SAND75-0622
Unlimited Release

ION IMPLANTATION RANGE AND ENERGY DEPOSITION CODES COREL, RASE4, AND DAMG2

D. K. Brice

Prepared by Sandia Laboratories, Albuquerque, New Mexico 87115 and Livermore, California 94550 for the United States Energy Research and Development Administration under Contract AT (29-1)-789

July 1977

Sandia Laboratories

SF 2900 Q(7-73)

When printing a copy of any digitized SAND Report, you are required to update the markings to current standards.

Issued by Sandia Laboratories, operated for the United States Energy Research and Development Administration by Sandia Corporation

---

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22151
Price: Printed Copy $6.50; Microfiche $2.25

ION IMPLANTATION RANGE AND ENERGY DEPOSITION CODES COREL, RASE4, AND DAMG2

D. K. Brice
Ion-Solid Interactions, Division 5111
Sandia Laboratories, Albuquerque, New Mexico 87115

ABSTRACT

The FORTRAN codes COREL, RASE4 and DAMG2 can be used to calculate quantities associated with ion implantation range and energy deposition distributions within an amorphous target, or for ions incident far from low index directions and planes in crystalline targets. RASE4 calculates the projected range, R_p, the root mean square spread in the projected range, ΔR_p, and the root mean square spread of the distribution perpendicular to the projected range $\Delta R_\perp$. These parameters are calculated as a function of incident ion energy, E, and the instantaneous energy of the ion, E'. They are sufficient to determine the three dimensional spatial distribution of the ions in the target in the Gaussian approximation when the depth distribution is independent of the lateral distribution. RASE4 can perform these calculations for targets having up to four different component atomic species. In addition, it calculates the rates at which energy is deposited into atomic motions and displacements (atomic processes), into electronic excitation and ionization (electronic processes), as well as the effects of target atom recoils in redistributing the deposited energy. The code COREL is a short, economical version of RASE4 which calculates the range and

straggling variables for E' = 0. Its primary use in the present package is to provide the average range and straggling variables for recoiling target atoms which are created by the incident ion. This information is used by RASE4 in calculating the redistribution of deposited energy by the target atom recoils. The code DAMG2 uses the output from RASE4 to calculate the depth distribution of energy deposition into either atomic processes or electronic processes. With other input DAMG2 can be used to calculate the depth distribution of any energy dependent interaction between the incident ions and target atoms. This report documents the basic theory behind COREL, RASE4 and DAMG2, including a description of codes, listings, and complete instructions for using the codes, and their limitations. The codes are written in FØRTRAN, Version 2.3 for CDC 6000 series computer systems, to be compiled and executed by FUN CØMPILER, SCOPE 3.3-SCM VER324 31 07/30/75. Copies of these codes are available from the Argonne Code Center, Argonne National Laboratory, as well as from the NEA Computer Programme Library in Ispra, Italy for member requesters.

This page intentionally left blank

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	9
II.	COREL	12
	A. General Equations	12
	B. Solutions to the Equations	15
	C. Scattering Cross Sections and Stopping Power	17
	D. User's Guide	18
	i. General	18
	ii. Input	20
	iii. Output	25
	E. Brief Programmer's Guide	27
	i. Methods of Integration	27
	ii. Main Program Variables	28
	iii. Subroutines	29
	F. Sample Output	30
III.	RASE4	33
	A. General Equations	33
	B. Compound Targets	35
	C. User's Guide	37
	i. General	37
	ii. Input	38
	iii. Output	42
	D. Brief Programmer's Guide	43
	i. General	43
	ii. Subroutines	44

Table of Contents (Continued)

	Page
E. Sample Output	44
IV. DAMG2	52
A. General Equations	52
B. User's Guide	53
i. General	53
ii. Input	55
iii. Output	58
iv. Subroutines	61
C. Brief Programmer's Guide	62
D. Sample Output	62
V. MAIN PROGRAM LISTINGS	68
A. COREL	68
B. RASE4	78
C. DAMG2	92
VI. LISTINGS OF SUBROUTINES	103
REFERENCES	110

List of Figures

Figure 1	Data block structure written on TAPE10 by COREL	25
Figure 2	Data block structure written on TAPE20 by RASE4	42
Figure 3	Data block structure written on TAPE30 by DAMG2	59

List of Tables

		Page
Table I	Values of $g(E,E')$ and $h(\psi)$ for R, R_p, R_c^2 and R_r^2.	15
Table II	Values of r and s^2 for the functions S_D, S_I and P.	53

This page intentionally left blank

I. INTRODUCTION

Over the past decade ion beams have found increased application in the study and alteration of the near-surface region of solids. Charged atomic projectiles (ions) with energies of tens or hundreds of keV will penetrate into target material from a few hundred angstroms (for heavier ions) up to a few microns (for lighter ions). The resultant implantation of the ions, along with the deposition and distribution within the target material of their initial kinetic energy, provides a useful and precisely controllable tool for altering the properties of the target in the near-surface region. Among the uses of ion implantation are the implantation doping of semiconductors, creation of near-surface alloys and compounds, sputter etching of surfaces, enhancement of diffusion rates in the implanted region, the creation and study of structural damage produced in crystalline or polycrystalline targets, and the simulation of neutron produced structural damage in nuclear reactor structural materials.

During the last six or seven years research workers in Division 5111 have been investigating the physics of ion implantation processes. By combining experimental measurement with the development of appropriate theoretical models an increasingly comprehensive understanding of ion-solid interactions has been obtained. The computer codes COREL, RASE4 and DAMG2 have evolved from this interplay of theory and experiment. Through these codes some of the theoretical quantities of interest in such studies can be readily evaluated for

comparison with experimental measurement, or as an aid in designing future experiments. The purpose of this report is to provide a detailed description of these codes so that they may be used by other groups who may have an interest in one or more of the quantities which are evaluated. In a previous report earlier versions of RASE4 (then named RASE3) and DAMG2 were documented.[1] In the intervening two and a half years, however, new theoretical developments,[2] as well as improvements in the codes themselves, have increased the power and applicability of these routines. Thus, it is necessary that these two codes be brought up to date along with the initial presentation of the code COREL.

The three codes should be considered as a single package which provides a variety of information concerning the spatial distribution of the incident ions in a target material, as well as the distribution of energy deposited into atomic displacements and motions (atomic processes) and into electronic excitation and ionization (electronic processes). In addition, the output of each individual code in the sequence COREL, RASE4 and DAMG2 is normally required as input for the calculations of the next code in the sequence.

The basic purpose of COREL in this sequence is to provide information concerning recoiling target atoms which are created through collisions by the incident ions. In particular, this code provides, as a function of recoil atom initial kinetic energy, the average distance traveled by the atom parallel to its initial direction of motion before it comes to rest, R_p,

(average projected range), the standard deviation in recoil projected ranges, ΔR_p the standard deviation in transverse displacement of the recoils from the initial direction of motion, $\Delta R_\perp$, and the average distance actually traveled by a recoil atom along its trajectory before coming to rest, R. COREL can also be used to obtain these range and straggling parameters for foreign atoms moving in the target material. In this respect COREL can be considered as a truncated, economical version of RASE4 which performs only these particular calculations.

RASE4 calculates range and straggling parameters for the incident ions. In addition to calculating these parameters for the ions after they have come to rest in the target RASE4 also evaluates them at a number of intermediate energies. Thus, all four of these parameters (Rp, ΔRp, $\Delta R_\perp$ and R) are presented as functions of two variables, the initial ion kinetic energy E and instantaneous ion kinetic energy E'. This code also provides the results of averaging certain quantities of interest over the elastic and inelastic scattering cross sections. In this latter category RASE4 gives as a function of instantaneous ion energy the inelastic contribution to the ion stopping power, the average range of the target atom recoils projected onto the ion direction of motion, the standard deviation in these particular recoil projected ranges and the partitioning of the elastic contribution to the stopping power into that fraction which will ultimately be deposited into atomic processes and that fraction which will ultimately be deposited into electronic processes.

Finally, the last code in the sequence, DAMG2, takes as input the results from RASE4, and calculates the distribution in depth of the energy deposited into electronic processes or atomic processes. These calculations are carried out as a function of incident ion energy, and provide for redistribution of the deposited energy by the recoiling target atoms. In addition, if other quantities of interest are produced through the interaction of the ions and target atoms, e.g., nuclear reaction products, x rays, etc., DAMG2 can also calculate the depth distribution of this production. In this latter case, however, the relevant averages over the production cross section must be obtained from external sources for input into DAMG2.

The physical theory which underlies the calculations of this code package have been presented in great detail in the literature.[2,3] In the present report the mathematical expressions to be evaluated will be emphasized at the expense of the physics of the ion implantation process. This report is meant primarily to provide that information which is necessary for other users to apply the three codes to their own research programs, and to maintain and alter the codes as their needs may require.

II. COREL

A. General Equations

Consider an atomic projectile with mass M_1, atomic number Z_1, and kinetic energy, E, incident on a solid target composed of atoms of mass M_2 and atomic number Z_2. The incident particle

loses its energy through interactions with the electrons and atomic cores of the target, and E' represents its instantaneous energy as it is slowing to a stop. The range and straggling parameters of interest are 1) the average distance a projectile will have traveled along its trajectory when its energy is E', R(E,E'); 2) the average distance it will have traveled in its original direction of motion (average projected range), $R_p(E,E')$; 3) the standard deviation in R_p, $\Delta R_p(E,E')$; and 4) the standard deviation in the transverse displacement from its initial trajectory, $\Delta R\ (E,E')$.

These parameters are sufficient to construct the spatial distribution of the ions at energy E' in the gaussian approximation. In particular, if $P_3(E,E',x,y,z)dxdydz$ is the probability of finding the ion in the volume dxdydz centered on (x,y,z) after its energy has been reduced to E' then approximately

$$P_3(E,E',x,y,z) = \frac{2^{-1/2}\pi^{-3/2}}{\Delta R_p \Delta R_\perp^{\ 2}} \exp\left\{ - \frac{(x - R_p)^2}{2\Delta R_p^{\ 2}} - \frac{(y^2 + z^2)}{\Delta R_\perp^{\ 2}} \right\} \qquad (1)$$

In (1) the cartesian co-ordinate system (x,y,z) has been chosen with the x-axis parellel to the initial direction of motion of the incident ion, and the origin is located at the target surface.

Let the parameters $R_c^2(E,E')$ and $R_r^2(E,E')$ be defined by

$$R_c^2 = R_p^2 + (\Delta R_\perp)^2 + (\Delta R_p)^2 \quad , \qquad (2a)$$

and

$$R_r^2 = R_p^2 - \tfrac{1}{2}(\Delta R_\perp)^2 + (\Delta R_p)^2 \quad . \qquad (2b)$$

For an amorphous target, or for incidence far from low index crystallographic axes and planes in crystalline targets, it can be shown[3] that R, R_p, R_c^2 and R_r^2 all satisfy differential equations of the same general form. In particular, if X(E,E') stands for R, R_p, R_c^2 or R_r^2, then

$$g(E,E') = \sum_{n=o}^{\infty} \beta_n(E,E')\ d^n X(E,E')/dE^n \tag{3}$$

where the source function, g(E,E'), and the coefficients, $\beta_n(E,E')$ are well defined functions of the energy variables, which are characteristic of the specific parameter represented by X.

The coefficients β_n are given by

$$\beta_o(E,E') = N\left\{\int_o^{Tm} \sigma(E,T)\ dT - \int_o^{T_1} h(\psi)\ \sigma(E,T)\ dT\right\} , \tag{4}$$

$$\beta_1(E,E') = S(E) + N\int_o^{T_1} T\ h(\psi)\ \sigma(E,T)\ dT , \tag{5}$$

and

$$\beta_n(E,E') = (-1)^{n+1}\ (N/n!)\int_o^{T_1} T^n\ h(\psi)\ \sigma(E,T)dT , \tag{6}$$

for n = 2,3, ... In Eqs. (4) through (6), N is the target atomic density, $\sigma(E,T)$ is the differential cross section for an incident particle with energy E to transfer energy T to a recoiling target atom in an elastic scattering event, S(E) is the electronic contribution to the stopping power of the target for the incident particle, and the upper limits on the integrals are given by

$$T_m = 4\, M_1\, M_2\, E \,/\, (M_1 + M_2)^2 \quad , \tag{7a}$$

and

$$T_1 = \text{minimum of } (T_m,\ E - E') \quad . \tag{7b}$$

The elastic energy exchange described by $\sigma(E,T)$ results in a deflection of the incident ion's trajectory through a laboratory angle ψ, where

$$\text{Cos}\ \psi = \left\{ 1 - \frac{(1 + \mu)}{2} \frac{T}{E} \right\} \Big/ \left\{ 1 - \frac{T}{E} \right\}^{1/2} \tag{8a}$$

with

$$\mu = M_2/M_1 \quad . \tag{8b}$$

The function $h(\psi)$ depends on which of the four parameters is to be represented by X. The functions $g(E,E')$ and $h(\psi)$ are given in Table I.

TABLE I

Values of $g(E,E')$ and $h(\psi)$ for R, R_p, R_c^2 and R_r^2.

X(E,E')	g(E,E')	h(ψ)
R(E,E')	1	1
$R_p(E,E')$	1	Cos ψ
$R_c^2(E,E')$	$2\ R_p(E,E')$	1
$R_r^2(E,E')$	$2\ R_p(E,E')$	$1 - (3/2)\ \text{Sin}^2\psi$

From the values of $g(E,E')$ in Table I it is seen that solutions to the equation for R_p must be obtained before the equations for R_r^2 and R_c^2 can be solved.

The expressions above all refer to monatomic targets. The results can be extended to polyatomic targets by summing the

β_n values obtained from Eqs. (4) through (6) for each individual atomic species present in the target. The resultant sum then replaces β_n in the differential equation, Eq. (3). COREL is designed to accept targets with up to four different atomic components.

B. Solutions to the Equations

First order solutions to Eq. (3) are represented by $X_1(E,E')$ where X_1 satisfies

$$g(E,E') = \beta_o(E,E')\, X_1(E,E') + \beta_1(E,E')\, d\, X_1(E,E')/dE \ . \quad (9)$$

This equation has the formal solution

$$X_1(E,E') = \int_{E'}^{E} [g(y,E')/\beta_1(y,E')]\ W(y,E,E')\ dy \quad , \quad (10)$$

where

$$W(y,E,E') = \exp\left\{ - \int_{y}^{E} [\beta_o(x,E')/\beta_1(x,E')]\ dx \right\} \ . \quad (11)$$

A second order correction to X_1 can be obtained by writing

$$X = X_1 + \Delta_1$$

and substituting this expression for X into Eq. (3). This substitution results in

$$- \beta_2(E,E')d^2X_1(E,E')/dE^2 = \beta_o(E,E')\Delta_1(E,E')$$

$$+ \beta_1(E,E')\ d\Delta_1(E,E')/dE + \text{terms of higher order in } X_1 \text{ and } \Delta_1. \quad (12)$$

Let $Y_2(E,E')$ be a solution to Eq. (12) when the terms of higher order are dropped. The approximate solution to Eq. (3) for X through second order is then given by $X_2(E,E')$ where

$$X_2 = X_1 + Y_2 \quad . \tag{13}$$

Note that the truncated version of Eq. (12) is formally the same as Eq. (9), the only change being that the source function $g(E,E')$ is replaced by the left hand side of Eq. (12). Thus, formal solutions for Y_2 can be obtained from Eqs. (10) and (11) by making this replacement for g. The iterative procedure described above can be continued to higher order, each time resulting in an equation of the form of Eq. (9) with a new source function which depends on the lower order solutions. COREL uses this procedure to obtain approximate solutions to Eq. (3) through second order.

C. Scattering Cross Section and Stopping Power

The elastic differential scattering cross section used by COREL is the Lindhard[4] expression, based on a Thomas-Fermi model of the interacting atoms. It is given by

$$\sigma(E,T) = \pi a^2 \, \varepsilon^2 \, t^{-3/2} \, f_L(t^{1/2})/(2\, T_m) \quad , \tag{14}$$

where

$$\varepsilon = E/E_L \quad , \tag{15}$$

$$E_L = Z_1 Z_2 \, e^2 (M_1 + M_2)/(M_2 a) \quad , \tag{16}$$

and

$$t = \varepsilon^2 \, T/T_m \quad . \tag{17}$$

In Eq. (16) e is the electronic unit of charge and a is given by

$$a = 0.8853 \, a_o/(Z_1^{2/3} + Z_2^{2/3})^{1/2} \tag{18}$$

where a_o is the Bohr radius of the hydrogen atom ($a_o = 0.529 \times 10^{-8}$ cm). The function $f_L(t^{1/2})$ in Eq. (14) is tabulated by Lindhard, et al.[4]

For the electronic contribution to the stopping power COREL offers the user two choices. The first, valid at sufficiently low incident energy, is given by

$$S(E) = NC_{Se}E^p \tag{19}$$

where C_{Se} and p are constants. The constants in Eq. (19) can come from experimental measurements, or from the theoretical expressions of Lindhard[4] or Firsov.[5] The former case requires input by the user, while the theoretical expressions are contained in the code.

For higher incident energy the user may choose to use the expression

$$S(E) = AN(Z_1 + Z_2)\, f_B(u) \left\{ u^{1/2} \left[\frac{30u^2 + 53u + 21}{3(1+u)^2} \right] + (10u + 1)\, \mathrm{Atan}(u^{1/2}) \right\} , \tag{20}$$

where $A = 0.60961 \times 10^{-15}$ eV cm^2/atom. The E dependence is given by u where

$$u = \frac{E}{Z^2 M_1 E_1} \tag{21}$$

with $E_1 = 99.20$ keV. The function $f_B(u)$ is given by

$$f_B(u) = [1 + (4Z^2 a'^2 u)^{n/2}]^{-1} . \tag{21}$$

This more complex expression for S(E) results from a modification[6] of the Firsov[5] theory and provides S(E) at all nonrelativistic velocities. This expression requires input of values

for the constants Z, a' and n. These parameters depend on the particular ion-target combination for which S(E) is desired, and they can be obtained by fitting Eq. (20) to experimental data.

D. User's Guide

i. General

COREL calculates $R_p(E,0)$, $R(E,0)$, $\Delta R_p(E,0)$ and $\Delta R_\perp(E,0)$ for a number of equally spaced incident energies up to some maximum energy. Both the energy spacing and the maximum energy may be selected by the user within certain restrictions described below. The code provides printed output, and output on TAPE10 for use as input to the code RASE4. The calculations may be carried out for any incident atomic projectile and for targets composed of up to four different atomic species.

The primary purpose of COREL is to provide the range and straggling parameters for <u>target atoms</u> incident on the target. This information allows the codes RASE4 and DAMG2 to account for the redistribution of deposited energy by the recoiling target atoms which are produced by any incident ion. Because of this basic purpose, the information written on TAPE10 does not identify the incident ion; it is assumed to be a target atom. For compound targets, the constituent atomic species in the target must be labeled with an index which provides an ordering of the species types. This same labeling scheme should be used in the codes RASE4 and DAMG2. Output on TAPE10 for compound targets must contain results for each species type as incident

ions, with the results in the same order as the labeling scheme. That is, the calculation of range and straggling variables must be done in the chosen order using each component atomic species as incident ion with all other input variables remaining the same. This is not done automatically by the code, otherwise it would not be possible to use COREL for incident ions of other types. The printed output identifies the incident ion. Writing on TAPE10 can be suppressed by the user by setting the value of NDIVE on card 4 of the input to a number greater than 1 (see below). The instructions which cause writing on TAPE10 are COR 138-139, COR 501 and COR 511-514 (see listings in Section V).

COREL uses the subroutines COPHI, FET, FSET, SECALC and WIND10 (See subroutine listings in Section VI). COREL writes an EOF on TAPE10 after completing a given set of calculations. WIND10 searches for this EOF mark on subsequent runs of the code so that COREL can begin writing at that point. This allows a library of range and straggling data for different targets to be compiled. If the user does not wish to maintain such a library, subroutine WIND10 and instruction COR 106 can be eliminated.

ii. Input

The input to COREL consists of five data cards for each incident ion. These cards, and the variables they contain are described below.

<u>Card 1</u>

Input (TITLE (I), I = 1,8)

Format (8A10)

TITLE(1) identifies the incident ion.

TITLE(2), TITLE(3) identifies the target material.

TITLE(5), through TITLE(8) identify atomic species present in the target.

Example

```
O-16     SILICON DIOXIDE SIO2          SILICON   OXYGEN
```

```
0000000000 0000000000 0000 000000000000 0000000000  00000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
111 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

Card 2

Input (IVAR(J),J = 1,8)

Format (8A10)

IVAR(J), J = 1,8 contains a variable format for printed output. The only changes from the example shown below should be in the number of digits desired after the decimal point in the floating point variables.

Example

```
(2(2F7.0,F9.2,3F12.2,7X))
```

```
 0 0000  0000 000000 0 00000000000000000000000000000000000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
1111111111111111 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

Card 3

Input ALAT,P1,Z1,(P2(J),Z2(J),J = 1,4), (CSE(K),K = 1,4), EQM

Format (16F5.0)

ALAT is the value in angstroms of the cube root of the volume of a unit cell of target material. The size of

the unit cell may be chosen by the user, but this must be consistent with the input on card 4.

P1 is the atomic mass (amu) of the incident ion.

Z1 is the atomic number of the incident ion.

P2(J) is the atomic mass (amu) of atoms of the J'th species in the target.

Z2(J) is their atomic number. For targets with fewer than four constituent atomic species the extra values for P2(J) and Z2(J) may be set to zero, or left blank.

CSE(K) is the value of C_{Se} in Eq. (19) for ions incident on atoms of the K'th atomic species in the target. COREL assumes Bragg's Rule in calculating S(E) for a compound target. That is, S(E) is calculated from Eq. (19) or (20) for each component atomic species in the target and the results are summed to obtain a total S(E). CSE(K) must have a value such that for energy in units of keV Eq. (19) gives S(E)/N in units of 10^{-18} keV-cm^2/atom. If theoretical values are to be used for CSE(K) or if S(E) is to be obtained from Eq. (20) this variable may be set to zero or left blank.

EQM is not used in the code COREL. It has been left as an input variable in order that the input to COREL would be identical in form to the first five cards of input to RASE4.

Example

```
3.566  16.   8.  28.  14.  16.   8.   0.   0.   0.   0.   0.   0.   0.   0.   0.

0000000000000000000000000000000000000000 0000 0000 0000 0000 0000 0000 0000 0000 C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
1111111 1111111111111111 1111 1111111111111111111111111111111111111111111111111111
```

Note that in this example CSE(K) has been set to zero. Input directing COREL to an appropriate calculation of S(E) is contained on the next card.

Card 4

Input NE,NSTP,NCOMP,(NFORM(J),J = 1,4),(NEXP(K),K = 1,4),

NDIV,NDIVE,NMULT,NDSKP,NRPT

Format (1615)

NE,NDIV,NDIVE and NMULT determine the various energy intervals associated with the calculation. The integration mesh size used in the trapezoidal integration of Eq. (10) and (11) is ΔE, where ΔE is given in keV by NMULT/NDIV until the value of the integration variable exceeds 5*NDIVE*NMULT/NDIV, after which ΔE is given by NDIVE*NMULT/NDIV. For accuracy in the integrations ΔE should initially be less than $E_L/2$, where E_L is given by Eq. (16). The primary purpose of NDIVE in the code COREL is as a flag variable for writing on TAPE10. For NDIVE>1 writing on TAPE10 will be suppressed. Printed output is given by COREL for ion energies E_m = mE in keV, m = 1,2,..., where E = NDIVE*NMULT/NDIV. The maximum ion energy considered in the calculation is E_{max} = NE*NMULT/NDIV. Thus,

E_{max}/E = NE/NDIVE should be chosen to be an integer. The maximum permissible value for NE is 1600.

NSTP,NDSKP are not used by COREL. They have been left as input variables so that the input to COREL would be identical in form to the first five cards of input to RASE4.

NCOMP is the number of component atomic species present in the target. $1 \leq NCOMP \leq 4$

NFORM(J) is the number of atoms of the J'th atomic species in a unit cell of the target. This value should be consistent with ALAT on card 3.

NEXP(K) is 100 times the value of p in Eq. (19), except when theoretical values are desired, or when Eq. (20) is to be used to compute S(E). To use the Lindhard theoretical values for CSE and p, set NEXP(K) = 0. To use the Firsov values for CSE and p, set NEXP(K) = -50. To use Eq. (20) set NEXP(K) = any negative number except 50 (see discussion of Card 5).

NRPT is a flag variable which tells COREL when the last set of input data is being read. For NRPT $\neq$ 0 the code will complete the calculations of the present five-card set of data and terminate. For NRPT = 0 the code will expect another five-card set of data to follow the present set.

Example

```
30   10    2    1    2    0    0    0    0    0    0    1    1    1    1    1

0000 0000 00000000000000000000 0000 0000 0000 0000 0000 0000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
11111111 1111111111 1111111111111111111111111111111111111111111 1111 1111 1111 1111
```

Card 5

Input (SETA(J), ALP(J), EN(J),J = 1,4)

Format (12F5.0)

SETA(J), ALP(J), and EN(J) are, respectively, values for Z,a' and n to be used in Eq. (20) for calculating S(E) for the J'th target atomic species if that option has been chosen on Card 4. If Z, a' and n are not needed then they may be left blank on this card. Note that the choice of method for calculating S(E) may be different for each individual species in the target. For example, Eq. (20) may be used for the first component, and Eq. (19) for the remaining three, with a mixed choice between experimental and theoretical values for CSE(K).

Example

```
2.533.46503.1872.134.44763.393

00000000 00000000000000000000000000000000000000000000000000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
11111111111 1111 1111111111111111111111111111111111111111111111111111111111111111
```

Note that this example has been chosen for protons incident on silicon dioxide. The four previous cards have been examples

for oxygen incident in SiO_2 for which Z, a' and n values are not available.

iii. Output

The output of COREL consists of information on TAPE10 (if NDIVE = 1) and printed output. Two logical blocks of data are written on TAPE10 for each incident ion considered. For compound targets there should be one set of two output blocks for each type of atom in the target material, and this output should be in the same order as the target components are ordered for the input data cards. This same ordering should also be used for the input cards to RASE4. The data blocks are indicated schematically in Fig. 1 for both elemental and compound targets.

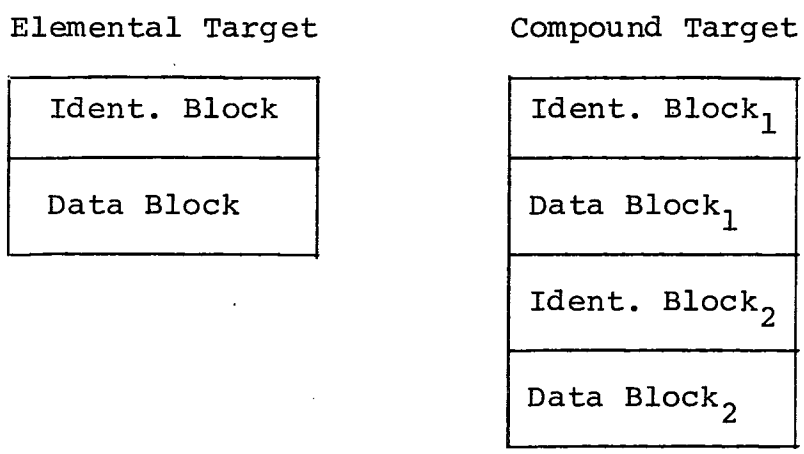


Figure 1. Data Block Structure Written on TAPE10 by COREL

The identification block contains NE,NDIV,NMULT,NCOMP, ALAT,(P2(J),Z2(J),SETA(J),ALP(J),EN(J),NFORM(J),NEXP(J), J = 1,4). This information is written by instruction COR138 and COR139.

The data block contains NE,NDIV,NMULT,NCOMP,(RP(J),DELTA(J), DP(J),J = 1,NE). This data is written by instruction COR501. The variables are RP, projected range, DELTA, standard deviation in RP, and DP, the standard deviation in the ion locations transverse to RP. The energy associated with these variables is E(J) = J*NMULT/NDIV.

After completing all requested calculations COREL writes an identification block and a data block which are internally generated. In its calculations, RASE4 searches TAPE10 for range and straggling variables for the target atoms. If it reaches the last data block and it has been unable to find the desired data, a flag is set which tells RASE4 to calculate approximate values for this information.

Sample printed output is shown in Section II.F. below for the example input cards previously shown. Note that because of the values of NEXP on card 4 the information on card 5 is ignored. The running time for this sample was 13 seconds. The running time should be roughly proportional to NE*NCOMP.

The quantities in the printed output which have not previously been discussed are:

ENERGY MAXIMUM = $Z_1^{4/3}$ * P1 * 25 keV. This is the upper limit of validity for S(E) given by Eq. (19).

T-F RADIUS = a in Eq. (18).

EFFECTIVE Z = denominator of Eq. (18) raised to the 3/2 power.

CHAR. ENERGY = E_L in Eq. (16).

CHARACTERISTIC LENGTH is a length defined by Lindhard[4] and is presented for information only.

ATOMIC DENSITY is the target atomic density calculated from the input variables.

U(I) is the variable $t^{1/2}$ in Eqs. (14) and (17) and F(U) is $f_L(t^{1/2})$.

SLOPE is the slope of F(U) vs LOG U.

The printed range and straggling variables are:

RP(E,EP) = $R_p(E,E')$

SPREAD IN RP-PARA = $\Delta R_p(E,E')$

R(E,EP) = R(E,E')

SPREAD IN RP-PERP. = $\Delta R_\perp(E,E')$

E. Brief Programmers Guide

i. Methods of Integration

The integrations indicated in Eqs. (10) and (11) are carried out by the trapezoidal method. They are executed in the loop beginning at instruction COR247. The index J steps the integration variable over intervals with width as described in the user's guide. (See input card 4.) Summation over the component atomic species in the target is performed by the loop beginning at instruction COR267. The loop index LJ is the component number.

The integrations indicated in Eqs. (4), (5) and (6) are executed in the loop beginning at instruction COR293. These integrations are performed by the rectangular method. The

integrand is evaluated in the middle of the interval, multiplied by the interval width, and summed. Integration is performed in this loop using $u = t^{1/2}$ from Eq. (17) as the integration variable. For this variable the upper limit to the integration is EPS = ε, from Eq. (15). A variable integration interval is used here. The first interval has a width of 10^{-NQT} where NQT is given in instruction COR274. The next 18 integration intervals have width $10^{-NQT}/2$; the next 18, $10^{1-NQT}/2$; the next 18, $10^{2-NQT}/2$; etc., until the upper limit is reached. The last interval has width OVER which is calculated internally. These intervals can be made narrower by increasing the value of X in instruction COR279. The value of X in this instruction should always be an integer.

ii. Main Program Variables

The values of β_n are stored as

β_o = ALPHA

β_1 = BETA

β_2 = GAM

with suffixes on ALPHA, BETA AND GAM to distinguish between β_n values for R, R_p, R_c^2 and R_r^2. Suffix 1 refers to R_p, suffix 2 to R_r^2 and suffix 3 to both R and R_c^2 since the β_n values are identical for these two parameters.

The variables EXINT(J) contain values of $[W(0,E,0)]^{-1}$ from Eq. (11). The suffix convention for EXINT is the same as for the β_n. Note that $W(y,E,0) = W(0,E,0)/W(0,y,0)$.

Inside the integration loops the variables R(J), RP(J), RC2(J) and RR2(J) contain first order solutions to the equations

for R, Rp, R_c^2 and R_r^2, respectively, multiplied by the corresponding EXINT(J). The variables DRO, DR, DRC2 and DRR2 contain the corresponding second order corrections to R, RP, RC2 and RR2. Values for R, ΔR_p, $\Delta R_\perp$, and R are calculated in the loop beginning at instruction COR460.

iii. Subroutines.

COPHI(CPHI,T,J,LJ)

CPHI is the cosine of the ion scattering angle in laboratory coordinates.

T is the energy transfer to a recoiling target atom in keV.

J is an index which gives the ion energy before the scattering event. E(J) = NDIVE*NMULT*J/NDIV.

LJ is an index which identifies the component atomic species of the target.

COPHI returns a value for CPHI for an input of T,J,LJ.

The Common Block PHIVAR contains E(J) values and P2(LJ) values.

FET(Y,X)

$Y = f_L(t^{1/2})$ from Eq. (14).

$X = t^{1/2}$

FET returns a value of Y for an input of X.

FSET

FSET contains a tabulation of f_L values.

SECALC(Z1,Z2,P1,E,SE,Z,A,EN)

Z1 = incident ion atomic number.

Z2 = target atom atomic number.

P1 = incident ion atomic mass in amu.

E = ion energy in keV

SE = S(E)/N from Eq. (20)

Z = Z from Eq. (20)

A = a' from Eq. (20)

EN = n from Eq. (20)

SECALC returns a value of SE in keV-cm^2/atom for an input of the remaining variables.

WIND10

WIND10 positions output TAPE10 so that writing begins after the last data block previously written on the tape.

F. Sample Output

The sample output on the next two pages was obtained using as input the example cards of Section II-Dii.

RANGE AND STRAGGLING DATA FOR O-16 IONS INCIDENT ON A/AN SILICON DIOXIDE SIO2 TARGET

INITIAL VALUES

INCIDENT ION O-16
ATOMIC NUMBER(Z1) = 8
ATOMIC MASS(M1) = 16.00 AMU
ENERGY MAXIMUM = 6348.97KEV

TARGET - SILICON DIOXIDE SIO2

COMPONENTS	SILICON	OXYGEN		
ATOMIC NUMBER	14	8	0	0
ATOMIC MASS	28.00 AMU	16.00 AMU	0.00 AMU	0.00 AMU
T-F RADIUS	1.497E-09 CM	1.657E-09 CM	0. CM	0. CM
EFFECTIVE Z	30.720	22.627	0.000	0.000
CHAR. ENERGY	16.932 KEV	11.121 KEV	0.000 KEV	0.000 KEV

CHARACTERISTIC LENGTH = *421.96 ANGSTROMS

ATOMIC DENSITY = 6.61572E+22 ATOMS/CC

THE FOLLOWING ELECTRONIC STOPPING CROSS SECTIONS WERE USED IN THE CALCULATIONS

SILICON - THOMAS-FERMI ATOMIC MODEL
CSE = 1.09373E-03

OXYGEN - THOMAS-FERMI ATOMIC MODEL
CSE = 1.69703E-03

NE	NSTP	NCOMP	NFORM(I) I = 1	2	3	4	NEXP(J) J = 1	2	3	4	NDIV	NDIVE	NMULT	NDSKP	NRPT
30	10	2	1	2	0	0	0	0	0	0	1	1	1	1	1

I	U(I)	F(U)	SLOPE
1	.002	.162	.06781
2	.004	.209	.07749
3	.010	.280	.07791
4	.020	.334	.07069
5	.040	.383	.05239
6	.100	.431	.00987
7	.150	.435	-.02433
8	.200	.428	-.06204
9	.400	.385	-.12005
10	1.000	.275	-.13129
11	2.000	.184	-.11109
12	4.000	.107	-.06221

- F(U) IS UNIVERSAL THOMAS-FERMI FUNCTION - FROM LSS

**** UNITS *****
ENERGY - KEV
RP,R,DELTA,DP - ANGSTROMS
EPS,DE/DR - DIMENSIONLESS
DAMAGE - EV/ANGSTROM

E	EP	RP(E,EP)	SPREAD IN RP-PARA	R(E,EP)	SPREAD IN RP-PERP
2	0	48.88	39.38	56.48	14.85
4	0	91.51	54.05	118.99	52.54
6	0	133.85	71.41	179.17	79.18
8	0	176.57	89.28	238.75	103.90
10	0	219.85	107.09	298.22	127.83
12	0	263.72	124.65	357.74	151.29
14	0	308.11	141.99	417.34	174.36
16	0	353.00	159.12	477.03	197.07
18	0	398.34	175.96	536.79	219.48
20	0	444.06	192.47	596.60	241.66
22	0	490.11	208.70	656.43	263.60
24	0	536.48	224.64	716.27	285.26
26	0	583.11	240.26	776.07	306.68
28	0	629.98	255.55	835.81	327.87
30	0	677.04	270.53	895.48	348.82

E	EP	RP(E,EP)	SPREAD IN RP-PARA	R(E,EP)	SPREAD IN RP-PERP
1	0	27.66	34.80	22.00	1.00
3	0	70.30	46.04	88.26	37.15
5	0	112.65	62.65	149.21	66.20
7	0	155.15	80.32	208.99	91.68
9	0	198.14	98.21	268.48	115.94
11	0	241.71	115.90	327.97	139.60
13	0	285.85	133.34	387.53	162.87
15	0	330.50	150.58	447.17	185.76
17	0	375.62	167.59	506.91	208.31
19	0	421.15	184.26	566.69	230.60
21	0	467.04	200.61	626.51	252.67
23	0	513.26	216.70	686.35	274.47
25	0	559.76	232.51	746.18	296.00
27	0	606.52	247.94	805.95	317.31
29	0	653.49	263.08	865.66	338.37

III. RASE4

A. General Equations

The equations governing the range and straggling parameters are solved by RASE4 for several values of E', the instantaneous ion energy. The solutions to these equations have already been outlined in the discussion under COREL in Section IIA. In addition, RASE4 evaluates the average of several other quantities over the elastic scattering cross section. Input to RASE4 from a previous run of COREL is required for this process, and the resultant averages are used by DAMG2 in the evaluation of the energy deposition distributions. The quantities which are averaged are discussed below.

One important average evaluated by RASE4 is $\Sigma(E)$ where

$$\Sigma(E) = N \int_0^{T_m} q(T)\ \sigma(E,T)\ dT \quad . \tag{23}$$

In Eq. (23) q(T) is that portion of the energy, T, of the recoiling target atom which will eventually wind up in atomic motions and displacements (atomic processes). All other symbols in Eq. (23) are as defined in Section II. For elemental targets the function q(T) is given by Lindhard[4] as

$$q(T) = T/(1 + k\ g_L(T/E_o)) \quad , \tag{24}$$

where

$$k = 0.1334\ Z_2^{2/3}/A_2^{1/2} \quad . \tag{25}$$

A_2 is the target atomic mass number. Robinson[7] gives $g_L(x)$ as

$$g_L(x) = x + 0.40244\ x^{3/4} + 3.4008\ x^{1/6} \quad , \tag{26}$$

and E_o is the value of E_L, Eq. (16), for $Z_1 = Z_2$ and $M_1 = M_2$. For compound targets, a simple expression for q(T) is not available, and the user must provide values for this function. Note, however, the iterative procedure described in Section IIIB.

A second quantity evaluated by RASE4 is $\Sigma_I(E)$ given by

$$\Sigma_I(E) = N\int_o^{T_m} q'(T)\ \sigma(E,T)\ dT \quad , \tag{27}$$

where $q'(T) = T - q(T)$. Note that $q'(T)$ is that portion of the recoil energy, T, which will ultimately be deposited into electronic processes.

Let $R_p(T,0)$, $\Delta R_p(T,0)$ and $\Delta R_\perp(T,0)$ be the range and straggling parameters for a recoiling target atom of energy T. Let η be the cosine of the recoil angle in laboratory coordinates, relative to the direction of motion of the ion before the collision which created the recoil atom. Define

$$r_p(T) = \eta\ R_p(T,0) \tag{28}$$

and

$$r_{p2}(T) = [\eta R_p(T,0)]^2 + [\eta \Delta R_p(T,0)]^2 + (1 - \eta^2)\ \Delta R_\perp^2(T,0)/2 \quad . \tag{29}$$

Then,

$$G_1(E) \equiv \int_o^{T_m} r_p(T)\ q(T)\ \sigma(E,T)\ dT \Big/ \int_o^{T_m} q(T)\ \sigma(E,T)\ dT \tag{30}$$

and

$$G_2(E) \equiv \int_0^{T_m} r_{p2}(T)\ q(T)\ \sigma(E,T)\ dT \Big/ \int_0^{T_m} q(T)\ \sigma(E,T)\ dT \tag{31}$$

represent weighted averages of the recoil range and its square, projected on the direction of motion of the incident ion. These averages are related to the redistribution of the energy deposited into atomic processes by the recoiling target atoms. In particular, if D is the average distance energy is carried by the recoils before deposition, and ΔD its standard deviation, then to a very good approximation,

$$D = 0.8\ G_1 \tag{32}$$

and

$$\Delta D^2 = 0.64(G_2 - G_1^2) = 0.64\ \Delta G^2 \quad . \tag{33}$$

Equations similar to Eq. (30) - (33) can also be written for energy deposited into ionization. For that case q(T) in Eq. (30) and (31) would be replaced by q'(T). In order to distinguish between the redistribution parameters D and ΔD for energy deposition into both atomic processes and ionization, the symbols I and ΔI will be used to represent the left hand sides of Eq. (32) and (33) when q'(T) is used in the evaluation of G_1 and G_2 (ionization) while D and ΔD will continue to represent the parameters when q(T) is used in the evaluation (atomic processes). RASE4 evaluates both sets of parameters, i.e., D, ΔD, I and ΔI.

B. Compound Targets

For elemental targets Eq. (24) gives q(T) for the evaluation of the averages discussed in the previous section. For compound targets a simple formula such as this is not available and functions $q_i(T)$ (one is required for each component atomic species of the target) must be obtained by other means for input into RASE4. An iterative procedure using the codes RASE4 and DAMG2 can be carried out to obtain this information.

As an example of the iterative procedure, consider a target of silicon dioxide, SiO_2. Each of the atoms Si and O is considered, in turn, as an ion incident on SiO_2, and to begin with $q(T)/T = 1$ is assumed. This last assumption means that the recoils created by the incident Si and O deposit all their energy into atomic processes. The incident Si and O, however, lose some of their energy to electronic excitation because of S(E). The calculation of the above averages is then carried out by RASE4 under these assumptions, and the output of RASE4 is input to DAMG2 which calculates the partitioning of the energy into atomic and electronic processes. As a result, DAMG2 outputs a function F(T) which is a first order approximation to q(T)/T. (Note: a function F(T) is obtained for Si ions and a different F(T) for O ions). These first order approximations to q(T)/T are then input to RASE4 and the procedure is repeated. After several iterations the functions F(T) converge and the functions $q_i(T)$ are determined over the energy range considered in the iterative procedure. Note

that the $q_i(T)/T$ are functions which describe the damage efficiency of the target atom recoils, so functions are needed only for those atoms which are present in the target material. Thus, once the iterative procedure has been completed for a given target material, calculations for incident foreign ions can be carried out without iteration, since the $q_i(T)/T$ have been determined.

C. User's Guide

i. General

The range and straggling parameters $R(E,E')$, $R_p(E,E')$ $\Delta R_p(E,E')$ and $\Delta R_\perp(E,E')$ are calculated by RASE4 for several values of E and E' which are under the control of the user as described below. These calculations are carried out in exactly the same manner as described in Section II for COREL. The code provides printed output and output on TAPE20 for input to the code DAMG2 for calculating energy deposition distributions.

In addition to calculating the range and straggling variables RASE4 also calculates the averages over the cross section $\Sigma(E)$, $\Sigma_I(E)$, $D(E)$, $\Delta D(E)$, $I(E)$ and $\Delta I(E)$ as described in Section IIIA. The user may choose to have only Σ, D, and ΔD evaluated; only Σ_I, I and ΔI evaluated; all six quantities evaluated; or none of the six quantities evaluated. (See discussion of input variable NDSKP.)

RASE4 uses the subroutines COPHI, FET, FSET, SECALC, DAMG, RANGER, TEN, and WIND20 (see subroutine listings in Section VI). RASE4 writes an End of File mark on TAPE20 after a given set

of calculations. The subroutine WIND20 positions output TAPE20 so that writing will begin on the tape after the last data block previously written. This allows the user to maintain a library of RASE4 output on tape. If it is not desired to maintain such a library the instructions RAS 122 and the subroutine WIND20 can be eliminated.

ii. Input

The input to RASE4 consists of recoil data from TAPE10, and five or more data cards. For elemental targets only five cards are required. For compound targets at least three additional input cards are required.

<u>Card 1</u> This card has the same form and function as for COREL.

<u>Card 2</u> This card has the same form and function as for COREL.

<u>Card 3</u> This card has the same form and function as for COREL. The variable ED (listed as EQM in the discussion of COREL) is functional in RASE4. ED is the threshold energy (in keV) for displacement of atoms of the target. When a recoiling target atom is created with energy T less than ED then all its energy is assumed to go into atomic processes. For T greater than ED an energy ED goes into atomic processes, and the recoiling target atom moves away from the site of the collision with an energy T-ED, which is

subsequently partitioned into atomic and electronic processes. Except for light ions incident on heavy atom targets, or at very low incident energy, realistic values of ED have no significant effect on the range or energy deposition distributions, i.e., generally such effects are much less than 1%.

Card 4 This card has the same form and input as for COREL. The variables NSTP and NDSKP are functional in RASE4.

NSTP in conjunction with NDIV, NMULT and NDIVE determines the energy spacing of the variables E and E' in the output of the range and straggling variables. All energies are in keV. The energy spacing in the variable E is given by NSTP*NMULT/NDIV. The low energy spacing in the variable E' is given by NDIVE*NMULT/NDIV up to E' = NSTP*NMULT/NDIV, after which the energy spacing in E' is the same as for E. The user should use values of NDIVE, NSTP and NE such that NE/NSTP = integer and NSTP/NDIVE = integer. For accuracy in DAMG2, NE/NSTP $\geq$ 10, NSTP/NDIVE $\geq$ 10. All other output has energy spacing in units of NMULT/NDIV. See discussion of this card for COREL for other limitations on these variables.

NDSKP determines which of the averages described in Section IIIA will be calculated. For NDSKP = 1, all six averages are calculated. For NDSKP = 2, Σ_I, I and ΔI are calculated using q'(T) in Eqs. (30) and (31). For any other positive value of NDSKP, Σ D and ΔD are calculated using q(T) in Eqns. (30 and (31). For NDSKP = 0 none of the averages are calculated. When RASE4 is used in the iterative procedure described above for compound targets, NDSKP should be set to a negative value ≤ -2. A value of NDSKP ≤ -2 written on TAPE20 in a previous calculation instructs the code to begin writing new data on TAPE20 at that point, since the old data is temporary in the iterative mode. The last calculation of an iteration sequence should have a positive value for NDSKP. NOTE the program DAMG2 will consider its output on TAPE30 to be temporary for any value of NDSKP except NDSKP = 1.

Card 5 This card has the same form and function as for COREL.

Card 6 This card is necessary only if NCOMP > 1 on card 4.

Input DE, NDAM

Format (F6.1,I4)

DE is the energy spacing at which equally spaced values of $q_i(T)/T$ are to be read from subsequent cards. DE is in keV. For accuracy DE should be

less than $E_L/4$ where E_L is calculated from Eq. (16) for the lightest target component.

NDAM is the number of equally spaced values of $q_i(T)/T$ which are to be entered for each component atomic species in the target. NDAM should be less than or equal to 100, and should be chosen so that NDAM*DE is greater than or equal to the maximum possible recoil atom energy. This value can be determined by evaluating Eq. (7a) for each of the atomic species in the target and using the largest of these values for determining NDAM.

Example

```
10.  10

0000 0000 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
111 1111 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

Additional Cards These cards are needed only if NCOMP > 1 on card 4. A set of these cards is required for each component atomic species present in the target.

Input (PART(J,K),K = 2, NDAM + 1)

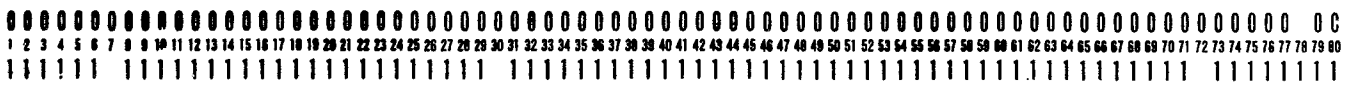
Format (10F8.5)

PART(J,K) contains the value of q(T)/T for the J'th target atom species with T = (K-1)*DE. The code reads cards for the J'th species through the instruction READ 1001, (PART(J,K),K = 2, NDAM + 1). This instruction is contained in a DO loop with J as loop variable (see subroutine DAMG, instructions DMG 25-DMG 28). Thus, for

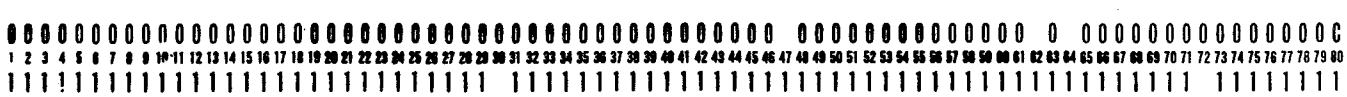
NDAM = 10, 1 input card is required for each target atom component, for 10 < NDAM $\leq$ 20 two cards are required for each component, etc. Up to 10 cards may be required for each component. Values for PART(J,K) may be obtained by the iterative procedure described in Section IIIB. Briefly, the iterative procedure begins by assuming PART(J,K) = 1 for all energies (K), and for each of the component atomic species (J) present in the target. The code RASE4 is then processed using these PART(J,K)-values and the resultant output is used to evaluate damage functions with the code DAMG2 (see Section IV). The resultant output of DAMG2 provides a first order estimate of the correct PART(J,K)-values. Cards containing these new PART(J,K)-values must then be punched for input to RASE4. The subsequent output is again used with DAMG2 to obtain a better estimate of the PART(J,K)-values This procedure continues until the PART(J,K)-values have converged to values which do not change significantly with further iteration.

Examples

.6616 .6539 .6385 .6162 .5945 .5742 .5555 .5382 .5221 .5072



.6246 .5974 .5529 .5135 .4797 .4507 .4253 .4030 .3831 .3653



iii. Output

The output of RASE4 consists of information on TAPE20 and printed output. Two, three, or four logical blocks of data are written on TAPE20 for each incident ion considered. The data blocks are indicated schematically in Fig. 2.

Ident. Block
Damage Block
Ionization Block
Range Block

Figure 2. Data Block Structure Written on TAPE20 by RASE4.

The identification block contains NE, NSTP, NDIV, NMULT, NCOMP, NDIVE, NDSKP, (NFORM(J), NEXP(J), J = 1,4), ALAT, P1, Z1, (P2(K), Z2(K), CSE(K), SETA(K), ALP(K), EN(K),K = 1,4). This information is written by instructions RAS 160-RAS 162.

The damage data block contains NE + 1, NDIV, NMULT, NDSKP, (D(J), DR(J), DRR2(J), J = 1, NE). The locations D, DR and DRR2 contain $\Sigma(E)$, D(E) and $\Delta D(E)$, respectively, from Eqs. (23), (32) and (33), with E(J) = J*NMULT/NDIV. This information is written by instruction RAS 432 and RAS 433.

The ionization data block contains NE + 1, NDIV, NMULT, NDSKP, (DRO(J), D(J), DR(J), DRR2(J), J = 1, NE). The locations DRO, D, DR and DRR2 contain, respectively, S(E) in eV/Å, $\Sigma_I(E)$ from Eq. (27), and I(E) and $\Delta I(E)$ from Eqs. (32) and (33) using q'(T) in the integrations. Again E(J) = J*NMULT/NDIV. This information is written by instructions RAS 434 and RAS 435.

The range data block contains ((DAMRP(J,K), DAMDEL(J,K), DAMR(J,K), DAMDP(J,K), K = 1,60), J = 1, 50). This information is written by instructions RAS 749 and RAS 750. The locations DAMRP, DAMDEL, DAMR, DAMDP contain, respectively, $R_p(E,E')$, $\Delta R_p(E,E')$, $R(E,E')$ and $\Delta R_\perp(E,E')$ from Eqs. (1), (2) and the solutions of Eq. (3). The variable E(J) = J*NSTP*NMULT/NDIV. For K $\leq$ NSTP/NDIVE, E'(K) = (K - 1)*NDIVE*NMULT/NDIV. For K > NSTP/NDIVE, E'(K) = (K - NSTP/NDIVE)*NSTP*NMULT/NDIV.

The printed output for RASE4 has essentially the same format as that for COREL. Also printed are DRO, D, DR and DRR2, as described for both the damage and ionization blocks of TAPE20 output. With this exception, the output variables all have the same meaning as for COREL. Sample output is shown in Section IIIE.

D. Brief Programmer's Guide

i. General

The programming of RASE4 is essentially the same as for COREL. The main differences are that E' has non-zero values, and the additional averages over the cross section are calculated as described in Section IIIA. The value for E' is set by the loop variable I in the DO loop beginning at instruction RAS 440. The averages over the cross section are carried out at the same time the β_n are evaluated, and by the same method. These integrations are performed in the loop beginning at instruction RAS 313. The energy integrations for Eq. (10) and (11) are performed by the loop beginning at instruction RAS 465.

The variables in RASE4 have the same meaning as in COREL. The locations DR and DRR2 are used in the loop beginning at instruction RAS 277 for temporary storage of G-values (Eq. (30), (31)).

ii. Subroutines

RASE4 uses the subroutines COPHI, FET, FSET and SECALC as described in Section IIEiii for COREL. In addition, the following subroutines are used by RASE4.

DAMG(Z, T, ALF, I)

Z = q(T) (See discussion in Section IIIA)

T = recoil atom energy in keV.

ALF = value of k in Eq. (25).

I = index to identify target atom species in compound target.

DAMG returns a value of Z for an input of the other variables. For elemental targets Z is calculated from Eq. (25) and the parameter I is ignored. For compound targets, DAMG reads input cards 6, 7, . . . etc. NDAM, NCOMP, DE and PART(J,K) are shared with the main program through the Common Block DFIR.

RANGER(E, LJ, CPHI, I, X, DELT)

E = recoil atom energy in keV.

LJ = index to identify target atom species in compound target.

CPHI = η in Eq. (28) and (29).

X = $r_p(E)$ from Eq. (28).

DELT = $r_{p2}(E)$ from Eq. (29).

RANGER returns a value for X and DELT for an input of the other variables. I = read index; for I = 1, RANGER reads the range and straggling variables for target atoms from TAPE10. These variables are stored in the Common Block DSEC.

WIND20

WIND20 positions output TAPE20 so that writing begins after the last data block previously written on the tape. If WIND20 encounters a data block with NDSKP < 0 it positions the tape to write over that data block. Negative values for NDSKP are used to indicate temporary data storage in the iterative procedure for compound targets.

TEN

TEN searches TAPE10 for target atom range and straggling variables consistent with the present calculations being carried out by RASE4. In particular TEN insures that i) the target atom variables have been calculated to sufficiently high energy, ii) the target is composed of the same constituents as indicated in the RASE4 input, and iii) that targets with the same density are used in both the COREL and RASE4 calculations.

E. SAMPLE OUTPUT

The sample output on page 46, et. seq., was obtained using as input the example cards 2, 4, 5 from Section II-Dii, the

example cards from Section III-Cii, and the following cards for card 1 and card 3.

```
SI-28    SILICON DIOXIDE SIO2          SILICON   OXYGEN

0 00000000 0000000000 0000 000000000000 0000000000  00000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111

3.566  28.  14.  28.  14.  16.   8.   0.   0.   0.   0.   0.   0.   0.   0.   0.

0000000000000000000000000000000000000000 0000 0000 0000 0000 0000 0000 0000 0000 C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
1111111111111 111111111 1111 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

RASE4 also used as input from tape 10 the sample output shown for COREL in Section II-F (along with similar data for silicon incident on SiO_2) to obtain the output presented on the next several pages.

RANGE AND STRAGGLING DATA FOR SI-28 IONS INCIDENT ON A/AN SILICON DIOXIDE SIO2 TARGET

INITIAL VALUES

INCIDENT ION SI-28
ATOMIC NUMBER(Z1) = 14
ATOMIC MASS(M1) = 28.00 AMU
ENERGY MAXIMUM = 23431.06KEV

TARGET - SILICON DIOXIDE SIO2

COMPONENTS	SILICON	OXYGEN		
ATOMIC NUMBER	14	8	0	0
ATOMIC MASS	28.00 AMU	16.00 AMU	0.00 AMU	0.00 AMU
T-F RADIUS	1.375E-09 CM	1.497E-09 CM	0. CM	0. CM
EFFECTIVE Z	39.598	30.720	0.000	0.000
CHAR. ENERGY	41.043 KEV	29.632 KEV	0.000 KEV	0.000 KEV

CHARACTERISTIC LENGTH = 239.04 ANGSTROMS

ATOMIC DENSITY = 6.61572E+22 ATOMS/CC

THE FOLLOWING ELECTRONIC STOPPING CROSS SECTIONS WERE USED IN THE CALCULATIONS

SILICON - THOMAS-FERMI ATOMIC MODEL , LINDHARD THEORY
CSE = 1.23221E-03

OXYGEN - THOMAS-FERMI ATOMIC MODEL , LINDHARD THEORY
CSE = 1.81522E-03

NE	NSTP	NCOMP	NFORM(I) I = 1	2	3	4	NEXP(J) J = 1	2	3	4	NDIV	NDIVE	NMULT	NDSKP	NRPT
30	10	2	1	2	0	0	0	0	0	0	1	1	1	1	0

I	U(I)	F(U)	SLOPE
1	.002	.162	.06781
2	.004	.209	.07749
3	.010	.280	.07791
4	.020	.334	.07069
5	.040	.383	.05239
6	.100	.431	.00987
7	.150	.435	-.02433
8	.200	.428	-.06204
9	.400	.385	-.12005
10	1.000	.275	-.13129
11	2.000	.184	-.11109
12	4.000	.107	-.06221

- F(U) IS UNIVERSAL THOMAS-FERMI FUNCTION - FROM LSS

***** UNITS *****
ENERGY - KEV
LENGTHS - ANGSTROMS
ENERGY DEPOSITION RATES - EV/ANGSTROM

ELECTRONIC STOPPING POWER , EV/ANGSTROM

E	DE/DX	E	DE/DX	E	DE/DX	E	DE/DX	E	DE/DX	E	DE/DX	E	DE/DX
1.0	3.0474E+00	2.0	4.3097E+00	3.0	5.2783E+00	4.0	6.0949E+00	5.0	6.8143E+00	6.0	7.4646E+00	7.0	8.0627E+00
8.0	8.6194E+00	9.0	9.1423E+00	10.0	9.6368E+00	11.0	1.0107E+01	12.0	1.0557E+01	13.0	1.0988E+01	14.0	1.1402E+01
15.0	1.1803E+01	16.0	1.2190E+01	17.0	1.2565E+01	18.0	1.2929E+01	19.0	1.3283E+01	20.0	1.3629E+01	21.0	1.3965E+01
22.0	1.4294E+01	23.0	1.4615E+01	24.0	1.4929E+01	25.0	1.5237E+01	26.0	1.5539E+01	27.0	1.5835E+01	28.0	1.6125E+01
29.0	1.6411E+01	30.0	1.6691E+01										

DEPOSITED ENERGY(EV/ANGSTROM) , DEP-E

E	DEP-E	E	DEP-E	E	DEP-E	E	DEP-E	E	DEP-E	E	DEP-E	E	DEP-E
1.0	3.9736E+01	2.0	4.5763E+01	3.0	4.8603E+01	4.0	5.0132E+01	5.0	5.0896E+01	6.0	5.1153E+01	7.0	5.1055E+01
8.0	5.0724E+01	9.0	5.0242E+01	10.0	4.9655E+01	11.0	4.9016E+01	12.0	4.8357E+01	13.0	4.7763E+01	14.0	4.7198E+01
15.0	4.6634E+01	16.0	4.6103E+01	17.0	4.5561E+01	18.0	4.5020E+01	19.0	4.4540E+01	20.0	4.4064E+01	21.0	4.3584E+01
22.0	4.3103E+01	23.0	4.2643E+01	24.0	4.2203E+01	25.0	4.1779E+01	26.0	4.1357E+01	27.0	4.0941E+01	28.0	4.0545E+01
29.0	4.0151E+01	30.0	3.9761E+01										

AVERAGE PROJECTED RECOIL RANGE(ANGSTROMS) RP-REC

E	RP-REC	E	RP-REC	E	RP-REC	E	RP-REC	E	RP-REC	E	RP-REC	E	RP-REC
1.0	6.2833E+00	2.0	1.1755E+01	3.0	1.6558E+01	4.0	2.0807E+01	5.0	2.4715E+01	6.0	2.8305E+01	7.0	3.1788E+01
8.0	3.4976E+01	9.0	3.7969E+01	10.0	4.0774E+01	11.0	4.3423E+01	12.0	4.5988E+01	13.0	4.8709E+01	14.0	5.1348E+01
15.0	5.3978E+01	16.0	5.6665E+01	17.0	5.9283E+01	18.0	6.1862E+01	19.0	6.4497E+01	20.0	6.7101E+01	21.0	6.9656E+01
22.0	7.2189E+01	23.0	7.4727E+01	24.0	7.7254E+01	25.0	7.9740E+01	26.0	8.2197E+01	27.0	8.4635E+01	28.0	8.7074E+01
29.0	8.9477E+01	30.0	9.1865E+01										

RMS SPREAD IN RP-REC

E	DRP-REC	E	DRP-REC	E	DRP-REC	E	DRP-REC	E	DRP-REC	E	DRP-REC	E	DRP-REC
1.0	1.2398E+01	2.0	2.0679E+01	3.0	2.5785E+01	4.0	3.0377E+01	5.0	3.4955E+01	6.0	3.9312E+01	7.0	4.4984E+01
8.0	4.9940E+01	9.0	5.4786E+01	10.0	5.9631E+01	11.0	6.4333E+01	12.0	6.8905E+01	13.0	7.3863E+01	14.0	7.8646E+01
15.0	8.3440E+01	16.0	8.8368E+01	17.0	9.3247E+01	18.0	9.8375E+01	19.0	1.0296E+02	20.0	1.0783E+02	21.0	1.1264E+02
22.0	1.1750E+02	23.0	1.2234E+02	24.0	1.2713E+02	25.0	1.3188E+02	26.0	1.3664E+02	27.0	1.4135E+02	28.0	1.4606E+02
29.0	1.5073E+02	30.0	1.5539E+02										

ELECTRONIC STOPPING POWER , EV/ANGSTROM

E	DE/DX	E	DE/DX	E	DE/DX	E	DE/DX	E	DE/DX	E	DE/DX	E	DE/DX
1.0	3.0474E+00	2.0	4.3097E+00	3.0	5.2783E+00	4.0	6.0949E+00	5.0	6.8143E+00	6.0	7.4646E+00	7.0	8.0627E+00
8.0	8.6194E+00	9.0	9.1423E+00	10.0	9.6368E+00	11.0	1.0107E+01	12.0	1.0557E+01	13.0	1.0988E+01	14.0	1.1402E+01
15.0	1.1803E+01	16.0	1.2190E+01	17.0	1.2565E+01	18.0	1.2929E+01	19.0	1.3283E+01	20.0	1.3629E+01	21.0	1.3965E+01
22.0	1.4294E+01	23.0	1.4615E+01	24.0	1.4929E+01	25.0	1.5237E+01	26.0	1.5539E+01	27.0	1.5835E+01	28.0	1.6125E+01
29.0	1.6411E+01	30.0	1.6691E+01										

DEPOSITED ENERGY(EV/ANGSTROM) , DEP-E

E	DEP-E	E	DEP-E	E	DEP-E	E	DEP-E	E	DEP-E	E	DEP-E	E	DEP-E
1.0	5.3499E-01	2.0	1.2219E+00	3.0	1.9433E+00	4.0	2.6657E+00	5.0	3.3772E+00	6.0	4.0619E+00	7.0	4.7393E+00
8.0	5.3855E+00	9.0	6.0132E+00	10.0	6.6209E+00	11.0	7.1991E+00	12.0	7.7418E+00	13.0	8.1619E+00	14.0	8.5037E+00
15.0	8.8092E+00	16.0	9.0581E+00	17.0	9.3001E+00	18.0	9.5193E+00	19.0	9.6641E+00	20.0	9.7933E+00	21.0	9.9206E+00
22.0	1.0044E+01	23.0	1.0145E+01	24.0	1.0226E+01	25.0	1.0292E+01	26.0	1.0358E+01	27.0	1.0421E+01	28.0	1.0466E+01
29.0	1.0513E+01	30.0	1.0559E+01										

AVERAGE PROJECTED RECOIL RANGE(ANGSTROMS) RP-REC

E	RP-REC	E	RP-REC	E	RP-REC	E	RP-REC	E	RP-REC	E	RP-REC	E	RP-REC
1.0	1.0987E+01	2.0	2.1217E+01	3.0	3.0553E+01	4.0	3.8085E+01	5.0	4.6147E+01	6.0	5.2952E+01	7.0	6.2280E+01
8.0	7.0307E+01	9.0	7.8506E+01	10.0	8.6804E+01	11.0	9.5072E+01	12.0	1.0281E+02	13.0	1.0948E+02	14.0	1.1595E+02
15.0	1.2193E+02	16.0	1.2755E+02	17.0	1.3308E+02	18.0	1.3842E+02	19.0	1.4368E+02	20.0	1.4890E+02	21.0	1.5403E+02
22.0	1.5905E+02	23.0	1.6401E+02	24.0	1.6899E+02	25.0	1.7410E+02	26.0	1.7917E+02	27.0	1.8415E+02	28.0	1.8919E+02
29.0	1.9420E+02	30.0	1.9916E+02										

RMS SPREAD IN RP-REC

E	DRP-REC	E	DRP-REC	E	DRP-REC	E	DRP-REC	E	DRP-REC	E	DRP-REC	E	DRP-REC
1.0	1.7187E+01	2.0	2.6423E+01	3.0	3.1994E+01	4.0	3.5596E+01	5.0	4.2376E+01	6.0	4.6861E+01	7.0	5.5666E+01
8.0	6.2454E+01	9.0	6.9056E+01	10.0	7.6356E+01	11.0	8.3753E+01	12.0	9.0132E+01	13.0	9.6627E+01	14.0	1.0300E+02
15.0	1.0887E+02	16.0	1.1485E+02	17.0	1.2079E+02	18.0	1.2657E+02	19.0	1.3263E+02	20.0	1.3871E+02	21.0	1.4462E+02
22.0	1.5058E+02	23.0	1.5662E+02	24.0	1.6268E+02	25.0	1.6883E+02	26.0	1.7503E+02	27.0	1.8112E+02	28.0	1.8735E+02
29.0	1.9358E+02	30.0	1.9976E+02										

E	EP	RP(E,EP)	SPREAD IN RP-PARA	R(E,EP)	SPREAD IN RP-PERP	E	EP	RP(E,EP)	SPREAD IN RP-PARA	R(E,EP)	SPREAD IN RP-PERP
20	1	238.78	102.27	273.37	99.36	10	1	114.45	53.38	124.66	41.67
30	1	363.32	148.30	422.32	153.42	20	1	238.78	102.27	273.37	99.36
20	2	234.04	101.74	267.81	95.89	10	2	108.64	52.80	118.84	38.75
30	2	359.06	147.66	416.80	149.86	20	2	234.04	101.74	267.81	95.89
20	3	227.31	100.41	257.52	90.37	10	3	100.18	50.63	108.13	33.17
30	3	353.64	146.46	406.56	144.53	20	3	227.31	100.41	257.52	90.37
20	4	219.67	98.58	246.06	84.23	10	4	90.32	47.50	95.97	26.99
30	4	346.24	144.96	395.20	138.61	20	4	219.67	98.58	246.06	84.23
20	5	211.31	96.36	234.07	77.79	10	5	79.19	43.50	82.85	20.68
30	5	338.84	143.20	383.36	132.36	20	5	211.31	96.36	234.07	77.79
20	6	202.37	93.76	221.78	71.23	10	6	66.72	38.59	68.79	14.48
30	6	330.98	141.19	371.30	125.93	20	6	202.37	93.76	221.78	71.23

20	7	192.87	90.77	209.24	64.63	10	7	52.57	32.75	53.47	8.46
30	7	322.68	138.94	359.07	119.39	20	7	192.87	90.77	209.24	64.63
20	8	182.84	87.40	196.48	58.07	10	8	35.96	26.15	36.09	2.19
30	8	313.97	136.47	346.71	112.79	20	8	182.84	87.40	196.48	58.07
20	9	172.28	83.64	183.48	51.57	10	9	14.57	20.01	14.35	1.00
30	9	304.90	133.76	334.24	106.18	20	9	172.28	83.64	183.48	51.57
30	10	295.46	130.81	321.64	99.57	20	10	161.18	79.45	170.20	45.18
30	20	179.81	86.54	195.27	37.40	20	20	0.00	0.00	0.00	0.00
20	0	243.52	102.80	278.93	102.83	10	0	120.26	53.95	130.49	44.59
30	0	367.58	148.95	427.85	156.99	20	0	243.52	102.80	278.93	102.83

IV. DAMG2

A. General equations

The code DAMG2 calculates the distribution in depth of the energy deposited into atomic processes, $Q_D(E,x)$, or the distribution in depth of the energy deposited into electronic processes, $Q_I(E,x)$. These quantities are obtained by integrating the equations

$$Q_D(E,x) = \int_0^E S_D(E,E',X)\ \Sigma(E')\ (dR/dE')\ dE' \tag{34}$$

and

$$Q_I(E,x) = \int_0^E [P(E,E',x)\ S(E) + S_I(E,E'x)\ \Sigma_I(E')]\ (dR/dE')\ dE' \quad . \tag{35}$$

The functions $S_D(E,E',x)$, $P(E,E',x)$, and $S_I(E,E',x)$ are all Gaussian functions of the form $w(x)$, where

$$w(x) = (2\pi s^2)^{-1/2} \exp[-\ (x - r)^2/2s^2] \quad . \tag{36}$$

The values of s^2 and r for each of the functions S_D, S_I and P are listed in Table II. The quantities D, ΔD, I and ΔI are as defined in Eqs. (32) and (33) and the subsequent discussion.

TABLE II

Values of r and s^2 for the functions S_D, S_I and P.

Function	r	s^2
$S_D(E,E',x)$	$R_p(E,E') + D(E')$	$\Delta R_p^2(E,E') + \Delta D^2(E')$
$S_I(E,E',x)$	$R_p(E,E') + I(E')$	$\Delta R_p^2(E,E') + \Delta I^2(E')$
$P(E,E',x)$	$R_p(E,E')$	$\Delta R_p^2(E,E')$

The quantity (dR/dE') appearing in both Eqs. (34) and (35) is simply $dR(E,E')/dE'$, and $\Sigma(E')$ and $\Sigma_I(E')$ are the averages from Eqs. (23) and (27). The functions S_D and S_I account for the redistribution of energy by recoiling target atoms. This redistribution is taken care of in a Gaussian approximation through the parameters r and s which appear in Table II. For no recoil redistribution the parameters D, ΔD, I, and ΔI would all be set to zero.

B. User's Guide

i. General

The code DAMG2 calculates either $Q_D(E,x)$ or $Q_I(E,x)$ from Eqs. (34) or (35). The user can choose which function is desired by setting the value of the input variable NTYPE (see input below). The range and straggling data from which the functions dR/dE', S_D, S_I, P are determined, along with the values of Σ and Σ_I, are obtained from TAPE20. In performing the integrations to obtain the Q functions, DAMG2 uses rectangular integration. Linear interpolation at incremented values of E' is used to obtain values of the integrands from

the specific energies at which calculations have been performed by RASE4. The actual integration is performed in the code by incrementing R, rather than E'.

Distributions Q are calculated for each of the energies, E, for which range and straggling data are available on TAPE20 in a given data block. This number is NE/NSTP for a given run of RASE4. The distributions at the lowest two or three energies will be less accurate than those at higher energies because of the linear interpolation described above. The user can correct the output at the lowest energies by noting that the fraction of energy which goes into atomic processes is a decreasing function of energy with

$$q_o(E)/E \simeq 1 - B\ E^{1/6} \tag{37}$$

at low incident energies. Here B is a constant and,

$$q_o(E) = \int_o^\infty Q_D(E,x)\ dx \quad . \tag{38}$$

The correction in the low energy Q_D distributions consists of calculating the value of B in Eq. (37) using a medium energy value of q_o/E. The low energy q_o/E values are then calculated using the above formula, Eq. (37). Finally, the low energy $Q_D(E,x)$ are multiplied by the ratio of the new q_o/E values to the old q_o/E values output by DAMG2. Overall accuracies of better than 5% should be obtained for the function Q_D when the low energy distributions are corrected in this manner.

ii. Input

The input to DAMG2 consists of data on TAPE20 and four data cards.

Card 1

Input This is a header card and can contain any titular material the user desires. This information will appear as the first line of output for each category of printed output.

Format (80H ----)

Example

```
S1 INCIDENT ON S102    30 KEV
```

```
0 000000000 0000 00000000 000 0000000000000000000000000000000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
1111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

Card 2

Input ALAT, P1, Z1, (P2(J), Z2(J), J = 1,4)

Format (11F5.0)

ALAT, P1, Z1, (P2(J), Z2(J), J = 1,4) all have the same meaning as on Input card 3 of COREL and RASE4. This information is used in a search of TAPE20 for appropriate input to DAMG2.

Example

```
3.566  28.  14.  28.  14.  16.   8.   0.   0.   0.   0.   0.   0.   0.   0.   0.
```

```
0000000000000000000000000000000000000000 0000 0000 0000 0000 0000 0000 0000 0000 C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
1111111111111 11111111 1111 1111111111111111111111111111111111111111111111111111
```

Card 3

Input NE, NSTP, NDIV, NDIVE, NMULT, NTYPE, JSTP, NXTND, KE

Format (9I5)

NE, NSTP, NDIV, NDIVE and NMULT all have the same meaning as on input card 4 of COREL and RASE4. They should have the same values here as they had in the RASE4 calculation. These variables are used to set the energy scales in DAMG2, and to search TAPE20 for correct input. Calculations of the energy deposition distributions are carried out for all incident energies for which RASE4 wrote range and straggling variables to TAPE20.

NTYPE identifies the type of energy deposition distribution desired. For NTYPE = 2 the depth distribution of energy into electronic processes is calculated. For all other values of NTYPE the depth distribution of energy into atomic processes is calculated.

JSTP is a variable which determines the spacing of a portion of the printed output. The main printed output consists of Q(E,x) vs x for equally spaced values of x. Output is also printed for Q(E,x) vs x for equally spaced values of Q. This latter output is useful in constructing contour plots of Q(E,x) on the (E,x) plane. The equally spaced values of Q are in multiples of JSTP/10, with units of eV/Angstrom. Note that JSTP is changed to a real variable before the division

by 10 so that Q-values can be spaced as closely as 0.1 eV/Angstrom.

NXTND is a variable which may be used to decrease the size of the integration interval for Eqs. (34), (35) and (37). The integration interval size is reduced by a factor of 1/NXTND.

KE is a variable which may be used to select closely spaced values of the averages of Section III for use in evaluating the integrals in Eqs. (34) and (35) when E' drops below a preset value. The preset value in keV is KE. For KE $\neq$ 0 DAMG2 will select from TAPE20 two sets of values for the averages. The main set will correspond to the incident energies determined by NE, NSTP, NMULT, NDIV and NDIVE. The secondary set will correspond to the energy KE. This variable proves useful only when the averages vary rapidly with energy at low energies. When using this variable it is necessary to have low energy data available on TAPE20 from a previous calculation by RASE4. It is also assumed that the low energy data precedes the high energy data on TAPE20. The search for low energy data occurs first, and TAPE20 is not rewound prior to searching for the high energy data.

Example

30 10 1 1 1 1 50 1 0 0

```
0000 0000 0000000000000000000000000 000000000 0000 0000000000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
11111111 11111 1111 1111 1111 111111111 1111111111111111111111111111111111111111
```

Card 4

Input NRPT

Format (I5)

NRPT determines when the last set of input data has been read. For NRPT $\leq$ 0, the program terminates after carrying out the calculations of this input data set. For NRPT > 0 the program reads another set of input data after completing the calculations for the present four cards.

Example

1

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000C
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
1111 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

iii. Output

The output from DAMG2 consists of output on TAPE30, and printed output. Two logical blocks of output are written on TAPE30 for each incident ion and energy distribution type considered. The output is indicated schematically in Fig. 3.

Ident. Block
Data Block

Figure 3. Data Block Structure Written on TAPE30 by DAMG2

The identification block contains NDAM, NE, NSTP, NDIV, NDIVE, NMULT, JSTP, NXTND, KE, ITER, ALAT, P1, Z1, (P2(J), Z2(J), J = 1,4). This information is written by instructions DAM 186 and DAM 187. The variable NDAM = NE/NSTP, and ITER is the value of |NDSKP| from the RASE4 calculation which provided the specific input used in the present calculations.

The data block contains ITER, (E(I), RP(I,1), DELTA(I,1), DP(I,1), R(I,1), F(I), G(I), XI00(I), XI01(I), XI02(I), XI03(I), XI20(I), XI21(I), XI30(I), XM0M1(I), XM0M2(I), XM0M3(I), DAMG(I,J), J = 1, 101), I = 1, NDAM). These variables are identified as follows:

E(I) = incident ion energy for I'th output group.

RP(I,1), DELTA(I,1), DP(I,1), R(I,1) = range and straggling variables for incident ions with E = E(I) and E' = 0.

F(I) = fraction of incident ion energy deposited into the energy category of interest (i.e., atomic or electronic processes).

G(I) = actual energy deposited into target in category of interest = $q_o(E)$ from Eq. (38).

XInm(I) variables are averages over the depth distribution of certain quantities of interest. These values include

averaging over that part of the distribution <u>which lies outside the target surface</u>. Specifically, the variable XInm(I) is obtained by replacing S_D, S_I, or P in Eqs. (34) or (35) by $s^n r^m$, where s and r are defined in Table II, for each of these functions. The variables XInm are used to evaluate the moments of the energy deposition distributions for comparison with other calculations of these quantities.

XM0M1, XM0M2 and XM0M3 are, respectively, the average depth of the energy deposition distribution, and the second and third central moments of the distribution. The units are Å, $Å^2$, $Å^3$, respectively. These are calculated using the variables XInm.

DAMG(I,J) = Q(E,x) where E = E(I) and x = RP(I,1)*(J - 1)/50.

The printed output is easily interpreted. First, all the input variables, along with values of NDAM and ITER are printed. Following this the depth distributions are output. For this output each page contains ten columns of information. The first column is a depth scale in units of 100*X/RP. This depth scale applies to the remaining nine columns of information which are subdivided into three groups of three columns. Within each group of three the first column contains P(E,0,x) from Eq. (36), but without the pre-exponential factor. The third column contains Q(E,x) from Eq. (34) or (35). The second column contains

$$\int_x^\infty Q(E,x')\, dx'/q_o(E),$$

where q_o is given by Eq. (38). Each group of three columns is preceded by header information which gives the incident energy, range and straggling variables, $q_o(E)$, and the energy partition. This last quantity is XI00/E, and represents the fraction of incident energy which is deposited into the category of interest, <u>including that which is deposited outside the target surface</u>.

The output described above is in equally spaced values of the depth variable, x. Next, Q(E,x) is output as a function of x, but at equally-spaced values of Q. This output is useful in constructing contours of equal Q-values in the (E-x) plane. After this output, some of the averages over the depth distributions are recapitulated as a function of incident ion energy. The final output consists of a tabulation as a function of I(E = E(I)) of the third central moment of the distribution as calculated, and a correction to the third moment which would result if non-Gaussian functions having skewness equal to standard deviation, s, were substituted for S_D, S_I and P in Eqs. (34) and (35).

Sample output, obtained from the example cards under input, is shown in Section IV-D. The sample output shown required 60 seconds of computation time. The code DAMG2 requires 154200_8 of core storage.

iv. Subroutines.

DAMG2 uses the subroutines TWENTY and WIND30.

The subroutine TWENTY searches TAPE20 for appropriate input from RASE4 calculations. The routine WIND30

position TAPE30 so that writing begins after the last data block written on a previous run of DAMG2. This allows the user to collect a library of output data. If it is not desired to maintain such a library the instruction DAM 112 and the subroutine WIND30 may be eliminated.

C. Brief Programmers Guide

Eqs. (34) or (35) are integrated by the rectangular method. The integrand is evaluated at the center of each integration interval, multiplied by the interval width, and summed. The integration interval is 1 angstrom near E' = E, and increases as E' decreases. As E' nears 0, the interval width decreases. Integration is accomplished by the DO loop beginning at instruction DAM 285. Integration is actually performed over the variable R(E,E') rather than E'. The variables in the loop have the following identification:

$$X = R(E,E')$$

$$Y = (dR/dE')dE'$$

$$DMG = \Sigma(E') \text{ from Eq. (23) or (27)}$$

$$XD = D(E') \text{ or } I(E') \text{ from Eq. (32)}$$

$$DXD = \Delta D(E') \text{ or } \Delta I(E') \text{ from Eq. (33)}$$

$$DMG1 = S(E') \text{ from Eq. (19) or (20)}$$

D. Sample Output

The sample output on the next five pages was obtained using as input the example cards of Section IV-Bii along with input from TAPE20 for the example output shown in Section III-E.

SI INCIDENT ON SIO2 30 KEV

ALAT	P1	Z1	P2(1)	Z2(1)	P2(2)	Z2(2)	P2(3)	Z2(3)	P2(4)	Z2(4)
3.566	28.000	14.000	28.000	14.000	16.000	8.000	0.000	0.000	0.000	0.000

NDAM	NE	NSTP	NDIV	NDIVE	NMULT	NTYPE	JSTP	NXTND	KE	ITER
3	30	10	1	1	1	1	50	1	0	1

SI INCIDENT ON SIO2 30 KEV

**** UNITS *****

ENERGY - KEV
DEPTH,RANGE,DELTA RANGE - ANGSTROMS
ENERGY PARTITION,ION DISTRIBUTION,INTEGRATED ENERGY DISTRIBUTION - DIMENSIONLESS
ENERGY DISTRIBUTION - EV/ANGSTROM

INCIDENT ENERGY	***	INCIDENT ENERGY	***	INCIDENT ENERGY
10.00-KEV	***	20.00-KEV	***	30.00-KEV
PROJ. RANGE	***	PROJ. RANGE	***	PROJ. RANGE
120-ANGSTROMS	***	244-ANGSTROMS	***	368-ANGSTROMS
DELTA RP	***	DELTA RP	***	DELTA RP
54-ANGSTROMS	***	103-ANGSTROMS	***	149-ANGSTROMS
DELTA PERPINDICULAR	***	DELTA PERPINDICULAR	***	DELTA PERPINDICULAR
45-ANGSTROMS	***	103-ANGSTROMS	***	157-ANGSTROMS

TOTAL ENERGY IN DISTRIBUTION

5.86-KEV

ENERGY PARTITION

.6333

X/RP (PCT)	FINAL ION DISTR.	INTEGRATED ENERGY DISTR.	ENERGY DISTR.
0	.0834	1.0000	16.03
2	.0920	.9932	17.07
4	.1013	.9860	18.14
6	.1113	.9783	19.23
8	.1221	.9702	20.35
10	.1337	.9616	21.49
12	.1460	.9525	22.64
14	.1592	.9430	23.81
16	.1732	.9330	24.99
18	.1881	.9225	26.17
20	.2039	.9115	27.35
22	.2206	.9000	28.53
24	.2381	.8881	29.70
26	.2565	.8757	30.86
28	.2758	.8627	32.00
30	.2960	.8494	33.11
32	.3170	.8356	34.20
34	.3388	.8213	35.25
36	.3614	.8066	36.27
38	.3848	.7915	37.25
40	.4088	.7761	38.17
42	.4335	.7602	39.05
44	.4588	.7440	39.87
46	.4846	.7275	40.63
48	.5108	.7107	41.33
50	.5373	.6936	41.97
52	.5641	.6762	42.53
54	.5911	.6587	43.02
56	.6182	.6409	43.44
58	.6451	.6230	43.79
60	.6720	.6050	44.05
62	.6985	.5869	44.24
64	.7247	.5687	44.35
66	.7503	.5505	44.38
68	.7754	.5322	44.33
70	.7996	.5141	44.21
72	.8230	.4960	44.01
74	.8454	.4780	43.73

TOTAL ENERGY IN DISTRIBUTION

12.44-KEV

ENERGY PARTITION

.6596

FINAL ION DISTR.	INTEGRATED ENERGY DISTR.	ENERGY DISTR.
.0605	1.0000	15.29
.0676	.9938	16.53
.0753	.9870	17.82
.0838	.9798	19.15
.0930	.9720	20.52
.1030	.9637	21.94
.1139	.9549	23.38
.1255	.9454	24.85
.1381	.9354	26.35
.1516	.9248	27.85
.1660	.9136	29.37
.1814	.9018	30.88
.1978	.8894	32.39
.2151	.8764	33.87
.2335	.8629	35.34
.2529	.8488	36.77
.2732	.8341	38.15
.2946	.8189	39.49
.3169	.8032	40.77
.3401	.7870	41.99
.3642	.7703	43.13
.3891	.7532	44.20
.4148	.7357	45.18
.4412	.7179	46.07
.4683	.6997	46.86
.4959	.6812	47.56
.5239	.6624	48.16
.5523	.6435	48.65
.5809	.6244	49.03
.6096	.6051	49.31
.6383	.5858	49.47
.6669	.5664	49.54
.6951	.5470	49.49
.7230	.5276	49.35
.7503	.5084	49.10
.7768	.4892	48.75
.8025	.4702	48.31
.8272	.4514	47.79

TOTAL ENERGY IN DISTRIBUTION

18.23-KEV

ENERGY PARTITION

.6411

FINAL ION DISTR.	INTEGRATED ENERGY DISTR.	ENERGY DISTR.
.0476	1.0000	14.07
.0537	.9941	15.25
.0604	.9877	16.47
.0678	.9808	17.74
.0760	.9734	19.06
.0849	.9654	20.42
.0946	.9569	21.81
.1052	.9478	23.23
.1167	.9382	24.67
.1291	.9279	26.13
.1424	.9171	27.61
.1568	.9056	29.08
.1723	.8936	30.56
.1887	.8810	32.02
.2063	.8678	33.46
.2249	.8540	34.88
.2446	.8397	36.26
.2654	.8248	37.61
.2873	.8093	38.90
.3102	.7934	40.14
.3341	.7770	41.31
.3590	.7601	42.41
.3848	.7428	43.44
.4115	.7251	44.38
.4390	.7070	45.24
.4671	.6886	46.01
.4958	.6699	46.68
.5250	.6510	47.25
.5546	.6318	47.73
.5844	.6125	48.10
.6143	.5930	48.37
.6442	.5735	48.53
.6739	.5539	48.59
.7033	.5343	48.55
.7321	.5148	48.41
.7603	.4953	48.16
.7876	.4759	47.82
.8140	.4567	47.39

76	.8667	.4601	43.38	.8508	.4328	47.17	.8391	.4377	46.87
78	.8867	.4424	42.95	.8730	.4145	46.48	.8630	.4189	46.26
80	.9054	.4248	42.46	.8938	.3964	45.71	.8853	.4004	45.57
82	.9227	.4075	41.91	.9131	.3787	44.87	.9061	.3822	44.80
84	.9384	.3904	41.28	.9307	.3613	43.97	.9250	.3643	43.96
86	.9525	.3736	40.60	.9465	.3443	43.00	.9421	.3467	43.05
88	.9649	.3571	39.87	.9604	.3276	41.99	.9571	.3296	42.08
90	.9755	.3409	39.08	.9723	.3114	40.93	.9700	.3128	41.06
92	.9842	.3250	38.24	.9822	.2956	39.83	.9807	.2965	39.98
94	.9911	.3095	37.36	.9899	.2802	38.69	.9891	.2806	38.86
96	.9960	.2944	36.44	.9955	.2653	37.52	.9951	.2651	37.70
98	.9990	.2796	35.48	.9989	.2509	36.32	.9988	.2502	36.50
100	1.0000	.2652	34.49	1.0000	.2369	35.11	1.0000	.2357	35.28
102	.9990	.2513	33.47	.9989	.2234	33.88	.9988	.2217	34.03
104	.9960	.2378	32.43	.9955	.2103	32.64	.9951	.2082	32.76
106	.9911	.2247	31.37	.9899	.1978	31.40	.9891	.1953	31.49
108	.9842	.2120	30.30	.9822	.1857	30.15	.9807	.1828	30.20
110	.9755	.1998	29.21	.9723	.1742	28.91	.9700	.1709	28.91
112	.9649	.1880	28.12	.9604	.1631	27.67	.9571	.1595	27.63
114	.9525	.1767	27.02	.9465	.1525	26.45	.9421	.1486	26.35
116	.9384	.1658	25.93	.9307	.1424	25.23	.9250	.1383	25.09
118	.9227	.1554	24.84	.9131	.1327	24.04	.9061	.1284	23.84
120	.9054	.1454	23.76	.8938	.1236	22.86	.8853	.1190	22.68
122	.8867	.1359	22.68	.8730	.1148	21.70	.8630	.1102	21.39
124	.8667	.1268	21.63	.8508	.1066	20.57	.8391	.1018	20.21
126	.8454	.1182	20.58	.8272	.0987	19.47	.8140	.0939	19.06
128	.8230	.1099	19.56	.8025	.0913	18.39	.7876	.0864	17.93
130	.7996	.1021	18.56	.7768	.0843	17.35	.7603	.0794	16.84
132	.7754	.0947	17.58	.7503	.0777	16.33	.7321	.0728	15.79
134	.7503	.0877	16.62	.7230	.0715	15.35	.7033	.0666	14.77
136	.7247	.0810	15.70	.6951	.0657	14.41	.6739	.0609	13.79
138	.6985	.0748	14.80	.6669	.0602	13.49	.6442	.0555	12.85
140	.6720	.0689	13.93	.6383	.0551	12.62	.6143	.0505	11.94
142	.6451	.0633	13.08	.6096	.0503	11.78	.5844	.0459	11.09
144	.6182	.0581	12.28	.5809	.0459	10.97	.5546	.0416	10.27
146	.5911	.0532	11.50	.5523	.0417	10.20	.5250	.0376	9.489
148	.5641	.0487	10.75	.5239	.0379	9.473	.4958	.0339	8.752
150	.5373	.0444	10.04	.4959	.0343	8.778	.4671	.0305	8.056
152	.5108	.0404	9.356	.4683	.0310	8.119	.4390	.0274	7.401
154	.4846	.0367	8.707	.4412	.0280	7.495	.4115	.0245	6.784
156	.4588	.0333	8.090	.4148	.0251	6.907	.3848	.0219	6.206
158	.4335	.0301	7.504	.3891	.0225	6.353	.3590	.0195	5.665
160	.4088	.0271	6.949	.3642	.0202	5.832	.3341	.0173	5.161
162	.3848	.0244	6.425	.3401	.0180	5.343	.3102	.0153	4.692
164	.3614	.0218	5.931	.3169	.0160	4.886	.2873	.0135	4.256
166	.3388	.0195	5.466	.2946	.0141	4.460	.2654	.0119	3.852
168	.3170	.0173	5.029	.2732	.0125	4.063	.2446	.0104	3.480
170	.2960	.0153	4.619	.2529	.0109	3.694	.2249	.0091	3.137
172	.2758	.0135	4.236	.2335	.0096	3.352	.2063	.0079	2.821
174	.2565	.0119	3.878	.2151	.0083	3.035	.1887	.0068	2.532
176	.2381	.0103	3.545	.1978	.0072	2.743	.1723	.0058	2.267
178	.2206	.0089	3.235	.1814	.0062	2.474	.1568	.0050	2.026
180	.2039	.0077	2.947	.1660	.0052	2.227	.1424	.0042	1.807
182	.1881	.0065	2.680	.1516	.0044	2.001	.1291	.0035	1.607
184	.1732	.0055	2.433	.1381	.0037	1.794	.1167	.0029	1.427
186	.1592	.0045	2.206	.1255	.0030	1.605	.1052	.0024	1.264

188	.1460	.0037	1.996	.1139	.0024	1.433	.0946	.0019	1.117
190	.1337	.0029	1.803	.1030	.0019	1.277	.0849	.0015	.9853
192	.1221	.0022	1.626	.0930	.0014	1.136	.0760	.0011	.8671
194	.1113	.0015	1.464	.0838	.0010	1.008	.0678	.0008	.7613
196	.1013	.0010	1.316	.0753	.0006	.8929	.0604	.0005	.6669
198	.0920	.0005	1.181	.0676	.0003	.7892	.0537	.0002	.5830
200	.0834	0.0000	1.058	.0605	0.0000	.6961	.0476	0.0000	.5084

SI INCIDENT ON SIO2 30 KEV

VALUES FOR CONTOUR PLOT ENERGY DISTRIBUTION(EV/ANGSTROM) DEPTH(ANGSTROMS)

ENERGY(KEV)	10		20		30
20.0	9	20.0	18	15.0	6
25.0	19	25.0	35	20.0	35
30.0	29	30.0	51	25.0	60
35.0	40	35.0	67	30.0	85
40.0	53	40.0	85	35.0	111
40.0	105	45.0	106	40.0	139
35.0	119	45.0	199	45.0	174
30.0	131	40.0	223	45.0	300
25.0	142	35.0	244	40.0	338
20.0	153	30.0	264	35.0	369
15.0	165	25.0	283	30.0	398
10.0	181	20.0	304	25.0	427
5.0	202	15.0	328	20.0	457
0.0	0	10.0	357	15.0	491
0.0	0	5.0	398	10.0	532
0.0	0	0.0	0	5.0	591

SI INCIDENT ON SIO2 30 KEV

ENERGY	X-AVERAGE	DELTA X-SQ AVERAGE	ENERGY PARTITION	TOTAL ENERGY IN DISTRIBUTION
10	82.076	3218.9	.63326	6.3326
			ENERGY DEPOSITED IN TARGET =	5.8589
20	164.14	11298	.65959	13.192
			ENERGY DEPOSITED IN TARGET =	12.439
30	249.45	24495	.64114	19.234
			ENERGY DEPOSITED IN TARGET =	18.227

SI INCIDENT ON SIO2 30 KEV

I	DPCT	XMOM3(I)	DXMOM3(I)
1	.633264	1.6023E+04	6.0347E+04
2	.659591	1.9583E+05	4.0147E+05
3	.641137	4.6916E+05	1.2600E+06

V. LISTINGS OF MAIN PROGRAMS

A. COREL

```
      PROGRAM COREL(INPUT,OUTPUT,TAPE10)                                COR 0001
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                    ISSUED BY SANDIA LABORATORIES.                *
C     *               A PRIME CONTRACTOR TO THE UNITED STATES            *
C     *           ENERGY RESEARCH AND DEVELOPEMENT ADMINISTRATION.       *
C     * * * * * * * * * * * * * * *   NOTICE   * * * * * * * * * * * * * *
C     * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE  *
C     * UNITED STATES GOVERNMENT. NEITHER THE UNITED STATES NOR THE      *
C     * UNITED STATES ENERGY RESEARCH AND DEVELOPEMENT ADMINISTRATION,   *
C     * NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS,        *
C     * SUBCONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS  *
C     * OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR *
C     * THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,     *
C     * APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS  *
C     * USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.                   *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * THE BASIC REFERENCE DOCUMENTS FOR THIS CODE ARE SLA-73-0416,     *
C     * APRIL,1973, AND SAND-75-0622, DECEMBER, 1975.                    *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * THIS CODE HAS BEEN RELEASED FOR UNLIMITED DISTRIBUTION.          *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

    1 FORMAT(16F5.0)                                                    COR 0043
    2 FORMAT(16I5)                                                      COR 0044
    3 FORMAT(2/)                                                        COR 0045
    4 FORMAT(8A10)                                                      COR 0046
    5 FORMAT(1H1,* RANGE AND STRAGGLING DATA FOR *,A10,* IONS INCIDENT OCOR 0047
     1N A/AN *,2A10,* TARGET *,4/)                                      COR 0048
    6 FORMAT(* INITIAL VALUES*3/)                                       COR 0049
    7 FORMAT(*  INCIDENT ION *,A10,/*  ATOMIC NUMBER(Z1) = *,F4.0,/*  ATCOR 0050
     1OMIC MASS(M1) = *,F7.2,* AMU*/*  ENERGY MAXIMUM = *,F10.2,*KEV*2/)COR 0051
    8 FORMAT(*  TARGET - *,2A10, /)                                     COR 0052
    9 FORMAT(*  COMPONENTS    *,4(5X,A10,5X),/*  ATOMIC NUMBER *,4(F10.0COR 0053
     1,10X),/*  ATOMIC MASS    *, 4(F8.2,* AMU *,7X)/*  T-F RADIUS    *,4COR 0054
     2(E10.3,* CM *,6X),/*  EFFECTIVE Z    *,4(F10.3,10X))              COR 0055
 1000 FORMAT(*    I      U(I)      F(U)      SLOPE    -   F(U) IS UNIVERSACOR 0056
     1L THOMAS-FERMI FUNCTION - FROM LSS*2/)                            COR 0057
 1001 FORMAT(I5,2F10.3,F10.5)                                           COR 0058
 1002 FORMAT(4/* THE FOLLOWING ELECTRONIC STOPPING CROSS SECTIONS WERE  COR 0059
     1USED IN THE CALCULATIONS*2/)                                      COR 0060
 1003 FORMAT(3X,A10,3X,*-   THOMAS-FERMI ATOMIC MODEL*)                 COR 0061
 1004 FORMAT(5/19H*****  UNITS  ***** ,/*  ENERGY  -  KEV*/*  RP,R,DELTACOR 0062
```

```
     1,DP  -  ANGSTROMS*/*  EPS,DE/DR  -  DIMENSIONLESS*/                COR 0063
     2*  DAMAGE  -  EV/ANGSTROM*2/)                                      COR 0064
 1005 FORMAT(3X,A10,3X,*-   FIRSOV THEORY*)                              COR 0065
 1006 FORMAT(3X,A10,3X,21H-   K*E**P , WITH K= ,E11.4,*   AND   P = *,F4COR 0066
     1.2,2/)                                                             COR 0067
 1007 FORMAT(7X,*CSE = *,E12.5,2/)                                       COR 0068
 1008 FORMAT(3X,A10,3X,*-   BRICE FORMULA*)                              COR 0069
 1009 FORMAT(7X,*Z = *,F6.3,5X,* A = *,F7.4,5X,* N = *,F6.3,2/)          COR 0070
 1031 FORMAT(A10,* INCIDENT ON A/AN *,2A10,* TARGET *)                   COR 0071
 1032 FORMAT(I4,3I2,/,(8E10.4))                                          COR 0072
 1033 FORMAT(3/,*      NE     NSTP  NCOMP          NFORM(I)              COR 0073
     1     NEXP(J)              NDIV  NDIVE  NMULT  NDSKP   NRPT*/21X,*  COR 0074
     2I = 1      2      3      4  J = 1      2      3      4*2/)         COR 0075
 1034 FORMAT(16I7,2/)                                                    COR 0076
 1040 FORMAT(3/*       E      EP RP(E,EP)   SPREAD IN     R(E,EP)   SPREADCOR 0077
     1 IN              E      EP RP(E,EP)   SPREAD IN     R(E,EP)   SPREADCOR 0078
     2 IN*/    *                             RP-PARA                  RP-PECOR 0079
     3RP                                     RP-PARA                  RP-PECOR 0080
     4RP*2/)                                                             COR 0081
 1041 FORMAT(*  CHAR. ENERGY  *,4(F8.3,* KEV *,7X),2/)                   COR 0082
 1042 FORMAT(2/,*  CHARACTERISTIC LENGTH = *,F7.2,* ANGSTROMS*,2/)       COR 0083
 1043 FORMAT(2/*  ATOMIC DENSITY = *,E12.5,* ATOMS/CC*2/)                COR 0084
      DIMENSION SETA(4),ALP(4),EN(4)                                     COR 0085
      DIMENSION A(4)                                                     COR 0086
      DIMENSION ALPHA(1600),ALPHA1(1600),ALPHA2(1600),ALPHA3(1600)       COR 0087
      DIMENSION BETA1(1600),BETA2(1600),BETA3(1600)                      COR 0088
      DIMENSION CSE(4),CEPS(4),CTM(4),CALPH(4)                           COR 0089
      DIMENSION              DELTA(1600),DP(1600),          DR(1600)    COR 0090
      DIMENSION DRO(1600),DRR2(1600),DRC2(1600),DENS(4)                  COR 0091
      DIMENSION EXINT1(1600),EXINT2(1600),EXINT3(1600)                   COR 0092
      DIMENSION E(1600),EF(50)                                           COR 0093
      DIMENSION F(50)                                                    COR 0094
      DIMENSION GAM1(1600),GAM2(1600),GAM3(1600)                         COR 0095
      DIMENSION IVAR(8),NEXP(4),NFORM(4)                                 COR 0096
      DIMENSION P2(4)                                                    COR 0097
      DIMENSION            RP(1600),          RP2(1600),RR2(1600)        COR 0098
      DIMENSION RC2(1600),R(1600)                                        COR 0099
      DIMENSION SE(1600),SLF(50)                                         COR 0100
      DIMENSION TITLE(8)                                                 COR 0101
      DIMENSION Z2(4),Z(4)                                               COR 0102
      COMMON/PHIVAR/E,P1,P2                                              COR 0103
      COMMON/FVAR/F,SLF,EF,NF,MM                                         COR 0104
      NRPT = 0                                                           COR 0105
      CALL WIND10                                                        COR 0106
  350 CONTINUE                                                           COR 0107
      IF(NRPT)255,355,255                                                COR 0108
  355 CONTINUE                                                           COR 0109
      PI = 3.141592654                                                   COR 0110
      HBAR = (6.6254E-27)/(2.*PI)                                        COR 0111
      Q = 4.80294E-10                                                    COR 0112
      AO = 0.529484E-8                                                   COR 0113
      ERGEV = 1.60203E-12                                                COR 0114
      AMU = 1.66035E-24                                                  COR 0115
      DO 115 J = 1,4                                                     COR 0116
      P2(J) = 0.                                                         COR 0117
      Z(J) = 0.                                                          COR 0118
      A(J) = 0.                                                          COR 0119
      NEXP(J) = 0                                                        COR 0120
      NFORM(J) = 0                                                       COR 0121
  115 CONTINUE                                                           COR 0122
      READ 4,(TITLE(I),I = 1,8)                                          COR 0123
      READ 4 ,(IVAR(J) , J = 1,8)                                        COR 0124
```

```
      READ 1,ALAT,P1,Z1,(P2(J),Z2(J),J = 1,4),(CSE(K),K = 1,4),EQM      COR 0125
      EX = ((Q**2/HBAR)**2)*AMU/(2000.*ERGEV)                           COR 0126
      X = 4./3.                                                         COR 0127
      X = Z1**X                                                         COR 0128
      EX = EX*P1*X                                                      COR 0129
      PRINT 5,(TITLE(J),J=1,3)                                          COR 0130
      PRINT 6                                                           COR 0131
      PRINT 7,TITLE(1),Z1,P1,EX                                         COR 0132
      PRINT 8,TITLE(2),TITLE(3)                                         COR 0133
      READ 2,NE,NSTP,NCOMP,(NFORM(J),J=1,4),(NEXP(K),K=1,4),NDIV,NDIVE  COR 0134
     1,NMULT,NDSKP,NRPT                                                 COR 0135
      READ 1,(SETA(J),ALP(J),EN(J),J=1,4)                               COR 0136
      IF(NDIVE.GT.1) GO TO 118                                          COR 0137
      WRITE TAPE10,NE,NDIV,NMULT,NCOMP,ALAT,(P2(J),Z2(J),CSE(J),SETA(J),COR 0138
     1ALP(J),EN(J),NFORM(J),NEXP(J),J=1,4)                              COR 0139
  118 CONTINUE                                                          COR 0140
      IF(NDIV)121,121,122                                               COR 0141
  121 CONTINUE                                                          COR 0142
      NDIV = 1                                                          COR 0143
  122 CONTINUE                                                          COR 0144
      IF(NDIVE)123,123,124                                              COR 0145
  123 CONTINUE                                                          COR 0146
      NDIVE = 1                                                         COR 0147
  124 CONTINUE                                                          COR 0148
      ALAT = ALAT*1.E-8                                                 COR 0149
      ALAT = 1./(ALAT**3)                                               COR 0150
      X = 2./3.                                                         COR 0151
      CRHO = 0.                                                         COR 0152
      DO 1250 J=1,NCOMP                                                 COR 0153
      DENS(J) = NFORM(J)*ALAT                                           COR 0154
      CSE(J) = CSE(J)*1.E-26*DENS(J)                                    COR 0155
      Z(J) = Z1**X + Z2(J)**X                                           COR 0156
      Z(J) = SQRT(Z(J))                                                 COR 0157
      A(J) = .8853*A0/Z(J)                                              COR 0158
      Z(J) = Z(J)**3                                                    COR 0159
      CEPS(J) = ((A(J)*P2(J))/(Z1*Z2(J)*(P1+P2(J))*Q**2))*ERGEV*1000.   COR 0160
      CTM(J) = (4.*P1*P2(J))/((P1+P2(J))**2)                            COR 0161
      CRHO = CRHO + PI*A(J)**2*DENS(J)*CTM(J)                           COR 0162
      IF(NEXP(J))1202,1200,1201                                         COR 0163
 1202 CONTINUE                                                          COR 0164
      CSE(J) = 2.34E-34*(Z1+Z2(J))*DENS(J)                              COR 0165
      GO TO 1203                                                        COR 0166
 1200 CONTINUE                                                          COR 0167
      CSE(J) = DENS(J)*8.*PI*A0*HBAR*Z2(J)/Z(J)                         COR 0168
      Y = 7./6.                                                         COR 0169
      Y = Z1**Y                                                         COR 0170
      CSE(J) = CSE(J)*Y/(ERGEV*1.E+11)                                  COR 0171
 1203 CONTINUE                                                          COR 0172
      Y = 2./(P1*AMU)                                                   COR 0173
      Y = Y*ERGEV*1000.                                                 COR 0174
      Y = SQRT(Y)                                                       COR 0175
      CSE(J) = Y*CSE(J)                                                 COR 0176
 1201 CONTINUE                                                          COR 0177
      CALPH(J) = PI*DENS(J)*1.E-8*(A(J)**2)                             COR 0178
 1250 CONTINUE                                                          COR 0179
      PRINT 9,(TITLE(I),I=5,8),(Z2(J),J=1,4),(P2(K),K=1,4),(A(L),L=1,4),COR 0180
     1(Z(M),M=1,4)                                                      COR 0181
      EPS = 0.                                                          COR 0182
      DO 1255 J = 1,NCOMP                                               COR 0183
      Z(J) = 1./CEPS(J)                                                 COR 0184
      EPS = EPS + DENS(J)                                               COR 0185
 1255 CONTINUE                                                          COR 0186
```

```
      PRINT 1041,(Z(J) , J=1,4)                                         COR 0187
      CRHO = 1.E+8/CRHO                                                 COR 0188
      PRINT 1042,CRHO                                                   COR 0189
      PRINT 1043,EPS                                                    COR 0190
      PRINT 1002                                                        COR 0191
      DO 1302 J = 1,NCOMP                                               COR 0192
      IF(NEXP(J))1299,1301,1300                                         COR 0193
 1299 CONTINUE                                                          COR 0194
      IF(NEXP(J))12995,12991,12995                                      COR 0195
12991 CONTINUE                                                          COR 0196
      PRINT 1005,TITLE(J+4)                                             COR 0197
      PRINT 1007,CSE(J)                                                 COR 0198
      NEXP(J) = 0                                                       COR 0199
      GO TO 1302                                                        COR 0200
12995 CONTINUE                                                          COR 0201
      PRINT 1008,TITLE(J+4)                                             COR 0202
      PRINT 1009,SETA(J),ALP(J),EN(J)                                   COR 0203
      GO TO 1302                                                        COR 0204
 1300 CONTINUE                                                          COR 0205
      Y = NEXP(J)                                                       COR 0206
      Y = Y/100.                                                        COR 0207
      PRINT 1006,TITLE(J+4),CSE(J),Y                                    COR 0208
      GO TO 1302                                                        COR 0209
 1301 CONTINUE                                                          COR 0210
      PRINT 1003,TITLE(J+4)                                             COR 0211
      PRINT 1007,CSE(J)                                                 COR 0212
 1302 CONTINUE                                                          COR 0213
      PRINT 1033                                                        COR 0214
      PRINT 1034,NE,NSTP,NCOMP,(NFORM(J),J=1,4),(NEXP(K),K=1,4),NDIV,NDICOR 0215
     1VE,NMULT,NDSKP,NRPT                                               COR 0216
      CALL FSET                                                         COR 0217
      L = NF-1                                                          COR 0218
      PRINT 1000                                                        COR 0219
      DO 12 I = 1,L                                                     COR 0220
      PRINT 1001,I,EF(I),F(I),SLF(I)                                    COR 0221
   12 CONTINUE                                                          COR 0222
      NENT = 100                                                        COR 0223
      PRINT 1004                                                        COR 0224
      PRINT 1040                                                        COR 0225
      I = 1                                                             COR 0226
      DELTA(I) = 0.                                                     COR 0227
      DP(I) = 0.                                                        COR 0228
      DR(I) = 0.                                                        COR 0229
      DRO(I) = 0.                                                       COR 0230
      DRR2(I) = 0.                                                      COR 0231
      DRC2(I) = 0.                                                      COR 0232
      Y = NDIV                                                          COR 0233
      U = 0.5/Y                                                         COR 0234
      X = NMULT                                                         COR 0235
      U = U*X                                                           COR 0236
      EXINT1(I) = 0.                                                    COR 0237
      EXINT2(I) = 0.                                                    COR 0238
      EXINT3(I) = 0.                                                    COR 0239
      R(I) = 0.                                                         COR 0240
      RP(I) = 0.                                                        COR 0241
      RR2(I) = 0.                                                       COR 0242
      RC2(I) = 0.                                                       COR 0243
      RP2(I) = 0.                                                       COR 0244
      JJJ = 1                                                           COR 0245
      NE = NE + 1                                                       COR 0246
      DO 150 J = I,NE,JJJ                                               COR 0247
      SE(J) = 0.                                                        COR 0248
```

```
      ALPHA(J) = 0.                                                     COR 0249
      E(J) = J-1                                                        COR 0250
      Y = NDIV                                                          COR 0251
      E(J) = E(J)/Y                                                     COR 0252
      X = NMULT                                                         COR 0253
      E(J) = E(J)*X                                                     COR 0254
      ALPHA1(J) = 0.                                                    COR 0255
      ALPHA2(J) = 0.                                                    COR 0256
      ALPHA3(J) = 0.                                                    COR 0257
      BETA1(J) = 0.                                                     COR 0258
      BETA2(J) = 0.                                                     COR 0259
      BETA3(J) = 0.                                                     COR 0260
      GAM1(J) = 0.                                                      COR 0261
      GAM2(J) = 0.                                                      COR 0262
      GAM3(J) = 0.                                                      COR 0263
      K = J-JJJ                                                         COR 0264
      IF(J-I)150,1410,1399                                              COR 0265
 1399 CONTINUE                                                          COR 0266
      DO 149 LJ = 1,NCOMP                                               COR 0267
      NINT = NENT                                                       COR 0268
      TM = CTM(LJ)*E(J)                                                 COR 0269
      EPS = CEPS(LJ)*E(J)                                               COR 0270
      W = TM/EPS**2                                                     COR 0271
      CD = CALPH(LJ)*W                                                  COR 0272
      CG = CD*W                                                         COR 0273
      NQT = 4                                                           COR 0274
      X = NQT                                                           COR 0275
      X = X + ALOG10(EPS)                                               COR 0276
      NT = X                                                            COR 0277
      NT = NT - NQT                                                     COR 0278
      X = 2.                                                            COR 0279
      Y = 10.**NT                                                       COR 0280
      T = X*(EPS/Y - 1.)                                                COR 0281
      NSTP = T                                                          COR 0282
      Y = NSTP                                                          COR 0283
      OVER = T-Y                                                        COR 0284
      NX = X                                                            COR 0285
      NT = (NT+NQT)*NX                                                  COR 0286
      NT = 9*NT +     NSTP + 1                                          COR 0287
      NX = 9.*X                                                         COR 0288
      NQT = -NQT                                                        COR 0289
      Y = 10.**NQT                                                      COR 0290
      X = 0.5*Y                                                         COR 0291
      NSTP = -1                                                         COR 0292
      DO 90 L = 1,NT                                                    COR 0293
      IF(L-2)85,86,87                                                   COR 0294
   86 CONTINUE                                                          COR 0295
      T = NX                                                            COR 0296
      T = T/9.                                                          COR 0297
      T = (10.**NQT)/T                                                  COR 0298
      Y = T                                                             COR 0299
      X = (10.**NQT)-0.5*Y                                              COR 0300
   87 CONTINUE                                                          COR 0301
      X = X+Y                                                           COR 0302
   85 CONTINUE                                                          COR 0303
      T = W*(X**2)                                                      COR 0304
      CALL FET(FT,X)                                                    COR 0305
      CALL COPHI(CPHI,T,J,LJ)                                           COR 0306
      SPHI = 1.5*CPHI**2-.5                                             COR 0307
      TX = FT*Y/X**2                                                    COR 0308
      ALPHA(J) = ALPHA(J) + TX*CALPH(LJ)                                COR 0309
      ALPHA1(J) = ALPHA1(J) - CPHI*TX*CALPH(LJ)                         COR 0310
```

```
      ALPHA2(J) = ALPHA2(J) - SPHI*TX*CALPH(LJ)                         COR 0311
      ALPHA3(J) = ALPHA3(J) - TX*CALPH(LJ)                              COR 0312
      TX=TX*CALPH(LJ)*T                                                 COR 0313
      BETA1(J) = BETA1(J) + CPHI*TX                                     COR 0314
      BETA2(J) = BETA2(J) + SPHI*TX                                     COR 0315
      BETA3(J) = BETA3(J) + TX                                          COR 0316
      TX=TX*T/2.                                                        COR 0317
      GAM1(J) = GAM1(J) + CPHI*TX                                       COR 0318
      GAM2(J) = GAM2(J) + SPHI*TX                                       COR 0319
      GAM3(J) = GAM3(J) + TX                                            COR 0320
      NSTP = NSTP + 1                                                   COR 0321
      IF(NSTP-NX)90,89,89                                               COR 0322
   89 CONTINUE                                                          COR 0323
      X = X + 0.5*Y                                                     COR 0324
      NSTP = 0                                                          COR 0325
      Y = 10.*Y                                                         COR 0326
      X = X-0.5*Y                                                       COR 0327
   90 CONTINUE                                                          COR 0328
      X = X+0.5*Y                                                       COR 0329
      X = X + 0.5*OVER*Y                                                COR 0330
      Y = OVER*Y                                                        COR 0331
      T = W*(X**2)                                                      COR 0332
      CALL FET(FT,X)                                                    COR 0333
      CALL COPHI(CPHI,T,J,LJ)                                           COR 0334
      SPHI = 1.5*CPHI**2-.5                                             COR 0335
      TX = FT*Y/X**2                                                    COR 0336
      ALPHA(J) = ALPHA(J) + TX*CALPH(LJ)                                COR 0337
      ALPHA1(J) = ALPHA1(J) - CPHI*TX*CALPH(LJ)                         COR 0338
      ALPHA2(J) = ALPHA2(J) - SPHI*TX*CALPH(LJ)                         COR 0339
      ALPHA3(J) = ALPHA3(J) - TX*CALPH(LJ)                              COR 0340
      TX=TX*CALPH(LJ)*T                                                 COR 0341
      BETA1(J) = BETA1(J) + CPHI*TX                                     COR 0342
      BETA2(J) = BETA2(J) + SPHI*TX                                     COR 0343
      BETA3(J) = BETA3(J) + TX                                          COR 0344
      TX=TX*T/2.                                                        COR 0345
      GAM1(J) = GAM1(J) + CPHI*TX                                       COR 0346
      GAM2(J) = GAM2(J) + SPHI*TX                                       COR 0347
      GAM3(J) = GAM3(J) + TX                                            COR 0348
      IF(NEXP(LJ))83,84,82                                              COR 0349
   82 CONTINUE                                                          COR 0350
      X = NEXP(LJ)                                                      COR 0351
      X = X/100.                                                        COR 0352
      SE(J) = CSE(LJ)*(E(J)**X) + SE(J)                                 COR 0353
      GO TO 149                                                         COR 0354
   83 CONTINUE                                                          COR 0355
      CALL SECALC(Z1,Z2(LJ),P1,E(J),SET,SETA(LJ),ALP(LJ),EN(LJ))        COR 0356
      SE(J) = SE(J) + DENS(LJ)*1.E-8*SET                                COR 0357
      GO TO 149                                                         COR 0358
   84 CONTINUE                                                          COR 0359
      SE(J) = CSE(LJ)*SQRT(E(J)) + SE(J)                                COR 0360
  149 CONTINUE                                                          COR 0361
 1410 CONTINUE                                                          COR 0362
      ALPHA1(J) = ALPHA1(J) + ALPHA(J)                                  COR 0363
      ALPHA2(J) = ALPHA2(J) + ALPHA(J)                                  COR 0364
      ALPHA3(J) = ALPHA3(J) + ALPHA(J)                                  COR 0365
      BETA1(J) = BETA1(J) + SE(J)*(1.-EQM)                              COR 0366
      BETA2(J) = BETA2(J) + SE(J)*(1.-EQM)                              COR 0367
      BETA3(J) = BETA3(J) + SE(J)*(1.-EQM)                              COR 0368
      IF(J-I)150,92,93                                                  COR 0369
   92 CONTINUE                                                          COR 0370
      BETA1(J) = 1.E+100                                                COR 0371
      BETA2(J) = 1.E+100                                                COR 0372
```

```
      BETA3(J) = 1.E+100                                                COR 0373
      A = 0.                                                            COR 0374
      Z22= 0.                                                           COR 0375
      AO = 0.                                                           COR 0376
      A1 = 0.                                                           COR 0377
      A2 = 0.                                                           COR 0378
      Q = 0.                                                            COR 0379
      GO TO 150                                                         COR 0380
   93 CONTINUE                                                          COR 0381
      X = ALPHA1(J)/BETA1(J)                                            COR 0382
      Y = ALPHA2(J)/BETA2(J)                                            COR 0383
      Z = ALPHA3(J)/BETA3(J)                                            COR 0384
      W = GAM1(J)/BETA1(J)                                              COR 0385
      T = GAM2(J)/BETA2(J)                                              COR 0386
      TX= GAM3(J)/BETA3(J)                                              COR 0387
      EXINT1(J) = EXINT1(K) + U*(X+A)                                   COR 0388
      A = X                                                             COR 0389
      EXINT2(J) = EXINT2(K) + U*(Y+Z22)                                 COR 0390
      Z22=Y                                                             COR 0391
      EXINT3(J) = EXINT3(K) + U*(Z + AO)                                COR 0392
      AO = Z                                                            COR 0393
      IF(K-I)150,931,933                                                COR 0394
  931 CONTINUE                                                          COR 0395
      IF(EXINT1(K)+600.)932,933,933                                     COR 0396
  932 CONTINUE                                                          COR 0397
      EXINT1(K) = 1.E-250                                               COR 0398
      EXINT2(K) = 1.E-250                                               COR 0399
      EXINT3(K) = 1.E-250                                               COR 0400
      GO TO 934                                                         COR 0401
  933 CONTINUE                                                          COR 0402
      EXINT1(K) = EXP(EXINT1(K))                                        COR 0403
      EXINT2(K) = EXP(EXINT2(K))                                        COR 0404
      EXINT3(K) = EXP(EXINT3(K))                                        COR 0405
  934 CONTINUE                                                          COR 0406
      X = EXP(EXINT1(J))                                                COR 0407
      Y = EXP(EXINT2(J))                                                COR 0408
      Z = EXP(EXINT3(J))                                                COR 0409
      RP(J) = RP(K) + U*(X/BETA1(J) + EXINT1(K)/BETA1(K))               COR 0410
      DR(J) = (1.-ALPHA1(J)*RP(J)/X)/BETA1(J)-(1.-ALPHA1(K)*RP(K)/EXINT1COR 0411
     1(K))/BETA1(K)                                                     COR 0412
      DR(J) = DR(J)*(W*X+A1*EXINT1(K))/2.                               COR 0413
      DR(J) = DR(J) + DR(K)                                             COR 0414
      RR2(J) = RR2(K)+(((RP(J)+DR(J))*Y)/(BETA2(J)*X)+((RP(K)+DR(K))*EXICOR 0415
     1NT2(K))/(BETA2(K)*EXINT1(K)))*2.*U                                COR 0416
      RC2(J) = RC2(K) + (((RP(J)+DR(J))*Z)/(BETA3(J)*X) + ((RP(K)+DR(K))COR 0417
     1*EXINT3(K))/(BETA3(K)*EXINT1(K)))*2.*U                            COR 0418
      Z11= ((2.*(RP(J)+DR(J))/X-ALPHA2(J)*RR2(J)/Y)/BETA2(J)-(2.*(RP(K)+COR 0419
     1DR(K))/EXINT1(K)-ALPHA2(K)*RR2(K)/EXINT2(K))/BETA2(K))            COR 0420
      Z11= Z11*(T*Y+A2*EXINT2(K))/2.                                    COR 0421
      DRR2(J) = Z11+ DRR2(K)                                            COR 0422
      Z11= ((2.*(RP(J)+DR(J))/X-ALPHA3(J)*RC2(J)/Z)/BETA3(J)-(2.*(RP(K)+COR 0423
     1DR(K))/EXINT1(K)-ALPHA3(K)*RC2(K)/EXINT3(K))/BETA3(K))            COR 0424
      Z11= Z11*(TX*Z+Q*EXINT3(K))/2.                                    COR 0425
      DRC2(J) = Z11+ DRC2(K)                                            COR 0426
      R(J) = R(K) + U*(Z/BETA3(J)+EXINT3(K)/BETA3(K))                   COR 0427
      Z11= (1.-ALPHA3(J)*R(J)/Z)/BETA3(J)-(1.-ALPHA3(K)*R(K)/EXINT3(K))/COR 0428
     1BETA3(K)                                                          COR 0429
      Z11= Z11*(TX*Z+Q*EXINT3(K))/2.                                    COR 0430
      DRO(J) = Z11+ DRO(K)                                              COR 0431
      A1 = W                                                            COR 0432
      A2 = T                                                            COR 0433
      Q = TX                                                            COR 0434
```

```
      K = 5*NDIVE + I                                                   COR 0435
      IF(J-K)150,14951,150                                              COR 0436
14951 CONTINUE                                                          COR 0437
      Y = NDIVE                                                         COR 0438
      U = U*Y                                                           COR 0439
      JJJ = NDIVE                                                       COR 0440
  150 CONTINUE                                                          COR 0441
      NINT = NENT                                                       COR 0442
      EXINT1(NE) = EXP(EXINT1(NE))                                      COR 0443
      EXINT2(NE) = EXP(EXINT2(NE))                                      COR 0444
      EXINT3(NE) = EXP(EXINT3(NE))                                      COR 0445
      JJJ = 1                                                           COR 0446
      K = 5*NDIVE + I                                                   COR 0447
      DO 155 J = I,NE,JJJ                                               COR 0448
      R(J) = R(J) + DRO(J)                                              COR 0449
      RP(J) = RP(J) + DR(J)                                             COR 0450
      RR2(J) = RR2(J) + DRR2(J)                                         COR 0451
      RC2(J) = RC2(J) + DRC2(J)                                         COR 0452
      RP2(J) = RP(J)**2                                                 COR 0453
      IF(J-K)155,15451,155                                              COR 0454
15451 CONTINUE                                                          COR 0455
      JJJ = NDIVE                                                       COR 0456
  155 CONTINUE                                                          COR 0457
      K = 0                                                             COR 0458
      NDI = NDIVE + I                                                   COR 0459
      DO 175 J = NDI,NE,NDIVE                                           COR 0460
      IF(I-J)165,175,175                                                COR 0461
  165 CONTINUE                                                          COR 0462
      IF(J-NE)167,166,175                                               COR 0463
  166 CONTINUE                                                          COR 0464
      K = 1                                                             COR 0465
  167 CONTINUE                                                          COR 0466
      RP(J) = RP(J)/EXINT1(J)                                           COR 0467
      DELTA(J) = (2.*RR2(J)/EXINT2(J)+RC2(J)/EXINT3(J))/3.              COR 0468
      DELTA(J) = DELTA(J)-RP2(J)/(EXINT1(J)**2)                         COR 0469
      DP(J) = 2.*(RC2(J)/EXINT3(J)-RR2(J)/EXINT2(J))/3.                 COR 0470
      IF(DELTA(J))170,170,171                                           COR 0471
  170 CONTINUE                                                          COR 0472
      DELTA(J) = 1.                                                     COR 0473
  171 CONTINUE                                                          COR 0474
      IF(DP(J))1715,1715,1716                                           COR 0475
 1715 CONTINUE                                                          COR 0476
      DP(J) = 1.                                                        COR 0477
 1716 CONTINUE                                                          COR 0478
      DP(J) = SQRT(DP(J))                                               COR 0479
      DELTA(J) = SQRT(DELTA(J))                                         COR 0480
      R(J) = R(J)/EXINT3(J)                                             COR 0481
      IF(K)172,172,173                                                  COR 0482
  172 CONTINUE                                                          COR 0483
      K = 1                                                             COR 0484
      GO TO 175                                                         COR 0485
  173 CONTINUE                                                          COR 0486
      NN = J-NDIVE                                                      COR 0487
      K = 0                                                             COR 0488
      Y = NDIV                                                          COR 0489
      Y = Y/NMULT                                                       COR 0490
      X = J-1                                                           COR 0491
      X = X/Y                                                           COR 0492
      T = I-1                                                           COR 0493
      T = T/Y                                                           COR 0494
      W = NN-1                                                          COR 0495
      W = W/Y                                                           COR 0496
```

```
      PRINT IVAR,X,T,RP(J),DELTA(J),R(J),DP(J),W,T,RP(NN),DELTA(NN),R(NNCOR 0497
     1),DP(NN)                                                          COR 0498
  175 CONTINUE                                                          COR 0499
      IF(NDIVE.GT.1)GO TO 350                                           COR 0500
      WRITE TAPE10,NE,NDIV,NMULT,NCOMP,(RP(J),DELTA(J),DP(J),J=1,NE)    COR 0501
      GO TO 350                                                         COR 0502
  255 CONTINUE                                                          COR 0503
      NE = 0                                                            COR 0504
      NDIV = 0                                                          COR 0505
      NMULT = 0                                                         COR 0506
      NCOMP = 0                                                         COR 0507
      RP(1) = 0.                                                        COR 0508
      DELTA(1) = 0.                                                     COR 0509
      DP(1) = 0.                                                        COR 0510
      WRITE TAPE10,NE,NDIV,NMULT,NCOMP,ALAT,(P2(J),Z2(J),CSE(J),SETA(J),COR 0511
     1ALP(J),EN(J),NFORM(J),NEXP(J),J=1,4)                              COR 0512
      WRITE TAPE10,NE,NDIV,NMULT,NCOMP,(RP(J),DELTA(J),DP(J),J=1,NE)    COR 0513
      ENDFILE 10                                                        COR 0514
      REWIND 10                                                         COR 0515
      END                                                               COR 0516
```

B. RASE4

```
      PROGRAM RASE4(INPUT,OUTPUT,TAPE10,TAPE20)                         RAS 0001
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
C     *                    ISSUED BY SANDIA LABORATORIES.                *
C     *               A PRIME CONTRACTOR TO THE UNITED STATES            *
C     *           ENERGY RESEARCH AND DEVELOPEMENT ADMINISTRATION.       *
C     * * * * * * * * * * * * * * *   NOTICE   * * * * * * * * * * * * * **
C     * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE  *
C     * UNITED STATES GOVERNMENT. NEITHER THE UNITED STATES NOR THE      *
C     * UNITED STATES ENERGY RESEARCH AND DEVELOPEMENT ADMINISTRATION,   *
C     * NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS,        *
C     * SUBCONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS  *
C     * OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR *
C     * THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,     *
C     * APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS  *
C     * USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.                   *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
C     * THE BASIC REFERENCE DOCUMENTS FOR THIS CODE ARE SLA-73-0416,     *
C     * APRIL,1973, AND SAND-75-0622, DECEMBER, 1975.                    *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
C     * THIS CODE HAS BEEN RELEASED FOR UNLIMITED DISTRIBUTION.          *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **

    1 FORMAT(16F5.0)                                                    RAS 0043
    2 FORMAT(16I5)                                                      RAS 0044
    3 FORMAT(2/)                                                        RAS 0045
    4 FORMAT(8A10)                                                      RAS 0046
    5 FORMAT(1H1,* RANGE AND STRAGGLING DATA FOR *,A10,* IONS INCIDENT ORAS 0047
     1N A/AN *,2A10,* TARGET *,4/)                                      RAS 0048
    6 FORMAT(* INITIAL VALUES*3/)                                       RAS 0049
    7 FORMAT(*  INCIDENT ION *,A10,/*  ATOMIC NUMBER(Z1) = *,F4.0,/*  ATRAS 0050
     1OMIC MASS(M1) = *,F7.2,* AMU*/*  ENERGY MAXIMUM = *,F10.2,*KEV*2/)RAS 0051
    8 FORMAT(*  TARGET - *,2A10, /)                                    RAS 0052
    9 FORMAT(*  COMPONENTS    *,4(5X,A10,5X),/*  ATOMIC NUMBER *,4(F10.0RAS 0053
     1,10X),/*  ATOMIC MASS    *, 4(F8.2,* AMU *,7X)/*  T-F RADIUS    *,4RAS 0054
     2(E10.3,* CM *,6X),/*  EFFECTIVE Z   *,4(F10.3,10X))               RAS 0055
 1000 FORMAT(*    I      U(I)      F(U)      SLOPE   -   F(U) IS UNIVERSARAS 0056
     1L THOMAS-FERMI FUNCTION - FROM LSS*2/)                            RAS 0057
 1001 FORMAT(I5,2F10.3,F10.5)                                           RAS 0058
 1002 FORMAT(4/* THE FOLLOWING ELECTRONIC STOPPING CROSS SECTIONS WERE  RAS 0059
     1USED IN THE CALCULATIONS*2/)                                      RAS 0060
 1003 FORMAT(3X,A10,3X,*-  THOMAS-FERMI ATOMIC MODEL , LINDHARD THEORY*)RAS 0061
 1004 FORMAT(5/20H *****  UNITS  ***** ,/*  ENERGY  -  KEV*/*  LENGTHS  RAS 0062
```

```
     1-  ANGSTROMS*/*  ENERGY DEPOSITION RATES  -  EV/ANGSTROM*2/)
1005 FORMAT(3X,A10,3X,*-  THOMAS-FERMI ATOMIC MODEL , FIRSOV THEORY*)
1006 FORMAT(3X,A10,3X,21H-   K*E**P , WITH K= ,E11.4,*   AND   P = *,F4
     1.2,2/)
1007 FORMAT(7X,*CSE = *,E12.5,2/)
1008 FORMAT(3X,A10,3X,*-   BRICE FORMULA*)
1009 FORMAT(7X,*Z = *,F6.3,5X,* A = *,F7.4,5X,* N = *,F6.3,2/)
1031 FORMAT(A10,* INCIDENT ON A/AN *,2A10,* TARGET *)
1032 FORMAT(I4,3I2,/,(8E10.4))
1033 FORMAT(3/,*     NE     NSTP  NCOMP          NFORM(I)
     1     NEXP(J)              NDIV  NDIVE  NMULT  NDSKP   NRPT*/21X,*
     2I = 1      2      3      4  J = 1      2      3      4*2/)
1034 FORMAT(16I7,2/)
1035 FORMAT(5/,*   DEPOSITED ENERGY(EV/ANGSTROM) , DEP-E*,2/,1X,7(*
     1 E     DEP-E*),2/)
1036 FORMAT(1X,7(F6.1,1X,E10.4))
1037 FORMAT(5/,*   RMS SPREAD IN  RP-REC*,2/,1X,7(*     E   DRP-REC*),2
     1/)
1038 FORMAT(5/,*   AVERAGE PROJECTED RECOIL RANGE(ANGSTROMS)      RP-RE
     1C*,2/,1X,7(*     E    RP-REC *),2/)
1039 FORMAT(5/,*   ELECTRONIC STOPPING POWER ,  EV/ANGSTROM*,2/,1X,7(*
     1    E      DE/DX*),2/)
1040 FORMAT(3/*      E     EP RP(E,EP)   SPREAD IN     R(E,EP)   SPREAD
     1 IN            E     EP RP(E,EP)   SPREAD IN     R(E,EP)   SPREAD
     2 IN*/    *                          RP-PARA                RP-PE
     3RP                                  RP-PARA                RP-PE
     4RP*2/)
1041 FORMAT(*  CHAR. ENERGY  *,4(F8.3,* KEV *,7X),2/)
1042 FORMAT(2/,*  CHARACTERISTIC LENGTH = *,F9.2,* ANGSTROMS*,2/)
1043 FORMAT(2/*  ATOMIC DENSITY = *,E12.5,* ATOMS/CC*2/)
      DIMENSION SETA(4),ALP(4),EN(4)
      DIMENSION A(4)
      DIMENSION ALPHA(1600),ALPHA1(1600),ALPHA2(1600),ALPHA3(1600)
      DIMENSION BETA1(1600),BETA2(1600),BETA3(1600)
      DIMENSION CSE(4),CEPS(4),CTM(4),CALPH(4)
      DIMENSION DELTAO(1600),DELTA(1600),DP(1600),DPO(1600),DR(1600)
      DIMENSION DRO(1600),DRR2(1600),DRC2(1600),DENS(4)
      DIMENSION D(1600)
      DIMENSION EXINT1(1600),EXINT2(1600),EXINT3(1600)
      DIMENSION E(1600),EF(50)
      DIMENSION F(50)
      DIMENSION GAM1(1600),GAM2(1600),GAM3(1600)
      DIMENSION IVAR(8),NEXP(4),NFORM(4)
      DIMENSION P2(4)
      DIMENSION RPO(1600),RP(1600),RO(1600),RP2(1600),RR2(1600)
      DIMENSION RC2(1600),R(1600)
      DIMENSION SE(1600),SLF(50)
      DIMENSION TITLE(8)
      DIMENSION Z2(4),Z(4)
      DIMENSION ALF(4),EO(4),SN(4),SK(4)
      DIMENSION DAMRP(50,60),DAMDEL(50,60),DAMR(50,60),DAMDP(50,60)
      COMMON/PHIVAR/E,P1,P2
      COMMON/FVAR/F,SLF,EF,NF,MM
      COMMON/DSEC/ALPHA1,ALPHA2,ALPHA3,BETA1,BETA2,BETA3,GAM1,GAM2,GAM3,
     1EXINT1,EXINT2,EXINT3,NSPC(4),ERAN
      COMMON/DFIR/NCOMP,NDAM,DE,PART(4,101)
      COMMON/TENVAR/NE,NDIV,NMULT,ALAT,Z2
      NSPC(1)=NSPC(2)=NSPC(3)=NSPC(4) = 0
      NRPT = 0
      CALL WIND20
  350 CONTINUE
      IF(NRPT)255,355,255
```

```
355 CONTINUE                                                            RAS 0125
    PI = 3.141592654                                                    RAS 0126
    HBAR = (6.6254E-27)/(2.*PI)                                         RAS 0127
    Q = 4.80294E-10                                                     RAS 0128
    AO = 0.529484E-8                                                    RAS 0129
    ERGEV = 1.60203E-12                                                 RAS 0130
    AMU = 1.66035E-24                                                   RAS 0131
    DO 115 J = 1,4                                                      RAS 0132
    P2(J) = 0.                                                          RAS 0133
    Z(J) = 0.                                                           RAS 0134
    A(J) = 0.                                                           RAS 0135
    NEXP(J) = 0                                                         RAS 0136
    NFORM(J) = 0                                                        RAS 0137
115 CONTINUE                                                            RAS 0138
    DO 116 J=1,50                                                       RAS 0139
    DO 116 K=1,60                                                       RAS 0140
    DAMRP(J,K)=0.                                                       RAS 0141
    DAMDEL(J,K)=0.                                                      RAS 0142
    DAMR(J,K)=0.                                                        RAS 0143
    DAMDP(J,K)=0.                                                       RAS 0144
116 CONTINUE                                                            RAS 0145
    READ 4,(TITLE(I),I = 1,8)                                           RAS 0146
    READ 4 ,(IVAR(J) , J = 1,8)                                         RAS 0147
    READ 1,ALAT,P1,Z1,(P2(J),Z2(J),J=1,4),(CSE(K),K=1,4),ED             RAS 0148
    EX = ((Q**2/HBAR)**2)*AMU/(2000.*ERGEV)                             RAS 0149
    X = 4./3.                                                           RAS 0150
    X = Z1**X                                                           RAS 0151
    EX = EX*P1*X                                                        RAS 0152
    PRINT 5,(TITLE(J),J=1,3)                                            RAS 0153
    PRINT 6                                                             RAS 0154
    PRINT 7,TITLE(1),Z1,P1,EX                                           RAS 0155
    PRINT 8,TITLE(2),TITLE(3)                                           RAS 0156
    READ 2,NE,NSTP,NCOMP,(NFORM(J),J=1,4),(NEXP(K),K=1,4),NDIV,NDIVE    RAS 0157
   1,NMULT,NDSKP,NRPT                                                   RAS 0158
    READ 1,(SETA(J),ALP(J),EN(J),J=1,4)                                 RAS 0159
    WRITE TAPE20,NE,NSTP,NDIV,NMULT,NCOMP,NDIVE,NDSKP,(NFORM(J),NEXP(JRAS 0160
   1),J=1,4),ALAT,P1,Z1,(P2(K),Z2(K),CSE(K),SETA(K),ALP(K),EN(K),K=1,4RAS 0161
   2)                                                                   RAS 0162
    CALL TEN                                                            RAS 0163
    CALL RANGER(X,MM,Y,1,U,V)                                           RAS 0164
    NDAM = 1                                                            RAS 0165
    CALL DAMG(Z,1.,1.,1)                                                RAS 0166
    IF(NDIV)121,121,122                                                 RAS 0167
121 CONTINUE                                                            RAS 0168
    NDIV = 1                                                            RAS 0169
122 CONTINUE                                                            RAS 0170
    IF(NDIVE)123,123,124                                                RAS 0171
123 CONTINUE                                                            RAS 0172
    NDIVE = 1                                                           RAS 0173
124 CONTINUE                                                            RAS 0174
    ALAT = ALAT*1.E-8                                                   RAS 0175
    ALAT = 1./(ALAT**3)                                                 RAS 0176
    X = 2./3.                                                           RAS 0177
    CRHO = 0.                                                           RAS 0178
    DO 1250 J=1,NCOMP                                                   RAS 0179
    DENS(J) = NFORM(J)*ALAT                                             RAS 0180
    CSE(J) = CSE(J)*1.E-26*DENS(J)                                      RAS 0181
    Z(J) = Z1**X + Z2(J)**X                                             RAS 0182
    Z(J) = SQRT(Z(J))                                                   RAS 0183
    A(J) = .8853*AO/Z(J)                                                RAS 0184
    Z(J) = Z(J)**3                                                      RAS 0185
    CEPS(J) = ((A(J)*P2(J))/(Z1*Z2(J)*(P1+P2(J))*Q**2))*ERGEV*1000.     RAS 0186
```

```
      CTM(J) = (4.*P1*P2(J))/((P1+P2(J))**2)                            RAS 0187
      CRHO = CRHO + PI*A(J)**2*DENS(J)*CTM(J)                           RAS 0188
      IF(NEXP(J))1202,1200,1201                                         RAS 0189
 1202 CONTINUE                                                          RAS 0190
      CSE(J) = 2.34E-34*(Z1+Z2(J))*DENS(J)                              RAS 0191
      GO TO 1203                                                        RAS 0192
 1200 CONTINUE                                                          RAS 0193
      CSE(J) = DENS(J)*8.*PI*AO*HBAR*Z2(J)/Z(J)                         RAS 0194
      Y = 7./6.                                                         RAS 0195
      Y = Z1**Y                                                         RAS 0196
      CSE(J) = CSE(J)*Y/(ERGEV*1.E+11)                                  RAS 0197
 1203 CONTINUE                                                          RAS 0198
      Y = 2./(P1*AMU)                                                   RAS 0199
      Y = Y*ERGEV*1000.                                                 RAS 0200
      Y = SQRT(Y)                                                       RAS 0201
      CSE(J) = Y*CSE(J)                                                 RAS 0202
 1201 CONTINUE                                                          RAS 0203
      CALPH(J) = PI*DENS(J)*1.E-8*(A(J)**2)                             RAS 0204
 1250 CONTINUE                                                          RAS 0205
      PRINT 9,(TITLE(I),I=5,8),(Z2(J),J=1,4),(P2(K),K=1,4),(A(L),L=1,4),RAS 0206
     1(Z(M),M=1,4)                                                      RAS 0207
      X = 7./6.                                                         RAS 0208
      Y = 2./3.                                                         RAS 0209
      W = 3./2.                                                         RAS 0210
      U = 1./3.                                                         RAS 0211
      V = 7./3.                                                         RAS 0212
      DO 1252 J = 1,NCOMP                                               RAS 0213
      SN(J) = 0.                                                        RAS 0214
      SK(J) = 0.                                                        RAS 0215
      EO(J) = 8.6939E-2*Z2(J)**V                                        RAS 0216
      RO(J) = 2.9008E24*Z2(J)**Y/DENS(J)                                RAS 0217
      ALF(J) = 1.3596E-26*Z2(J)**X*DENS(J)                              RAS 0218
      ALF(J) = ALF(J)*RO(J)/SQRT(EO(J)*P2(J))                           RAS 0219
      DO 1251 K = 1,NCOMP                                               RAS 0220
      Z = SQRT(Z2(J)**Y+Z2(K)**Y)                                       RAS 0221
      SN(J) = SN(J) + 2.7666E-26*Z2(K)/(Z*(P2(J)+P2(K)))                RAS 0222
      SK(J) = SK(J) + 3.8456E-26*Z2(K)/(Z*Z*Z)                          RAS 0223
 1251 CONTINUE                                                          RAS 0224
      SN(J) = SN(J)*DENS(J)*Z2(J)*P2(J)                                 RAS 0225
      SK(J) = SK(J)*(Z2(J)**X)*DENS(J)                                  RAS 0226
      SN(J) = SN(J)/SK(J)                                               RAS 0227
      DO 1252 K = 1,NDAM                                                RAS 0228
      PART(J,K) = PART(J,K)/EO(J)                                       RAS 0229
 1252 CONTINUE                                                          RAS 0230
      EPS = 0.                                                          RAS 0231
      DO 1255 J = 1,NCOMP                                               RAS 0232
      Z(J) = 1./CEPS(J)                                                 RAS 0233
      EPS = EPS + DENS(J)                                               RAS 0234
 1255 CONTINUE                                                          RAS 0235
      PRINT 1041,(Z(J) , J=1,4)                                         RAS 0236
      CRHO = 1.E+8/CRHO                                                 RAS 0237
      PRINT 1042,CRHO                                                   RAS 0238
      PRINT 1043,EPS                                                    RAS 0239
      PRINT 1002                                                        RAS 0240
      DO 1302 J = 1,NCOMP                                               RAS 0241
      IF(NEXP(J))1299,1301,1300                                         RAS 0242
 1299 CONTINUE                                                          RAS 0243
      IF(NEXP(J)+50)12995,12991,12995                                   RAS 0244
12991 CONTINUE                                                          RAS 0245
      PRINT 1005,TITLE(J+4)                                             RAS 0246
      PRINT 1007,CSE(J)                                                 RAS 0247
      NEXP(J) = 0                                                       RAS 0248
```

```
      GO TO 1302                                                        RAS 0249
12995 CONTINUE                                                          RAS 0250
      PRINT 1008,TITLE(J+4)                                             RAS 0251
      PRINT 1009,SETA(J),ALP(J),EN(J)                                   RAS 0252
      GO TO 1302                                                        RAS 0253
 1300 CONTINUE                                                          RAS 0254
      Y = NEXP(J)                                                       RAS 0255
      Y = Y/100.                                                        RAS 0256
      PRINT 1006,TITLE(J+4),CSE(J),Y                                    RAS 0257
      GO TO 1302                                                        RAS 0258
 1301 CONTINUE                                                          RAS 0259
      PRINT 1003,TITLE(J+4)                                             RAS 0260
      PRINT 1007,CSE(J)                                                 RAS 0261
 1302 CONTINUE                                                          RAS 0262
      PRINT 1033                                                        RAS 0263
      PRINT 1034,NE,NSTP,NCOMP,(NFORM(J),J=1,4),(NEXP(K),K=1,4),NDIV,NDIRAS 0264
     1VE,NMULT,NDSKP,NRPT                                               RAS 0265
      CALL FSET                                                         RAS 0266
      L = NF-1                                                          RAS 0267
      PRINT 1000                                                        RAS 0268
      DO 12 I = 1,L                                                     RAS 0269
      PRINT 1001,I,EF(I),F(I),SLF(I)                                    RAS 0270
   12 CONTINUE                                                          RAS 0271
      PRINT 1004                                                        RAS 0272
      GO TO 1304                                                        RAS 0273
 1303 CONTINUE                                                          RAS 0274
      NDSKP = 2                                                         RAS 0275
 1304 CONTINUE                                                          RAS 0276
      DO 100 I = 1,NE                                                   RAS 0277
      DRO(I) = 0.                                                       RAS 0278
      DRR2(I) = 0.                                                      RAS 0279
      DR(I) = 0.                                                        RAS 0280
      D(I) = 0.                                                         RAS 0281
      SE(I) = 0.                                                        RAS 0282
      ALPHA(I) = 0.                                                     RAS 0283
      E(I) = I                                                          RAS 0284
      Y = NDIV                                                          RAS 0285
      E(I) = E(I)/Y                                                     RAS 0286
      X = NMULT                                                         RAS 0287
      E(I) = E(I)*X                                                     RAS 0288
      DO 999 LJ = 1,NCOMP                                               RAS 0289
      SPHI = (P1+P2(LJ))/(2.*SQRT(P1*P2(LJ)*E(I)))                      RAS 0290
      EPS = CEPS(LJ)*E(I)                                               RAS 0291
      X = EPS                                                           RAS 0292
      TM = CTM(LJ)*E(I)                                                 RAS 0293
      W = TM/EPS**2                                                     RAS 0294
      NQT = 4                                                           RAS 0295
      X = NQT + ALOG10(EPS)                                             RAS 0296
      NT = X                                                            RAS 0297
      NT = NT - NQT                                                     RAS 0298
      X = 2.                                                            RAS 0299
      Y = 10.**NT                                                       RAS 0300
      T = X*(EPS/Y - 1.)                                                RAS 0301
      NQP = T                                                           RAS 0302
      Y = NQP                                                           RAS 0303
      OVER = T - Y                                                      RAS 0304
      NX = X                                                            RAS 0305
      NT = (NT + NQT)*NX                                                RAS 0306
      NT = 9*NT+NQP    +1                                               RAS 0307
      NX = 9*NX                                                         RAS 0308
      NQT = -NQT                                                        RAS 0309
      Y = 10.**NQT                                                      RAS 0310
```

```
      X = 0.5*Y                                                         RAS 0311
      NQP = -1                                                          RAS 0312
      DO 80 J = 1,NT                                                    RAS 0313
      IF(J-2)793,791,792                                                RAS 0314
  791 CONTINUE                                                          RAS 0315
      T = NX                                                            RAS 0316
      T = T/9.                                                          RAS 0317
      T = Y/T                                                           RAS 0318
      X = Y - 0.5*T                                                     RAS 0319
      Y = T                                                             RAS 0320
  792 CONTINUE                                                          RAS 0321
      X = X+Y                                                           RAS 0322
  793 CONTINUE                                                          RAS 0323
      CALL FET(FT,X)                                                    RAS 0324
      TX = FT*Y                                                         RAS 0325
      ALPHA(I) = ALPHA(I) + (TX/X**2)*CALPH(LJ)                         RAS 0326
      IF(NDSKP)795,7985,795                                             RAS 0327
  795 CONTINUE                                                          RAS 0328
      T = W*X*X                                                         RAS 0329
      T = SQRT(T)                                                       RAS 0330
      CPHI = SPHI*T                                                     RAS 0331
      IF(NSPC(1).EQ.0)GO TO 7953                                        RAS 0332
      T = T*T-ED                                                        RAS 0333
       CALL RANGER(T,LJ,CPHI,0,REC,DREC)                                RAS 0334
      GO TO 7954                                                        RAS 0335
 7953 CONTINUE                                                          RAS 0336
      REC = CPHI*2.*(T-SN(LJ)*ALOG(T/SN(LJ)+1.))/SK(LJ)                 RAS 0337
      DREC = REC*REC                                                    RAS 0338
      T = T*T-ED                                                        RAS 0339
 7954 CONTINUE                                                          RAS 0340
      IF(NDAM.GT.1)CALL DAMG(Z,T,ALF(LJ),LJ)                            RAS 0341
      IF(T.LE.0.)T=-ED                                                  RAS 0342
      T = T/EO(LJ)                                                      RAS 0343
      IF(NDAM.LE.1)CALL DAMG(Z,T,ALF(LJ),LJ)                            RAS 0344
      IF(NDSKP.EQ.2) GO TO 796                                          RAS 0345
      U = TX*CALPH(LJ)*(Z*EO(LJ)+ED)/(X*X)                              RAS 0346
      GO TO 798                                                         RAS 0347
  796 CONTINUE                                                          RAS 0348
      U = TX*CALPH(LJ)*(T-Z)*EO(LJ)/(X*X)                               RAS 0349
  798 CONTINUE                                                          RAS 0350
      D(I) = D(I) + U                                                   RAS 0351
      DR(I) = DR(I) + REC*U                                             RAS 0352
      DRR2(I) = DRR2(I) + DREC*U                                        RAS 0353
 7985 CONTINUE                                                          RAS 0354
      NQP = NQP + 1                                                     RAS 0355
      IF(NQP-NX)80,799,799                                              RAS 0356
  799 CONTINUE                                                          RAS 0357
      X = X + 0.5*Y                                                     RAS 0358
      NQP = 0.                                                          RAS 0359
      Y = 10.*Y                                                         RAS 0360
      X = X - 0.5*Y                                                     RAS 0361
   80 CONTINUE                                                          RAS 0362
      X = X+0.5*Y*(1.+OVER)                                             RAS 0363
      Y = Y*OVER                                                        RAS 0364
      CALL FET(FT,X)                                                    RAS 0365
      TX = FT*Y                                                         RAS 0366
      ALPHA(I) = ALPHA(I) + (TX/X**2)*CALPH(LJ)                         RAS 0367
      IF(NDSKP)814,816,814                                              RAS 0368
  814 CONTINUE                                                          RAS 0369
      X = X**2                                                          RAS 0370
      T = W*X                                                           RAS 0371
      T = SQRT(T)                                                       RAS 0372
```

```
      CPHI = SPHI*T                                                     RAS 0373
      IF(NSPC(1).EQ.0)GO TO 8143                                        RAS 0374
      T = T*T-ED                                                        RAS 0375
       CALL RANGER(T,LJ,CPHI,C,REC,DREC)                                RAS 0376
      GO TO 8144                                                        RAS 0377
 8143 CONTINUE                                                          RAS 0378
      REC = CPHI*2.*(T-SN(LJ)*ALOG(T/SN(LJ)+1.))/SK(LJ)                 RAS 0379
      DREC = REC*REC                                                    RAS 0380
      T = T*T-ED                                                        RAS 0381
 8144 CONTINUE                                                          RAS 0382
      IF(NDAM.GT.1)CALL DAMG(Z,T,ALF(LJ),LJ)                            RAS 0383
      IF(T.LE.0.)T=-ED                                                  RAS 0384
      T = T/EO(LJ)                                                      RAS 0385
      IF(NDAM.LE.1)CALL DAMG(Z,T,ALF(LJ),LJ)                            RAS 0386
      IF(NDSKP.EQ.2) GO TO 8145                                         RAS 0387
      U = TX*CALPH(LJ)*(Z*EO(LJ)+ED)/X                                  RAS 0388
      GO TO 815                                                         RAS 0389
 8145 CONTINUE                                                          RAS 0390
      U = TX*CALPH(LJ)*(T-Z)*EO(LJ)/X                                   RAS 0391
  815 CONTINUE                                                          RAS 0392
      D(I) = D(I) + U                                                   RAS 0393
      DR(I) = DR(I) + REC*U                                             RAS 0394
      DRR2(I) = DRR2(I) + DREC*U                                        RAS 0395
  816 CONTINUE                                                          RAS 0396
      IF(NEXP(LJ))85,84,82                                              RAS 0397
   82 CONTINUE                                                          RAS 0398
      X = NEXP(LJ)                                                      RAS 0399
      X = X/100.                                                        RAS 0400
      SE(I) = CSE(LJ)*(E(I)**X) + SE(I)                                 RAS 0401
      GO TO 999                                                         RAS 0402
   84 CONTINUE                                                          RAS 0403
      SE(I) = CSE(LJ)*SQRT(E(I)) + SE(I)                                RAS 0404
      GO TO 999                                                         RAS 0405
   85 CONTINUE                                                          RAS 0406
      CALL SECALC(Z1,Z2(LJ),P1,E(I),SET,SETA(LJ),ALP(LJ),EN(LJ))        RAS 0407
      SE(I) = SE(I) + DENS(LJ)*1.E-8*SET                                RAS 0408
  999 CONTINUE                                                          RAS 0409
      IF(NDSKP)9995,100,9995                                            RAS 0410
 9995 CONTINUE                                                          RAS 0411
      DRO(I) = 1000.*SE(I)                                              RAS 0412
      IF(D(I).EQ.0.) GO TO 100                                          RAS 0413
      DR(I) = DR(I)/D(I)                                                RAS 0414
      DRR2(I) = DRR2(I)/D(I)                                            RAS 0415
      DRR2(I) = DRR2(I)-DR(I)*DR(I)                                     RAS 0416
      DRR2(I) = SQRT(DRR2(I))                                           RAS 0417
      D(I) = 1000.*D(I)                                                 RAS 0418
  100 CONTINUE                                                          RAS 0419
      IF(NDSKP)1096,1100,1096                                           RAS 0420
 1096 CONTINUE                                                          RAS 0421
      KKK = NE+1                                                        RAS 0422
      PRINT 1039                                                        RAS 0423
      PRINT 1036,(E(J),DRO(J),J=1,NE)                                   RAS 0424
      PRINT 1035                                                        RAS 0425
      PRINT 1036,(E(J),D(J),J=1,NE)                                     RAS 0426
      PRINT 1038                                                        RAS 0427
      PRINT 1036,(E(J),DR(J),J=1,NE)                                    RAS 0428
      PRINT 1037                                                        RAS 0429
      PRINT 1036,(E(J),DRR2(J),J=1,NE)                                  RAS 0430
      PRINT 3                                                           RAS 0431
      IF(NDSKP.EQ.2)WRITE TAPE20,KKK,NDIV,NMULT,NDSKP,(DRO(J),D(J),DR(J)RAS 0432
     1,DRR2(J),J=1,NE)                                                  RAS 0433
      IF(NDSKP.NE.2)WRITE TAPE20,KKK,NDIV,NMULT,NDSKP,(D(J),DR(J),DRR2(JRAS 0434
```

```
     1),J=1,NE)
      IF(NDSKP.EQ.1) GO TO 1303
 1100 CONTINUE
      PRINT 1040
      KKK = NDIVE
      DO 200 I = NDIVE,NE,KKK
      DELTA(I) = 0.
      DP(I) = 0.
      DR(I) = 0.
      DRO(I) = 0.
      DRR2(I) = 0.
      DRC2(I) = 0.
      Y = NDIV
      U = 0.5/Y
      X = NMULT
      U = U*X
      Y = -U*(ALPHA(I)/SE(I))
      EXINT1(I) = Y
      EXINT2(I) = Y
      EXINT3(I) = Y
      R(I) = 0.
      RP(I) = 0.
      RR2(I) = 0.
      RC2(I) = 0.
      RP2(I) = 0.
      IF(I-NSTP)102,101,101
  101 CONTINUE
      KKK = NSTP
  102 CONTINUE
      JJJ = 1
      DO 150 J = I,NE,JJJ
      ALPHA1(J) = 0.
      ALPHA2(J) = 0.
      ALPHA3(J) = 0.
      BETA1(J) = 0.
      BETA2(J) = 0.
      BETA3(J) = 0.
      GAM1(J) = 0.
      GAM2(J) = 0.
      GAM3(J) = 0.
      K = J-JJJ
      IF(J-I)150,1410,1399
 1399 CONTINUE
      DO 149 LJ = 1,NCOMP
      TM = CTM(LJ)*E(J)
      EPS = CEPS(LJ)*E(J)
      W = TM/EPS**2
      CD = CALPH(LJ)*W
      CG = CD*W
      X = J-I
      Y = NDIV
      Y = Y/NMULT
      X = X/Y
      IF(X-TM)1402,1401,1401
 1401 CONTINUE
      X = TM
 1402 CONTINUE
      X = X*EPS**2/TM
      X = SQRT(X)
      NQT = 4
      Y = NQT + ALOG10(X)
      NT = Y
```

```
      NT = NT - NQT                                                     RAS 0497
      Y = 2.                                                            RAS 0498
      T = Y*(X/10.**NT - 1.)                                            RAS 0499
      NQP = T                                                           RAS 0500
      NX = Y                                                            RAS 0501
      Y = NQP                                                           RAS 0502
      OVER = T-Y                                                        RAS 0503
      NT = (NT+NQT)*NX                                                  RAS 0504
      NT = 9*NT+NQP    +1                                               RAS 0505
      NX = 9*NX                                                         RAS 0506
      NQT = -NQT                                                        RAS 0507
      Y = 10.**NQT                                                      RAS 0508
      X = 0.5*Y                                                         RAS 0509
      NQP = -1                                                          RAS 0510
      DO 90 L = 1,NT                                                    RAS 0511
      IF(L-2)893,891,892                                                RAS 0512
  891 CONTINUE                                                          RAS 0513
      T = NX                                                            RAS 0514
      T = T/9.                                                          RAS 0515
      T = Y/T                                                           RAS 0516
      X = Y - 0.5*T                                                     RAS 0517
      Y = T                                                             RAS 0518
  892 CONTINUE                                                          RAS 0519
      X = X+Y                                                           RAS 0520
  893 CONTINUE                                                          RAS 0521
      T = W*(X**2)                                                      RAS 0522
      CALL FET(FT,X)                                                    RAS 0523
      CALL COPHI(CPHI,T,J,LJ)                                           RAS 0524
      SPHI = 1.5*CPHI**2-.5                                             RAS 0525
      TX = FT*Y/X**2                                                    RAS 0526
      ALPHA1(J) = ALPHA1(J) - CPHI*TX*CALPH(LJ)                         RAS 0527
      ALPHA2(J) = ALPHA2(J) - SPHI*TX*CALPH(LJ)                         RAS 0528
      ALPHA3(J) = ALPHA3(J) - TX*CALPH(LJ)                              RAS 0529
      TX = FT*Y                                                         RAS 0530
      BETA1(J) = BETA1(J) + CPHI*TX*CD                                  RAS 0531
      BETA2(J) = BETA2(J) + SPHI*TX*CD                                  RAS 0532
      BETA3(J) = BETA3(J) + TX*CD                                       RAS 0533
      TX = FT*Y*X**2/2.                                                 RAS 0534
      GAM1(J) = GAM1(J) + CPHI*TX*CG                                    RAS 0535
      GAM2(J) = GAM2(J) + SPHI*TX*CG                                    RAS 0536
      GAM3(J) = GAM3(J) + TX*CG                                         RAS 0537
      NQP = NQP + 1                                                     RAS 0538
      IF(NQP-NX)90,899,899                                              RAS 0539
  899 CONTINUE                                                          RAS 0540
      X = X + 0.5*Y                                                     RAS 0541
      NQP = 0.                                                          RAS 0542
      Y = 10.*Y                                                         RAS 0543
      X = X - 0.5*Y                                                     RAS 0544
   90 CONTINUE                                                          RAS 0545
      X = X + 0.5*Y*(1.+OVER)                                           RAS 0546
      Y = OVER*Y                                                        RAS 0547
      T = W*X**2                                                        RAS 0548
      CALL FET(FT,X)                                                    RAS 0549
      CALL COPHI(CPHI,T,J,LJ)                                           RAS 0550
      SPHI = 1.5*CPHI**2-.5                                             RAS 0551
      TX = FT*Y/X**2                                                    RAS 0552
      ALPHA1(J) = ALPHA1(J) - CPHI*TX*CALPH(LJ)                         RAS 0553
      ALPHA2(J) = ALPHA2(J) - SPHI*TX*CALPH(LJ)                         RAS 0554
      ALPHA3(J) = ALPHA3(J) - TX*CALPH(LJ)                              RAS 0555
      TX = FT*Y                                                         RAS 0556
      BETA1(J) = BETA1(J) + CPHI*TX*CD                                  RAS 0557
      BETA2(J) = BETA2(J) + SPHI*TX*CD                                  RAS 0558
```

```
      BETA3(J) = BETA3(J) + TX*CD                                       RAS 0559
      TX = TX*X**2/2.                                                   RAS 0560
      GAM1(J) = GAM1(J) + CPHI*TX*CG                                    RAS 0561
      GAM2(J) = GAM2(J) + SPHI*TX*CG                                    RAS 0562
      GAM3(J) = GAM3(J) + TX*CG                                         RAS 0563
  149 CONTINUE                                                          RAS 0564
 1410 CONTINUE                                                          RAS 0565
      ALPHA1(J) = ALPHA1(J) + ALPHA(J)                                  RAS 0566
      ALPHA2(J) = ALPHA2(J) + ALPHA(J)                                  RAS 0567
      ALPHA3(J) = ALPHA3(J) + ALPHA(J)                                  RAS 0568
      BETA1(J) = BETA1(J) + SE(J)                                       RAS 0569
      BETA2(J) = BETA2(J) + SE(J)                                       RAS 0570
      BETA3(J) = BETA3(J) + SE(J)                                       RAS 0571
      IF(J-I)150,92,93                                                  RAS 0572
   92 CONTINUE                                                          RAS 0573
      A = ALPHA1(J)/BETA1(J)                                            RAS 0574
      Z2= ALPHA2(J)/BETA2(J)                                            RAS 0575
      AO= ALPHA3(J)/BETA3(J)                                            RAS 0576
      A1 = GAM1(J)/BETA1(J)                                             RAS 0577
      A2 = GAM2(J)/BETA2(J)                                             RAS 0578
      Q  = GAM3(J)/BETA3(J)                                             RAS 0579
      GO TO 150                                                         RAS 0580
   93 CONTINUE                                                          RAS 0581
      X = ALPHA1(J)/BETA1(J)                                            RAS 0582
      Y = ALPHA2(J)/BETA2(J)                                            RAS 0583
      Z = ALPHA3(J)/BETA3(J)                                            RAS 0584
      W = GAM1(J)/BETA1(J)                                              RAS 0585
      T = GAM2(J)/BETA2(J)                                              RAS 0586
      TX= GAM3(J)/BETA3(J)                                              RAS 0587
      EXINT1(J) = EXINT1(K) + U*(X+A)                                   RAS 0588
      A = X                                                             RAS 0589
      EXINT2(J) = EXINT2(K) + U*(Y+Z2)                                  RAS 0590
      Z2 = Y                                                            RAS 0591
      EXINT3(J) = EXINT3(K) + U*(Z + AO)                                RAS 0592
      AO = Z                                                            RAS 0593
      IF(K-I)150,931,933                                                RAS 0594
  931 CONTINUE                                                          RAS 0595
      IF(EXINT1(K)+600.)932,933,933                                     RAS 0596
  932 CONTINUE                                                          RAS 0597
      EXINT1(K) = 1.E-250                                               RAS 0598
      EXINT2(K) = 1.E-250                                               RAS 0599
      EXINT3(K) = 1.E-250                                               RAS 0600
      GO TO 934                                                         RAS 0601
  933 CONTINUE                                                          RAS 0602
      EXINT1(K) = EXP(EXINT1(K))                                        RAS 0603
      EXINT2(K) = EXP(EXINT2(K))                                        RAS 0604
      EXINT3(K) = EXP(EXINT3(K))                                        RAS 0605
  934 CONTINUE                                                          RAS 0606
      X = EXP(EXINT1(J))                                                RAS 0607
      Y = EXP(EXINT2(J))                                                RAS 0608
      Z = EXP(EXINT3(J))                                                RAS 0609
      RP(J) = RP(K) + U*(X/BETA1(J) + EXINT1(K)/BETA1(K))               RAS 0610
      DR(J) = (1.-ALPHA1(J)*RP(J)/X)/BETA1(J)-(1.-ALPHA1(K)*RP(K)/EXINT1RAS 0611
     1(K))/BETA1(K)                                                     RAS 0612
      DR(J) = DR(J)*(W*X+A1*EXINT1(K))/2.                               RAS 0613
      DR(J) = DR(J) + DR(K)                                             RAS 0614
      RR2(J) = RR2(K)+(((RP(J)+DR(J))*Y)/(BETA2(J)*X)+((RP(K)+DR(K))*EXIRAS 0615
     1NT2(K))/(BETA2(K)*EXINT1(K)))*2.*U                                RAS 0616
      RC2(J) = RC2(K) + (((RP(J)+DR(J))*Z)/(BETA3(J)*X) + ((RP(K)+DR(K))RAS 0617
     1*EXINT3(K))/(BETA3(K)*EXINT1(K)))*2.*U                            RAS 0618
      Z1= ((2.*(RP(J)+DR(J))/X-ALPHA2(J)*RR2(J)/Y)/BETA2(J)-(2.*(RP(K)+DRAS 0619
     1R(K))/EXINT1(K)-ALPHA2(K)*RR2(K)/EXINT2(K))/BETA2(K))             RAS 0620
```

```
      Z1 = Z1*(T*Y+A2*EXINT2(K))/2.                                     RAS 0621
      DRR2(J) = Z1 + DRR2(K)                                            RAS 0622
      Z1 = ((2.*(RP(J)+DR(J))/X-ALPHA3(J)*RC2(J)/Z)/BETA3(J)-(2.*(RP(K)+RAS 0623
     1DR(K))/EXINT1(K)-ALPHA3(K)*RC2(K)/EXINT3(K))/BETA3(K))            RAS 0624
      Z1 = Z1*(TX*Z+Q*EXINT3(K))/2.                                     RAS 0625
      DRC2(J) = Z1 + DRC2(K)                                            RAS 0626
      R(J) = R(K) + U*(Z/BETA3(J)+EXINT3(K)/BETA3(K))                   RAS 0627
      Z1 = (1.-ALPHA3(J)*R(J)/Z)/BETA3(J)-(1.-ALPHA3(K)*R(K)/EXINT3(K))/RAS 0628
     1BETA3(K)                                                          RAS 0629
      Z1 = Z1*(TX*Z+Q*EXINT3(K))/2.                                     RAS 0630
      DRO(J) = Z1 + DRO(K)                                              RAS 0631
      A1 = W                                                            RAS 0632
      A2 = T                                                            RAS 0633
      Q = TX                                                            RAS 0634
      K = 5*NDIVE + I                                                   RAS 0635
      IF(J-K)150,14951,150                                              RAS 0636
14951 CONTINUE                                                          RAS 0637
      Y = NDIVE                                                         RAS 0638
      U = U*Y                                                           RAS 0639
      JJJ = NDIVE                                                       RAS 0640
  150 CONTINUE                                                          RAS 0641
      EXINT1(NE) = EXP(EXINT1(NE))                                      RAS 0642
      EXINT2(NE) = EXP(EXINT2(NE))                                      RAS 0643
      EXINT3(NE) = EXP(EXINT3(NE))                                      RAS 0644
      JJJ = 1                                                           RAS 0645
      K = 5*NDIVE + I                                                   RAS 0646
      DO 155 J = I,NE,JJJ                                               RAS 0647
      R(J) = R(J) + DRO(J)                                              RAS 0648
      RP(J) = RP(J) + DR(J)                                             RAS 0649
      RR2(J) = RR2(J) + DRR2(J)                                         RAS 0650
      RC2(J) = RC2(J) + DRC2(J)                                         RAS 0651
      RP2(J) = RP(J)**2                                                 RAS 0652
      IF(J-K)155,15451,155                                              RAS 0653
15451 CONTINUE                                                          RAS 0654
      JJJ = NDIVE                                                       RAS 0655
  155 CONTINUE                                                          RAS 0656
      IF(I.GT.NSTP) KOAM = I/NSTP+NSTP/NDIVE                            RAS 0657
      IF(I.LE.NSTP) KOAM = I/NDIVE + 1                                  RAS 0658
      KWR = 0                                                           RAS 0659
      K = 0                                                             RAS 0660
      DO 175 J = NSTP,NE,NSTP                                           RAS 0661
      KWR = KWR + 1                                                     RAS 0662
      IF(I-J)165,175,175                                                RAS 0663
  165 CONTINUE                                                          RAS 0664
      IF(J-NE)167,166,175                                               RAS 0665
  166 CONTINUE                                                          RAS 0666
      K = 1                                                             RAS 0667
  167 CONTINUE                                                          RAS 0668
      RP(J) = RP(J)/EXINT1(J)                                           RAS 0669
      DELTA(J) = (2.*RR2(J)/EXINT2(J)+RC2(J)/EXINT3(J))/3.              RAS 0670
      DELTA(J) = DELTA(J)-RP2(J)/(EXINT1(J)**2)                         RAS 0671
      DP(J) = 2.*(RC2(J)/EXINT3(J)-RR2(J)/EXINT2(J))/3.                 RAS 0672
      IF(DELTA(J))170,170,171                                           RAS 0673
  170 CONTINUE                                                          RAS 0674
      DELTA(J) = 1.                                                     RAS 0675
  171 CONTINUE                                                          RAS 0676
      IF(DP(J))1715,1715,1716                                           RAS 0677
 1715 CONTINUE                                                          RAS 0678
      DP(J) = 1.                                                        RAS 0679
 1716 CONTINUE                                                          RAS 0680
      DP(J) = SQRT(DP(J))                                               RAS 0681
      DELTA(J) = SQRT(DELTA(J))                                         RAS 0682
```

```
      R(J) = R(J)/EXINT3(J)                                             RAS 0683
      DAMRP(KWR,KDAM)=RP(J)                                             RAS 0684
      DAMDEL(KWR,KDAM)=DELTA(J)                                         RAS 0685
      DAMR(KWR,KDAM)=R(J)                                               RAS 0686
      DAMDP(KWR,KDAM)=DP(J)                                             RAS 0687
      IF(K)172,172,173                                                  RAS 0688
  172 CONTINUE                                                          RAS 0689
      K = 1                                                             RAS 0690
      GO TO 175                                                         RAS 0691
  173 CONTINUE                                                          RAS 0692
      NN = J-NSTP                                                       RAS 0693
      K = 0                                                             RAS 0694
      Y = NDIV                                                          RAS 0695
      Y = Y/NMULT                                                       RAS 0696
      X = J                                                             RAS 0697
      X = X/Y                                                           RAS 0698
      T = I                                                             RAS 0699
      T = T/Y                                                           RAS 0700
      W = NN                                                            RAS 0701
      W = W/Y                                                           RAS 0702
      PRINT IVAR,X,T,RP(J),DELTA(J),R(J),DP(J),W,T,RP(NN),DELTA(NN),R(NNRAS 0703
     1),DP(NN)                                                          RAS 0704
  175 CONTINUE                                                          RAS 0705
      IF(I-2*NDIVE)176,176,190                                          RAS 0706
  176 CONTINUE                                                          RAS 0707
      KWR=0                                                             RAS 0708
      DO 185 J = NSTP,NE,NSTP                                           RAS 0709
      KWR=KWR+1                                                         RAS 0710
      IF(I-2*NDIVE)177,178,185                                          RAS 0711
  177 CONTINUE                                                          RAS 0712
      RPO(J) = 2.*RP(J)                                                 RAS 0713
      DELTAO(J) = 2.*DELTA(J)                                           RAS 0714
      RO(J) = 2.*R(J)                                                   RAS 0715
      DPO(J) = 2.*DP(J)                                                 RAS 0716
      GO TO 185                                                         RAS 0717
  178 CONTINUE                                                          RAS 0718
      RPO(J) = RPO(J) - RP(J)                                           RAS 0719
      DELTAO(J) = DELTAO(J) - DELTA(J)                                  RAS 0720
      RO(J) = RO(J) - R(J)                                              RAS 0721
      DPO(J) = DPO(J) - DP(J)                                           RAS 0722
      DAMRP(KWR,1)=RPO(J)                                               RAS 0723
      DAMDEL(KWR,1)=DELTAO(J)                                           RAS 0724
      DAMR(KWR,1)=RO(J)                                                 RAS 0725
      DAMDP(KWR,1)=DPO(J)                                               RAS 0726
  185 CONTINUE                                                          RAS 0727
  190 CONTINUE                                                          RAS 0728
      PRINT 3                                                           RAS 0729
  200 CONTINUE                                                          RAS 0730
      K = 0                                                             RAS 0731
      L = 2*NSTP                                                        RAS 0732
      DO 250 J = L,NE,L                                                 RAS 0733
      IF(J+NSTP-NE)202,201,202                                          RAS 0734
  201 CONTINUE                                                          RAS 0735
      L = NSTP                                                          RAS 0736
  202 CONTINUE                                                          RAS 0737
      NN = J - NSTP                                                     RAS 0738
      Y = NDIV                                                          RAS 0739
      Y = Y/NMULT                                                       RAS 0740
      X = J                                                             RAS 0741
      X = X/Y                                                           RAS 0742
      W = NN                                                            RAS 0743
      W = W/Y                                                           RAS 0744
```

```
      T = 0.                                                            RAS 0745
      PRINT IVAR,X,T,RPO(J),DELTAO(J),RO(J),DPO(J),W,T,RPO(NN),DELTAO(NNRAS 0746
     1),RO(NN),DPO(NN)                                                  RAS 0747
  250 CONTINUE                                                          RAS 0748
      WRITE TAPE20,((DAMRP(J,K),DAMDEL(J,K),DAMR(J,K),DAMDP(J,K),K=1,60)RAS 0749
     1,J=1,50)                                                          RAS 0750
      REWIND 10                                                         RAS 0751
      GO TO 350                                                         RAS 0752
  255 CONTINUE                                                          RAS 0753
      ENDFILE 20                                                        RAS 0754
      REWIND 20                                                         RAS 0755
      END                                                               RAS 0756
```

C. DAMG2

```
      PROGRAM DAMG2(INPUT,OUTPUT,TAPE20,TAPE30)                         DAM 0001
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                    ISSUED BY SANDIA LABORATORIES.                *
C     *              A PRIME CONTRACTOR TO THE UNITED STATES             *
C     *          ENERGY RESEARCH AND DEVELOPEMENT ADMINISTRATION.        *
C     * * * * * * * * * * * * * * *   NOTICE   * * * * * * * * * * * * * *
C     * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE  *
C     * UNITED STATES GOVERNMENT. NEITHER THE UNITED STATES NOR THE      *
C     * UNITED STATES ENERGY RESEARCH AND DEVELOPEMENT ADMINISTRATION,   *
C     * NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS,        *
C     * SUBCONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS  *
C     * OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR *
C     * THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,     *
C     * APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS  *
C     * USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.                   *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * THE BASIC REFERENCE DOCUMENTS FOR THIS CODE ARE SLA-73-0416,     *
C     * APRIL,1973, AND SAND-75-0622, DECEMBER, 1975.                    *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * THIS CODE HAS BEEN RELEASED FOR UNLIMITED DISTRIBUTION.          *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

    1 FORMAT(I4,3I2,/,(8E10.4))                                          DAM 0043
    2 FORMAT(8A10)                                                       DAM 0044
    3 FORMAT(80H                                                         DAM 0045
     1                                  )                                DAM 0046
    4 FORMAT(1H1)                                                        DAM 0047
    5 FORMAT(//)                                                         DAM 0048
    6 FORMAT(16I5)                                                       DAM 0049
    7 FORMAT(5/)                                                         DAM 0050
    8 FORMAT(*      NDAM        NE      NSTP      NDIV     NDIVE     NMUDAM 0051
      READ 6,NE,NSTP,NDIV,NDIVE,NMULT,NTYPE,JSTP,NXTND,KE                DAM 0052
    9 FORMAT(I7,2(F8.4,F9.4,G14.4,7X),F8.4,F9.4,G14.4)                   DAM 0053
   10 FORMAT(19H*****  UNITS  ***** ,2/*  ENERGY  -  KEV*/*  DEPTH,RANGEDAM 0054
     1,DELTA RANGE  -  ANGSTROMS*/*  ENERGY PARTITION,ION DISTRIBUTION,IDAM 0055
     2NTEGRATED ENERGY DISTRIBUTION  -  DIMENSIONLESS*/*  ENERGY DISTRIBDAM 0056
     3UTION  -  EV/ANGSTROM*,3/)                                         DAM 0057
   11 FORMAT(6X,2(6X,15HINCIDENT ENERGY,10X,3H***,4X),6X,15HINCIDENT ENEDAM 0058
     1RGY,/6X,2(31X,3H***,4X)/6X,2(F14.2   ,*-KEV*,13X,3H***,4X),F14.2  DAM 0059
     2 ,*-KEV*2(/6X,31X,3H***,35X,3H***))                                DAM 0060
   12 FORMAT(6X,2(8X,11HPROJ. RANGE,12X,3H***,4X),8X,11HPROJ. RANGE,/6X,DAM 0061
     12(31X,3H***,4X),/6X,2(6X,F5.0,*-ANGSTROMS*,10X,3H***,4X),6X,F5.0,*DAM 0062
```

```
  2-ANGSTROMS*,2(/6X,31X,3H***,35X,3H***))                              DAM 0063
13 FORMAT(6X,2(8X,11HDELTA RP    ,12X,3H***,4X),8X,11HDELTA RP    ,/6X,DAM 0064
  12(31X,3H***,4X),/6X,2(6X,F5.0,*-ANGSTROMS*,10X,3H***,4X),6X,F5.0,*DAM 0065
  2-ANGSTROMS*,2(/37X,3H***,35X,3H***))                                 DAM 0066
14 FORMAT(6X,2(4X,19HDELTA PERPINDICULAR,8X,3H***,4X),4X,19HDELTA PERDAM 0067
  1PINDICULAR,/6X,2(31X,3H***,4X),/6X,2(6X,F5.0,*-ANGSTROMS*,10X,3H**DAM 0068
  2*,4X),6X,F5.0,*-ANGSTROMS*,2(/37X,3H***,35X,3H***))                  DAM 0069
15 FORMAT(6X,2(1X,28HTOTAL ENERGY IN DISTRIBUTION,2X,3H***,4X),1X,28HDAM 0070
  1TOTAL ENERGY IN DISTRIBUTION,/6X,2(31X,3H***,4X),/6X,2(F14.2,*-KEVDAM 0071
  2*,13X,3H***,4X),F14.2,*-KEV*,2(/37X,3H***,35X,3H***))                DAM 0072
16 FORMAT(6X,2(5X,16HENERGY PARTITION,10X,3H***,4X),5X,16HENERGY PARTDAM 0073
  1ITION,/6X,2(31X,3H***,4X),/6X,2(9X,F9.4,13X,3H***,4X),9X,F9.4,2(/3DAM 0074
  27X,3H***,35X,3H***))                                                 DAM 0075
17 FORMAT(3(37X,3H***,35X,3H*** /),* X/RP *,2(2X,5HFINAL,1X,11HINTEGRDAM 0076
  1ATED,3X,6HENERGY,4X,3H***,4X),2X,5HFINAL,1X,11HINTEGRATED,3X,6HENEDAM 0077
  2RGY,/6X,2(3X,3HION,4X,6HENERGY,5X,7HDISTR. ,3X,3H***,4X),3X,3HION,DAM 0078
  34X,6HENERGY,5X,7HDISTR. ,/* (PCT)*,2(2X,6HDISTR.,2X,6HDISTR.,15X  DAM 0079
  4 ,3X,3H***,4X),,2X,6HDISTR.,2X,6HDISTR.,15X,          2(/37X,3H***,DAM 0080
  535X,3H***))                                                          DAM 0081
18 FORMAT(*  VALUES FOR CONTOUR PLOT     ENERGY DISTRIBUTION(EV/ANGSTDAM 0082
  1ROM)    DEPTH(ANGSTROMS)*3/)                                         DAM 0083
19 FORMAT(* ENERGY(KEV)*,F6.1,4(6X,F10.0),2/2X,5(*   DISTR.  DEPTH*))DAM 0084
20 FORMAT(2X,5(1X,F8.1,F7.0))                                           DAM 0085
22 FORMAT(I3,11F7.2)                                                    DAM 0086
23 FORMAT(7H ENERGY,6X,9HX-AVERAGE,12X,10HDELTA X-SQ,11X,6HENERGY,15XDAM 0087
  1,12HTOTAL ENERGY,/34X,7HAVERAGE,14X,9HPARTITION,12X,15HIN DISTRIBUDAM 0088
  2TION,3/)                                                             DAM 0089
24 FORMAT(I6,2X,4(G15.5,6X))                                            DAM 0090
25 FORMAT(*    I       DPCT        XMOM3(I)       DXMOM3(I)*,2/,(I5,F10.DAM 0091
  16,2E15.4))                                                           DAM 0092
26 FORMAT(50X,*ENERGY DEPOSITED IN TARGET =*,G15.5,2/)                  DAM 0093
27 FORMAT(16F5.0)                                                       DAM 0094
28 FORMAT(1X,*       ALAT          P1          Z1      P2(1)      Z2(1)      DAM 0095
  1P2(2)      Z2(2)      P2(3)      Z2(3)      P2(4)      Z2(4)*,2/,11F10.DAM 0096
  23,3/)                                                                DAM 0097
   DIMENSION D(1600),TEST(80),RP(50,60),DELTA(50,60),R(50,60)           DAM 0098
   DIMENSION DP(50,60),DAMG(25,101),DIST(25,101),DINT(25,101),G(25)     DAM 0099
   DIMENSION D1(1600)                                                   DAM 0100
   DIMENSION E(50),EP(60)                                               DAM 0101
   DIMENSION INDEX(25),F(25)                                            DAM 0102
   DIMENSION RC(25,250,2)                                               DAM 0103
   DIMENSION XMOM1(25),XMOM2(25)                                        DAM 0104
   DIMENSION XI00(25)                                                   DAM 0105
   DIMENSION XI01(25),XI02(25),XI03(25),XI20(25),XI21(25),XI30(25)      DAM 0106
   DIMENSION XMOM3(25)                                                  DAM 0107
   DIMENSION RD(1600),DRD(1600),RD1(1600),DRD1(1600)                    DAM 0108
   DIMENSION SE(1600),SE1(1600)                                         DAM 0109
   TYPE REAL LEN(25)                                                    DAM 0110
   COMMON/TWENVAR/P2(4),Z2(4),P1,Z1,NTYPE,EM,ITER,ALAT                  DAM 0111
   CALL WIND30                                                          DAM 0112
30 CONTINUE                                                             DAM 0113
   A = 1./2.506628274                                                   DAM 0114
   DO 40 I = 1,25                                                       DAM 0115
   XI01(I)=0.                                                           DAM 0116
   XI02(I)=0.                                                           DAM 0117
   XI03(I)=0.                                                           DAM 0118
   XI20(I)=0.                                                           DAM 0119
   XI21(I)=0.                                                           DAM 0120
   XI30(I)=0.                                                           DAM 0121
   DO 40 J = 1,250                                                      DAM 0122
   RC(I,J,1) = 0.                                                       DAM 0123
   RC(I,J,2) = 0.                                                       DAM 0124
```

```
      IF(J-101)35,35,40                                                 DAM 0125
   35 CONTINUE                                                          DAM 0126
      DIST(I,J) = 0.                                                    DAM 0127
      DAMG(I,J) = 0.                                                    DAM 0128
      DINT(I,J) = 0.                                                    DAM 0129
   40 CONTINUE                                                          DAM 0130
      DO 405 I = 1,50                                                   DAM 0131
      DO 405 J = 1,60                                                   DAM 0132
      RP(I,J) = 0.                                                      DAM 0133
      R(I,J) = 0.                                                       DAM 0134
      DELTA(I,J) = 0.                                                   DAM 0135
      DP(I,J) = 0.                                                      DAM 0136
  405 CONTINUE                                                          DAM 0137
      DO 407 I = 1,160                                                  DAM 0138
      RD(I) = 0.                                                        DAM 0139
      RD1(I) = 0.                                                       DAM 0140
      DRD(I) = 0.                                                       DAM 0141
      DRD1(I) = 0.                                                      DAM 0142
  407 CONTINUE                                                          DAM 0143
      PRINT 4                                                           DAM 0144
      READ 3                                                            DAM 0145
      PRINT 3                                                           DAM 0146
      PRINT 5                                                           DAM 0147
      READ 27,ALAT,P1,Z1,(P2(J),Z2(J),J=1,4)                            DAM 0148
      PRINT 28,ALAT,P1,Z1,(P2(J),Z2(J),J=1,4)                           DAM 0149
      PRINT 5                                                           DAM 0150
      READ 6,NE,NSTP,NDIV,NDIVE,NMULT,NTYPE,JSTP,NXTND,KE               DAM 0151
      NDAM = NE/NSTP                                                    DAM 0152
      STP = JSTP                                                        DAM 0153
      STP = STP/10.                                                     DAM 0154
      IF(NDAM)41,41,42                                                  DAM 0155
   41 CONTINUE                                                          DAM 0156
      NDAM = 1                                                          DAM 0157
   42 CONTINUE                                                          DAM 0158
      IF(KE.NE.0)43,44                                                  DAM 0159
   43 CONTINUE                                                          DAM 0160
      EM = KE                                                           DAM 0161
      CALL TWENTY                                                       DAM 0162
      IF(ITER.EQ.0)455,432                                              DAM 0163
  432 CONTINUE                                                          DAM 0164
      IF(NTYPE.NE.2)READ TAPE20,KKK,NDI,NMU,NTY,(D1(J),RD1(J),DRD1(J),J=DAM 0165
     12,KKK)                                                            DAM 0166
      IF(ITER.EQ.1) READ(20)                                            DAM 0167
      IF(NTYPE.EQ.2)READ TAPE20,KKK,NDI,NMU,NTY,(SE1(J),D1(J),RD1(J),DRDDAM 0168
     11(J),J=2,KKK)                                                     DAM 0169
      READ(20)                                                          DAM 0170
      COMP1=NMU                                                         DAM 0171
      COMP1=COMP1/NDI                                                   DAM 0172
   44 CONTINUE                                                          DAM 0173
      EM = NE*NMULT                                                     DAM 0174
      EM = EM/NDIV                                                      DAM 0175
      CALL TWENTY                                                       DAM 0176
      IF(ITER.EQ.0)455,45                                               DAM 0177
   45 CONTINUE                                                          DAM 0178
      IF(NTYPE.NE.2)READ TAPE20,KKK,NDI,NMU,NTY,(D(J),RD(J),DRD(J),J=2,KDAM 0179
     1KK)                                                               DAM 0180
      IF(ITER.EQ.1) READ(20)                                            DAM 0181
      IF(NTYPE.EQ.2)READ TAPE20,KKK,NDI,NMU,NTY,(SE(J),D(J),RD(J),DRD(J)DAM 0182
     1,J=2,KKK)                                                         DAM 0183
      COMP = NMU                                                        DAM 0184
      COMP = COMP/NDI                                                   DAM 0185
      WRITE TAPE30,NDAM,NE,NSTP,NDIV,NDIVE,NMULT,NTYPE,JSTP,NXTND,KE,ITEDAM 0186
```

```
     1R,ALAT,P1,Z1,(P2(J),Z2(J),J=1,4)                                  DAM 0187
      PRINT 5                                                           DAM 0188
      PRINT 8,NDAM,NE,NSTP,NDIV,NDIVE,NMULT,NTYPE,JSTP,NXTND,KE,ITER    DAM 0189
      SE(1) = 0.                                                        DAM 0190
      SE1(1) = 0.                                                       DAM 0191
      RD(1) = 0.                                                        DAM 0192
      RD1(1) = 0.                                                       DAM 0193
      DRD(1) = 0.                                                       DAM 0194
      DRD1(1) = 0.                                                      DAM 0195
      D(1) = 0.                                                         DAM 0196
      D1(1) = 0.                                                        DAM 0197
      NE = NE/NSTP                                                      DAM 0198
      NSTP = NSTP/NDIVE                                                 DAM 0199
      READ TAPE20,((RP(J,K),DELTA(J,K),R(J,K),DP(J,K),K=1,60),J=1,50)   DAM 0200
      REWIND 20                                                         DAM 0201
      IF(NXTND.GT.1)535,545                                             DAM 0202
  535 CONTINUE                                                          DAM 0203
      DO 54 J=1,50                                                      DAM 0204
      DO 54 K=1,60                                                      DAM 0205
      RP(J,K)=NXTND*RP(J,K)                                             DAM 0206
      DELTA(J,K)=NXTND*DELTA(J,K)                                       DAM 0207
      R(J,K) = NXTND*R(J,K)                                             DAM 0208
      DP(J,K)=NXTND*DP(J,K)                                             DAM 0209
   54 CONTINUE                                                          DAM 0210
  545 CONTINUE                                                          DAM 0211
      DO 5006 I = 1,NE                                                  DAM 0212
      JRP = 1                                                           DAM 0213
      JDELT = 1                                                         DAM 0214
      DO 5006 J = 2,NSTP                                                DAM 0215
      L = NSTP+1-J                                                      DAM 0216
      X = DELTA(I,L) - DELTA(I,L+1)                                     DAM 0217
      IF(X)5001,5001,5002                                               DAM 0218
 5001 CONTINUE                                                          DAM 0219
      X = JDELT                                                         DAM 0220
      JDELT = JDELT + 1                                                 DAM 0221
      DELTA(I,L) =(DELTA(I,L+1)-DELTA(I,L+2))/X + DELTA(I,L+1)          DAM 0222
 5002 CONTINUE                                                          DAM 0223
      X = RP(I,L)-RP(I,L+1)                                             DAM 0224
      IF(X)5003,5003,5004                                               DAM 0225
 5003 CONTINUE                                                          DAM 0226
      X = JRP                                                           DAM 0227
      JRP = JRP + 1                                                     DAM 0228
      RP(I,L) = (RP(I,L+1)-RP(I,L+2))/X + RP(I,L+1)                     DAM 0229
      R(I,L) = (R(I,L+1) - R(I,L+2))/X + R(I,L+1)                       DAM 0230
 5004 CONTINUE                                                          DAM 0231
      X = R(I,L) - R(I,L+1)                                             DAM 0232
      IF(X)5005,5005,5006                                               DAM 0233
 5005 CONTINUE                                                          DAM 0234
      X = JRP                                                           DAM 0235
      JRP = JRP + 1                                                     DAM 0236
      RP(I,L) = (RP(I,L+1)-RP(I,L+2))/X + RP(I,L+1)                     DAM 0237
      R(I,L) = (R(I,L+1) - R(I,L+2))/X + R(I,L+1)                       DAM 0238
 5006 CONTINUE                                                          DAM 0239
      X = NDIV                                                          DAM 0240
      Y = NDIVE*NMULT                                                   DAM 0241
      DO 59 I = 1,60                                                    DAM 0242
      IF(I-NSTP)591,591,592                                             DAM 0243
  591 CONTINUE                                                          DAM 0244
      EP(I) = I-1                                                       DAM 0245
      EP(I) = EP(I)*Y/X                                                 DAM 0246
      GO TO 59                                                          DAM 0247
  592 CONTINUE                                                          DAM 0248
```

```
      EP(I) = (I-NSTP)*NSTP                                             DAM 0249
      EP(I) = EP(I)*Y/X                                                 DAM 0250
   59 CONTINUE                                                          DAM 0251
      X = NDIV                                                          DAM 0252
      Y = NDIVE*NMULT                                                   DAM 0253
      DO 69 I = 1,50                                                    DAM 0254
      E(I) = NSTP*I                                                     DAM 0255
      E(I) = E(I)*Y/X                                                   DAM 0256
      J = I+NSTP                                                        DAM 0257
      RP(I,J) = 0.                                                      DAM 0258
      R(I,J) = 0.                                                       DAM 0259
      DELTA(I,J) = 0.                                                   DAM 0260
      DP(I,J) = 0.                                                      DAM 0261
   69 CONTINUE                                                          DAM 0262
      IF(NE-NDAM)601,602,602                                            DAM 0263
  601 CONTINUE                                                          DAM 0264
      NDAM = NE                                                         DAM 0265
  602 CONTINUE                                                          DAM 0266
      KK = NE/NDAM                                                      DAM 0267
      LL = 1                                                            DAM 0268
      KXKK = RP(NE,1) + 10.*DELTA(NE,1)                                 DAM 0269
      DO 300 I = KK,NE,KK                                               DAM 0270
      IF(I-II)613,613,610                                               DAM 0271
  610 CONTINUE                                                          DAM 0272
      II = II + 3*KK                                                    DAM 0273
      JJ = II                                                           DAM 0274
  611 CONTINUE                                                          DAM 0275
      IF(RP(JJ,1))612,612,613                                           DAM 0276
  612 CONTINUE                                                          DAM 0277
      JJ = JJ-KK                                                        DAM 0278
      GO TO 611                                                         DAM 0279
  613 CONTINUE                                                          DAM 0280
      LEN(LL) = RP(I,1)/50.                                             DAM 0281
      K = I+NSTP                                                        DAM 0282
      KX = 1                                                            DAM 0283
      JJ = 1                                                            DAM 0284
      DO 150 L = 1,KXKK,KX                                              DAM 0285
      X = L                                                             DAM 0286
      Y = KX                                                            DAM 0287
      IF(X-R(I,1))73,73,72                                              DAM 0288
   72 CONTINUE                                                          DAM 0289
      Y = Y-X+R(I,1)                                                    DAM 0290
      Z = KX                                                            DAM 0291
      X = X-Z + Y                                                       DAM 0292
   73 CONTINUE                                                          DAM 0293
      X = X-.5*Y                                                        DAM 0294
      KKK = 1                                                           DAM 0295
      DO 80 M = JJ,K,KKK                                                DAM 0296
      MM = K+1-M                                                        DAM 0297
      Z = R(I,MM)-X                                                     DAM 0298
      IF(Z)80,80,75                                                     DAM 0299
   75 CONTINUE                                                          DAM 0300
      KKK = K                                                           DAM 0301
   80 CONTINUE                                                          DAM 0302
      IF(MM-1)140,81,81                                                 DAM 0303
   81 CONTINUE                                                          DAM 0304
      IF(KKK-K)140,82,140                                               DAM 0305
   82 CONTINUE                                                          DAM 0306
      JJ = K+1-MM                                                       DAM 0307
      Z = Z/(R(I,MM) - R(I,MM+1))                                       DAM 0308
      DELT = Z*DELTA(I,MM+1) + (1.-Z)*DELTA(I,MM)                       DAM 0309
      DR = Z*RP(I,MM+1) + (1.-Z)*RP(I,MM)                               DAM 0310
```

```
      LM = EP(MM+1)                                                     DAM 0311
      IF(KE-LM)825,825,821                                              DAM 0312
  821 CONTINUE                                                          DAM 0313
      EY = 1./COMP1                                                     DAM 0314
      EY = (Z*EP(MM+1) + (1.-Z)*EP(MM))*EY + 1.                         DAM 0315
      LM = EY                                                           DAM 0316
      LN = LM+1                                                         DAM 0317
      Z = LM                                                            DAM 0318
      Z = EY-Z                                                          DAM 0319
      DMG = Z*D1(LN) + (1.-Z)*D1(LM)                                    DAM 0320
      XD = Z*RD1(LN)+(1.-Z)*RD1(LM)                                     DAM 0321
      DXD = Z*DRD1(LN)+(1.-Z)*DRD1(LM)                                  DAM 0322
      DMG1 = Z*SE1(LN)+(1.-Z)*SE1(LM)                                   DAM 0323
      GO TO 826                                                         DAM 0324
  825 CONTINUE                                                          DAM 0325
      EY = 1./COMP                                                      DAM 0326
      EY = (Z*EP(MM+1) + (1.-Z)*EP(MM))*EY + 1.                         DAM 0327
      LM = EY                                                           DAM 0328
      LN = LM+1                                                         DAM 0329
      Z = LM                                                            DAM 0330
      Z = EY-Z                                                          DAM 0331
      DMG = Z*D(LN)+(1.-Z)*D(LM)                                        DAM 0332
      XD = Z*RD(LN)+(1.-Z)*RD(LM)                                       DAM 0333
      DXD = Z*DRD(LN)+(1.-Z)*DRD(LM)                                    DAM 0334
      DMG1 = Z*SE(LN)+(1.-Z)*SE(LM)                                     DAM 0335
  826 CONTINUE                                                          DAM 0336
      DR1 = DR                                                          DAM 0337
      DELT1 = DELT                                                      DAM 0338
      DR = DR + 0.8*XD*NXTND                                            DAM 0339
      DXD = 0.8*DXD*NXTND                                               DAM 0340
      DELT = SQRT(DELT*DELT + DXD*DXD)                                  DAM 0341
      XI00(LL) = XI00(LL) + Y*DMG                                       DAM 0342
      XI01(LL)=XI01(LL) + Y*DMG*DR                                      DAM 0343
      XI02(LL) = XI02(LL) + Y*DMG*DR*DR                                 DAM 0344
      XI03(LL) = XI03(LL) + Y*DMG*DR*DR*DR                              DAM 0345
      XI20(LL) = XI20(LL) + Y*DMG*DELT*DELT                             DAM 0346
      XI21(LL) = XI21(LL) + Y*DMG*DELT*DELT*DR                          DAM 0347
      XI30(LL)=XI30(LL)+Y*DMG*DELT1*DELT1*DELT1                         DAM 0348
      IF(NTYPE.EQ.1) GO TO 827                                          DAM 0349
      XI00(LL) = XI00(LL) + Y*DMG1                                      DAM 0350
      XI01(LL) = XI01(LL) + Y*DMG1*DR1                                  DAM 0351
      XI02(LL) = XI02(LL) + Y*DMG1*DR1*DR1                              DAM 0352
      XI03(LL) = XI03(LL) + Y*DMG1*DR1*DR1*DR1                          DAM 0353
      XI20(LL) = XI20(LL) + Y*DMG1*DELT1*DELT1                          DAM 0354
      XI21(LL) = XI21(LL) + Y*DMG1*DELT1*DELT1*DR1                      DAM 0355
      XI30(LL) = XI30(LL) + Y*DMG1*DELT1*DELT1*DELT1                    DAM 0356
  827 CONTINUE                                                          DAM 0357
      DO 200 J = 1,101                                                  DAM 0358
      RO = (J-1)*LEN(LL)                                                DAM 0359
      Z = (DR-RO)/DELT                                                  DAM 0360
      Z1 = (DR1-RO)/DELT1                                               DAM 0361
      IF(Z+25.)200,83,83                                                DAM 0362
   83 CONTINUE                                                          DAM 0363
      IF(Z-25.)84,84,200                                                DAM 0364
   84 CONTINUE                                                          DAM 0365
      IF(DELT-10.)820,830,830                                           DAM 0366
  820 CONTINUE                                                          DAM 0367
      Y = .1                                                            DAM 0368
      X = L-1                                                           DAM 0369
      X = X-.5*Y                                                        DAM 0370
      DO 190 KR = 1,10                                                  DAM 0371
      X = X+Y                                                           DAM 0372
```

```
      Z = (R(I,MM)-X)/(R(I,MM)-R(I,MM+1))                               DAM 0373
      DELT = Z*DELTA(I,MM+1) + (1.-Z)*DELTA(I,MM)                       DAM 0374
      DR = Z*RP(I,MM+1) + (1.-Z)*RP(I,MM)                               DAM 0375
      LM = EP(MM+1)                                                     DAM 0376
      IF(KE-LM)835,835,831                                              DAM 0377
  831 CONTINUE                                                          DAM 0378
      EY = 1./COMP1                                                     DAM 0379
      EY = (Z*EP(MM+1) + (1.-Z)*EP(MM))*EY + 1.                         DAM 0380
      LM = EY                                                           DAM 0381
      LN = LM+1                                                         DAM 0382
      Z = LM                                                            DAM 0383
      Z = EY-Z                                                          DAM 0384
      DMG = Z*D1(LN) + (1.-Z)*D1(LM)                                    DAM 0385
      XD = Z*RD1(LN)+(1.-Z)*RD1(LM)                                     DAM 0386
      DXD = Z*DRD1(LN)+(1.-Z)*DRD1(LM)                                  DAM 0387
      DMG1 = Z*SE1(LN)+(1.-Z)*SE1(LM)                                   DAM 0388
      GO TO 836                                                         DAM 0389
  835 CONTINUE                                                          DAM 0390
      EY = 1./COMP                                                      DAM 0391
      EY = (Z*EP(MM+1) + (1.-Z)*EP(MM))*EY + 1.                         DAM 0392
      LM = EY                                                           DAM 0393
      LN = LM+1                                                         DAM 0394
      Z = LM                                                            DAM 0395
      Z = EY-Z                                                          DAM 0396
      DMG = Z*D(LN)+(1.-Z)*D(LM)                                        DAM 0397
      XD = Z*RD(LN)+(1.-Z)*RD(LM)                                       DAM 0398
      DXD = Z*DRD(LN)+(1.-Z)*DRD(LM)                                    DAM 0399
      DMG1 = Z*SE(LN)+(1.-Z)*SE(LM)                                     DAM 0400
  836 CONTINUE                                                          DAM 0401
      DR1 = DR                                                          DAM 0402
      DELT1 = DELT                                                      DAM 0403
      DR = DR + 0.8*XD*NXTND                                            DAM 0404
      DXD = 0.8*DXD*NXTND                                               DAM 0405
      DELT = SQRT(DELT*DELT + DXD*DXD)                                  DAM 0406
      Z1 = (DR1-RO)/DELT1                                               DAM 0407
      Z1 = Z1*Z1/2.                                                     DAM 0408
      Z = (DR-RO)/DELT                                                  DAM 0409
      Z = Z**2/2.                                                       DAM 0410
      DAMG(LL,J) = (A*Y*DMG/DELT)*EXP(-Z) + DAMG(LL,J)                  DAM 0411
      IF(NTYPE.EQ.2) DAMG(LL,J) = DAMG(LL,J) + (A*Y*DMG1/DELT1)*EXP(-Z1)DAM 0412
  190 CONTINUE                                                          DAM 0413
      GO TO 200                                                         DAM 0414
  830 CONTINUE                                                          DAM 0415
      Z = Z**2/2.                                                       DAM 0416
      Z1 = Z1*Z1/2.                                                     DAM 0417
      DAMG(LL,J) = (A*Y*DMG/DELT)*EXP(-Z) + DAMG(LL,J)                  DAM 0418
      IF(NTYPE.EQ.2) DAMG(LL,J) = DAMG(LL,J) + (A*Y*DMG1/DELT1)*EXP(-Z1)DAM 0419
  200 CONTINUE                                                          DAM 0420
      Y = DELT1/20.                                                     DAM 0421
      LM = Y                                                            DAM 0422
      IF(LM)150,150,85                                                  DAM 0423
   85 CONTINUE                                                          DAM 0424
      IF(MM-NSTP-1)89,88,89                                             DAM 0425
   88 CONTINUE                                                          DAM 0426
      MM = MM-1                                                         DAM 0427
   89 CONTINUE                                                          DAM 0428
      Y = R(I,MM)-R(I,MM+1)                                             DAM 0429
      Y = Y/10.                                                         DAM 0430
      Y = ABS(Y) + 1.                                                   DAM 0431
      LN = Y                                                            DAM 0432
      IF(LN-LM)86,87,87                                                 DAM 0433
   86 CONTINUE                                                          DAM 0434
```

```
      KX = LN                                                           DAM 0435
      GO TO 150                                                         DAM 0436
   87 CONTINUE                                                          DAM 0437
      KX = LM                                                           DAM 0438
      GO TO 150                                                         DAM 0439
  140 CONTINUE                                                          DAM 0440
      KX = KXKK                                                         DAM 0441
  150 CONTINUE                                                          DAM 0442
      DO 130 J = 1,101                                                  DAM 0443
      RO = (J-1)*LEN(LL)                                                DAM 0444
      X = ((RO-RP(I,1))/DELTA(I,1))**2/2.                               DAM 0445
      DIST(LL,J) = EXP(-X)                                              DAM 0446
  130 CONTINUE                                                          DAM 0447
      DAMG(LL,1) = 3.*(DAMG(LL,2)-DAMG(LL,3))+ DAMG(LL,4)               DAM 0448
      Y = LEN(LL)/2.                                                    DAM 0449
      DO 270 J = 1,100                                                  DAM 0450
      LM = 101-J                                                        DAM 0451
      LN = LM+1                                                         DAM 0452
      DINT(LL,LM) = Y*(DAMG(LL,LM)+DAMG(LL,LN)) + DINT(LL,LN)           DAM 0453
  270 CONTINUE                                                          DAM 0454
      X = NXTND                                                         DAM 0455
      G(LL) = (DINT(LL,1)/1000.)/X                                      DAM 0456
      INDEX(LL) = I                                                     DAM 0457
      DO 280 J = 2,100                                                  DAM 0458
      DINT(LL,J) = DINT(LL,J)/DINT(LL,1)                                DAM 0459
  280 CONTINUE                                                          DAM 0460
      XI01(LL) = XI01(LL)/XI00(LL)                                      DAM 0461
      XI02(LL) = XI02(LL)/XI00(LL)                                      DAM 0462
      XI03(LL) = XI03(LL)/XI00(LL)                                      DAM 0463
      XI20(LL) = XI20(LL)/XI00(LL)                                      DAM 0464
      XI21(LL) = XI21(LL)/XI00(LL)                                      DAM 0465
      XI30(LL) = XI30(LL)/XI00(LL)                                      DAM 0466
      XI00(LL) = XI00(LL)/1000.                                         DAM 0467
      F(LL) = XI00(LL)/E(I)                                             DAM 0468
      DINT(LL,1) = 1.                                                   DAM 0469
      L = 1                                                             DAM 0470
      DO 290 J = 1,100                                                  DAM 0471
      KRC = DAMG(LL,J)/STP                                              DAM 0472
      JRC = DAMG(LL,J+1)/STP                                            DAM 0473
 2805 CONTINUE                                                          DAM 0474
      IF(KRC-JRC)281,290,282                                            DAM 0475
  281 CONTINUE                                                          DAM 0476
      RC(LL,L,1) = JRC*STP                                              DAM 0477
      JRC = JRC-1                                                       DAM 0478
      GO TO 283                                                         DAM 0479
  282 CONTINUE                                                          DAM 0480
      RC(LL,L,1) = KRC*STP                                              DAM 0481
      KRC = KRC-1                                                       DAM 0482
  283 CONTINUE                                                          DAM 0483
      X = DAMG(LL,J)-RC(LL,L,1)                                         DAM 0484
      X = X/(DAMG(LL,J)-DAMG(LL,J+1))                                   DAM 0485
      Y = (J-1)*LEN(LL)                                                 DAM 0486
      Z = J*LEN(LL)                                                     DAM 0487
      RC(LL,L,2) = X*Z + (1.-X)*Y                                       DAM 0488
      L = L+1                                                           DAM 0489
      GO TO 2805                                                        DAM 0490
  290 CONTINUE                                                          DAM 0491
      LL = LL+1                                                         DAM 0492
  300 CONTINUE                                                          DAM 0493
      KK = 1                                                            DAM 0494
      KKK = 3                                                           DAM 0495
      X = NXTND                                                         DAM 0496
```

```
310 CONTINUE                                                            DAM 0497
    IF(NDAM-KK)420,320,320                                              DAM 0498
320 CONTINUE                                                            DAM 0499
    IF(NDAM-KKK)330,340,340                                             DAM 0500
330 CONTINUE                                                            DAM 0501
    KKK = NDAM                                                          DAM 0502
340 CONTINUE                                                            DAM 0503
    PRINT 4                                                             DAM 0504
    PRINT 3                                                             DAM 0505
    PRINT 7                                                             DAM 0506
    PRINT 10                                                            DAM 0507
    DO 350 I = KK,KKK                                                   DAM 0508
    J = INDEX(I)                                                        DAM 0509
    TEST(I) = E(J)                                                      DAM 0510
350 CONTINUE                                                            DAM 0511
    PRINT 11,(TEST(J),J=KK,KKK)                                         DAM 0512
    DO 360 I = KK,KKK                                                   DAM 0513
    J = INDEX(I)                                                        DAM 0514
    TEST(I) = RP(J,1)/X                                                 DAM 0515
360 CONTINUE                                                            DAM 0516
    PRINT 12,(TEST(J),J=KK,KKK)                                         DAM 0517
    DO 370 I = KK,KKK                                                   DAM 0518
    J = INDEX(I)                                                        DAM 0519
    TEST(I) = DELTA(J,1)/X                                              DAM 0520
370 CONTINUE                                                            DAM 0521
    PRINT 13,(TEST(J),J=KK,KKK)                                         DAM 0522
    DO 380 I = KK,KKK                                                   DAM 0523
    J = INDEX(I)                                                        DAM 0524
    TEST(I) = DP(J,1)/X                                                 DAM 0525
380 CONTINUE                                                            DAM 0526
    PRINT 14,(TEST(J),J=KK,KKK)                                         DAM 0527
    PRINT 15,(G(J),J = KK,KKK)                                          DAM 0528
    PRINT 16,(F(J),J = KK,KKK)                                          DAM 0529
    PRINT 17                                                            DAM 0530
    DO 400 J = 1,101                                                    DAM 0531
    LM = 2*(J-1)                                                        DAM 0532
    PRINT 9,LM,(DIST(K,J),DINT(K,J),DAMG(K,J),K = KK,KKK)               DAM 0533
400 CONTINUE                                                            DAM 0534
    KK = KK+3                                                           DAM 0535
    KKK = KKK+3                                                         DAM 0536
    GO TO 310                                                           DAM 0537
420 CONTINUE                                                            DAM 0538
    L = 1                                                               DAM 0539
430 CONTINUE                                                            DAM 0540
    LL = L+4                                                            DAM 0541
    IF(NDAM-LL)435,436,436                                              DAM 0542
435 CONTINUE                                                            DAM 0543
    LL = NDAM                                                           DAM 0544
436 CONTINUE                                                            DAM 0545
    PRINT 4                                                             DAM 0546
    PRINT 3                                                             DAM 0547
    PRINT 7                                                             DAM 0548
    PRINT 18                                                            DAM 0549
    J = INDEX(2) - INDEX(1)                                             DAM 0550
    PRINT 19,(E(I*J),I=L,LL)                                            DAM 0551
    PRINT 5                                                             DAM 0552
    Y = NXTND                                                           DAM 0553
    DO 438 J = 1,250                                                    DAM 0554
    X = 0.                                                              DAM 0555
    DO 4365 I = L,LL                                                    DAM 0556
    X = X + RC(I,J,1)                                                   DAM 0557
    RC(I,J,2) = RC(I,J,2)/Y                                             DAM 0558
```

```
4365 CONTINUE                                                           DAM 0559
     IF(X)438,438,437                                                   DAM 0560
 437 CONTINUE                                                           DAM 0561
     PRINT 20,(RC(I,J,1),RC(I,J,2),I=L,LL)                              DAM 0562
 438 CONTINUE                                                           DAM 0563
     L = L+5                                                            DAM 0564
     IF(NDAM-L)440,430,430                                              DAM 0565
 440 CONTINUE                                                           DAM 0566
     PRINT 4                                                            DAM 0567
     PRINT 3                                                            DAM 0568
     PRINT 7                                                            DAM 0569
     PRINT 23                                                           DAM 0570
     J = INDEX(2) - INDEX(1)                                            DAM 0571
     DO 450 I = 1,NDAM                                                  DAM 0572
     L = E(I*J)                                                         DAM 0573
     XMOM1(I) = XI01(I)                                                 DAM 0574
     XMOM2(I) = XI20(I)+XI02(I)-XI01(I)*XI01(I)                         DAM 0575
     XMOM3(I)=3.*(XI21(I)-XI01(I)*(XI20(I)+XI02(I)))+2.*XI01(I)*XI01(I)DAM 0576
    1*XI01(I) + XI03(I)                                                 DAM 0577
     PRINT 24,L,XMOM1(I),XMOM2(I),F(I),XI00(I)                          DAM 0578
     PRINT 26,G(I)                                                      DAM 0579
 450 CONTINUE                                                           DAM 0580
     WRITE TAPE30,ITER,(E(I),RP(I,1),DELTA(I,1),DP(I,1),R(I,1),F(I),G(IDAM 0581
    1),XI00(I),XI01(I),XI02(I),XI03(I),XI20(I),XI21(I),XI30(I),XMOM1(I)DAM 0582
    2,XMOM2(I),XMOM3(I),(DAMG(I,J),J=1,101),I=1,NDAM)                   DAM 0583
     PRINT 4                                                            DAM 0584
     PRINT 3                                                            DAM 0585
     PRINT 7                                                            DAM 0586
     PRINT 25,(I,F(I),XMOM3(I),XI30(I),I=1,NDAM)                        DAM 0587
 455 CONTINUE                                                           DAM 0588
     READ 6,NRPT                                                        DAM 0589
     IF(NRPT)460,460,30                                                 DAM 0590
 460 CONTINUE                                                           DAM 0591
     ENDFILE 30                                                         DAM 0592
     REWIND 30                                                          DAM 0593
     END                                                                DAM 0594
```

VI. LISTINGS OF SUBROUTINES

All subroutines used by COREL, RASE4 and DAMG2 are listed here in alphabetical order.

```
      SUBROUTINE COPHI(CPHI,T,J,LJ)                                     COP 0001
      DIMENSION E(1600)                                                 COP 0002
      DIMENSION P2(4)                                                   COP 0003
      COMMON/PHIVAR/E,P1,P2                                             COP 0004
      X = P1-P2(LJ)                                                     COP 0005
      IF(X)20,10,20                                                     COP 0006
   10 CONTINUE                                                          COP 0007
      X = 1. - T/E(J)                                                   COP 0008
      CPHI = SQRT(X)                                                    COP 0009
      RETURN                                                            COP 0010
   20 CONTINUE                                                          COP 0011
      X = 1. - T/E(J)                                                   COP 0012
      CPHI = (1.-(1.+P2(LJ)/P1)*T/(2.*E(J)))/SQRT(X)                    COP 0013
      RETURN                                                            COP 0014
      END                                                               COP 0015
```

```
      SUBROUTINE DAMG(Z,T,ALF,I)                                        DMG 0001
      COMMON/DFIR/NCOMP,NDAM,DE,PART(4,101)                             DMG 0002
 1000 FORMAT(F6.1,I4,I5)                                                DMG 0003
 1001 FORMAT(10F8.5)                                                    DMG 0004
      IF(NCOMP.GT.1)GO TO 10                                            DMG 0005
      IF(T)1,1,2                                                        DMG 0006
    1 CONTINUE                                                          DMG 0007
      Z = T                                                             DMG 0008
      RETURN                                                            DMG 0009
    2 CONTINUE                                                          DMG 0010
      G = SQRT(T)                                                       DMG 0011
      G = SQRT(G)                                                       DMG 0012
      G = G**3                                                          DMG 0013
      G = 0.40244*G + T                                                 DMG 0014
      X = 1./6.                                                         DMG 0015
      G = G + 3.4008*T**X                                               DMG 0016
      G = ALF*G + 1.                                                    DMG 0017
      Z = T/G                                                           DMG 0018
      RETURN                                                            DMG 0019
   10 CONTINUE                                                          DMG 0020
      IF(NDAM.GT.1) GO TO 30                                            DMG 0021
      IF(NDAM.EQ.1)READ 1000,DE,NDAM                                    DMG 0022
      IF(NDAM.EQ.0)GO TO 1                                              DMG 0023
      NDAM = NDAM + 1                                                   DMG 0024
      DO 20 J = 1,NCOMP                                                 DMG 0025
      READ 1001,(PART(J,K),K=2,NDAM)                                    DMG 0026
      PART(J,1)=1.                                                      DMG 0027
   20 CONTINUE                                                          DMG 0028
      RETURN                                                            DMG 0029
   30 CONTINUE                                                          DMG 0030
      Z = T/DE                                                          DMG 0031
      N = Z                                                             DMG 0032
      Z = Z-N                                                           DMG 0033
      N = N+1                                                           DMG 0034
      M = N+1                                                           DMG 0035
      Z = Z*PART(I,M) + (1.-Z)*PART(I,N)                                DMG 0036
      Z = Z*T                                                           DMG 0037
      RETURN                                                            DMG 0038
      END                                                               DMG 0039
```

```
      SUBROUTINE FET(Y,X)                                               FET 0001
    1 FORMAT(* ERROR IN SUBROUTINE FET.   MM = *,I5)                    FET 0002
      DIMENSION F(50),EF(50),SLF(50)                                    FET 0003
      COMMON/FVAR/F,SLF,EF,NF,MM                                        FET 0004
      N = 0                                                             FET 0005
      MM = 0                                                            FET 0006
      DO 10 I = 1,NF                                                    FET 0007
      Y = (X-EF(I))                                                     FET 0008
      IF(Y)10,5,5                                                       FET 0009
    5 CONTINUE                                                          FET 0010
      N = I                                                             FET 0011
   10 CONTINUE                                                          FET 0012
      NN = 10                                                           FET 0013
      IF(N)40,20,15                                                     FET 0014
   15 CONTINUE                                                          FET 0015
      MM = 15                                                           FET 0016
      IF(N-NF)16,18,40                                                  FET 0017
   16 CONTINUE                                                          FET 0018
      Z = X/EF(N)                                                       FET 0019
      Z = ALOG(Z)                                                       FET 0020
      Y = F(N) + SLF(N)*Z                                               FET 0021
      GO TO 50                                                          FET 0022
   18 CONTINUE                                                          FET 0023
      Y = 1./(2.*X)                                                     FET 0024
      RETURN                                                            FET 0025
   20 CONTINUE                                                          FET 0026
      Z = .35                                                           FET 0027
      Y = 1.43*(X**Z)                                                   FET 0028
      GO TO 50                                                          FET 0029
   40 CONTINUE                                                          FET 0030
      PRINT 1, MM                                                       FET 0031
   50 CONTINUE                                                          FET 0032
      RETURN                                                            FET 0033
      END                                                               FET 0034
```

```
      SUBROUTINE FSET                                                   FSE 0001
      DIMENSION F(50),SLF(50),EF(50)                                    FSE 0002
      COMMON/FVAR/F,SLF,EF,NF,MM                                        FSE 0003
      NF = 13                                                           FSE 0004
      SLF(1)  =  .06781                                                 FSE 0005
      SLF(2)  =  .07749                                                 FSE 0006
      SLF(3)  =  .07791                                                 FSE 0007
      SLF(4)  =  .07069                                                 FSE 0008
      SLF(5)  =  .05239                                                 FSE 0009
      SLF(6)  =  .00987                                                 FSE 0010
      SLF(7)  = -.02433                                                 FSE 0011
      SLF(8)  = -.06204                                                 FSE 0012
      SLF(9)  = -.12005                                                 FSE 0013
      SLF(10) = -.13129                                                 FSE 0014
      SLF(11) = -.11109                                                 FSE 0015
      SLF(12) = -.06221                                                 FSE 0016
      EF(1) = .002                                                      FSE 0017
      EF(2) = .004                                                      FSE 0018
      EF(3) = .010                                                      FSE 0019
      EF(4) = .020                                                      FSE 0020
      EF(5) = .040                                                      FSE 0021
      EF(6) = .100                                                      FSE 0022
```

```
      EF(7) = .150                                                      FSE 0023
      EF(8) = .200                                                      FSE 0024
      EF(9) = .400                                                      FSE 0025
      EF(10) = 1.000                                                    FSE 0026
      EF(11) = 2.000                                                    FSE 0027
      EF(12) = 4.000                                                    FSE 0028
      EF(13) = 10.000                                                   FSE 0029
      F(1) = .162                                                       FSE 0030
      F(2) = .209                                                       FSE 0031
      F(3) = .260                                                       FSE 0032
      F(4) = .334                                                       FSE 0033
      F(5) = .383                                                       FSE 0034
      F(6) = .431                                                       FSE 0035
      F(7) = .435                                                       FSE 0036
      F(8) = .428                                                       FSE 0037
      F(9) = .385                                                       FSE 0038
      F(10) = .275                                                      FSE 0039
      F(11) = .184                                                      FSE 0040
      F(12) = .107                                                      FSE 0041
      F(13) = .050                                                      FSE 0042
      RETURN                                                            FSE 0043
      END                                                               FSE 0044
```

```
      SUBROUTINE  RANGER(E,LJ,CPHI,I,X,DELT)                            RGR 0001
      DIMENSION R(4,1600),DR(4,1600),DP(4,1600)                         RGR 0002
      COMMON/DSEC/R,DR,DP,NE,NDIV,NMULT,NCOMP,EN                        RGR 0003
      IF(I.EQ.1) GO TO 50                                               RGR 0004
      Z = E/EN                                                          RGR 0005
      N = Z                                                             RGR 0006
      Z = Z-N                                                           RGR 0007
      Y = 1.-Z                                                          RGR 0008
      N = N+1                                                           RGR 0009
      M = N+1                                                           RGR 0010
      X = (Y*R(LJ,N)+Z*R(LJ,M))*CPHI                                    RGR 0011
      DELT = (Y*DR(LJ,N)+Z*DR(LJ,M))*CPHI                               RGR 0012
      Z = (Y*DP(LJ,N)+Z*DP(LJ,M))                                       RGR 0013
      DELT = DELT*DELT + Z*Z*(1.-CPHI*CPHI)/2. + X*X                    RGR 0014
      RETURN                                                            RGR 0015
   50 CONTINUE                                                          RGR 0016
      NCOMP = 1                                                         RGR 0017
      DO 60 J = 1,NCOMP                                                 RGR 0018
      READ TAPE10,NE,NDIV,NMULT,NCOMP,(R(J,K),DR(J,K),DP(J,K),K=1,NE)   RGR 0019
      IF(NE.EQ.0) RETURN                                                RGR 0020
      READ(10)                                                          RGR 0021
   60 CONTINUE                                                          RGR 0022
      EN = NMULT                                                        RGR 0023
      EN = EN/NDIV                                                      RGR 0024
      RETURN                                                            RGR 0025
      END                                                               RGR 0026
```

```
      SUBROUTINE SECALC(Z1,Z2,P1,E,SE,Z,A,EN)                           SEC 0001
C     Z1,Z2,ARE THE ATOMIC NUMBERS OF THE INCIDENT ION AND TARGET ATOMS,SEC 0002
C     RESPECTIVELY. THE INCIDENT ION MASS IN AMU IS P1, WHILE E IS THE  SEC 0003
C     INCIDENT ENERGY IN KEV.                                           SEC 0004
C     Z,A,EN ARE , RESPECTIVELY, THE PARAMETERS Z,A,N IN THE BRICE FORMUSEC 0005
C     SE IS THE INELASTIC STOPPING CROSS SECTION IN KEV-CM2/ATOM        SEC 0006
      B = 1.21918E-18                                                   SEC 0007
      X = E/(100.*P1)                                                   SEC 0008
      Y = X/(Z**2)                                                      SEC 0009
      SQ = SQRT(Y)                                                      SEC 0010
      AT = ATAN(SQ)                                                     SEC 0011
      SE = (((30.*Y+83.)*Y+74.)*Y+21.)                                  SEC 0012
      SE = SE/(3.*(1.+Y)**3)                                            SEC 0013
      SE = SQ*SE +AT*(10.*Y+1.)                                         SEC 0014
      SE = B*SE                                                         SEC 0015
      SE = SE*(Z1+Z2)/2.                                                SEC 0016
      SQ = 2.*SQRT(X)*A                                                 SEC 0017
      Y = 1.+SQ**EN                                                     SEC 0018
      SE = SE/Y                                                         SEC 0019
      RETURN                                                            SEC 0020
      END                                                               SEC 0021
```

```
      SUBROUTINE TEN                                                    TEN 0001
      COMMON/PHIVAR/E,P1,P2                                             TEN 0002
      COMMON/TENVAR/NE,NDIV,NMULT,ALAT,Z2                               TEN 0003
      DIMENSION E(1600),P2(4),Z2(4),N(4),P(33)                          TEN 0004
    1 FORMAT(5/,5(1X,*RECOIL DATA NOT AVAILABLE FOR THIS TARGET*,/),    TEN 0005
     12/,1X,*CALCULATION PROCEEDING WITH A THEORETICAL ESTIMATE FOR THISTEN 0006
     2 INFORMATION*,2/)                                                 TEN 0007
      EP = NE*NMULT                                                     TEN 0008
      EP = EP/NDIV                                                      TEN 0009
      PP = 1.                                                           TEN 0010
      DO 5 J = 1,4                                                      TEN 0011
      P(J) = (P1-P2(J))/(P1+P2(J))                                      TEN 0012
      P(J)=P(J)*P(J)                                                    TEN 0013
      IF(P(J).LT.PP)PP=P(J)                                             TEN 0014
    5 CONTINUE                                                          TEN 0015
      EP=EP*(1.-PP)                                                     TEN 0016
   10 CONTINUE                                                          TEN 0017
      READ TAPE10,(N(J),J=1,4),(P(K),K=1,33)                            TEN 0018
      IF(N(1).EQ.0) GO TO 100                                           TEN 0019
      ER = N(1)*N(3)                                                    TEN 0020
      ER = ER/N(2)                                                      TEN 0021
      IF(ER.GE.EP)30,20                                                 TEN 0022
   20 CONTINUE                                                          TEN 0023
      READ(10)                                                          TEN 0024
      GO TO 10                                                          TEN 0025
   30 CONTINUE                                                          TEN 0026
      IF(ALAT.NE.P(1))GO TO 20                                          TEN 0027
      IF(P2(1).NE.P(2))GO TO 20                                         TEN 0028
      IF(Z2(1).NE.P(3))GO TO 20                                         TEN 0029
      IF(P2(2).NE.P(10))GO TO 20                                        TEN 0030
      IF(Z2(2).NE.P(11))GO TO 20                                        TEN 0031
      IF(P2(3).NE.P(18))GO TO 20                                        TEN 0032
      IF(Z2(3).NE.P(19))GO TO 20                                        TEN 0033
      IF(P2(4).NE.P(26))GO TO 20                                        TEN 0034
```

```
      IF(Z2(4).NE.P(27))GO TO 20                                        TEN 0035
      RETURN                                                            TEN 0036
  100 CONTINUE                                                          TEN 0037
      PRINT 1                                                           TEN 0038
      RETURN                                                            TEN 0039
      END                                                               TEN 0040
```

```
      SUBROUTINE TWENTY                                                 TWE 0001
      DIMENSION N(15),P(27)                                             TWE 0002
      COMMON/TWENVAR/P2(4),Z2(4),P1,Z1,NTYPE,EM,ITER,ALAT               TWE 0003
    1 FORMAT(5/,1X,*RANGE AND STRAGGLING DATA NOT AVAILABLE ON TAPE20*,/TWE 0004
     1,1X,*CALCULATION ABORTED------PROCEEDING TO NEXT SET OF INPUT DATATWE 0005
     2*,/,1H1)                                                          TWE 0006
   10 CONTINUE                                                          TWE 0007
      READ TAPE20,(N(J),J=1,15),(P(K),K=1,27)                           TWE 0008
      IF(EOF,20)22,15                                                   TWE 0009
   15 CONTINUE                                                          TWE 0010
      E = N(1)*N(4)                                                     TWE 0011
      E = E/N(3)                                                        TWE 0012
      IF(E.GT.EM/1.01)16,25                                             TWE 0013
   16 CONTINUE                                                          TWE 0014
      IF(E.LT.1.01*EM)30,25                                             TWE 0015
   22 CONTINUE                                                          TWE 0016
      PRINT 1                                                           TWE 0017
      ITER=0                                                            TWE 0018
      REWIND 20                                                         TWE 0019
      RETURN                                                            TWE 0020
   25 CONTINUE                                                          TWE 0021
      IF(N(7).GT.0) READ(20)                                            TWE 0022
      READ(20)                                                          TWE 0023
      READ(20)                                                          TWE 0024
      GO TO 10                                                          TWE 0025
   30 CONTINUE                                                          TWE 0026
      IF(ALAT.NE.P(1))GO TO 25                                          TWE 0027
      IF(P1.NE.P(2)) GO TO 25                                           TWE 0028
      IF(Z1.NE.P(3)) GO TO 25                                           TWE 0029
      IF(P2(1).NE.P(4)) GO TO 25                                        TWE 0030
      IF(Z2(1).NE.P(5)) GO TO 25                                        TWE 0031
      IF(P2(2).NE.P(10)) GO TO 25                                       TWE 0032
      IF(Z2(2).NE.P(11)) GO TO 25                                       TWE 0033
      IF(P2(3).NE.P(16)) GO TO 25                                       TWE 0034
      IF(Z2(3).NE.P(17)) GO TO 25                                       TWE 0035
      IF(P2(4).NE.P(22)) GO TO 25                                       TWE 0036
      IF(Z2(4).NE.P(23)) GO TO 25                                       TWE 0037
      IF(N(7).LT.0) ITER=-N(7)                                          TWE 0038
      IF(N(7).GT.0) ITER=N(7)                                           TWE 0039
      RETURN                                                            TWE 0040
      END                                                               TWE 0041
```

```
      SUBROUTINE WIND10                                                 W10 0001
      DIMENSION P(37)                                                   W10 0002
   10 CONTINUE                                                          W10 0003
      READ TAPE10,(P(J),J=1,37)                                         W10 0004
      IF(EOF,10)30,20                                                   W10 0005
   20 CONTINUE                                                          W10 0006
      READ(10)                                                          W10 0007
      GO TO 10                                                          W10 0008
   30 CONTINUE                                                          W10 0009
      BACKSPACE 10                                                      W10 0010
      BACKSPACE 10                                                      W10 0011
      BACKSPACE 10                                                      W10 0012
      RETURN                                                            W10 0013
      END                                                               W10 0014
```

```
      SUBROUTINE WIND20                                                 W20 0001
      DIMENSION N(15),P(27)                                             W20 0002
   10 CONTINUE                                                          W20 0003
      READ TAPE20,(N(J),J=1,15),(P(K),K=1,27)                           W20 0004
      IF(EOF,20)20,30                                                   W20 0005
   20 CONTINUE                                                          W20 0006
      BACKSPACE 20                                                      W20 0007
      RETURN                                                            W20 0008
   30 CONTINUE                                                          W20 0009
      IF(N(7).LT.0) GO TO 20                                            W20 0010
      READ(20)                                                          W20 0011
      IF(EOF,20)42,31                                                   W20 0012
   31 CONTINUE                                                          W20 0013
      READ(20)                                                          W20 0014
      IF(EOF,20)41,32                                                   W20 0015
   32 CONTINUE                                                          W20 0016
      READ(20)                                                          W20 0017
      IF(EOF,20)40,10                                                   W20 0018
   40 CONTINUE                                                          W20 0019
      BACKSPACE 20                                                      W20 0020
   41 CONTINUE                                                          W20 0021
      BACKSPACE 20                                                      W20 0022
   42 CONTINUE                                                          W20 0023
      BACKSPACE 20                                                      W20 0024
      GO TO 20                                                          W20 0025
      END                                                               W20 0026
```

```
      SUBROUTINE WIND30                                                 W30 0001
      DIMENSION N(11),P(11)                                             W30 0002
   10 CONTINUE                                                          W30 0003
      READ TAPE30,(N(J),J=1,11),(P(K),K=1,11)                           W30 0004
      IF(EOF,30)30,20                                                   W30 0005
   20 CONTINUE                                                          W30 0006
      IF(N(11).GT.1)GO TO 30                                            W30 0007
      READ(30)                                                          W30 0008
      IF(EOF,30)25,10                                                   W30 0009
   25 CONTINUE                                                          W30 0010
      BACKSPACE 30                                                      W30 0011
   30 CONTINUE                                                          W30 0012
      BACKSPACE 30                                                      W30 0013
      RETURN                                                            W30 0014
      END                                                               W30 0015
```

REFERENCES

1. D. K. Brice, Sandia Laboratories Research Report, SLA73-0416, April 1973.
2. D. K. Brice, J. Appl. Phys. 46, 3385 (1975).
3. D. K. Brice, Rad. Effects 11, 227 (1971); 6, 77 (1970).
4. J. Lindhard, M. Scharff and H. E. Schiott, K. Danske. Videnskab. Selskab Mt.-Fys. Medd. 33, No. 14 (1963).
5. O. B. Firsov, Zh. Eksperim. i Teor. Fiz. 36, 1517 (1959). (Translated as Sov. Phys. JETP 36, 1076 (1959).
6. D. K. Brice, Phys. Rev. A 6, 1791 (1972).

Distribution:

Dr. J. C. Bourgoin
GPS, Ecole Normale Superieure
Fac. Sciences Tour 23
9 Quai Saint Bernard
Paris-(5)
75 France

Dr. Seijiro Furukawa
Tokyo Institute of Technology
Department of Physical Electronics
O-Okayama, Meguro-Ku, Tokyo
Japan

Dr. Varga Gabor
H-1052 Mudapest V.
Becsi u.1
Hungary

Dr. J. F. Gibbons
Electrical Engineering Department
Stanford University
Stanford, CA 94305

Dr. Thomas O. Hunter
Nuclear Engineering Department
153 Engineering Research Building
1500 Johnson Drive
Madison, WI 53706

Dr. G. L. Kulcinski
Nuclear Engineering Department
153 Engineering Research Bldg.
1500 Johnson Drive
Madison, WI 53706

Dr. Young-Hoon Lee
Department of Physics
State University of New York
at Albany
Albany, NY 12222

Dr. G. Linker
Kernforschungszentrum Karlsruhe
Postfach 3640
75 Karlsruhe
Germany

Dr. I. Manning
Naval Research Laboratory
Washington, DC 20375

Dr. Jos Odeurs
Katolieke Universiteit Leuven
Department Natuurkunde
Institut Voor Kern-En
Stralingsfysika
Celestijnenlaan 200 D
3030 Heverlee
Belgium

Dr. O. S. Oen
Oak Ridge National Laboratory
Solid State Division
P. O. Box X
Oak Ridge, TN 37830

Dr. D. M. Parkin
Mail Stop 734
Los Alamos Scientific
Laboratory
Los Alamos, NM 87544

Dr. J. Rosolowski
Building K-1, Rm 3C20
General Electric Research and
Development Center
P. O. Box 8
Schenectady, NY 12301

Prof. B. S. H. Royce
Solid State and Materials
Laboratory
The Engineering Quadrangle
Princeton University
Princeton, NJ 08540

Dr. P. Sigmund
Physics Laboratory II
H. C. Ørsted Institute
University of Copenhagen
Universitetsparken 5
2100 Copenhagen
Denmark

Dr. D. A. Thompson
Department of Engineering
Physics
McMaster University
Hamilton, Ontario
Canada

Mr. Frank Vozzo, c/o D. Kaplan
Physics Department
State University of New York
at Albany
1400 Washington Avenue
Albany, NY 12222

Dr. K. B. Winterbon
Theoretical Physics Division
Chalk River Nuclear Laboratories
Atomic Energy of Canada Limited
Chalk River, Ontario
Canada

Dr. E. A. Wolicki
Naval Research Laboratory
Washington, DC 20375

2100 W. J. Spencer
2140 B. L. Gregory
2300 L. D. Smith
2350 J. C. Crawford
2352 G. W. McClure
2353 W. G. Perkins
2353 D. F. Cowgill
5000 A. Narath
Attn: 5200 E. H. Beckner
5400 A. W. Snyder
5700 J. H. Scott
5800 R. S. Claassen
5100 J. K. Galt
Attn: 5120 G. J. Simmons
5130 G. A. Samara
5150 J. E. Schirber
5160 W. Herrmann
5110 F. L. Vook
5111 S. T. Picraux
5111 D. K. Brice (30)
5112 P. S. Peercy
5112 G. Krefft
5114 J. E. Houston
8266 E. A. Aas (2)
8340 J. L. Wirth
8341 W. D. Wilson
8341 L. G. Haggmark
8347 W. Bauer
3141 C. A. Pepmueller (Actg.) (5)
3151 W. L. Garner (3)
For: ERDA/TIC (Unlimited Release)
ERDA/TIC (25)
(R. P. Campbell 3171-1)