.YM)GO TO 10
ISN 0022                 IF (DABS(Y(1)/Y(2)).LE.2.D0)GO TO 10
ISN 0024                 FLAG=1
ISN 0025            50   CALL ABAFIX(IEV,NX,X,Y,NN,XX,YY,JJ)
ISN 0026                 DO 60 I=1,NN
ISN 0027                    IF (YY(I))60,60,70
ISN 0028            60   CONTINUE
                    C
```

```
                C     All the YY(NN) are negative, change the sign
                C     and then take the logarithmic of them.
ISN 0029              DO 80 I=1,NN
ISN 0030                 YY(I)=DABS(YY(I))
ISN 0031        80    CONTINUE
ISN 0032              IF (FLAG)100,90,100
ISN 0033        100   DO 110 I=1,NN
ISN 0034                 JI=JJ+I
ISN 0035                 YS(I)=DLOG(YY(I))*R(JI)/(1.0D0+R(JI))
ISN 0036        110   CONTINUE
ISN 0037              CALL CSPLN(XX,YS,NN,XL,YL)
ISN 0038              FBMFAC=-DEXP(YL*(1.D0+XE)/XE)
ISN 0039              RETURN
                C
ISN 0040        90    YAM=DMIN1(YY(2),YY(3))
ISN 0041              DO 120 I=4,NN
ISN 0042                 YAM=DMIN1(YAM,YY(I))
ISN 0043        120   CONTINUE
                C
ISN 0044              SHIFT=YAM/2.D0+1.0D0
ISN 0045              DO 130 I=1,NN
ISN 0046                 JI=JJ+I
ISN 0047                 YS(I)=DLOG(YY(I)+SHIFT)*R(JI)/(1.D0+R(JI))
ISN 0048        130   CONTINUE
ISN 0049              CALL CSPLN(XX,YS,NN,XL,YL)
ISN 0050              FBMFAC=-(DEXP(YL*(1.D0+XE)/XE)-SHIFT)
ISN 0051              RETURN
                C
ISN 0052        70    IF (FLAG.EQ.0)YMIN=DMIN1(YY(2),YY(3))
ISN 0054              IF(FLAG.EQ.1)YMIN=DMIN1(YY(1),YY(2),YY(3))
ISN 0056              YAM=DMIN1(DABS(YY(2)),DABS(YY(3)))
ISN 0057              DO 150 I=4,NN
ISN 0058                 YMIN=DMIN1(YY(I),YMIN)
ISN 0059                 YAM=DMIN1(DABS(YY(I)),YAM)
ISN 0060        150   CONTINUE
ISN 0061              SHIFT=DABS(YMIN)+YAM/2.D0+1.0D0
ISN 0062              DO 160 I=1,NN
ISN 0063                 JI=JJ+I
ISN 0064                 YS(I)=DLOG(YY(I)+SHIFT)*R(JI)/(1.D0+R(JI))
ISN 0065        160   CONTINUE
ISN 0066              CALL CSPLN(XX,YS,NN,XL,YL)
ISN 0067              FBMFAC=DEXP(YL*(1.D0+XE)/XE)-SHIFT
ISN 0068              RETURN
ISN 0069              END
```

```
VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
              C
              C
ISN 0002            SUBROUTINE ENDPEN(X1,X2,X3,Y1,Y2,Y3,A,B)
              C
ISN 0003            REAL*8 X1,X2,X3,Y1,Y2,Y3,A,B
ISN 0004            B=(Y1-Y2)/(X1-X2)
ISN 0005            A=(B-(Y1-Y3)/(X1-X3))/(X2-X3)
ISN 0006            B=B-A*(X1+X2)
ISN 0007            RETURN
ISN 0008            END
```

```
:VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE CSPLN(X,Y,NX,XX,YY)
                    C
                    C
ISN 0003            REAL*8 B,X(NX),Y(NX),BPAR(4),C(9,3),XX,YY,
                   1       AL,BL,AR,BR,D
ISN 0004            INTEGER INTRVD,IC,NX1,NX2
                    C
ISN 0005            IC=NX-1
                    C
                    C     FIT THE END POINTS WITH QUADRATIC POLYNOMIALS.
ISN 0006            CALL ENDPRM(X(1),X(2),X(3),Y(1),Y(2),Y(3),AL,BL)
ISN 0007            BPAR(1)=0.D0
ISN 0008            BPAR(2)=4.D0*AL
                    C
ISN 0009            NX1=NX-1
ISN 0010            NX2=NX-2
ISN 0011            CALL ENDPRM(X(NX2),X(NX1),X(NX),Y(NX2),Y(NX1),Y(NX),AR,BR)
ISN 0012            BPAR(3)=0.D0
ISN 0013            BPAR(4)=4.D0*AR
                    C
ISN 0014            CALL ICSICU(X,Y,NX,BPAR,C,IC,IER)
ISN 0015            IEVL=INTRVD(NX,X,XX)
ISN 0016            D=XX-X(IEVL)
ISN 0017            YY=((C(IEVL,3)*D+C(IEVL,2))*D+C(IEVL,1))
                   1          *D+Y(IEVL)
                    C
ISN 0018            RETURN
ISN 0019            END
```

```
!VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            INTEGER FUNCTION INTRVD(N,D,DD)
ISN 0003            REAL*8 DD,D(N)
ISN 0004            INTEGER LEFT,RIGHT
ISN 0005            INTRVD=0
ISN 0006            IF (DD.LT.D(1)) GO TO 100
ISN 0008            IF (DD.GT.D(N)) GO TO 100
ISN 0010            LEFT=1
ISN 0011            RIGHT=N
ISN 0012      10    MID=(LEFT+RIGHT)/2
ISN 0013            IF (DD-D(MID))20,30,40
ISN 0014      20    RIGHT=MID
ISN 0015            GO TO 50
ISN 0016      30    INTRVD=MID
ISN 0017            RETURN
ISN 0018      40    LEFT=MID
ISN 0019      50    LEFT1=LEFT+1
ISN 0020            IF (LEFT1.LT.RIGHT) GO TO 10
ISN 0022            INTRVD=LEFT
ISN 0023            IF (DD.EQ.D(RIGHT))INTRVD=RIGHT
ISN 0025            RETURN
ISN 0026      100   WRITE(3,1000)DD
ISN 0027      1000  FORMAT(//' ** THE VALUE IS OUT OF THE'
                   1   ,'ARRAY RANGE **',E15.6)
ISN 0028            RETURN
ISN 0029            END
```

```
VEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
              C
ISN 0002            SUBROUTINE ARAFIX(IEV,NX,X,Y,NN,XX,YY,JJ)
              C
              C     Construct the array (XX(NN),YY(NN)) pairs to
              C     do the cubic spline fit.
              C
ISN 0003            REAL*8 XX(NN),YY(NN),X(22),Y(22)
              C
ISN 0004            NL=NN/2
ISN 0005            NR=NX-NL
              C     If IEV in near the right end.
ISN 0006            IF (IEV-NR) 20,10,10
ISN 0007      10    JJ=NX-NN
ISN 0008            GO TO 50
              C
              C     If IEV is near the left end.
ISN 0009      20    IF (IEV-NL) 30,30,40
ISN 0010      30    JJ=0
ISN 0011            GO TO 50
ISN 0012      40    JJ=IEV-NL
ISN 0013      50    CONTINUE
ISN 0014            DO 60 I=1,NN
ISN 0015               XX(I)=X(JJ+I)
ISN 0016               YY(I)=Y(JJ+I)
ISN 0017      60    CONTINUE
              C
ISN 0018            RETURN
ISN 0019            END
```

```
VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
ISN 0002            SUBROUTINE DIRTRN(R)
                C
                C  PURPOSE:
                C     Construct the direct transition matrix elements.
                C
                C  INPUT:
                C     R:  Internuclear distance.
                C
                C  INPUT THROUGH COMMON BLOCK:
                C     Z1,Z2,MZ1,MZ2:
                C         Charges and masses of the projectile and target nuclei.
                C     MRATIO: Mass Ratio of proton to electron (1 a.u.).
                C     V1: Incident velocity in a.u.
                C     EA: Impact parameter in a.u.
                C
                C  OUTPUT THROUGH COMMON BLOCK:
                C     DTHREL(2,16), DTHIMG(2,16):
                C         Real and imaginary parts of the direct transition
                C         matrix elements.
                C
                C  FUNCTIONS CALLED BY THIS PROGRAM:
                C     GAMA,BETA
                C
                C     DATE:  Aug. 8, 1979
                C     PROGRAMMER: T.K. WU
                C
ISN 0003            REAL*8 Z(2),M(2),Z12,MRATIO,G(9),B(4),
                   1    V1,BA,R,DTHREL(2,16),DTHIMG(2,16),
                   2    GAMA,BETA,T1,A1,A2,A3,A4,A5,R2,DP(16),
                   3    TWOSQ,RR,B0,E,C,BA2,MU
ISN 0004            INTEGER Z1,Z2,MZ1,MZ2
ISN 0005            COMMON /CONST1/Z1,Z2,MZ1,MZ2
ISN 0006            COMMON /CONST2/MRATIO,V1,EA,B0,E,C,BA2,MU
ISN 0007            COMMON /DTMTRX/DTHREL,DTHIMG
ISN 0008            COMMON /INTNUC/RR
                C
                C
ISN 0009            RR=R
ISN 0010            Z(1)=DFLOAT(Z1)
ISN 0011            Z(2)=DFLOAT(Z2)
ISN 0012            Z12=Z(1)*Z(2)
ISN 0013            M(1)=DFLOAT(MZ1)*MRATIO
ISN 0014            M(2)=DFLOAT(MZ2)*MRATIO
                C
ISN 0015            II=2
ISN 0016            DO 100 I=1,2
                C
                C        Construct the basic formula for  DP  elements.
ISN 0017                 DO 10 J=1,5
ISN 0018                    G(J)=GAMA(J,Z(I))
ISN 0019        10       CONTINUE
                C
ISN 0020                 K=J
ISN 0021                 T1=Z(I)*1.5D0
```

```
ISN 0022                DO 20 J=1,3
ISN 0023                   JJ=J+1
ISN 0024                   G(K)=GAMA(J,T1)
ISN 0025                   B(J)=BETA(JJ,Z(I))
ISN 0026                   K=K+1
ISN 0027      20        CONTINUE
ISN 0028                B(4)=BETA(2,T1)
              C
ISN 0029                T1=2.D0*Z(I)
ISN 0030                G(9)=GAMA(1,T1)
              C
              C         Initialize all the coefficients.
              C
ISN 0031                TWOSQ=DSQRT(2.D0)*2.D0
ISN 0032                A1=Z(I)
ISN 0033                A2=A1*A1
ISN 0034                A3=A2*A1
ISN 0035                A4=A3*A1
ISN 0036                A5=A4*A1/240.D0
ISN 0037                R2=R*R
              C
ISN 0038                DO 30 J=1,16
ISN 0039                   DP(J)=0.D0
ISN 0040                   DTHREL(I,J)=0.D0
ISN 0041                   DTHIMG(I,J)=0.D0
ISN 0042      30        CONTINUE
              C
ISN 0043                DP(1)=2.D0*A3*G(9)
ISN 0044                DP(2)=A3/TWOSQ*(2.D0*G(6)-A1*G(7))
ISN 0045                DP(3)=A4/TWOSQ*(G(8)/R+R*G(6)-B(4))/3.D0
ISN 0046                DP(6)=A3*(4.D0*(G(1)-A1*G(2))+A2*G(3))/16.D0
ISN 0047                DP(7)=A4/48.D0*((-A1*G(4)+2.D0*G(3))/R+
                    1          R*(-A1*G(2)+2.D0*G(1))-2.D0*B(1)+A1*B(2))
ISN 0048                DP(11)=A5*(2.D0*(G(5)/R-B(3))/R+2.D0*R*(R*G(1)
                    1            -B(1))+7.D0*G(3))
ISN 0049                DP(16)=A5*(4.D0*G(3)+(B(3)-G(5)/R)/R+R*
                    1            (B(1)-R*G(1)))
ISN 0050                IF (I.EQ.1)GO TO 40
ISN 0052                   DP(3)=-DP(3)
ISN 0053                   DP(7)=-DP(7)
ISN 0054      40        A1=-Z(II)
ISN 0055                A2=-Z12/(W(I)*R2)
ISN 0056                A3=Z12*W(II)/(W(I)*(W(1)+W(2))*R*2.D0)
ISN 0057                A4=V1*BA/R2
              C
ISN 0058                DTHREL(I,1)=A1*DP(1)+A3
ISN 0059                DTHREL(I,2)=A1*DP(2)
ISN 0060                DTHREL(I,3)=A1*DP(3)+A2*TWOSQ*64.D0/(2.43D2*Z(I))
ISN 0061                DTHREL(I,6)=A1*DP(6)+A3
ISN 0062                DTHREL(I,7)=A1*DP(7)+A2*(-3.D0/Z(I))
ISN 0063                DTHREL(I,11)=A1*DP(11)+A3
ISN 0064                DTHREL(I,16)=A1*DP(16)+A3
ISN 0065                DTHREL(I,5)=DTHREL(I,2)
ISN 0066                DTHREL(I,10)=DTHREL(I,7)
ISN 0067                DTHREL(I,9)=DTHREL(I,3)
              C
```

```
ISN 0068                DTHIMG(I,12)=A4
ISN 0069                DTHIMG(I,15)=-A4
                C
ISN 0070                II=1
                C
ISN 0071        100     CONTINUE
                C
ISN 0072                RETURN
ISN 0073                END
```

```
:VEL 21.8 ( JUN 74 )                        OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
                C
                C
ISN 0002                DOUBLE PRECISION FUNCTION BETA(N,B)
                C
                C       Evaluate a series expansion.
                C
ISN 0003                REAL*8 B,R,BR,FAC,S,S1
ISN 0004                INTEGER N1,N
ISN 0005                COMMON /INTNUC/B
                C
ISN 0006                N1=N+1
ISN 0007                BR=B*R
ISN 0008                IF (N) 20,10,20
ISN 0009        10      BETA=1.D0+DEXP(-BR)/BR
ISN 0010                GO TO 40
ISN 0011        20      S=DFLOAT(N1)
ISN 0012                S1=1.D0
ISN 0013                DO 30 M=1,N
ISN 0014                   S1=S1*BR/DFLOAT(M)
ISN 0015                   S=S+S1*DFLOAT(N1-M)
ISN 0016        30      CONTINUE
ISN 0017                BETA=1.D0+DEXP(-BR)*S/BR
ISN 0018        40      FAC=1.D0/B
ISN 0019                DO 50 I=1,N
ISN 0020        50         FAC=FAC*DFLOAT(I)/B
ISN 0021                BETA=2.D0*BETA*FAC
ISN 0022                RETURN
ISN 0023                END
```

```
:VEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XREF
              C
              C
ISN 0002            DOUBLE PRECISION FUNCTION GAMA(N,G)
              C
              C     Evaluate a series expansion.
              C
ISN 0003            REAL*8 G,B,GB,S,S1,S2,FAC
ISN 0004            INTEGER N,N1,M,I
ISN 0005            COMMON /INTNUC/B
              C
ISN 0006            N1=N-1
ISN 0007            GB=G*B
ISN 0008            S=DFLOAT(N)
ISN 0009            S1=1.D0
ISN 0010            IF (N.EQ.1)GO TO 20
ISN 0012            DO 10 M=1,N1
ISN 0013               S1=S1*GB/DFLOAT(M)
ISN 0014               S=S+S1*(N-M)/DFLOAT(M+1)
ISN 0015      10    CONTINUE
ISN 0016      20    S2=DFLOAT(N+1)/GB
ISN 0017            S=S2-(S+S2)*DEXP(-GB)
ISN 0018            FAC=1.D0/G
ISN 0019            DO 30 I=1,N
ISN 0020      30       FAC=FAC*I/G
ISN 0021            GAMA=2.D0*FAC*S
ISN 0022            RETURN
ISN 0023            END
```

```
LEVEL 21.8 ( JUN 74 )                          OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,XEEP
ISN 0002            SUBROUTINE TIMENG(X,R)
                    C
                    C PURPOSE:
                    C     TIMENG calculates time by given internuclear distance and
                    C     evaluates the real and imaginary parts of the energy phase
                    C     factor  exp(i(Em-En)t).
                    C
                    C INPUT:
                    C     X:   an independent varaible, it may be internuclear distance
                    C          or the angle PHI.
                    C
                    C OUTPUT THROUGH COMMON BLOCKS:
                    C     /PHASE/
                    C     /DIRFAC/
                    C
                    C INPUT THROUGH COMMON BLOCKS:
                    C     /FLAG/
                    C     /CONST2/
                    C     /ENERGY/
                    C
                    C PROGRAMMER: TEH-HSIN K. HU
                    C DATE:       MAR. 1980
                    C
ISN 0003            REAL*8 X,R
ISN 0004            REAL*8 THETA,TANGEN,TIME
                    C
ISN 0005            COMMON /FLAG/INDT,IRT
ISN 0006            INTEGER INDT,IRT
                    C
ISN 0007            COMMON /CONST2/MRATIO,V1,BA,B0,E,C,BA2,MU
ISN 0008            REAL*8 MRATIO,V1,BA,B0,E,C,BA2,MU
                    C
ISN 0009            COMMON /ENERGY/EAA,EBB,EAB1,EAB2,EBA1,EBA2
ISN 0010            REAL*8 EAA,EBB,EAB1,EAB2,EBA1,EBA2
                    C
ISN 0011            COMMON /PHASE/RELAB1,IMGAB1,RELBA1,IMGBA1,
                   1              RELAB2,IMGAB2,RELBA2,IMGBA2,
                   2              SI,SR
ISN 0012            REAL*8 RELAB1,IMGAB1,RELBA1,IMGBA1,RELAB2,IMGAB2,RELBA2,
                   1       IMGBA2,SI(2,16),SR(2,16)
                    C
ISN 0013            COMMON /DIRFAS/RELEAA,IMGEAA,RELEBB,IMGEBB
ISN 0014            REAL*8 RELEAA,IMGEAA,RELEEB,IMGEBB
                    C
                    C
ISN 0015            INDT=-1
ISN 0016            IF (X.GT.0.D0)INDT=1
ISN 0018            R=DABS(X)
ISN 0019            IF (IRT.NE.0)R=BA2/(-B0+C*DCOS(X))
                    C     CALCULATE TIME
ISN 0021            THETA=X
ISN 0022            IF (IRT.EQ.0)THETA=INDT*DARCOS((BA2/R+B0)/C)
ISN 0024            TANGEN=DTAN(THETA/2.D0)*BA
ISN 0025            TIME=C-B0
```

```
ISN 0026            TIME=(B*C*DSIN(THETA)/BA+
                   1      BO*DLOG(-(TANGEN+TIME)/(TANGEN-TIME)))
ISN 0027            TIME=TIME/V1
ISN 0028            WRITE(3,1020)R,THETA,TANGEN,TIME
ISN 0029      1020  FORMAT(/' R=',D20.10,' THETA=',D20.10,' TANGEN=',
                   1       D20.10,' TIME=',D20.10)
              C
              C     CALCULATE REAL AND IMAGINARY PARTS OF THE ENERGY PHASE FACTOR
ISN 0030            CALL RELIMG(EAB1,TIME,RELAB1,IMGAB1)
ISN 0031            CALL RELIMG(EAB2,TIME,RELAB2,IMGAB2)
ISN 0032            CALL RELIMG(EBA1,TIME,RELBA1,IMGBA1)
ISN 0033            CALL RELIMG(EBA2,TIME,RELBA2,IMGBA2)
ISN 0034            CALL RELIMG(EAA,TIME,RELEAA,IMGEAA)
ISN 0035            CALL RELIMG(EBB,TIME,RELEBB,IMGEBB)
              C
ISN 0036            RETURN
ISN 0037            END
```

```
//ELECAP JOB UNC.P.Snnnn,WU.T-H,P=50,T=30,DEST=UNC,PRTY=9
//*PW=xxxxx
//   EXEC FTHGD,R.G=190K
//G.DECK DD DSN=UNC.P.S2158.WUTH.HE2H.ELECAP,DISP=OLD
//G.FT05F001 DD DSN=UNC.P.S6502.WUTH.HE2H.BACKUP,DISP=OLD
//G.FT07F001 DD DSN=UNC.P.S6502.WUTH.HE2H.DBLINT,DISP=OLD
//G.SYSIN DD *
    1     5     7     6
    2     1
    4     1
   42   114
  163     1
    3     1
    2     1
    3     1
    2     1
    3     1
    2     1
    3     2
    1     3
    3     2
    2     2
    3     2
    1     3
    1     4
    0     0
    3     3
    1     4
    1     4
    0     0
    1     5
    0     0
    2     2
    0     0
    3     2
```

2	2
1	4
0	0
3	3
1	4
2	2
0	0
1	4
0	0
1	5
0	0
2	3
0	0
1	5
0	0
2	3
0	0
1	5
0	0
1	1
0	1
1	1
0	1
1	1
0	1
0	1
1	1
1	1
0	1
0	1
0	1
0	1
0	1
0	1

```
    0     1
          1.0D+2          5.0D+0
           0.5D0          1.0D00
          1.0D-5          5.2D00
          1.0D+2          5.0D+0
          1.0D00          0.5D00
          1.0D-5          5.2D00
//
```

This JCL will produce the double integrals for R=5.2 a.u. . The results will be stored on disk 'UNC.P.S6502.WUTH.HE2H.DBLINT'.

```
//TRANMAX JOB UNC.P.Snnnn,WU.T-H,T=3,P=100,DEST=UNC,
//  PRTY=0
//*PW=xxxxx
//  EXEC FTHGD,R.G=200K
//G.DECK DD DSN=UNC.P.S6502.WUTH.HE2H.TRANOBJ,DISP=OLD
//G.FT05F001 DD DSN=UNC.P.S2158.WUTH.HE2H.TRANMAX,DISP=OLD
//G.SYSIN DD *
    1    5  114    1
(22 data sets of the double integral results from ELECAP)
//
```

```
//DCOUPL JOB UNC.P.Snnnn,WU.T-H,P=200,T=10,PRTY=0,DEST=UNC
//*PW=xxxxx
//G EXEC FTHGD,R.G=300K
//G.DECK DD DSN=UNC.P.S2158.WUTH.HE2H.DGEAR,DISP=OLD
//G.FT05F001 DD DSN=UNC.P.S2158.WUTH.HE2H.TRANMAX,DISP=OLD
//G.FT07F001 DD DSN=UNC.P.S2158.WUTH.HE2H.DIFF,DISP=OLD
//G.SYSIN DD *
  2  1
  4  1
   22    5    7
  1  0
 16 -1
500  1
      6.00
     .0050
     .0050
    1.0E-6
    1.0E-5
    1.0D-2
    0
     -6.D0
        0.
        0.
        0.
        0.
        0.
        0.
        0.
        0.
        0.
       1.0
        0.
        0.
        0.
        0.
        0.
        0.
        0.
//
```