ISSN 0915-6364

NATIONAL INSTITUTE FOR FUSION SCIENCE

# ELECTRAN - Monte Carlo Program of Secondary Electron Emission from Monoatomic Solids under the Impact of 0.1 - 10 keV Electrons

K. Ohya, A. Chen, J. Kawata, K. Nishimura,
D. Kato, T. Tanabe and T. Kato

(Received - Feb.18, 2004 )

NIFS-DATA-84 Mar. 2004

This report was prepared as a preprint of work performed as a collaboration research of the National Institute for Fusion Science (NIFS) of Japan. The views presented here are solely those of the authors. This document is intended for information only and may be published in a journal after some rearrangement of its contents in the future.

TOKI, JAPAN

# ELECTRAN – Monte Carlo Program of Secondary Electron Emission from Monoatomic Solids under the Impact of 0.1 – 10 keV Electrons

Kaoru Ohya [1], Allen Chen [2] , Jun Kawata [3], Kenji Nishimura [4], Daiji Kato [5],
Tetsuo Tanabe [2] and Takako Kato [5]

[1] *Faculty of Engineering, The University of Tokushima, Tokushima 770-7506, Japan*
[2] *Department of Nuclear Engineering, Nagoya University, Nagoya 464-8603, Japan*
[3] *Takuma National College of Technology, Takuma 769-1192, Japan*
[4] *Numazu College of Technology, Numazu 410-8501, Japan*
[5] *National Institute for Fusion Science, Toki 509-5292, Japan*

## Abstract

We present the ELECTRAN (ELECtron TRANsport) computer code, which treats the interaction of 0.1 – 10 keV electrons with monoatomic solids. The program calculates the energy and angular distributions of backscattered primaries and emitted secondary electrons from the solids, as well as the total yields. After describing the theoretical background of electron-solid interactions, source listings of a FORTRAN program of the ELECTRAN are given along with short comments for the lines. Finally, some calculation results are shown as an example.

**Keywords**
ELECTRAN, secondary electron emission, electron backscattering, Monte Carlo simulation

## Contents

## 1. Introduction

When fast charged particles penetrate into a solid, they produce kinetic electrons. If these electrons escape from the solid, they give rise to an outgoing electron current which can be measured and are called "Secondary Electrons (SE)". When the incident particles are also electrons this phenomena is called "Secondary Electron Emission (SEE)"; for incident ions it is called "Ion-Induced Electron Emission (IIEE)". Some of the incident particles are re-emitted immediately after impact or after migrating in the solid; this is called "backscattering" or "reflection". Such incident electrons are called "backscattered electrons (BSE)". All these electrons are of acute importance in such fields as radiation effects and transport phenomena in solids (e.g., radiation biology), plasma-surface interactions, microtechnology, surface analysis, microscopy, and particle detector development, etc. Refer to recent books on developments in the field of SEE and IIEE in experimental and theoretical researches [e.g., 1-3].

Computer simulations have been used recently in studies of ion implantation, radiation damage, sputtering, and the reflection and transmission of energetic particles [4]. The Monte Carlo (MC) method applied in these simulations has a number of distinct advantages over analytical formulations based on transport theory. It allows more rigorous treatment of elastic scattering, explicit consideration of surfaces and interfaces, and easy determination of energy and angular distributions.

In the 1970's, the direct simulation MC model based on differential cross-sections for elastic and inelastic collisions of particles in a medium was applied to electron backscattering and secondary electron emission [5]. Since then, the direct MC model has been further developed by several groups [6-11]. The MC computer program [12] presented here has been named ELECTRAN (ELECtron TRANsport). The program is applicable to monoatomic metals, semiconductors and insulators. It provides information on energy and angular distributions of BSEs and SEs, as well as total yields. In Sec. 2, we give a brief overview on the electron interaction models in ELECTRAN with the theoretical background for SEE. Section 3 presents the FORTRAN computer program along with a short explanation of the lines. Finally, some calculation results are shown in Sec. 4.

No claim is made that the programs given here represent the best or even the only way to do SEE simulations. There are a large number of other approaches, partly cited in the text. This report will have achieved its purpose if you –the user—feel ready, willing, and able to use the programs as the basis for your own experimentation and development.

# 2. Computational procedure

## 2.1 Electron transport

A primary beam of electrons impinging on a solid target suffers inelastic and elastic collisions with all components of the solid. Both types of interactions play an important role in the SEE process. The inelastic collisions excite the electrons in the solid, giving rise to SEs. The motion of a primary electron and SEs is treated in the same way. The transport of electrons in the solid is very sensitive to the energy and angular dependence of the collisional cross-sections. Hence a realistic description of the interaction processes is required to model SEE.

### *2.1.1 Elastic collision*

Elastic scattering is caused by the interactions of electrons with the real potential *V*(r) surrounding each ionic core. Various approximations for the potential and different techniques to evaluate the elastic cross section have been proposed. Due to its simplicity, the first-order Born approximation (FBA) is often used to calculate the elastic cross section. The screened Rutherford scattering cross section is an example of the FBA. It is derived from a screened Coulomb potential, which can be written as:

$$V(r) = -\frac{1}{4\pi\varepsilon_0}\frac{Ze}{r}\exp\left(-\frac{r}{a_s}\right) \tag{2.1}$$

where $Z$ is the target atomic number, $e$ the electronic charge, $\varepsilon_0$ the permittivity of vacuum and $a_s$ the screening radius. With this potential, the FBA gives the well-known screened Rutherford cross section [13]:

$$\frac{d\sigma_{el}}{d\Omega} = \frac{1}{(4\pi\varepsilon_0)^2}\frac{Z^2e^4}{4E^2}\frac{1}{(1+2\beta-\cos\vartheta)^2}\ , \tag{2.2}$$

and the total cross section is

$$\sigma_{el} = \frac{1}{(4\pi\varepsilon_0)^2}\frac{Z^2e^4}{4E^2}\frac{\pi}{\beta(1+\beta)}\ . \tag{2.3}$$

In these relations, $E$ is the electron energy, $\vartheta$ the scattering angle of the deflected electron, and the screening radius

$$a_s = a_B Z^{-1/3}\left(\frac{0.885}{\tau}\right) = \frac{a_{TF}}{\tau}\ , \tag{2.4}$$

with the Bohr radius $a_B$, the Thomas-Fermi radius $a_{TF}$, the screening parameter $\tau$, and the resulting screening value

$$\beta = \frac{h}{8mE}\frac{1}{a_s} \tag{2.5}$$

where $m$ is the rest mass of the electron and $h$ the Dirac constant (the Planck's constant divided by 2π).

The total cross section, eq. (2.3), is connected with the elastic mean free path (MFP)

$$\lambda_{el} = \frac{1}{N\sigma_{el}} \quad , \tag{2.6}$$

where $N$ is the atomic density of the target solid.

For high energies the screening parameter $\tau$ is nearly $\tau \sim 1$ but with decreasing energy it has to be corrected to higher values. Although the FBA is expected to fail at low energy, it can be used with a modified form of the screening parameter, instead of using the more precise method, i.e., the partial wave expansion (PWE). The PWE method gives the exact elastic cross section, the potential being incorporated in the phase shifts. However, it has been shown that, at 1 keV, noticeable difference between the FBA and the PWE results are only seen at large scattering angles for heavy elements [14].

Fitting and Reinhardt [15] tried to fit $\tau$ to the total cross sections, which have been calculated by the PWE methods. They found an appropriate approximation;

$$\tau(E) = 0.9 + \exp\left(-\frac{E}{E_\tau}\right) \quad , \tag{2.7}$$

for $E$ >100 eV with the only parameter $E_\tau$ in eV: 350 (C, $Z$=6), 300 (O, $Z$=8), 250 (Al, $Z$=13), 1600 (Cu, $Z$=29), 2500 (Ag, $Z$=47), 7600 (Au, $Z$=79). Assuming a monotonous increasing with $Z$, except for low $Z$, we can approximate the values of $E_\tau$ for different solid atoms with the fitted values by interpolation. The estimated values are listed in Table 1. The elastic MFPs using the parameters are shown for some metals in Fig. 1.

| Target (Z) | $E_\tau$ (eV) | $E_F$ (eV) | Φ (eV) |
|---|---|---|---|
| Be (4) | 350 | 11.76 | 4.98 |
| Al (13) | 250 | 11.07 | 4.17 |
| Si (14) | 330 | 7.83 | 4.79 |
| V (23) | 1100 | 6.35 | 4.30 |
| Cr (24) | 1180 | 7.25 | 4.50 |
| Mn (25) | 1270 | 7.51 | 4.10 |
| Fe (26) | 1350 | 7.76 | 4.74 |
| Co (27) | 1420 | 7.97 | 5.00 |
| Ni (28) | 1520 | 8.83 | 5.20 |
| Cu (29) | 1600 | 8.29 | 4.76 |
| Nb (41) | 2200 | 5.54 | 4.33 |
| Mo (42) | 2250 | 6.71 | 4.57 |
| Pd (46) | 2450 | 6.20 | 5.22 |
| Ag (47) | 2500 | 7.61 | 4.63 |
| Ta (73) | 6620 | 5.51 | 4.25 |
| W (74) | 6800 | 6.89 | 4.55 |
| Pt (78) | 7440 | 7.90 | 5.64 |
| Au (79) | 7600 | 9.11 | 5.38 |
| Pb (82) | 8100 | 9.37 | 4.25 |

**Table 1.** Screening energy parameter $E_\tau$ [15], Fermi energy $E_F$ [22] and work function [23] for some solids.

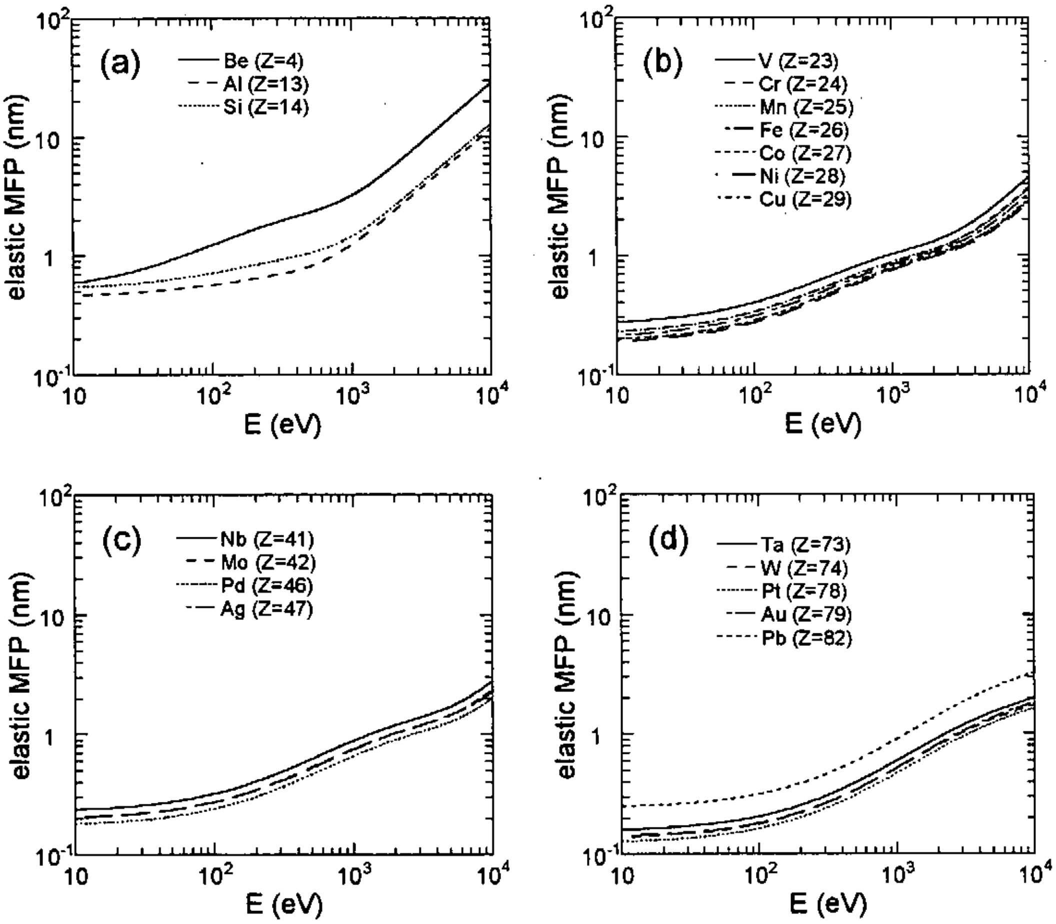

**Figure 1**. Energy dependence of elastic MFP of electrons in solids.

### *2.1.2 Inelastic collision*

A rigorous description of the inelastic interactions in a solid will be rather sophisticated. An essential simplification of this problem is to consider that the solid reacts as a whole to an external charge, in this case an incident electron. The dielectric theory of the response of a solid to a point charge is regarded as having a sufficiently wide range of validity [16]. The problem for SEE calculations is that not only the information on inelastic scattering is needed but also the information on the excitation of electrons. For this reason, most authors use the Lindhard dielectric function or an improved version of it. In this context, only nearly-free-electron-metals, such as aluminum, can be studied.

The theoretical formulation for inelastic scattering provides the probability of an energy loss $\omega$ per unit distance traveled by an electron of energy $E$ as [17]

$$\tau(E,\omega)=\frac{1}{\pi E}\int_{q-}^{q+}\frac{dq}{q}\operatorname{Im}\left[-\frac{1}{\varepsilon(q,\omega)}\right] , \tag{2.8}$$

in atomic units, where $\varepsilon(q,\omega)$ is a complex dielectric function and the integration with respect to the momentum transfer $q$ is limited to be

$$q_{\pm}=\sqrt{2}\left[\sqrt{E}\pm\sqrt{E-\omega}\right] . \tag{2.9}$$

The integration of $\tau(E,\omega)$ over the allowed values of $\omega$ yields the inelastic MFP $\lambda_{inel}$ through

$$\frac{1}{\lambda_{inel}}=\int\tau(E,\omega)d\omega . \tag{2.10}$$

Instead of the Lindhard dielectric function, Penn [18] and Ashley [19] approximated $\varepsilon(q,\omega)$ from the optical constant for the $q$=0 limit, $\varepsilon(0,\omega)$, where $\varepsilon(0,\omega)=(n+ik)^2$; $n$ is the index of reflection and $k$ the extinction coefficient which are obtained by the optical absorption. Different approaches using the experimentally determined optical constants have been developed to reach a detailed description of the interactions between energetic electrons and solids. The advantage of the approximate treatment is that since $\varepsilon(0,\omega)$ is based on the experimental results, it is applicable to different material species such as normal metals, noble metals, semiconductors, insulators, and compound materials. Another advantage is that it includes not only the bulk plasmon excitation and the conduction band electron excitation, which can be treated by the Lindhard dielectric function, but also the complicated processes of the inter-band, intra-band and some other transition mechanisms automatically.

According to the optical data model developed by Ashley [19], the extension of the dielectric response function, Im[-1/ $\varepsilon(q,\omega)$], to $q$>0 from the optical limit ($q$=0) is made through

$$\mathrm{Im}\left[-\frac{1}{\varepsilon(q,\omega)}\right]=\int_0^\infty \mathrm{Im}\left[-\frac{1}{\varepsilon(0,\omega')}\right]\frac{\delta\left[\omega-\left(\omega'+q^2/2\right)\right]}{\omega}\omega' d\omega' \ , \qquad (2.11)$$

which assumes a simple quadratic extension into the energy- and momentum-transfer plane. Including exchange corrections between the incident and electrons in the solid, the analytical integration over $\omega$ in eq. (2.10) combining with eq. (2,11) leads to the approximate expression for the inelastic inverse MFP. For metals, where states in the conduction band are filled up to the Fermi energy $E_F$, there is an additional restriction. The maximum value of $\omega$ should be the smaller of $\{(3/4)E; E\text{-}E_F\}$, which restrict the validity of the results for metals to $E\text{–}E_F>3E_F$. As a result, the inelastic inverse MFP is given as:

$$\frac{1}{\lambda_{inel}}=\frac{1}{2\pi E}\int_0^{E/2}\mathrm{Im}\left[-\frac{1}{\varepsilon(0,\omega')}\right]L_e\left(\frac{\omega'}{E}\right)d\omega' \ , \qquad (2.12)$$

where

$$L_e(a)=(1-a)\ln\left(\frac{4}{a}\right)-\frac{7}{4}a+a^{3/2}-\frac{33}{32}a^2 \ . \qquad (2.13)$$

From the above expressions, the values of $\lambda_{inel}(E)$ and $\tau(E,\omega)$ for different solid species listed in Table 1 are calculated for use in ELECTRAN calculations. The optical data of these solids are taken from [20,21]. In Fig. 2, the inelastic MFPs, $\lambda_{inel}$, are shown for these solids.

### 2.2 Escape of secondary electrons

The surface barrier influences the energy and angular distributions of SEs in the same way that light refracts at the interface between two media. For a primary electron penetrating the surface from the vacuum, the energy of the electron before the first inelastic collision is given by

$$E=E_p+U\,, \qquad (2.14)$$

where the primary energy, $E_\mathrm{p}$, is measured from the vacuum level and $E$ from the top of the conduction band. The inner potential, $U$, is approximated by the sum of the Fermi energy, $E_F$, and the work function, $\Phi$:

$$U=E_F+\Phi\,, \qquad (2.15)$$

In the case of electron ejection into vacuum from the solid, the energy measured from the vacuum level is

$$E=E'-U\,, \qquad (2.16)$$

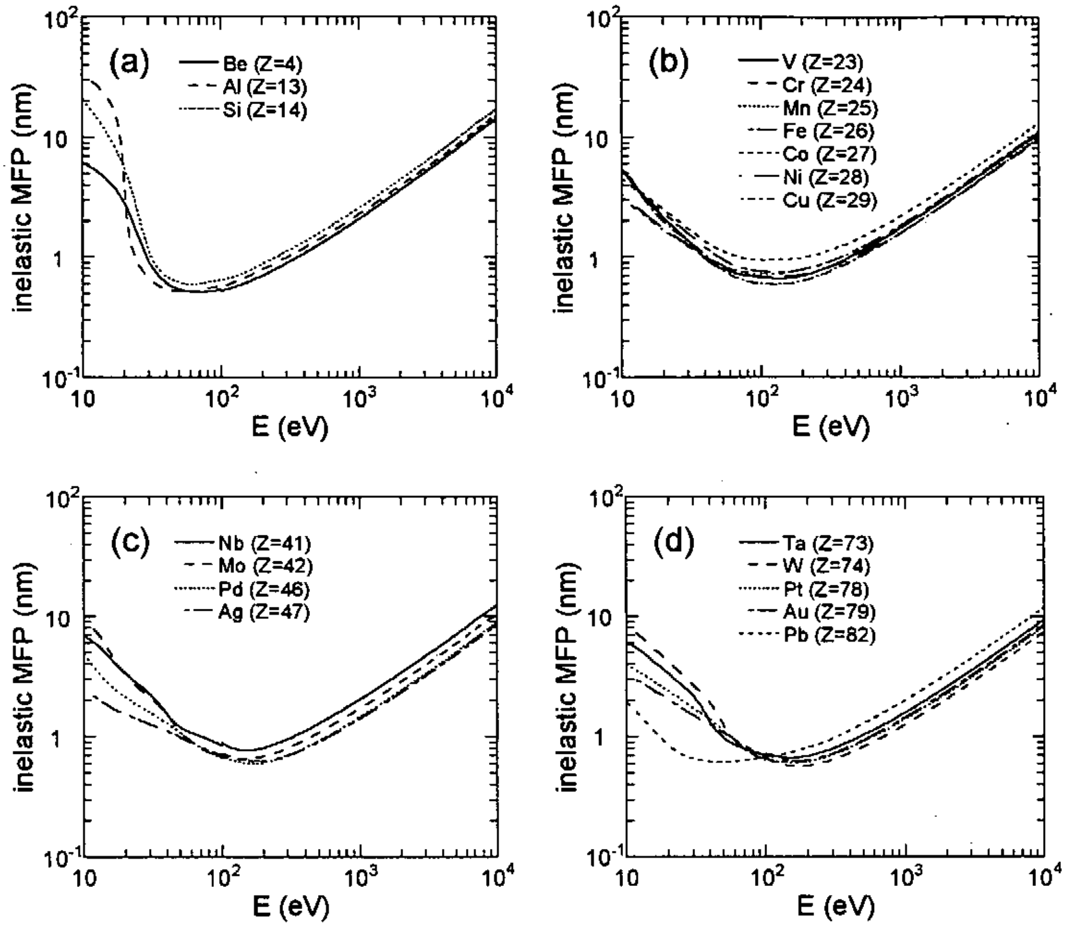

**Figure 2.** Energy dependence of inelastic MFP of electrons in solids.

and the ejection angle, $\theta$, measured from the normal to the surface is found by momentum conservation parallel to the surface:

$$\sqrt{E}\sin\theta = \sqrt{E'}\sin\theta' \quad , \tag{2.17}$$

where the prime denotes the corresponding quantity inside the solid. It should be noted that the surface barrier prohibits an electron from escaping from the surface into vacuum with an angle $\theta' > \theta_c$, where

$$\theta_c = \sin^{-1}\sqrt{1-\frac{U}{E'}} = \cos^{-1}\sqrt{\frac{U}{E'}} \quad . \tag{2.18}$$

The values of $E_F$ and $\Phi$ used in ELECTRAN are listed in Table 1. For 19 species of solids, the values of $E_F$ obtained from different theoretical studies [22] are averaged, and the values of $\Phi$ are taken from [23].

### 2.3 Monte Carlo model

A mathematical explanation of the calculation procedure for Monte Carlo simulation of SEE, which is performed in ELECTRAN, is given here. The detailed description of the program will be presented in the next section, along with the program listings.

The step length, $s$, of a scattering electron between two successive collision events is assumed to obey the Poisson stochastic process with the probability distribution

$$P(s)=\frac{1}{\lambda_{tot}}\exp\left(-\frac{s}{\lambda_{tot}}\right) , \tag{2.19}$$

where $\lambda_{tot}$ is the total MFP and is related to the corresponding elastic MFP, $\lambda_{el}$, and inelastic MFP, $\lambda_{inel}$, through

$$\frac{1}{\lambda_{tot}}=1\Big/\left(\int_0^{\infty} sP(s)ds\right)=\frac{1}{\lambda_{el}}+\frac{1}{\lambda_{inel}} , \tag{2.20}$$

By using a uniform random number, $R_1 \in [0,1]$, we obtain a value of $s$,

$$R_1=\int_0^{s} P(s')ds'=1-\exp\left(-\frac{s}{\lambda_{tot}}\right) , \tag{2.21}$$

and thus,

$$s=-\lambda_{tot}\ln(1-R_1)\equiv-\lambda_{tot}\ln(R_1) , \tag{2.22}$$

since $1\text{-}R_1 \in [0,1]$. This process selects a step length regardless of the type of individual scattering event. The event type is instead determined by another random number, $R_2$:

$$\text{elastic if } R_2<\frac{1/\lambda_{el}}{1/\lambda_{tot}}, \text{ and inelastic otherwise.} \tag{2.23}$$

If the scattering is elastic, the scattering angle, $\vartheta$, is calculated using a third random number,$R_3$:

$$R_3=\int_0^{\vartheta}\frac{d\sigma_{el}}{d\Omega}\sin\vartheta' d\vartheta' \Big/ \int_0^{\pi}\frac{d\sigma_{el}}{d\Omega}\sin\vartheta' d\vartheta' . \tag{2.24}$$

In the case of the screened Rutherford cross-section, i.e. eq. (2.3), the integration in eq. (2.4) can be done analytically, and the scattering angle is calculated by [13]

$$\cos\vartheta=1-\frac{2\beta R_3}{(1+\beta-R_3)} . \tag{2.25}$$

If inelastic scattering is selected by eq. (2.23), we determine the amount of energy loss $\Delta E$ (=$\omega$ in eq. (2.8)) by a fourth random number, $R_4$:

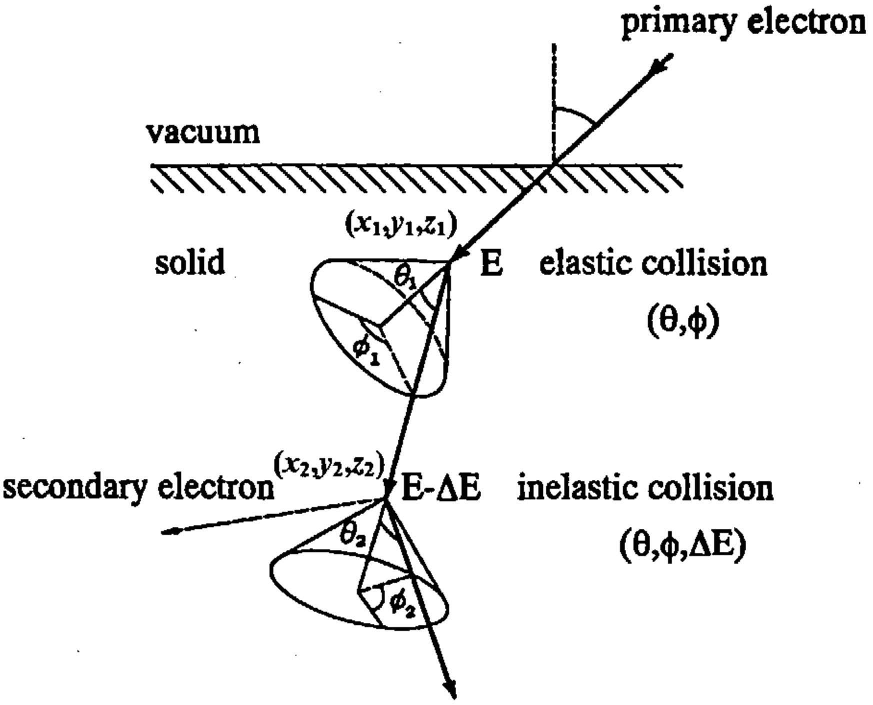

**Figure 3.** Schematic view of an electron trajectory in Monte Carlo simulation.

$$R_4 = \int_{E_{\min}}^{\Delta E} \tau(E,\Delta E)d(\Delta E) \Big/ \int_{E_{\min}}^{E_{\max}} \tau(E,\Delta E)d(\Delta E), \tag{2.26}$$

where the integration limits are set to $E_{min}$=0 and $E_{max}$=$E$–$E_F$. With respect to the scattering angle of an inelastically scattered electron in the direct simulation model the inelastic scattering angle in an individual event is simply related to energy loss through the relation

$$\sin\vartheta = \sqrt{\frac{\Delta E}{E}}. \tag{2.27}$$

A fifth random number, $R_5$, selects the azimuthal angle

$$\varphi = 2\pi R_5\ , \tag{2.28}$$

either for elastic or inelastic scattering.

On the SE generation in the model [24], it is assumed that the energy loss $\Delta E$ in an individual inelastic collision is transferred to a Fermi sea electron, resulting in a knock-on SE. Hence, each inelastic event produces a SE with the energy of $\Delta E$+$E_F$, whereas the primary electron loses its energy by $\Delta E$. The initial directional angle is determined according to classical collision scheme for the two electrons.

After a randomly occurring scattering event, the electron goes forward one step, and its position at the next scattering point is given by the equation

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} + s_n \begin{pmatrix} \sin\theta_n \cos\theta_n \\ \sin\theta_n \sin\theta_n \\ \cos\theta_n \end{pmatrix} , \tag{2.29}$$

where the angles $(\theta_n, \phi_n)$ in the coordinate system fixed with the solid are related to $(\theta, \varphi)$ in the coordinate system moving with the electron by the transformation relations

$$\begin{gathered} \cos\theta_n = \cos\theta_{n-1} \cos\vartheta - \sin\theta_{n-1} \sin\vartheta \cos\varphi \\ \sin(\phi_n - \phi_{n-1}) = \sin\vartheta \sin\varphi / \sin\theta_n \\ \cos(\phi_n - \phi_{n-1}) = (\cos\vartheta - \cos\theta_{n-1} \cos\theta_n)/(\sin\theta_{n-1} \sin\theta_n) \end{gathered} \tag{2.30}$$

We have made an assumption in the simulation that an electron transverses the first step, $s_0$, without scattering at the solid-vacuum interface, as shown in Fig. 3. The terminal point of the first flight is the first scattering point, $(x_1, y_1, z_1)$. By repeating the procedures described by eqs. (2.22)–(2.30), we can then derive the second scattering position, and so on, forming an electron trajectory which is terminated only after its kinetic energy falls below a cut-off energy, $E_c$.

Once a SE is generated by a primary electron in an excitation event, its energy, coordinates and direction of movement are stored in the memory of a computer. After finishing tracing the primary electrons, all the stored information on the SEs is recalled and the trajectory is simulated in a same way as for a primary electron. A large number of low-energy SEs are then successively generated. This cascade process is repeated until all the electrons either escape from the surface as true SEs or backscattered electrons, or come to rest within the solid, and treated as described in the last section.

## 3. Computer programs for secondary electron emission

This section presents FORTRAN computer programs and subroutines which implement the theory of the SEE in ELECTRAN. Since no special application software is not used in the code, ELECTRAN can operate with different types and versions of FORTRAN compilers on any personal-computer systems. The programs shown in this section only calculate results. No graphics programs are included in ELECTRAN, hence special graphics drivers are not necessary either. The calculated data are readily transferable to other computer systems for graphical presentation such as plotting.

The programs are:

Main program:

This is a FORTRAN main program which provides an input data on electron irradiation conditions and on materials irradiated by electrons for the electron transport subroutine "primary".

Subroutine: primary

This is a FORTRAN program which calculates the 3-dimensional electron penetration in a solid using MC simulation. The program calculates the backscattered electron (BSE) yield, and the energy and angular distributions of BSEs. It needs the subroutine program, "rannum", for the calculation. It also calls the subroutine "secondary" for SEE process.

Subroutine: secondary

This is a FORTRAN subroutine program which calculates the SE yield, and the energy and angular distributions of SEs. The program is called by the electron penetration subroutine "primary".

Subroutine: rannum

This is a FORTRAN program for generating a random number.

Data file: /e04opt.dat e04opts.dat /e13opt.dat e13opts.dat /e14opt.dat e14opts.dat /
/e23opt.dat e23opts.dat /e24opt.dat e24opts.dat /e26opt.dat e26opts.dat /
/e27opt.dat e27opts.dat /e28opt.dat e28opts.dat /e29opt.dat e29opts.dat /
/e41opt.dat e41opts.dat /e42opt.dat e42opts.dat /e46opt.dat e46opts.dat /
/e47opt.dat e47opts.dat /e73opt.dat e73opts.dat /e74opt.dat e74opts.dat /
/e78opt.dat e78opts.dat /e79opt.dat e79opts.dat /e82opt.dat e82opts.dat /

Data files of the MFP and the accumulated probability distribution function for electron excitation by electrons in materials. They are calculated by means of Ashley's optical-data model from optical constants compiled by Palik [21,22]. The file "e**opt.dat" corresponds to the data for a material with the atomic number Z of ** from 100 eV to 10 keV at 10 eV intervals, and the file "e**opts.dat" from 1 eV to 100 eV at 1 eV intervals. Both data allow for the calculation of electron excitation of high-energy PEs and low-energy SEs in materials.

### 3.1 Main program

The program is a main program of the ELECTRAN which calculates BSE and SE yields of monoatomic solids and the energy and angular distributions of BSEs and SEs. It calls the subroutine program "primary" for electron trajectory simulation.

Line 0032-0034 Set up physical constants

Line 0042-0043 Set up two seeds for generation of random numbers

Line 0045-0054 Read information on a primary electron and a target material for calculation and a seed for generation of random numbers.

e0: Incident energy of primary electron in eV: 100 – 10000 eV

an0: Incident angle of primary electron measured from surface normal in deg.: 0 – 89.999 deg.

z: Atomic number of target material. Available materials are currently:

4 (Be), 13 (Al), 14 (Si), 23 (V), 24 (Cr), 25 (Mn), 26 (Fe), 27 (Co), 28 (Ni), 29 (Cu), 41 (Nb), 42 (Mo), 46 (Pd), 47 (Ag), 73 (Ta), 74(W), 78 (Pt), 79 (Au) and 82 (Pb).

imax: Number of histories, i.e. the number of times to cycle the calculation with a new incident electron

seed1: An integer seed for random number generation.

Typical input for 1000 electrons (1 keV) into Al at an incident angle of 60 deg. would be:

```
1000.d0
60.d0
13.d0
10000
1234
```

Line 0058-0478 Set up material constants for calculation and read data files of the MFP and the accumulated probability distribution function for electron excitation by electrons in the materials.

wrfu: Work function in eV.

efev: Fermi energy in eV.

etau: Screening energy parameter $E_\tau$.

opmfp: One-dimensional array of MFP for electron excitation by electrons from 100 eV to 10 keV at 10 eV intervals.

opdmfp: One-dimensional array of MFP for electron excitation by electrons from 1 eV to 100 eV at 1 eV intervals.

oprnd: Two-dimensional array of the accumulative probability distribution for the energy loss of an electron due to electron excitation from 100 eV to 10 keV at 10 eV intervals.

opdrnd: Two-dimensional array of the accumulative probability distribution for the energy loss of an electron due to electron excitation from 1 eV to 100 eV at 1 eV intervals.

Line 0505-0508 Call the program "primary" and the Monte Carlo calculation is repeated by the number of primary electron.

Line 0512-0526 Output data. The file name for output is "electran.dat"; an example for 10000 electrons (1 keV) into Al at an incident angle of 60 deg. would be:

```
incident energy(eV)=     1000.00                  out00001
incident angle(deg)=       60.00                  out00002
incident number=     10000                        out00003
backscattered primary electrons=      5114       out00004
emitted secondary electrons=     22240            out00005
energy distribution                               out00006
       556                                        out00007
      1309                                        out00008
      1831                                        out00009
      2125                                        out00010
      2028                                        out00011
      1938                                        out00012
      1815                                        out00013
      1719                                        out00014
      1509                                        out00015
      1406                                        out00016
      1210                                        out00017
      1014                                        out00018
       820                                        out00019
       607                                        out00020
       467                                        out00021
       252                                        out00022
       137                                        out00023
       100                                        out00024
       67                                         out00025
       71                                         out00026
--------------
       22                                         out00988
       24                                         out00989
```

```
                  29                                  out00990
                  28                                  out00991
                  17                                  out00992
                  24                                  out00993
                  24                                  out00994
                  26                                  out00995
                  23                                  out00996
                   2                                  out00997
                   0                                  out00998
                   0                                  out00999
                   0                                  out01000
                   0                                  out01001
                   0                                  out01002
                   0                                  out01003
                   0                                  out01004
                   0                                  out01005
                   0                                  out01006
                 368                                  out01007
                   0                                  out01008
          ------------
                   0                                  out10008
angular distribution                                  out10009
                   0                                  out10010
                 131                                  out10011
                 351                                  out10012
                 628                                  out10013
                 795                                  out10014
                 995                                  out10015
                1095                                  out10016
                1241                                  out10017
                1184                                  out10018
                1210                                  out10019
                1247                                  out10020
                1240                                  out10021
                1104                                  out10022
                1026                                  out10023
```

```
 833                                   out10024
 696                                   out10025
 474                                   out10026
 309                                   out10027
 100                                   out10028
 104                                   out10029
 306                                   out10030
 504                                   out10031
 659                                   out10032
 772                                   out10033
 947                                   out10034
 985                                   out10035
1090                                   out10036
1105                                   out10037
1070                                   out10038
1061                                   out10039
 971                                   out10040
 954                                   out10041
 754                                   out10042
 607                                   out10043
 455                                   out10044
 264                                   out10045
  87                                   out10046
   0                                   out10047
```

Output-Row 00004: Number of backscattered electrons for 10000 primaries.

Output-Row 00005: Number of secondary electrons for 10000 primaries.

Output-Row 00007-10008: Number of electrons emitted with the energies from 0 eV to 10001 eV at 1 eV intervals, for 10000 primaries.

Output-Row 10010-10047: Number of electrons emitted with the angles from –90 deg. to +90 deg. at the interval of 5 deg., for 10000 primaries.

```
c*********************************************************************
c*****      Main Program of ELECtron TRANsport in Solids      ******
c*****              <<<<<<<ELECTRAN>>>>>>>                    ******
c*********************************************************************
      implicit real*8 (a-h,o-z)
      implicit integer*4 (i,j,k,l,m,n)
c*********************************************************************
      dimension iedatb(0:100001),iandatb(0:37)
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      dimension oprnd(0:550000),opmfp(0:1001)
      dimension opdrnd(0:5500),opdmfp(0:101)
c>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
c*********************************************************************
      common /ransu/random1
      common /tane/seed1,seed2,seed3
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      common /map2/oprnd,opmfp
      common /map8/opdrnd,opdmfp
      common /con1/efev,plaev,wrfu
      common /con2/dene,deie,out1,out2
      common /con3/deied,outd1,outd2
      common /con4/dened
      common /fitting/atf,etau,zt,dent
      common /flag1/ips,ipar,peeff
      common /hant1/ee,esur,rcosz,rsinxy,rcosx,rsiny,refra
      common /trabac/itraj,ibp,it,itr,iedatb,iandatb
      common /incident/e00,an
c>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
      integer*4 seed1,seed2,seed3
      integer*4 out1,out2,outd1,outd2,peeff
c*********************************************************************
      parameter(ev=1.60219d-12,em=9.0953d-28,a0=5.29177d-9)
      parameter(ec=4.80324d-10,pi=3.14159d0,h=1.05459d-27)
      parameter(amug=1.66057d-24)
c*********************************************************************
c     z   : atomic number of the material
c     amu : average atomic mass of the material (a.m.u.)
c     ad0 : weight density of the material (g/cm-3)
c     e0  : electron energy (eV)
c     an0 : incidence angle measured from the normal (deg.)
c*********************************************************************
      seed2=5988
      seed3=29493
C***********************************************************************
      write(6,*) 'incidence energy(eV)'
      read(5,*) e0
      write(6,*) 'incidence angle(degree)'
      read(5,*) an0
      write(6,*) 'atomic number of material'
      read(5,*) z
      write(6,*) 'incidence electron number'
      read(5,*) imax
      write(6,*) 'seed for generation of random number'
      read(5,*) seed1
c*********************************************************************
c Get material and optical data
c*********************************************************************
c***** Be (Z=4) *****
      if (nint(z) .eq. 4) then
         zt=dble(z)
         amu=9.012182d0
         ad0=1.848d0
         wrfu=4.98d0
         efev=11.76d0
         etau=350.d0
         open(1,file='e04opt.dat',status='old')
           read(1,1) ienediv,deie
    1      format(1x,i5,d15.8)
           do 40 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
    2        format(1x,i5,6d15.8)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
    3        format(1x,5d15.8)
   40      continue
         close(1)
         open(1,file='e04opts.dat',status='old')
           read(1,1) ienediv,deied
           do 45 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
   45      continue
         close(1)
c***** Al (Z=13) *****
      elseif (nint(z) .eq. 13) then
         zt=dble(z)
         amu=26.981538d0
         ad0=2.700d0
         wrfu=4.17d0
         efev=11.07d0
         etau=250.d0
         open(1,file='el3opt.dat',status='old')
           read(1,1) ienediv,deie
           do 130 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  130      continue
         close(1)
         open(1,file='el3opts.dat',status='old')
           read(1,1) ienediv,deied
           do 135 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  135      continue
         close(1)
c***** Si (Z=14) *****
      elseif (nint(z) .eq. 14) then
         zt=dble(z)
         amu=28.0855d0
         ad0=2.330d0
         wrfu=4.79d0
         efev=7.83d0
         etau=330.d0
         open(1,file='el4opt.dat',status='old')
           read(1,1) ienediv,deie
           do 140 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  140      continue
         close(1)
         open(1,file='el4opts.dat',status='old')
```

```
          read(1,1) ienediv,deied
          do 145 i=0,ienediv
            read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
            read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  145     continue
        close(1)
c***** V (Z=23) *****
      elseif (nint(z) .eq. 23) then
        zt=dble(z)
        amu=50.9415d0
        ad0=6.110d0
        wrfu=4.30d0
        efev=6.35d0
        etau=1100.d0
        open(1,file='e23opt.dat',status='old')
          read(1,1) ienediv,deie
          do 230 i=0,ienediv
            read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
            read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  230     continue
        close(1)
        open(1,file='e23opts.dat',status='old')
          read(1,1) ienediv,deied
          do 235 i=0,ienediv
            read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
            read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  235     continue
        close(1)
c***** Cr (Z=24) *****
      elseif (nint(z) .eq. 24) then
        zt=dble(z)
        amu=51.9961d0
        ad0=7.140d0
        wrfu=4.50d0
        efev=7.25d0
        etau=1180.d0
        open(1,file='e24opt.dat',status='old')
          read(1,1) ienediv,deie
          do 240 i=0,ienediv
            read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
            read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  240     continue
        close(1)
        open(1,file='e24opts.dat',status='old')
          read(1,1) ienediv,deied
          do 245 i=0,ienediv
            read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
            read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  245     continue
        close(1)
c***** Mn (Z=25) *****
      elseif (nint(z) .eq. 25) then
        zt=dble(z)
        amu=54.938049d0
        ad0=7.470d0
        wrfu=4.10d0
        efev=7.51d0
        etau=1270.d0
        open(1,file='e24opt.dat',status='old')
          read(1,1) ienediv,deie
          do 250 i=0,ienediv
            read(1,2) outl,edum0,edum1,edum2,efevm,plaev,opmfp(i)
            read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  250     continue
        close(1)
        open(1,file='e24opts.dat',status='old')
          read(1,1) ienediv,deied
          do 255 i=0,ienediv
            read(1,2) outdl,edum0,edum1,edum2,efevm,plaev,opdmfp(i)
            read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  255     continue
        close(1)
c***** Fe (Z=26) *****
      elseif (nint(z) .eq. 26) then
        zt=dble(z)
        amu=55.845d0
        ad0=7.874d0
        wrfu=4.74d0
        efev=7.76d0
        etau=1350.d0
        open(1,file='e26opt.dat',status='old')
          read(1,1) ienediv,deie
          do 260 i=0,ienediv
            read(1,2) outl,edum0,edum1,edum2,efevm,plaev,opmfp(i)
            read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  260     continue
        close(1)
        open(1,file='e26opts.dat',status='old')
          read(1,1) ienediv,deied
          do 265 i=0,ienediv
            read(1,2) outdl,edum0,edum1,edum2,efevm,plaev,opdmfp(i)
            read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  265     continue
        close(1)
c***** Co (Z=27) *****
      elseif (nint(z) .eq. 27) then
        zt=dble(z)
        amu=58.933200d0
        ad0=8.900d0
        wrfu=5.00d0
        efev=7.97d0
        etau=1420.d0
        open(1,file='e27opt.dat',status='old')
          read(1,1) ienediv,deie
          do 270 i=0,ienediv
            read(1,2) outl,edum0,edum1,edum2,efevm,plaev,opmfp(i)
            read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  270     continue
        close(1)
        open(1,file='e27opts.dat',status='old')
          read(1,1) ienediv,deied
          do 275 i=0,ienediv
            read(1,2) outdl,edum0,edum1,edum2,efevm,plaev,opdmfp(i)
            read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  275     continue
        close(1)
c***** Ni (Z=28) *****
      elseif (nint(z) .eq. 28) then
        zt=dble(z)
        amu=58.6934d0
```

```
         ad0=8.908d0
         wrfu=5.20d0
         efev=8.83d0
         etau=1520.d0
         open(1,file='e28opt.dat',status='old')
           read(1,1) ienediv,deie
           do 280 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  280      continue
         close(1)
         open(1,file='e28opts.dat',status='old')
           read(1,1) ienediv,deied
           do 285 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  285      continue
         close(1)
c***** Cu (Z=29) *****
      elseif (nint(z) .eq. 29) then
         zt=dble(z)
         amu=63.546d0
         ad0=8.920d0
         wrfu=4.76d0
         efev=8.29d0
         etau=1600.d0
         open(1,file='e29opt.dat',status='old')
           read(1,1) ienediv,deie
           do 290 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  290      continue
         close(1)
         open(1,file='e29opts.dat',status='old')
           read(1,1) ienediv,deied
           do 295 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  295      continue
         close(1)
c***** Nb (Z=41) *****
      elseif (nint(z) .eq. 41) then
         zt=dble(z)
         amu=92.90638d0
         ad0=8.570d0
         wrfu=4.33d0
         efev=5.54d0
         etau=2200.d0
         open(1,file='e41opt.dat',status='old')
           read(1,1) ienediv,deie
           do 410 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  410      continue
         close(1)
         open(1,file='e41opts.dat',status='old')
           read(1,1) ienediv,deied
           do 415 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  415      continue
         close(1)
c***** Mo (Z=42) *****
      elseif (nint(z) .eq. 42) then
         zt=dble(z)
         amu=95.94d0
         ad0=10.280d0
         wrfu=4.57d0
         efev=6.71d0
         etau=2250.d0
         open(1,file='e42opt.dat',status='old')
           read(1,1) ienediv,deie
           do 420 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  420      continue
         close(1)
         open(1,file='e42opts.dat',status='old')
           read(1,1) ienediv,deied
           do 425 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  425      continue
         close(1)
c***** Pd (Z=46) *****
      elseif (nint(z) .eq. 46) then
         zt=dble(z)
         amu=106.42d0
         ad0=12.023d0
         wrfu=5.22d0
         efev=6.20d0
         etau=2450.d0
         open(1,file='e46opt.dat',status='old')
           read(1,1) ienediv,deie
           do 460 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  460      continue
         close(1)
         open(1,file='e46opts.dat',status='old')
           read(1,1) ienediv,deied
           do 465 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  465      continue
         close(1)
c***** Ag (Z=47) *****
      elseif (nint(z) .eq. 47) then
         zt=dble(z)
         amu=107.8682d0
         ad0=10.490d0
         wrfu=4.63d0
         efev=7.61d0
         etau=2500.d0
         open(1,file='e47opt.dat',status='old')
           read(1,1) ienediv,deie
           do 470 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  470      continue
```

```
         close(1)
         open(1,file='e47opts.dat',status='old')
           read(1,1) ienediv,deied
           do 475 i=0,ienediv
             read(1,2) outd1,edum0,edum1,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outd1)
  475      continue
         close(1)
c***** Ta (Z=73) *****
      elseif (nint(z) .eq. 73) then
         zt=dble(z)
         amu=180.9479d0
         ad0=16.650d0
         wrfu=4.25d0
         efev=5.51d0
         etau=6620.d0
         open(1,file='e73opt.dat',status='old')
           read(1,1) ienediv,deie
           do 730 i=0,ienediv
             read(1,2) out1,edum0,edum1,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,out1)
  730      continue
         close(1)
         open(1,file='e73opts.dat',status='old')
           read(1,1) ienediv,deied
           do 735 i=0,ienediv
             read(1,2) outd1,edum0,edum1,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outd1)
  735      continue
         close(1)
c***** W (Z=74) *****
      elseif (nint(z) .eq. 74) then
         zt=dble(z)
         amu=183.84d0
         ad0=19.293287768d0
         wrfu=4.55d0
         efev=6.89d0
         etau=6800.d0
         open(1,file='e74opt.dat',status='old')
           read(1,1) ienediv,deie
           do 740 i=0,ienediv
             read(1,2) out1,edum0,edum1,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,out1)
  740      continue
         close(1)
         open(1,file='e74opts.dat',status='old')
           read(1,1) ienediv,deied
           do 745 i=0,ienediv
             read(1,2) outd1,edum0,edum1,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outd1)
  745      continue
         close(1)
c***** Pt (Z=78) *****
      elseif (nint(z) .eq. 78) then
         zt=dble(z)
         amu=195.078d0
         ad0=21.090d0
         wrfu=5.64d0
         efev=7.90d0
         etau=7440.d0
         open(1,file='e78opt.dat',status='old')
           read(1,1) ienediv,deie
           do 780 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  780      continue
         close(1)
         open(1,file='e78opts.dat',status='old')
           read(1,1) ienediv,deied
           do 785 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  785      continue
         close(1)
c***** Au (Z=79) ***
      elseif (nint(z) .eq. 79) then
         zt=dble(z)
         amu=196.96655d0
         ad0=19.300d0
         wrfu=5.38d0
         efev=9.11d0
         etau=7600.d0
         open(1,file='e79opt.dat',status='old')
           read(1,1) ienediv,deie
           do 790 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  790      continue
         close(1)
         open(1,file='e79opts.dat',status='old')
           read(1,1) ienediv,deied
           do 795 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  795      continue
         close(1)
c***** Pb (Z=82) *****
      elseif (nint(z) .eq. 82) then
         zt=dble(z)
         amu=207.2d0
         ad0=11.340d0
         wrfu=4.25d0
         efev=9.37d0
         etau=8100.d0
         open(1,file='e82opt.dat',status='old')
           read(1,1) ienediv,deie
           do 820 i=0,ienediv
             read(1,2) outl,edum0,eduml,edum2,efevm,plaev,opmfp(i)
             read(1,3) (oprnd(((i*(i+1))/2)+ii),ii=0,outl)
  820      continue
         close(1)
         open(1,file='e82opts.dat',status='old')
           read(1,1) ienediv,deied
           do 825 i=0,ienediv
             read(1,2) outdl,edum0,eduml,edum2,efevm,plaev,opdmfp(i)
             read(1,3) (opdrnd(((i*(i+1))/2)+ii),ii=0,outdl)
  825      continue
         close(1)
c***** data end *****
      else
```

```
         goto 10000
      endif
c*******************************************************************
      e00=e0*ev
      an=an0*pi/180.d0
      r0=z**(1.d0/3.d0)/0.885d0/a0
      dent=ad0/amu/1.66057d-24
      am=amu*amug
      rm=am/(am+em)*em
c>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
c*******************************************************************
      do 999 ici=1,3000
         call rannum
         rini=random1
  999 continue
c*******************************************************************
      ibp=0
      ips=0
      it=0
      itr=0
c*******************************************************************
c*******************************************************************
c  Start of main loop, processing all histories
c*******************************************************************
      do 1000 itraj=1,imax
c*******************************************************************
      call primary
 1000 continue
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
c  Write Output Data
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      open(2,file='electran.dat',status='new')
      write(2,2000)e0,an0,imax
      write(2,2003)ibp
      write(2,2004)ips
      write(2,*)'energy distribution'
      write(2,2002)(iedatb(i),i=0,10001)
      write(2,*)'angular distributions'
      write(2,2002)(iandatb(i),i=0,37)
 2000 format(1x,'incident energy(eV)=',f12.2/1x,
     /          'incident angle(deg)=',f12.2/1x,
     /          'incident number=',i10)
 2002 format(1x,i10)
 2003 format(1x,'backscattered primary electrons=',i10)
 2004 format(1x,'emitted secondary electrons=',i10)
      close(2)
      write(6,*) '*************************************************'
      write(6,*) 'End Simulation'
      write(6,*) '*************************************************'
c>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
10000 stop
      end
```

### 3.2 Subroutine “primary”

The program calculates the electron penetration of a solid and provides the backscattered electrons and their final trajectory angle and energy. It allows the treatment of monoatomic solids of 19 species from Be (the atomic number, $Z$=4) to Pb ($Z$=82). The mathematics of this MC program has been described in the last section. In this section, the program listing is presented along with some comments.

Line 0009 iedatb: One-dimensional array of energy of backscattered and secondary electrons.

iandatb: One-dimensional array of angle of backscattered and secondary electrons.

Line 0046-0051 Set up physical constants for the calculation.

pi: π (=3.141592654)

angst: Conversion of units in length from cm to angstrom.

abohr: Bohr radius in cm.

esu2: Electron charge in esu, squared.

dirac: Dirac constant (the Planck constant divided by 2) in erg s.

me: Electron rest mass in g.

elv: Conversion for units in energy from eV to erg.

dirac2: Dirac constant in eV s

atf: Thomas-Fermi radius in cm (see eq.(2.4)).

Line 0053-0057 Sep up energy and angle of a primary electron.

ee0: Incident energy of a primary electron in eV.

ee: Incident energy of a primary electron in erg.

an: Incident angle in degree, measured from the surface normal.

rsinxy: Sine of the angle between the direction of a moving electron between the surface normal

rcosz: Cosine of the angle between the direction of a moving electron between the surface normal

Line 0059-0067 Initialization of the quantities which are used in following the electron trajectory

ise: Index of generated SEs, which is used in the subroutine, second.f.

fp: Step length of a scattering electron

x, y, z: The position where a penetrating electron is scattered.

xold, yold, zold: The last position where the electron was scattered.

esur: The surface binding energy in eV, defined as $E_F+\phi$.

Line 0069-0087 Set up energy and angle of a penetrating electron from the vacuum to the solid through the surface barrier.

Line 0109 Start to search the position where the next collision occurs.

Line 0136-0140 Calculate elastic MFP from the screened Rutherford cross section (see eqs. (2.3)-(2.6)) for next collision.

Line 0141-0159 Determine inelastic MFP from the data files "e**opt.dat" and "e**opts.dat" for next collision.

Line 0160 Calculate total MFP for next collision by using eq. (2.20).

Line 0161-0176 Calculate the step length of a penetrating electron (eq. (2.22)) and the position where the next collision occurs.

Line 0191 If the penetrating electron returns to the surface, the calculation jumps to Line 0408.

Line 0224-0225 Select either an elastic collision or an inelastic collision by comparing the inverse MFPs.

Line 0226-0227 If an elastic collision is selected, the polar scattering angle is calculated by using eq. (2.25) and the azimuthal scattering angle is randomly selected between 0 and 2π (eq. (2.28)).

Line 0232-0331 If the inelastic collision is selected, the energy loss, $\Delta E$, of the penetrating electron is determined from the data files "e**opt.dat" and "e**opts.dat". The polar scattering angle is calculated from eq. (2.27), whereas the azimuthal angle is randomly selected. The energy of the penetrating electrons is reduced by the energy loss. Furthermore, the energy of an excited SE is set to be $\Delta E + U'$. The surface binding energy $U'$ is measured from the energy level of the SE excited in the conduction band.

Line 0354-0397 Determine the direction of the penetrating electron after the collision in the laboratory coordinate system.

Line 0400 If the electron energy is above the surface binding energy after the collision, the calculation jumps to the next collision loop for repetition.

Line 0403 If the electron energy is below the surface binding energy, the calculation stops and jumps to the trajectory simulation of the next incident electron.

Line 0421-0442 If the directional angle with respect to the surface normal is below the critical angle $\theta_c$ in eq. (2.18), the electron is ejected with the energy reduced by $U$ and with the refracted angle. If not, the calculation jumps to the simulation of the next incident electron. The ejected electrons are stored in a variable “ibp” whereas their energy and ejection angle ere stored arrays, “iedatb” and “iandatb”, respectively. After this procedure, the calculation jumps to the next simulation of the net incident electron.

```
c*******************************************************************     primar0001
c*****          Transport of primary electrons in solids      ******     primar0002
c*******************************************************************     primar0003
      subroutine primary                                                 primar0004
c*******************************************************************     primar0005
      implicit real*8 (a-h,o-z)                                          primar0006
      implicit integer*4 (i,j,k,l,m,n)                                   primar0007
c*******************************************************************     primar0008
      dimension iedatb(0:100001),iandatb(0:37)                           primar0009
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     primar0010
      dimension oprnd(0:550000),opmfp(0:1001)                            primar0011
      dimension opdrnd(0:5500),opdmfp(0:101)                             primar0012
c<<<<<<<<<<<<<<<<local arrays <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     primar0013
      dimension gxse(10000),gyse(10000),gzse(10000)                      primar0014
      dimension gsca(10000),gazi(10000)                                  primar0015
      dimension gene(10000),gcosx(10000),gsiny(10000),gsinxy(10000)      primar0016
      dimension gcosz(10000)                                             primar0017
c*******************************************************************     primar0018
      common /ransu/random1                                              primar0019
      common /tane/seed1,seed2,seed3                                     primar0020
      common /posi/xse,yse,zse/dire/ssinxy,scosx,ssiny,scosz             primar0021
      common /sanran/sescaan,seazian,seene                               primar0022
      common /map2/oprnd,opmfp                                           primar0023
      common /map8/opdrnd,opdmfp                                         primar0024
      common /con1/efev,plaev,wrfu                                       primar0025
      common /con2/dene,deie,out1,out2                                   primar0026
      common /con3/deied,outd1,outd2                                     primar0027
      common /con4/dened                                                 primar0028
      common /flag1/ips,ipar,peeff                                       primar0029
      common /flag5/ise                                                  primar0030
      common /adse1/gxse,gyse,gzse,gsca,gazi,gene                        primar0031
      common /adse2/gcosx,gsiny,gsinxy,gcosz                             primar0032
      common /plapro/conpla                                              primar0033
      common /fitting/atf,etau,zt,dent                                   primar0034
      common /hant1/ee,esur,rcosz,rsinxy,rcosx,rsiny,refra               primar0035
      common /hant2/xe,ye,ze,xeo,yeo,zeo                                 primar0036
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     primar0037
      common /trabac/itraj,ibp,it,itr,iedatb,iandatb                     primar0038
      common /incident/e00,an                                            primar0039
c>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>     primar0040
c*******************************************************************     primar0041
      real*8 me,mfpt,iopmfp,normal                                       primar0042
      integer*4 seed1,seed2,seed3                                        primar0043
      integer*4 out1,out2,outd1,outd2,peeff                              primar0044
c*******************************************************************     primar0045
      parameter(pi=3.141592654d0,angst=1.d-8,abohr=0.529d-8)             primar0046
      parameter(esu2=2.30711d-19,dirac=1.05459d-27)                      primar0047
      parameter(me=9.10953d-28,elv=1.60219d-12)                          primar0048
c*******************************************************************     primar0049
      dirac2=dirac/elv                                                   primar0050
      atf=0.885d0*abohr/(zt**(1.d0/3.d0))                                primar0051
c*******************************************************************     primar0052
      ee=e00/elv                                                         primar0053
      rcosx=-1.d0                                                        primar0054
      rsiny=0.d0                                                         primar0055
      rsinxy=sin(an)                                                     primar0056
      rcosz=cos(an)                                                      primar0057
c*******************************************************************     primar0058
      ise=0                                                              primar0059
      fp=0.d0                                                            primar0060
      x=0.d0
      y=0.d0
      z=0.d0
      xold=0.d0
      yold=0.d0
      zold=0.d0
      esur=(efev+wrfu)
c*******************************************************************
      rcosz=sqrt((ee*rcosz*rcosz+esur)/(ee+esur))
      ee=ee+esur
      if (abs(rcosz) .ge. 1.d0) rcosz=rcosz/abs(rcosz)
      if (abs(rcosz) .eq. 0.d0) then
         rcosz=0.d0
         rsinxy=1.d0
         rcosx=rcosx
         rsiny=rsiny
      elseif (abs(rcosz) .eq. 1.d0) then
         rcosz=rcosz
         rsinxy=1.d-8
         rcosx=rcosx
         rsiny=rsiny
      else
         rcosz=rcosz
         rsinxy=sqrt(1.d0-rcosz*rcosz)
         rcosx=rcosx
         rsiny=rsiny
      endif
c*******************************************************************
      xe=0.d0
      ye=0.d0
      ze=0.d0
      xeo=0.d0
      yeo=0.d0
      zeo=0.d0
      eev=ee
      cosz=rcosz
      sinxy=rsinxy
      cosx=rcosx
      siny=rsiny
      rcosz=0.d0
      rsinxy=0.d0
      rcosx=0.d0
      rsiny=0.d0
      peeff=peeff+1
c*******************************************************************
c Electron transport in solid
c     Starting main loop
c*******************************************************************
 500  idnum=0
      diel=0.d0
      xold=x
      yold=y
      zold=z
c*******************************************************************
      deev=eev-esur
      edivf=deev/deie
      ieif=int(edivf)
      idnum=1
      if ((ieif .lt. 10) .or. (deev .lt. 100.d0)
     /    .or. (edivf .lt. 10.d0)) then
```

```
         edivf=deev/deied                                               primar0121
         ieif=int(edivf)                                                primar0122
         idnum=2                                                        primar0123
      endif                                                             primar0124
      if ((idnum .eq. 1) .and. (ieif .ge. outl)) ieif=outl              primar0125
      if ((idnum .eq. 2) .and. (ieif .le. 0)) then                      primar0126
         if ((eev-esur) .le. 0.d0) then                                 primar0127
            goto 120                                                    primar0128
         else                                                           primar0129
            ieif=1                                                      primar0130
         endif                                                          primar0131
      endif                                                             primar0132
c******************************************************************      primar0133
      mfpt=0.d0                                                         primar0134
      elfmfp=0.d0                                                       primar0135
      ve2=2.d0*eev*elv/me                                               primar0136
      tau=0.9d0+exp(-eev/etau)                                          primar0137
      asf=1.d0/4.d0*((tau/atf*dirac/me)**2.d0)/ve2                      primar0138
      elfmfp=asf*(1.d0+asf)*me*ve2/elv*me*ve2/elv                       primar0139
     /       /pi/zt/(zt)/dent/esu2*elv/esu2*elv/angst                   primar0140
      if (idnum .eq. 1) then                                            primar0141
         if (ieif .ge. outl) then                                       primar0142
            iopmfp=opmfp(outl)                                          primar0143
         elseif ((ieif .le. 1) .and. (edivf .lt. 1.d0)) then            primar0144
            iopmfp=opmfp(ieif)                                          primar0145
         else                                                           primar0146
            iopmfp=opmfp(ieif)+(opmfp(ieif+1)-opmfp(ieif))/(1.d0*deie)  primar0147
     /         *(edivf-real(ieif))*deie                                 primar0148
         endif                                                          primar0149
      elseif (idnum .eq. 2) then                                        primar0150
         if (ieif .ge. outdl) then                                      primar0151
            iopmfp=opdmfp(ieif)                                         primar0152
         elseif ((ieif .le. 1) .and. (edivf .lt. 1.d0)) then            primar0153
            iopmfp=opdmfp(ieif)                                         primar0154
         else                                                           primar0155
            iopmfp=opdmfp(ieif)+(opdmfp(ieif+1)-opdmfp(ieif))           primar0156
     /         /(1.d0*deied)*(edivf-real(ieif))*deied                   primar0157
         endif                                                          primar0158
      endif                                                             primar0159
      mfpt=1.d0/(1.d0/elfmfp+1.d0/iopmfp)                               primar0160
  121 call rannum                                                       primar0161
      rm=random1                                                        primar0162
      if ((rm .eq. 0.d0) .or. (rm .eq. 1.d0)) goto 121                  primar0163
      fp=-mfpt*log(rm)                                                  primar0164
      if (((cosx**2)+(siny**2)) .ge. 1.d0) then                         primar0165
         rxy=sqrt(cosx**2+siny**2)                                      primar0166
         cosx=cosx/rxy                                                  primar0167
         siny=siny/rxy                                                  primar0168
         x=x+fp*sinxy*cosx                                              primar0169
         y=y+fp*sinxy*siny                                              primar0170
         z=z+fp*cosz                                                    primar0171
      else                                                              primar0172
         x=x+fp*sinxy*cosx                                              primar0173
         y=y+fp*sinxy*siny                                              primar0174
         z=z+fp*cosz                                                    primar0175
      endif                                                             primar0176
      xe=x                                                              primar0177
      ye=y                                                              primar0178
      ze=z                                                              primar0179
      xeo=xold                                                          primar0180
      yeo=yold
      zeo=zold
      rcosz=cosz
      rsinxy=sinxy
      rcosx=cosx
      rsiny=siny
      ee=eev
      esur=(efev+wrfu)
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
c Electron leaves the metal surface and backscaters into water
      if (ze .le. 0.d0) goto 417
c>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
 1478 call rannum
      rsele=randoml
      call rannum
      rscan=randoml
      call rannum
      razi=randoml
      call rannum
      rgene=randoml
      call rannum
      serazi=randoml
      call rannum
      rseiti=randoml
      deev=eev-esur
      edivf=deev/deie
      ieif=int(edivf)
      idnum=1
      if ((ieif .lt. 10) .or. (deev .lt. 100.d0)
     /     .or. (edivf .lt. 10.d0)) then
         edivf=deev/deied
         ieif=int(edivf)
         idnum=2
      endif
      if ((idnum .eq. 1) .and. (ieif .ge. outl)) ieif=outl
      if ((idnum .eq. 2) .and. (ieif .le. 0)) then
         if ((eev-esur) .le. 0.d0) then
            goto 120
         else
            ieif=1
         endif
      endif
      temp=mfpt/elfmfp
      if (rsele .le. temp) goto 124
      if (rsele .le. (temp+(mfpt/iopmfp))) goto 125
  124 scaan=acos(1.d0-((2.d0*asf*rscan)/(1.d0+asf-rscan)))
      azian=razi*2.d0*pi
      seene=0.d0
      sescaan=0.d0
      seazian=0.d0
      goto 131
  125 if ((idnum .eq. 1) .and. (ieif .ge. outl)) ieif=outl
      temp=ieif*(ieif+1)/2
      istart=temp
      ifinish=temp+ieif
      do 1023 ikm=istart,ifinish
         if (idnum .eq. 1) then
            if (rgene .le. oprnd(ikm)) goto 1022
         elseif (idnum .eq. 2) then
            if (rgene .le. opdrnd(ikm)) goto 1022
```

```
        endif
1023 continue
1022 if (idnum .eq. 1) then
        diel12=real(ikm-istart)*deie
        diel11=real((ikm-1)-istart)*deie
        if ((oprnd(ikm).eq.0.d0) .and. (oprnd(ikm-1).eq.0.d0)) then
        diel1=diel12
        else
        diel1=(diel12-diel11)/(oprnd(ikm)-oprnd(ikm-1))*
    /         (rgene-oprnd(ikm-1))+diel11
        endif
     elseif (idnum .eq. 2) then
        diel12=real(ikm-istart)*deied
        diel11=real((ikm-1)-istart)*deied
        if ((opdrnd(ikm) .eq. 0.d0) .and. (opdrnd(ikm-1) .eq. 0.d0))
    /      then
        diel1=diel12
        else
        diel1=(diel12-diel11)/(opdrnd(ikm)-opdrnd(ikm-1))*
    /         (rgene-opdrnd(ikm-1))+diel11
        endif
     endif
     if (((idnum .eq. 1) .and. (ieif .ge. out1)) .or.
    /    ((idnum .eq. 2) .and. (ieif .le. 1)
    /    .and. (edivf .lt. 1.d0))) then
        diel=diel1
        goto 1124
     endif
     istart=((ieif+1)*((ieif+1)+1)/2)
     ifinish=((ieif+1)*((ieif+1)+1)/2)+(ieif+1)
     do 1123 ikm=istart,ifinish
        if (idnum .eq. 1) then
           if (rgene .le. oprnd(ikm)) goto 1122
        elseif (idnum .eq. 2) then
           if (rgene .le. opdrnd(ikm)) goto 1122
        endif
1123 continue
1122 if (idnum .eq. 1) then
        diel22=real(ikm-istart)*deie
        diel21=real((ikm-1)-istart)*deie
        if ((oprnd(ikm).eq.0.d0) .and. (oprnd(ikm-1).eq.0.d0)) then
        diel2=diel22
        else
        diel2=(diel22-diel21)/(oprnd(ikm)-oprnd(ikm-1))*
    /         (rgene-oprnd(ikm-1))+diel21
        endif
     elseif (idnum .eq. 2) then
        diel22=real(ikm-istart)*deied
        diel21=real((ikm-1)-istart)*deied
        if((opdrnd(ikm).eq.0.d0) .and. (opdrnd(ikm-1).eq.0.d0)) then
        diel2=diel22
        else
        diel2=(diel22-diel21)/(opdrnd(ikm)-opdrnd(ikm-1))*
    /         (rgene-opdrnd(ikm-1))+diel21
        endif
     endif
     if (idnum .eq. 1) then
        diel=diel1+(diel2-diel1)/(1.d0*deie)*(edivf-real(ieif))*deie
     elseif (idnum .eq. 2) then
        diel=diel1+(diel2-diel1)/(1.d0*deied)*(edivf-real(ieif))*deied
```

```
      endif
 1124 if (diel .ge. (eev-esur)) diel=(eev-esur)
      if (diel .le. 0.d0) then
         scaan=0.d0
         sescaan=0.d0
         azian=0.d0
         seazian=0.d0
         eev=eev
         seene=0.d0
      else
         if (diel .ge. eev) then
            scaan=acos(0.d0)
         else
            scaan=acos(sqrt((eev-diel)/eev))
         endif
         sescaan=acos(sqrt(diel/eev))
         azian=razi*2.d0*pi
         seazian=2.d0*serazi*pi
         eev=eev-diel
         if (eev .le. 0.d0) eev=0.d0
         call rannum
         rge=random1
         eeg=efev*(rge**(2.d0/3.d0))
         seene=diel+efev-eeg
      endif
  132 if (seene .le. esur) then
         sescaan=0.d0
         seazian=0.d0
         seene=0.d0
         goto 131
      endif
c*******************************************************************
c Create a new secondary electron!!
c*******************************************************************
      xse=xold+fp*sinxy*cosx
      yse=yold+fp*sinxy*siny
      zse=zold+fp*cosz
      ssinxy=sinxy
      scosx=cosx
      ssiny=siny
      scosz=cosz
      prim=1
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      call secondary
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      ise=0
      ssinxy=0.d0
      scosx=0.d0
      ssiny=0.d0
      scosz=0.d0
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
c electron stopping
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
  131 temp1=sin(scaan)
      temp2=cos(azian)
      temp3=sin(azian)
      temp4=cos(scaan)
      uan=(temp1*temp2*cosz+temp4*sinxy)
      van=(temp3*temp1)
      bcosx=uan*cosx-van*siny
      bsiny=uan*siny+van*cosx                                           primar0361
      cosz=cosz*temp4-sinxy*temp1*temp2                                 primar0362
      if (abs(cosz) .ge. 1.d0) cosz=cosz/abs(cosz)                      primar0363
      if (abs(cosz) .eq. 0.d0) then                                     primar0364
         cosz=0.d0                                                      primar0365
         sinxy=1.d0                                                     primar0366
         bcosx=cosx*sinxy                                               primar0367
         bsiny=siny*sinxy                                               primar0368
         cosx=cosx                                                      primar0369
         siny=siny                                                      primar0370
      elseif (abs(cosz) .eq. 1.d0) then                                 primar0371
         cosz=cosz                                                      primar0372
         sinxy=1.d-8                                                    primar0373
         bcosx=cosx*sinxy                                               primar0374
         bsiny=siny*sinxy                                               primar0375
         cosx=cosx                                                      primar0376
         siny=siny                                                      primar0377
      else                                                              primar0378
         cosz=cosz                                                      primar0379
         sinxy=sqrt(1.d0-cosz*cosz)                                     primar0380
         cosx=cosx                                                      primar0381
         siny=siny                                                      primar0382
      endif                                                             primar0383
      normal=(cosz*cosz+bcosx*bcosx+bsiny*bsiny)                        primar0384
      if (normal .ge. 1.d0) then                                        primar0385
         cosz=cosz/sqrt(normal)                                         primar0386
         bcosx=bcosx/sqrt(normal)                                       primar0387
         bsiny=bsiny/sqrt(normal)                                       primar0388
         sinxy=sinxy/sqrt(normal)                                       primar0389
      endif                                                             primar0390
      cosx=bcosx/sinxy                                                  primar0391
      siny=bsiny/sinxy                                                  primar0392
      rxy=sqrt(cosx*cosx+siny*siny)                                     primar0393
      cosx=cosx/rxy                                                     primar0394
      siny=siny/rxy                                                     primar0395
      bcosx=cosx*sinxy                                                  primar0396
      bsiny=siny*sinxy                                                  primar0397
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< primar0398
c Electron still has energy, keep going....                             primar0399
      if (eev .gt. esur+0.01d0) goto 500                                primar0400
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< primar0401
c Primary electron completely stops, return                             primar0402
      if (eev .le. esur+0.01d0) goto 120                                primar0403
c*********************************************************************** primar0404
c*********************************************************************** primar0405
c Electron leaves the metal surface and backscaters into vacuum         primar0406
c*********************************************************************** primar0407
  417 xe=x                                                              primar0408
      ye=y                                                              primar0409
      ze=z                                                              primar0410
      xeo=xold                                                          primar0411
      yeo=yold                                                          primar0412
      zeo=zold                                                          primar0413
      rcosz=cosz                                                        primar0414
      rsinxy=sinxy                                                      primar0415
      rcosx=cosx                                                        primar0416
      rsiny=siny                                                        primar0417
      ee=eev                                                            primar0418
      esur=(efev+wrfu)                                                  primar0419
      if(abs(rcosz) .ge. 1.d0) rcosz=rcosz/abs(rcosz)                   primar0420
      egamma=pi-acos(rcosz)                                             primar0421
      if ((ee*rcosz*rcosz) .ge. esur) then                              primar0422
         eevac=ee-esur                                                  primar0423
         refra=sin(egamma)*sqrt(ee/eevac)                               primar0424
         refra=asin(refra)                                              primar0425
         if (rcosx .le. 0.d0) refra=-refra                              primar0426
         ee=eevac                                                       primar0427
         ibp=ibp+1                                                      primar0428
         eevvac=ee                                                      primar0429
         exitan=abs(refra)/pi*180.d0                                    primar0430
         ied=int(eevvac/1.d0)                                           primar0431
         if (ied .gt. 10000) then                                       primar0432
            iedatb(10001)=iedatb(10001)+1                               primar0433
         else                                                           primar0434
            iedatb(ied)=iedatb(ied)+1                                   primar0435
         endif                                                          primar0436
         iad=int(exitan/5.d0)                                           primar0437
         if (rcosx .le. 0.d0) iad=19-(iad+1)                            primar0438
         if (rcosx .gt. 0.d0) iad=(iad+1)+18                            primar0439
         if (iad .le. 0) iad=0                                          primar0440
         if (iad .gt. 36) iad=37                                        primar0441
         iandatb(iad)=iandatb(iad)+1                                    primar0442
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<    primar0443
         if (0 .eq. mod(peeff,1000)) then                               primar0444
            write(6,101) peeff                                          primar0445
            write(6,191) ibp                                            primar0446
            write(6,801) eevvac                                         primar0447
            write(6,811) exitan                                         primar0448
            write(6,1911) ips                                           primar0449
         endif                                                          primar0450
  101    format(1x,'primary electron number',i10,'(particles)')         primar0451
  191    format(1x,'escape primary electron number',i10,'(particles)')  primar0452
  801    format(1x,'escape primary electron energy',f10.3,' (eV)')      primar0453
  811    format(1x,'escape primary electron angle',f7.2,' (degree)')    primar0454
 1911    format(1x,'total escape secondary number',i10,'(particles)')   primar0455
      endif                                                             primar0456
c*******************************************************************    primar0457
c Electron completely stops                                             primar0458
c*******************************************************************    primar0459
  120 return                                                            primar0460
      end                                                               primar0461
```

### 3.2 Subroutine “secondary”

Basically, this program is similar to the program “primary”. It calculates the transport of SEs, produced on the trajectory of a penetrating PE, and their ejection into the vacuum. The program is called from the subroutine “primary”
Different points from the program “primary” are presented here.

Line 0052 The valuable “ise” is the index which corresponds to the number of SEs produced by the cascade multiplication. Initially, ise=0.

Line 0054-0056 The excitation energy and angle of SE produced by a primary electron are set.

Line 0349-0362 If an SE generates another SE (i.e., a cascade electron), the index ise is increased by one and the excited position and energy and the initial polar and azimuthal angle are stored.

Line 0501-0503 If ise=0, the calculation stops and returns to the subroutine program “primary.f”.

Othewise, the information of another SE stored in Line 0479-0488 is recalled and the calculation jumps to the trajectory simulation. The index ise is decreased by one.

```
c***********************************************************************
c*****      Transport of secondary electrons in solids         *******
c***********************************************************************
      subroutine secondary
c***** direct monte carlo simulation for electron scattering *******
c***********************************************************************
c***********************************************************************
      implicit real*8 (a-h,o-z)
      implicit integer*4 (i,j,k,l,m,n)
c***********************************************************************
      dimension iedatb(0:100001),iandatb(0:37)
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      dimension oprnd(0:550000),opmfp(0:1001)
      dimension opdrnd(0:5500),opdmfp(0:101)
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      dimension gxse(10000),gyse(10000),gzse(10000)
      dimension gsca(10000),gazi(10000)
      dimension gene(10000),gcosx(10000),gsiny(10000),gsinxy(10000)
      dimension gcosz(10000)
c***********************************************************************
      common /ransu/random1
      common /tane/seed1,seed2,seed3
      common /posi/xse,yse,zse/dire/ssinxy,scosx,ssiny,scosz
      common /sanran/sescaan,seazian,seene
      common /map2/oprnd,opmfp
      common /map8/opdrnd,opdmfp
      common /con1/efev,plaev,wrfu
      common /con2/dene,deie,out1,out2
      common /con3/deied,outd1,outd2
      common /con4/dened
      common /flag1/ips,ipar,peeff
      common /flag5/ise
      common /adse1/gxse,gyse,gzse,gsca,gazi,gene
      common /adse2/gcosx,gsiny,gsinxy,gcosz
      common /plapro/conpla
      common /fitting/atf,etau,zt,dent
      common /hant1/ee,esur,rcosz,rsinxy,rcosx,rsiny,refra
      common /hant2/xe,ye,ze,xeo,yeo,zeo
      common /trabac/itraj,ibp,it,itr,iedatb,iandatb
c***********************************************************************
      real*8 me,mfpt,iopmfp,normal
      integer*4 seed1,seed2,seed3
      integer*4 out1,out2,outd1,outd2,peeff
c***********************************************************************
      parameter(pi=3.141592654d0,angst=1.d-8,abohr=0.529d-8)
      parameter(esu2=2.30711d-19,dirac=1.05459d-27)
      parameter(me=9.10953d-28,elv=1.60219d-12)
c***********************************************************************
      dirac2=dirac/elv
c***********************************************************************
c Start transporting a new secondary electron
c         ise: secondary electron index
c***********************************************************************
  489 e00ev=seene
      scaan=sescaan
      azian=seazian
      x=xse
      y=yse
      z=zse
      xold=x
```

```
yoldy=y                                                                 second0061
zold=z                                                                  second0062
eev=e00ev                                                               second0063
sinxy=ssinxy                                                            second0064
siny=ssiny                                                              second0065
cosx=scosx                                                              second0066
cosz=scosz                                                              second0067
exsev=seene                                                             second0068
idnum=0                                                                 second0069
xe=x                                                                    second0070
ye=y                                                                    second0071
ze=z                                                                    second0072
xeo=xold                                                                second0073
yeo=yold                                                                second0074
zeo=zold                                                                second0075
rcosz=cosz                                                              second0076
rsinxy=sinxy                                                            second0077
rcosx=cosx                                                              second0078
rsiny=siny                                                              second0079
ee=eev                                                                  second0080
esur=(efev+wrfu)                                                        second0081
uan=(sin(scaan)*cos(azian)*cosz+cos(scaan)*sinxy)                       second0082
van=(sin(azian)*sin(scaan))                                             second0083
bcosx=uan*cosx-van*siny                                                 second0084
bsiny=uan*siny+van*cosx                                                 second0085
cosz=cosz*cos(scaan)-sinxy*sin(scaan)*cos(azian)                        second0086
if (abs(cosz) .ge. 1.d0) cosz=cosz/abs(cosz)                            second0087
if (abs(cosz) .eq. 1.d0) then                                           second0088
   sinxy=1.d-8                                                          second0089
elseif (abs(cosz) .eq. 0.d0) then                                       second0090
   sinxy=1.d0                                                           second0091
else                                                                    second0092
   sinxy=sqrt(1.d0-cosz**2)                                             second0093
endif                                                                   second0094
cosx=bcosx/sinxy                                                        second0095
siny=bsiny/sinxy                                                        second0096
if (abs(cosz) .ge. 1.d0) cosz=cosz/abs(cosz)                            second0097
if (abs(cosz) .eq. 0.d0) then                                           second0098
   cosz=0.d0                                                            second0099
   sinxy=1.d0                                                           second0100
   bcosx=cosx*sinxy                                                     second0101
   bsiny=siny*sinxy                                                     second0102
   cosx=cosx                                                            second0103
   siny=siny                                                            second0104
elseif (abs(cosz) .eq. 1.d0) then                                       second0105
   cosz=cosz                                                            second0106
   sinxy=1.d-8                                                          second0107
   bcosx=cosx*sinxy                                                     second0108
   bsiny=siny*sinxy                                                     second0109
   cosx=cosx                                                            second0110
   siny=siny                                                            second0111
else                                                                    second0112
   cosz=cosz                                                            second0113
   sinxy=sqrt(1.d0-cosz*cosz)                                           second0114
   bcosx=cosx*sinxy                                                     second0115
   bsiny=siny*sinxy                                                     second0116
   cosx=cosx                                                            second0117
   siny=siny                                                            second0118
endif                                                                   second0119
normal=(cosz*cosz+bcosx*bcosx+bsiny*bsiny)                              second0120
```

```
      if (normal .ge. 1.d0) then
         cosz=cosz/sqrt(normal)
         bcosx=bcosx/sqrt(normal)
         bsiny=bsiny/sqrt(normal)
         sinxy=sinxy/sqrt(normal)
      endif
      cosx=bcosx/sinxy
      siny=bsiny/sinxy
      cosx=cosx/sqrt(cosx*cosx+siny*siny)
      siny=siny/sqrt(cosx*cosx+siny*siny)
      bcosx=cosx*sinxy
      bsiny=siny*sinxy
c********************************************************************
c Transport current secondary electron
c********************************************************************
 500  idnum=0
      diel=0.d0
      xold=x
      yold=y
      zold=z
      ieif=int((eev-efev)/deie)
      edivf=((eev-efev)/deie)
      idnum=1
      if ((ieif .lt. 10) .or. ((eev-efev) .lt. 100.d0)
     /    .or. (edivf .lt. 10.d0)) then
         ieif=int((eev-efev)/deied)
         edivf=((eev-efev)/deied)
         idnum=2
      endif
      if ((idnum .eq. 2) .and. (ieif .le. 0)) then
         if ((eev-efev) .le. 0.d0) then
            goto 120
         else
            ieif=1
         endif
      endif
      mfpt=0.d0
      elfmfp=0.d0
      ve2=2.d0*eev*elv/me
      tau=0.9d0+exp(-eev/etau)
      asf=1.d0/4.d0*((tau/atf*dirac/me)**2.d0)/ve2
      elfmfp=asf*(1.d0+asf)*me*ve2/elv*me*ve2/elv
     /        /pi/zt/(zt)/dent/esu2*elv/esu2*elv/angst
      if (idnum .eq. 1) then
         if (ieif .ge. outl) then
            iopmfp=opmfp(outl)
         elseif ((ieif .le. 1) .and. (edivf .lt. 1.d0)) then
            iopmfp=opmfp(ieif)
         else
            iopmfp=opmfp(ieif)+(opmfp(ieif+1)-opmfp(ieif))/(1.d0*deie)
     /          *(edivf-real(ieif))*deie
         endif
      elseif (idnum .eq. 2) then
         if (ieif .ge. outdl) then
            iopmfp=opdmfp(outdl)
         elseif ((ieif .le. 1) .and. (edivf .lt. 1.d0)) then
            iopmfp=opdmfp(ieif)
         else
            iopmfp=opdmfp(ieif)+(opdmfp(ieif+1)-opdmfp(ieif))
     /          /(1.d0*deied)*(edivf-real(ieif))*deied
```

```
         endif
      endif
      mfpt=1.d0/(1.d0/elfmfp+1.d0/iopmfp)
  121 call rannum
      rm=random1
      if ((rm .eq. 0.) .or. (rm .eq. 1.d0)) goto 121
      fp=-mfpt*log(rm)
      if (((cosx**2)+(siny**2)) .ge. 1.d0) then
         cosx=cosx/sqrt(cosx**2+siny**2)
         siny=siny/sqrt(cosx**2+siny**2)
         x=x+fp*sinxy*cosx
         y=y+fp*sinxy*siny
         z=z+fp*cosz
      else
         x=x+fp*sinxy*cosx
         y=y+fp*sinxy*siny
         z=z+fp*cosz
      endif
      xe=x
      ye=y
      ze=z
      xeo=xold
      yeo=yold
      zeo=zold
      rcosz=cosz
      rsinxy=sinxy
      rcosx=cosx
      rsiny=siny
      ee=eev
      esur=(efev+wrfu)
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
c Electron leaves the metal surface and backscaters into water
      if (ze .le. 0.d0) goto 417
c>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
 1478 call rannum
      rsele=random1
      call rannum
      rscan=random1
      call rannum
      razi=random1
      call rannum
      rgene=random1
      call rannum
      serazi=random1
      call rannum
      rseiti=random1
      ieif=int((eev-efev)/deie)
      ediv2=((eev-efev)/deie)
      idnum=1
      if ((ieif .lt. 10) .or. ((eev-efev) .lt. 100.d0)
     /    .or. (ediv2 .lt. 10.d0)) then
         ieif=int((eev-efev)/deied)
         ediv2=((eev-efev)/deied)
         idnum=2
      endif
      if ((idnum .eq. 1) .and. (ieif .ge. out1)) ieif=out1
      if ((idnum .eq. 2) .and. (ieif .le. 0)) then
         if ((eev-efev) .le. 0.d0) then
            goto 120
         else
```

```
            ieif=1                                                      second0241
         endif                                                          second0242
      endif                                                             second0243
      temp=mfpt/elfmfp                                                  second0244
      if (rsele .le. temp) goto 124                                     second0245
      if (rsele .le. (temp+(mfpt/iopmfp))) goto 125                     second0246
  124 scaan=acos(1.d0-((2.d0*asf*rscan)/(1.d0+asf-rscan)))              second0247
      azian=razi*2.d0*pi                                                second0248
      seene=0.d0                                                        second0249
      sescaan=0.d0                                                      second0250
      seazian=0.d0                                                      second0251
      goto 131                                                          second0252
  125 if ((idnum .eq. 1) .and. (ieif .ge. outl)) ieif=outl              second0253
      temp=ieif*(ieif+1)/2                                              second0254
      istart=temp                                                       second0255
      ifinish=temp+ieif                                                 second0256
      do 1023 ikm=istart,ifinish                                        second0257
         if (idnum .eq. 1) then                                         second0258
            if (rgene .le. oprnd(ikm)) goto 1022                        second0259
         elseif (idnum .eq. 2) then                                     second0260
            if (rgene .le. opdrnd(ikm)) goto 1022                       second0261
         endif                                                          second0262
 1023 continue                                                          second0263
 1022 if (idnum .eq. 1) then                                            second0264
         diel12=real(ikm-istart)*deie                                   second0265
         diel11=real((ikm-1)-istart)*deie                               second0266
         diel1=(diel12-diel11)/(oprnd(ikm)-oprnd(ikm-1))*               second0267
     /         (rgene-oprnd(ikm-1))+diel11                              second0268
      elseif (idnum .eq. 2) then                                        second0269
         diel12=real(ikm-istart)*deied                                  second0270
         diel11=real((ikm-1)-istart)*deied                              second0271
         if ((opdrnd(ikm).eq.0.d0) .and. (opdrnd(ikm-1).eq.0.d0)) then  second0272
         diel1=diel12                                                   second0273
         else                                                           second0274
         diel1=(diel12-diel11)/(opdrnd(ikm)-opdrnd(ikm-1))*             second0275
     /         (rgene-opdrnd(ikm-1))+diel11                             second0276
         endif                                                          second0277
      endif                                                             second0278
      if (((idnum .eq. 1) .and. (ieif .ge. outl)) .or.                  second0279
     /    ((idnum .eq. 2) .and. (ieif .le. 1)                           second0280
     /    .and. (edivf .lt. 1.d0))) then                                second0281
         diel=diel1                                                     second0282
         goto 1124                                                      second0283
      endif                                                             second0284
      istart=((ieif+1)*((ieif+1)+1)/2)                                  second0285
      ifinish=((ieif+1)*((ieif+1)+1)/2)+(ieif+1)                        second0286
      do 1123 ikm=istart,ifinish                                        second0287
         if (idnum .eq. 1) then                                         second0288
            if (rgene .le. oprnd(ikm)) goto 1122                        second0289
         elseif (idnum .eq. 2) then                                     second0290
            if (rgene .le. opdrnd(ikm)) goto 1122                       second0291
         endif                                                          second0292
 1123 continue                                                          second0293
 1122 if (idnum .eq. 1) then                                            second0294
         diel22=real(ikm-istart)*deie                                   second0295
         diel21=real((ikm-1)-istart)*deie                               second0296
         diel2=(diel22-diel21)/(oprnd(ikm)-oprnd(ikm-1))*               second0297
     /         (rgene-oprnd(ikm-1))+diel21                              second0298
      elseif (idnum .eq. 2) then                                        second0299
         diel22=real(ikm-istart)*deied                                  second0300
         diel21=real((ikm-1)-istart)*deied
         if ((opdrnd(ikm).eq.0.d0) .and. (opdrnd(ikm-1).eq.0.d0)) then
         diel2=diel22
         else
         diel2=(diel22-diel21)/(opdrnd(ikm)-opdrnd(ikm-1))*
     /         (rgene-opdrnd(ikm-1))+diel21
         endif
      endif
      if (idnum .eq. 1) then
         diel=diel1+(diel2-diel1)/(1.d0*deie)*(edivf-real(ieif))*deie
      elseif (idnum .eq. 2) then
         diel=diel1+(diel2-diel1)/(1.d0*deied)*(edivf-real(ieif))*deied
      endif
 1124 if (diel .ge. (eev-efev)) diel=(eev-efev)
      if (diel .le. 0.d0) then
         scaan=0.d0
         sescaan=0.d0
         azian=0.d0
         seazian=0.d0
         eev=eev
         seene=0.d0
      else
         if (diel .ge. eev) then
            scaan=acos(0.d0)
         else
            scaan=acos(sqrt((eev-diel)/eev))
         endif
         sescaan=acos(sqrt(diel/eev))
         azian=razi*2.d0*pi
         seazian=2.d0*serazi*pi
         eev=eev-diel
         if (eev .le. 0.d0) eev=0.d0
         call rannum
         rge=random1
         eeg=efev*(rge**(2./3.))
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
         seene=diel+efev-eeg
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      endif
  132 if (seene .le. esur) then
         sescaan=0.d0
         seazian=0.d0
         seene=0.d0
         goto 131
      endif
c*******************************************************************
c Create a new secondary electron!!
c*******************************************************************
      xse=xold+fp*sinxy*cosx
      yse=yold+fp*sinxy*siny
      zse=zold+fp*cosz
      ise=ise+1
      gxse(ise)=xse
      gyse(ise)=yse
      gzse(ise)=zse
      gsca(ise)=sescaan
      gazi(ise)=seazian
      gene(ise)=seene
      gcosx(ise)=cosx
      gsiny(ise)=siny
      gsinxy(ise)=sinxy                                                     second0361
      gcosz(ise)=cosz                                                       second0362
  131 temp1=sin(scaan)                                                      second0363
      temp2=cos(azian)                                                      second0364
      temp3=sin(azian)                                                      second0365
      temp4=cos(scaan)                                                      second0366
      uan=(temp1*temp2*cosz+temp4*sinxy)                                    second0367
      van=(temp3*temp1)                                                     second0368
      bcosx=uan*cosx-van*siny                                               second0369
      bsiny=uan*siny+van*cosx                                               second0370
      cosz=cosz*temp4-sinxy*temp1*temp2                                     second0371
      if (abs(cosz) .ge. 1.d0) cosz=cosz/abs(cosz)                          second0372
      if (abs(cosz) .eq. 1.d0) then                                         second0373
         sinxy=1.d-8                                                        second0374
      elseif (abs(cosz) .eq. 1.d-18) then                                   second0375
         sinxy=1.d0                                                         second0376
      else                                                                  second0377
         sinxy=sqrt(1.d0-cosz**2)                                           second0378
      endif                                                                 second0379
      cosx=bcosx/sinxy                                                      second0380
      siny=bsiny/sinxy                                                      second0381
      if (abs(cosz) .ge. 1.d0) cosz=cosz/abs(cosz)                          second0382
      if (abs(cosz) .eq. 0.d0) then                                         second0383
         cosz=0.d0                                                          second0384
         sinxy=1.d0                                                         second0385
         bcosx=cosx*sinxy                                                   second0386
         bsiny=siny*sinxy                                                   second0387
         cosx=cosx                                                          second0388
         siny=siny                                                          second0389
      elseif (abs(cosz) .eq. 1.d0) then                                     second0390
         cosz=cosz                                                          second0391
         sinxy=1.d-8                                                        second0392
         bcosx=cosx*sinxy                                                   second0393
         bsiny=siny*sinxy                                                   second0394
         cosx=cosx                                                          second0395
         siny=siny                                                          second0396
      else                                                                  second0397
         cosz=cosz                                                          second0398
         sinxy=sqrt(1.d0-cosz*cosz)                                         second0399
         cosx=cosx                                                          second0400
         siny=siny                                                          second0401
      endif                                                                 second0402
      normal=(cosz*cosz+bcosx*bcosx+bsiny*bsiny)                            second0403
      if (normal .ge. 1.d0) then                                            second0404
         cosz=cosz/sqrt(normal)                                             second0405
         bcosx=bcosx/sqrt(normal)                                           second0406
         bsiny=bsiny/sqrt(normal)                                           second0407
         sinxy=sinxy/sqrt(normal)                                           second0408
      endif                                                                 second0409
      cosx=bcosx/sinxy                                                      second0410
      siny=bsiny/sinxy                                                      second0411
      rxy=sqrt(cosx*cosx+siny*siny)                                         second0412
      cosx=cosx/rxy                                                         second0413
      siny=siny/rxy                                                         second0414
      bcosx=cosx*sinxy                                                      second0415
      bsiny=siny*sinxy                                                      second0416
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     second0417
c Current electron still has energy, keep going....                         second0418
      if (eev .gt. esur+0.01d0) goto 500                                    second0419
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     second0420
```

```
c Electron completely stops.
      if (eev .le. esur+0.01d0) goto 120
c********************************************************************
c********************************************************************
c Electron leaves the metal surface and backscaters into water
c********************************************************************
  417 xe=x
      ye=y
      ze=z
      xeo=xold
      yeo=yold
      zeo=zold
      rcosz=cosz
      rsinxy=sinxy
      rcosx=cosx
      rsiny=siny
      ee=eev
      esur=(efev+wrfu)
      if (abs(rcosz) .ge. 1.d0) rcosz=rcosz/abs(rcosz)
      egamma=pi-acos(rcosz)
      if ((ee*rcosz*rcosz) .ge. esur) then
         eevac=ee-esur
         refra=sin(egamma)*sqrt(ee/eevac)
         refra=asin(refra)
         if (rcosx .le. 0.d0) refra=-refra
         ee=eevac
         ips=ips+1
         seevvac=ee
         sexitan=(abs(refra))/pi*180.d0
         ied=int(seevvac/1.d0)
         if (ied .gt. 10000) then
            iedatb(10001)=iedatb(10001)+1
         else
            iedatb(ied)=iedatb(ied)+1
         endif
         iad=int(sexitan/5.d0)
         if (rcosx .le. 0.d0) iad=19-(iad+1)
         if (rcosx .gt. 0.d0) iad=(iad+1)+18
         if (iad .le. 0) iad=0
         if (iad .gt. 36) iad=37
         iandatb(iad)=iandatb(iad)+1
         if (0 .eq. mod(peeff,1000)) then
            write(6,101) peeff
            write(6,111) ips
            write(6,801) seevvac
            write(6,811) sexitan
  101       format(1x,'primary electron number',i10,'(particles)')
  111       format(1x,'total escape secondary number',i10,'(particles)')
  801       format(1x,'escape secondary electron energy',f10.3,'(eV)')
  811       format(1x,'escape secondary electron angle',f7.2,'(degree)')
         endif
      endif
c********************************************************************
c Current electron completely stops, check next
c********************************************************************
c if there are no more secondary electrons, stop and return
  120 if (ise .eq. 0) goto 143
c check next secondary electron
  168 xse=gxse(ise)
      yse=gyse(ise)
      zse=gzse(ise)
      sescaan=gsca(ise)
      seazian=gazi(ise)
      seene=gene(ise)
      scosx=gcosx(ise)
      ssiny=gsiny(ise)
      ssinxy=gsinxy(ise)
      scosz=gcosz(ise)
      gxse(ise)=0.d0
      gyse(ise)=0.d0
      gzse(ise)=0.d0
      gsca(ise)=0.d0
      gazi(ise)=0.d0
      gene(ise)=0.d0
      gcosx(ise)=0.d0
      gsiny(ise)=0.d0
      gsinxy(ise)=0.d0
      gcosz(ise)=0.d0
      ise=ise-1
c If there are more s.e. but energy is too low, get next
      if ((ise .ne. 0) .and. (seene .le. esur)) goto 168
c If there are no more electrons, finish
      if ((ise .eq. 0) .and. (seene .le. 0.d0)) goto 143
c otherwise, keep going
      goto 489
  143 return
      end
```

### 3.3 Subroutine “rannum”

The program is a random number generator. The inputs are three integers, seed1, seed2 and seed3, for the seeds. The output is a real number in double precision, random1. The users can replace the program with any other random number generator.

```
c*******************************************************************
c***** Randum Number Ganerator *************************************
c*******************************************************************
c*******************************************************************
      subroutine rannum
c*******************************************************************
c*******************************************************************
      implicit real*8 (a-h,o-z)
      implicit integer*4 (i,j,k,l,m,n)
c*******************************************************************
      common /ransu/random1
      common /tane/seed1,seed2,seed3
c      save /tane/
c*******************************************************************
      real*8 r,random1
      integer*4 seed1,seed2,seed3
c*******************************************************************
c*******************************************************************
      seed1=171*mod(seed1,177)- 2*int(seed1/177)
      if (seed1 .lt. 0) then
         seed1=seed1+30269
      endif
      seed2=172*mod(seed2,176)-35*int(seed2/176)
      if (seed2 .lt. 0) then
         seed2=seed2+30307
      endif
      seed3=170*mod(seed3,178)-63*int(seed3/178)
      if (seed3 .lt. 0) then
         seed3=seed3+30323
      endif
      r=seed1/30269.d0 +seed2/30307.d0+seed3/30323.d0
      if (r .ge. 1.0d0) then
         r=r-int(r)
      endif
      random1=r
      return
      end
```

# 4. Examples of calculation results

## 4.1 Dependence of electron backscattering coefficient and secondary electron emission yield on the primary electron energy

Although backscattered electrons are experimentally determined as having exit energies of $E$>50 eV, the calculated backscattering coefficient (i.e., the ratio of backscattered primary electrons to incident primary electrons) involves electrons with the energies of $E$<50 eV. The calculated secondary electron yield (i.e., the ratio of all secondary electrons to incident primary electrons) also involves not only low-energy secondary electrons (<50 eV) but also high-energy secondary electrons (>50 eV) unlike the standard practice.

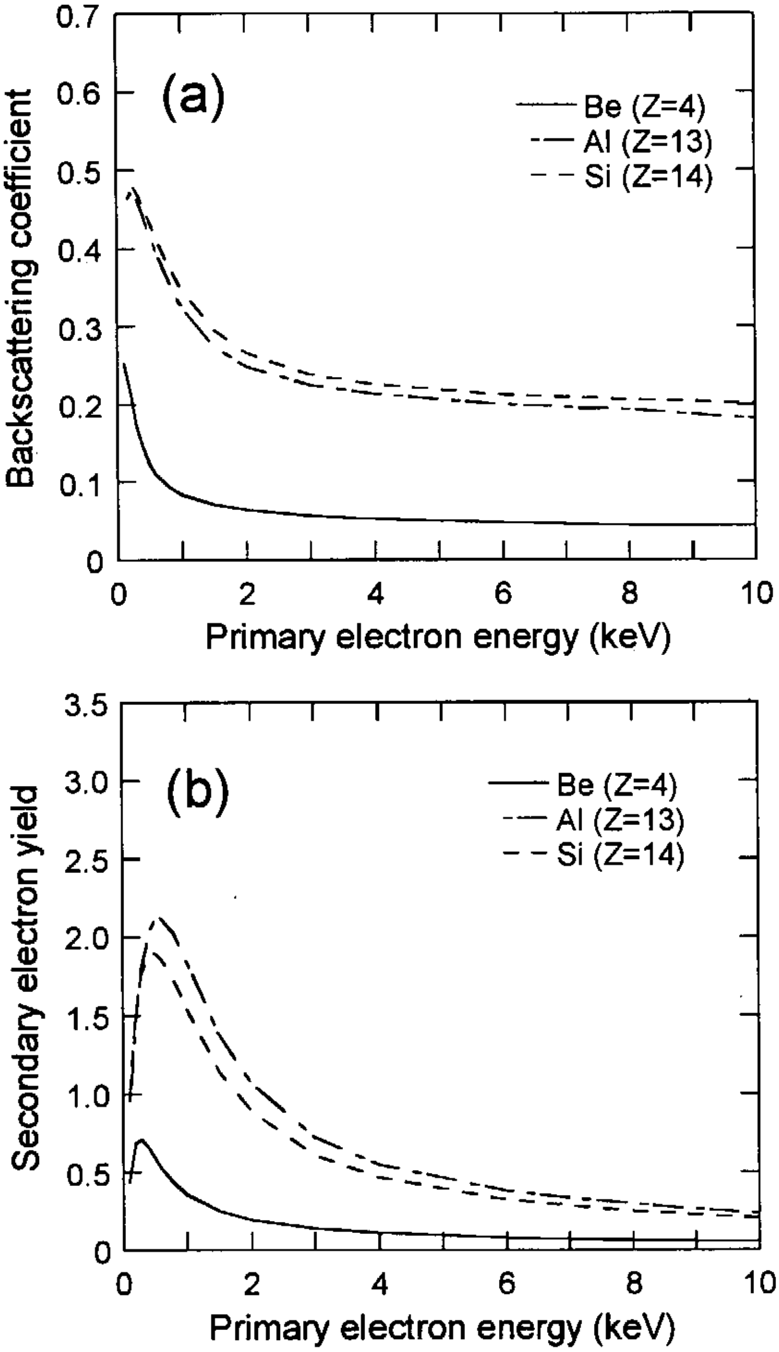

**Figure 4.** Calculated (a) backscattering coefficients and (b) secondary electron yields of Be, Al and Si.

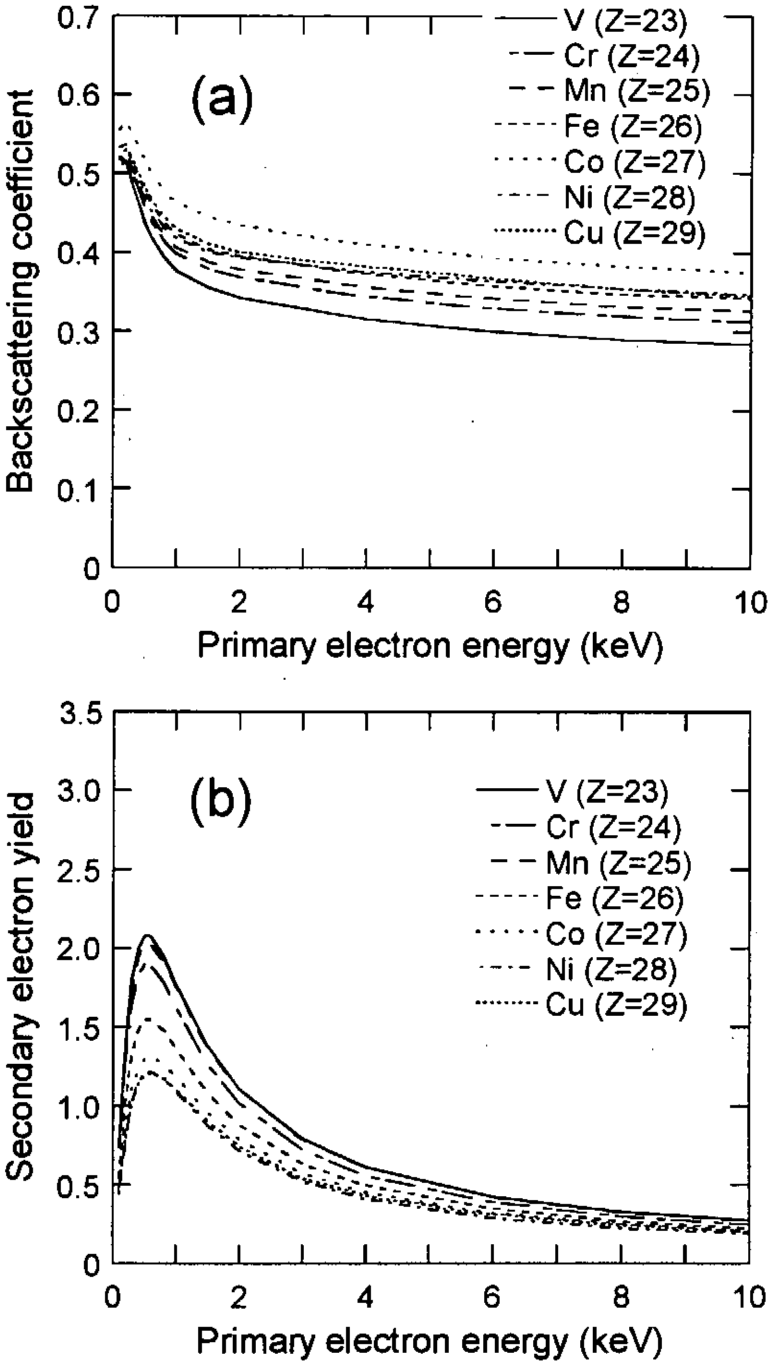

**Figure 5.** Calculated (a) backscattering coefficients and (b) secondary electron yields of V, Cr, Mn, Fe, Co, Ni and Cu.

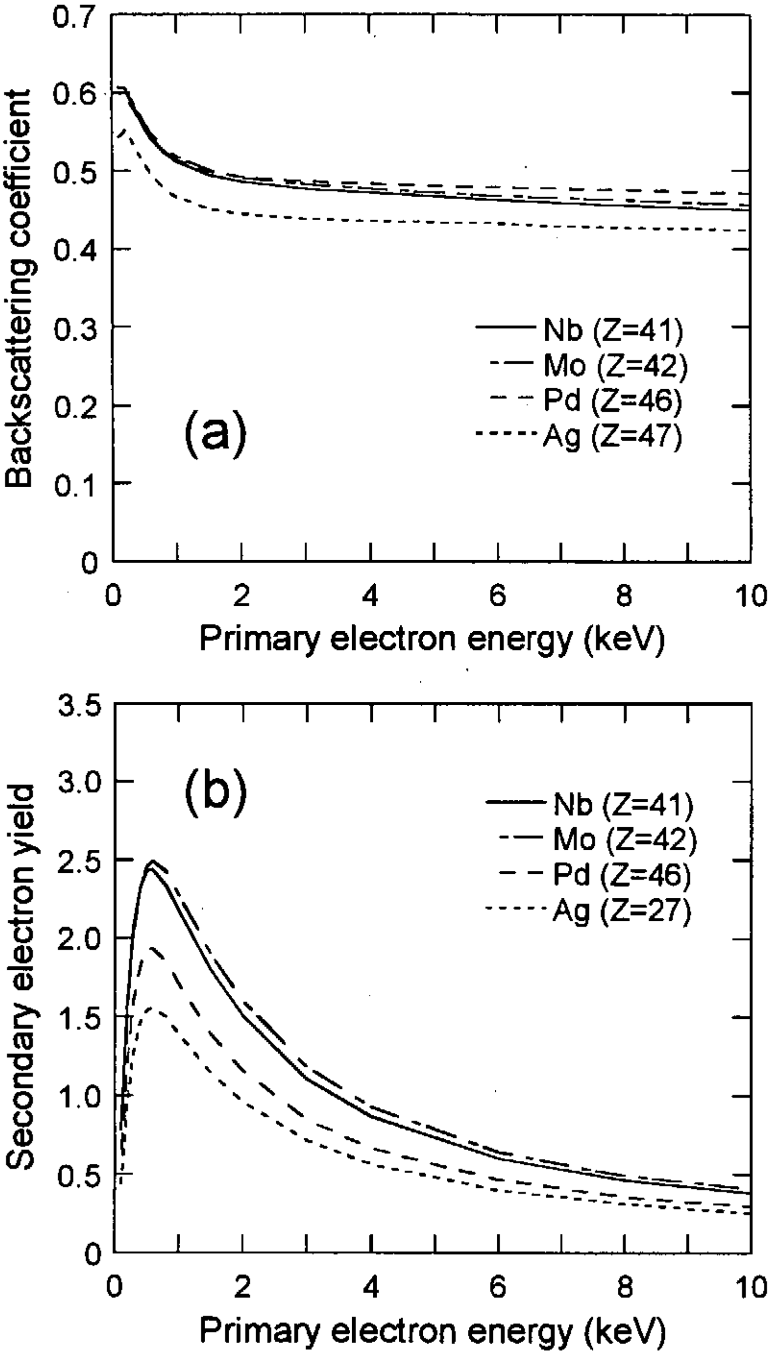

**Figure 6.** Calculated (a) backscattering coefficients and (b) secondary electron yields of Nb, Mo, Pd and Ag.

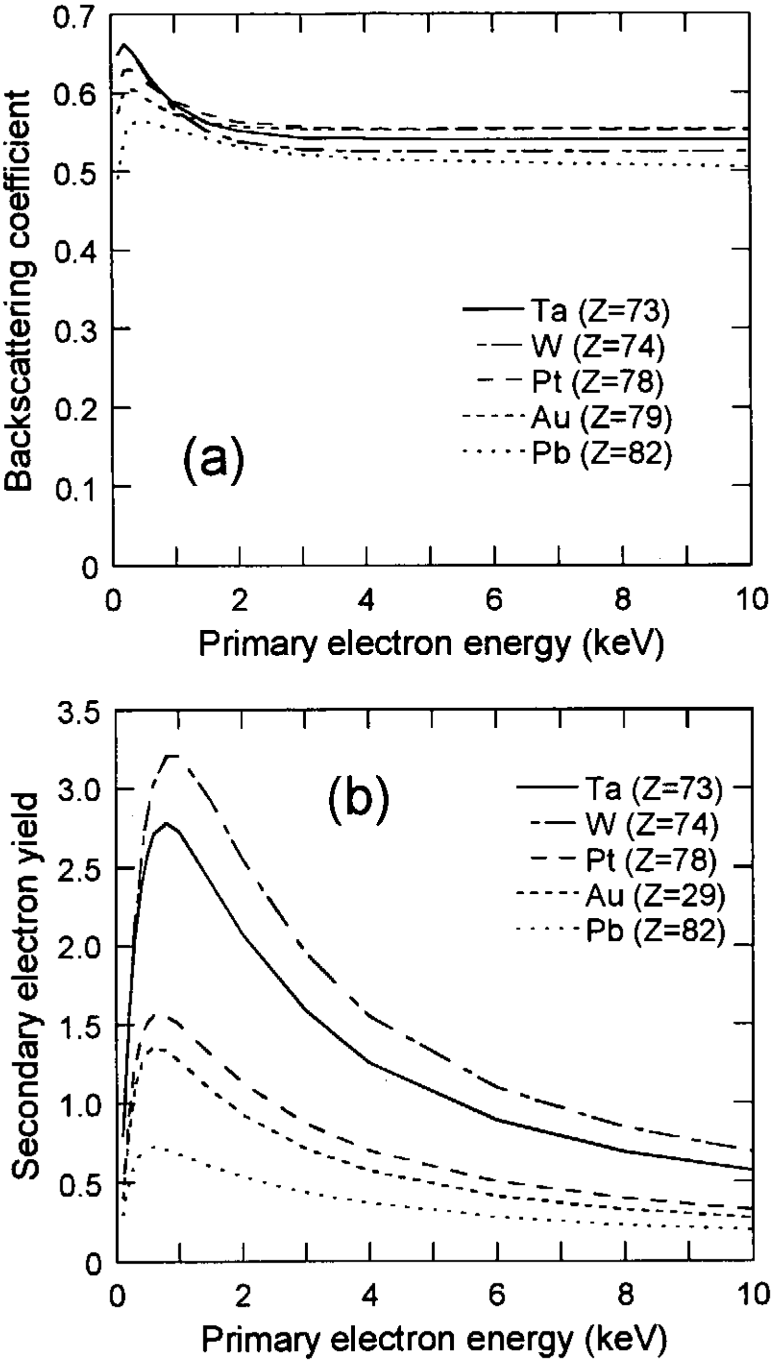

**Figure 7.** Calculated (a) backscattering coefficients and (b) secondary electron yields of Ta, W, Pt, Au and Pb.

### 4.2 Energy distributions of emitted electrons

The energy distributions are composed of two parts. The first part, corresponding to backscattered electrons, includes an elastic peak at the primary energy and a large background (sometimes only this background is called backscattered electrons). The second part is a secondary electron peak produced by primary electrons (it's also called "true" secondary electrons to distinguish with those backscattered electrons). In Fig. 8 the electron yield is normalized at the secondary electron peak.

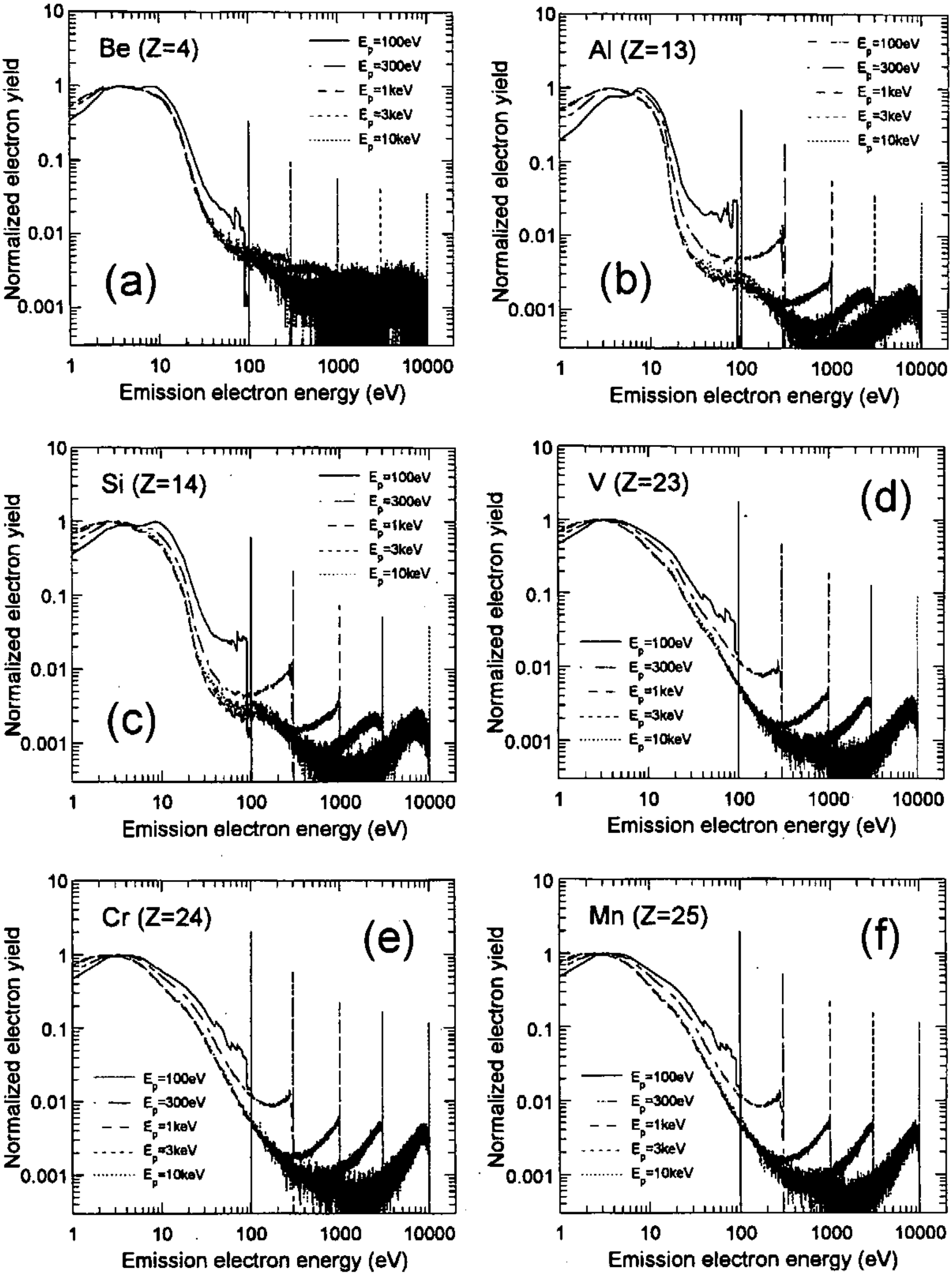

**Figure 8.** Calculated energy distributions of emitted electrons (backscattered electrons + secondary electrons) from (a) Be, (b) Al, (c) Si, (d) V, (e) Cr, (f) Mn, (g) Fe, (h) Co, (i) Ni, (j) Cu, (k) Nb, (l) Mo, (m) Pd, (n) Ag, (o) Ta, (p) W, (q) Pt, (r) Au, and (s) Pb.

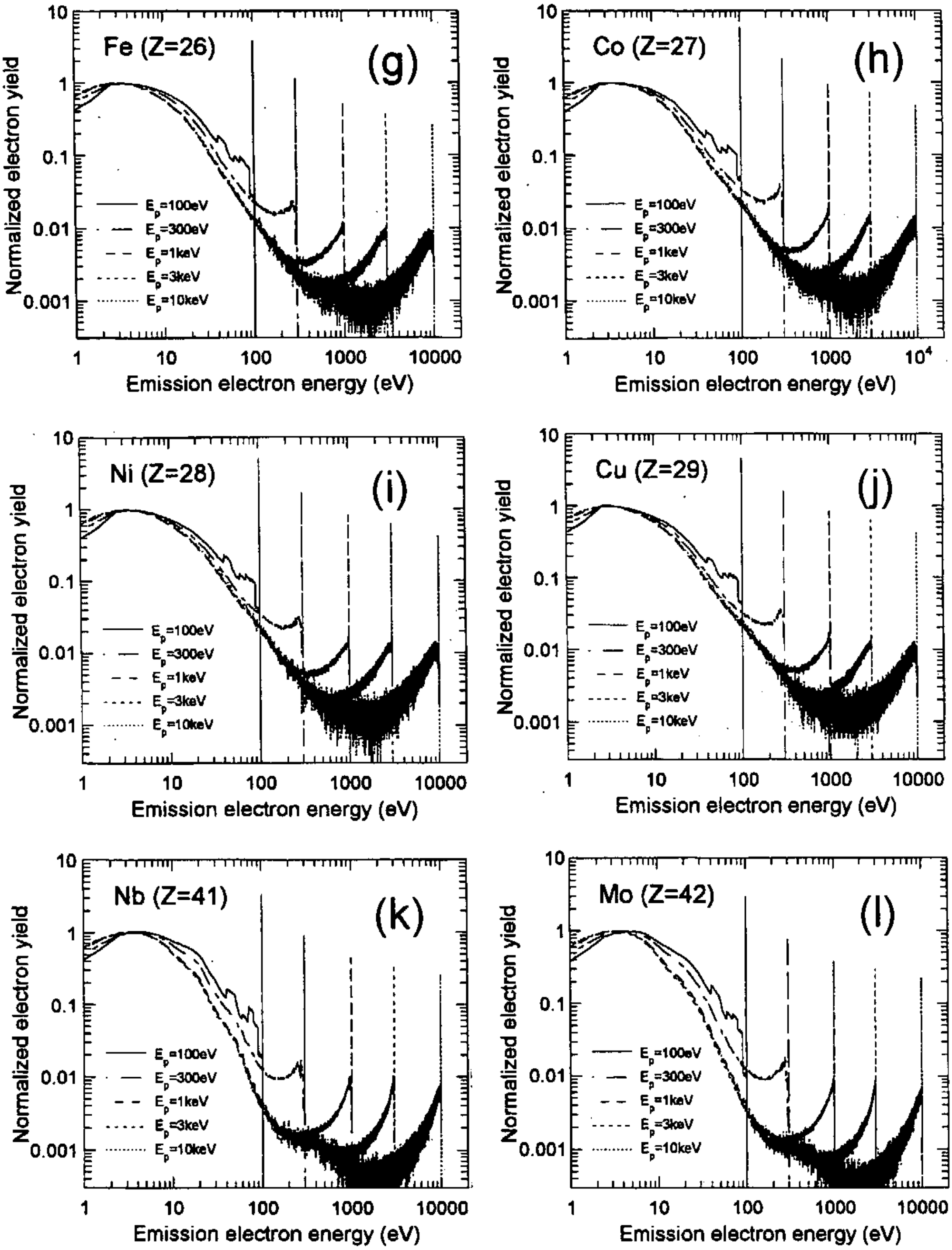

**Figure 8.** – Continued.

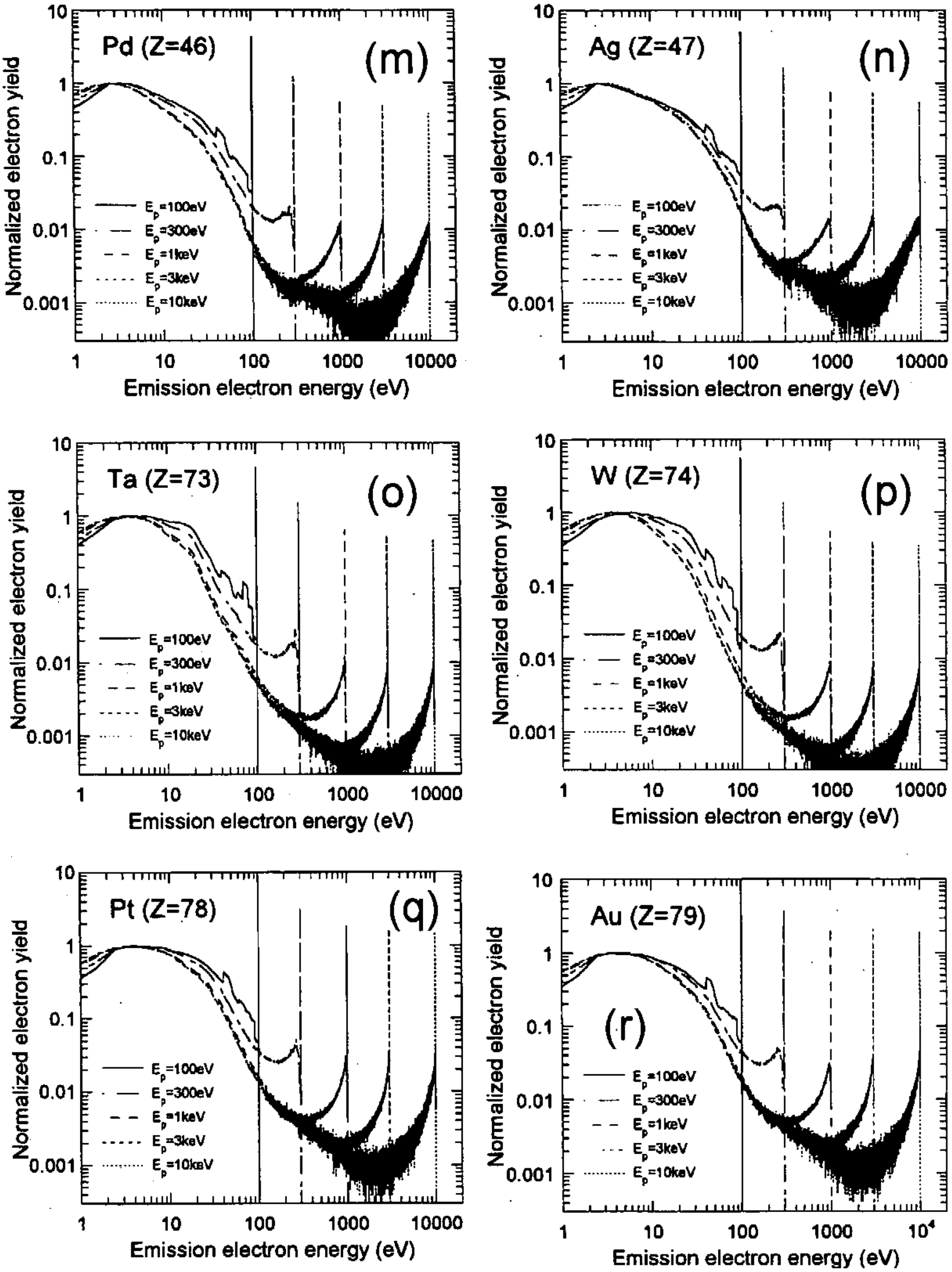

Figure 8. – Continued.

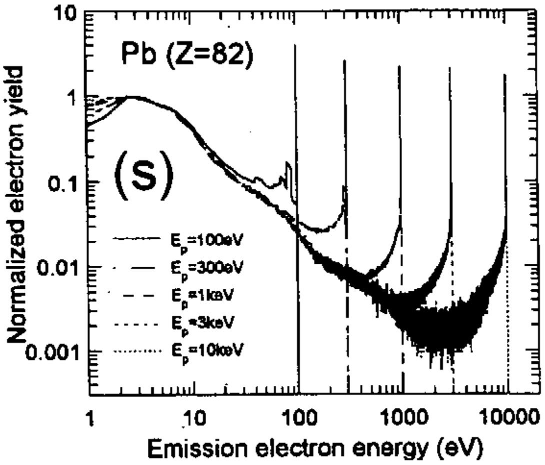

**Figure 8.** – Continued.

### 4.3 Angular distribution of emitted electrons

Assuming an isotropic momentum distribution of electrons in a solid and no refraction of the electrons escaped from the surface potential barrier, the emission of the electrons obeys a simple cosine law: the number of electrons emitted in a solid angle is proportional to cosθ, where θ is the emission angle measured from the surface normal. Therefore, the calculated results in Fig. 9, which show the number of electrons emitted at the intervals of 5 deg., are deduced to that in a solid angle and then normalized.

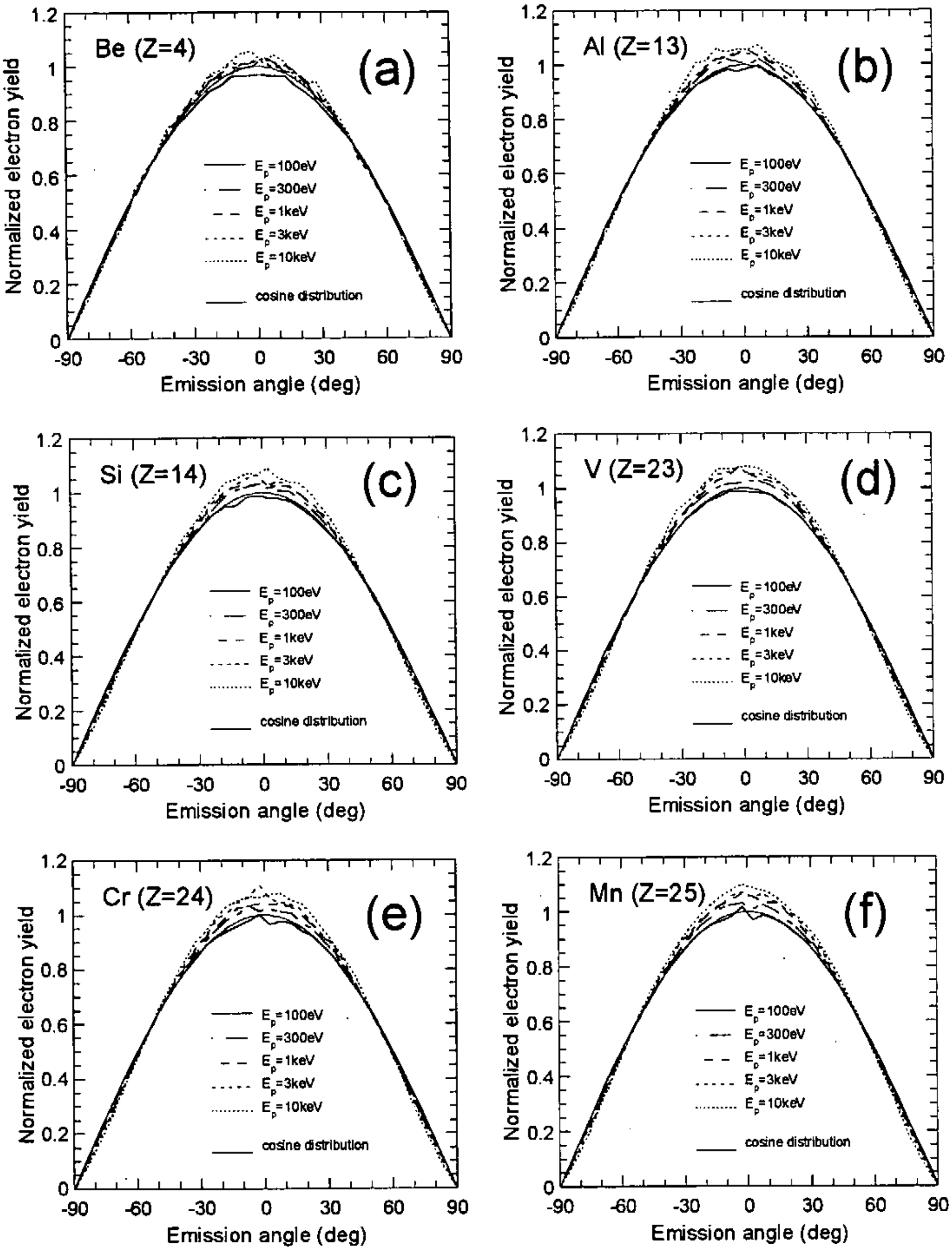

**Figure 9.** Calculated angular distributions of emitted electrons (backscattered electrons + secondary electrons) from (a) Be, (b) Al, (c) Si, (d) V, (e) Cr, (f) Mn, (g) Fe, (h) Co, (i) Ni, (j) Cu, (k) Nb, (l) Mo, (m) Pd, (n) Ag, (o) Ta, (p) W, (q) Pt, (r) Au, and (s) Pb.

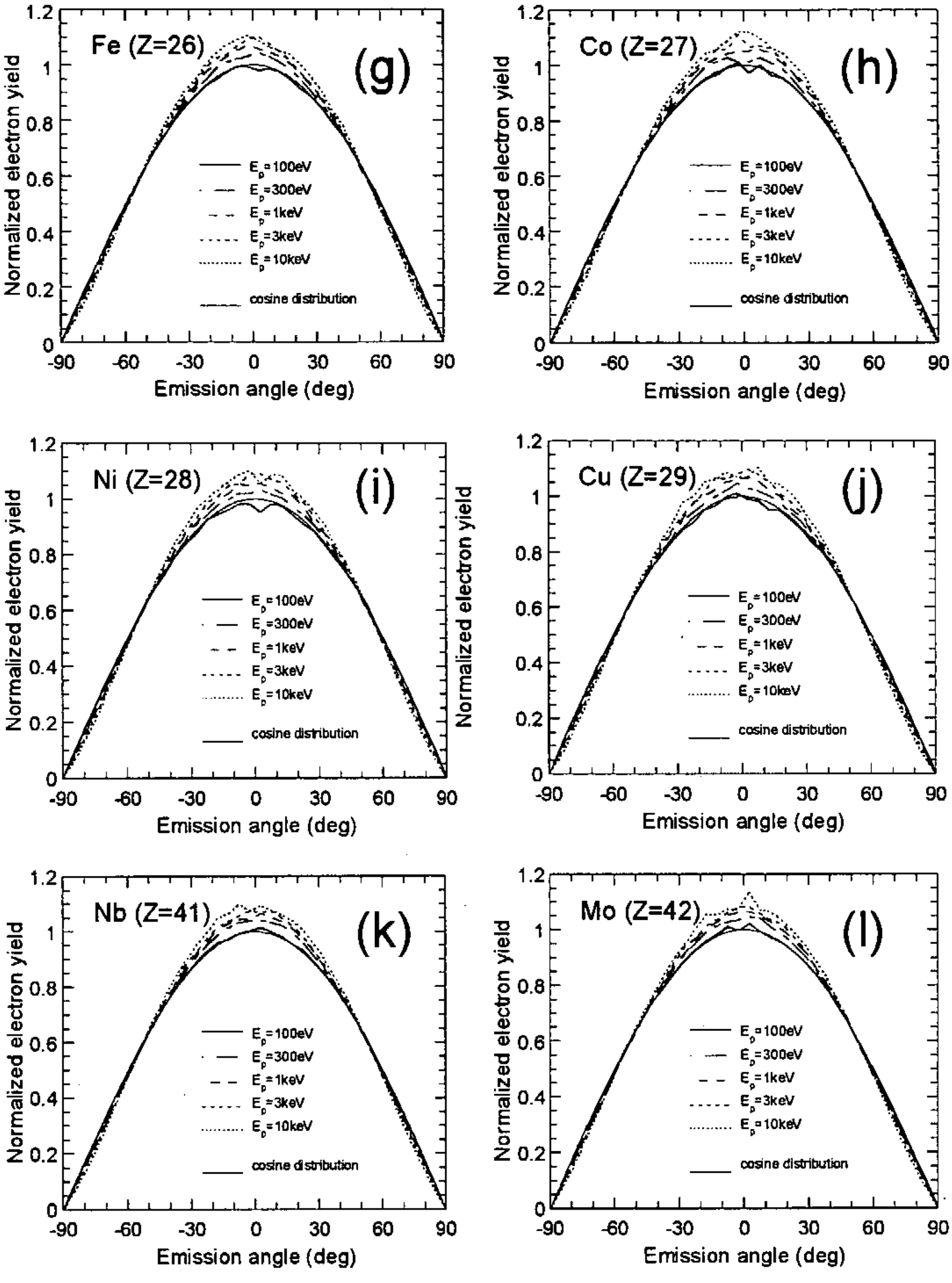

**Figure 9.** – Continued.

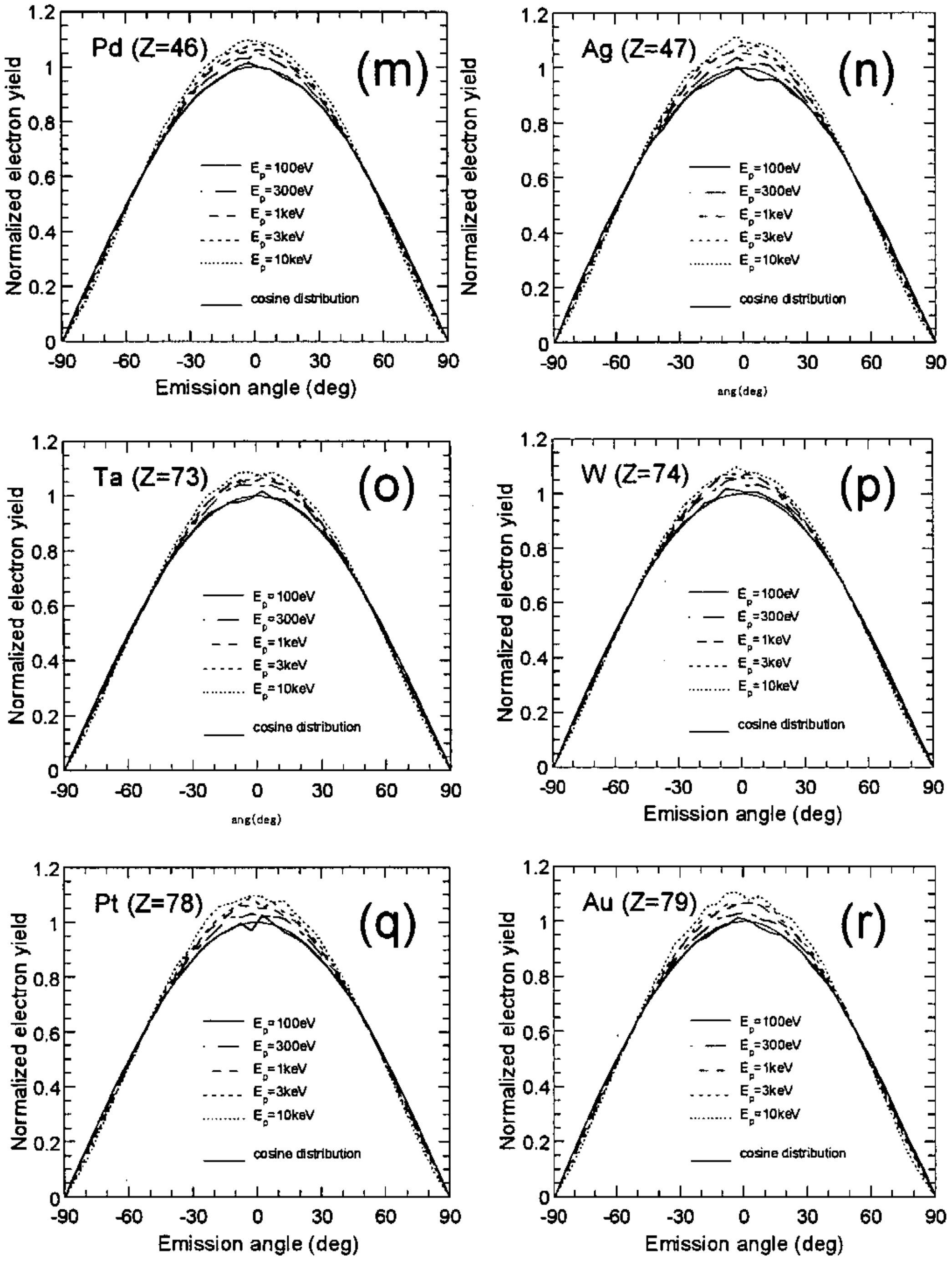

**Figure 9.** – Continued.

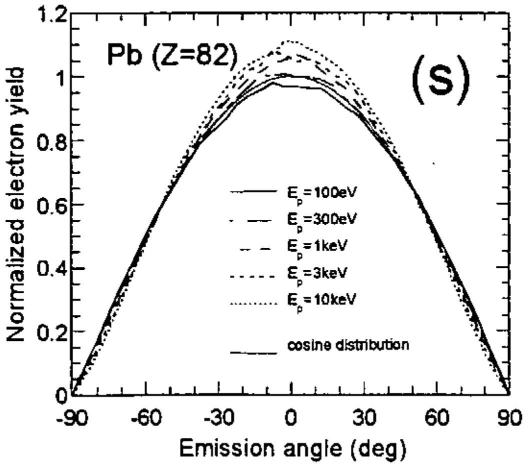

**Figure 9. – Continued.**

## Acknowledgements

We wish to acknowledge contributions from taskgroup members of collaboration research of the Data and Planning Center at the National Institute for Fusion Science on charged and neutral particle behavior at and near plasma facing material surfaces (http//dpsalvia.nifs.ac.jp/dkato). One of the authors (K.O.) thanks to Mr. S. Ebisu for performing a number o f calculations using ELECTRAN in Sec. 4.

## List of symbols

| | |
|---|---|
| $a_B$ | Bohr radius (0.53 Å) |
| $a_s$ | screening radius in the screened Rutherford cross section for elastic scattering |
| $a_{TF}$ | Thomas-Fermi radius $(0.885Z^{1/3}a_B)$ |
| $e$ | electron charge |
| $E$ | instantaneous energy of primary electron and secondary electron in a solid |
| $E'$ | energy of an electron, which reaches the material surface, in a solid |
| $E_b$ | energy of primary electron backscattered in vacuum |
| $E_F$ | Fermi energy of the free-electron gas |
| $E_p$ | energy of primary electron of incidence |
| $E_s$ | energy of secondary electron emitted in vacuum |
| $E_\tau$ | screening energy parameter in $\tau(E)$ in the screened Rutherford cross section |
| $m_e$ | electron mass |
| $h$ | Dirac constant (Planck's constant divide by $2\pi$) |
| $k$ | extinction coefficient |
| $N$ | atomic density of target material atoms |
| $n$ | index of reflection |
| $q$ | momentum transfer from an electron to material electron in inelastic collision |
| $R_1$,$R_2$,$R_3$ | uniform random number |
| $s$ | step length of an electron between two successive collisions |
| $U$ | surface barrier energy of target material |
| $x_n$,$y_n$,$z_n$ | position of the $n$th scattering point in target material |
| $Z$ | atomic number of target atom |
| $\beta$ | screening value in the screened Rutherford cross section for elastic scattering |
| $\Delta E$ | inelastic energy loss of an electron in a solid |
| $\varepsilon_0$ | permittivity of vacuum |

| | |
|---|---|
| $\varepsilon(q, \omega)$ | complex dielectric function of target material |
| $\Phi$ | work function of target material |
| $\lambda_{el}$ | mean free path for elastic scattering of electrons from a solid atom |
| $\lambda_{inel}$ | mean free path for electron excitation by an electron in a solid |
| $\lambda_{tot}$ | total mean free path of an electron for elastic and inelastic collisions |
| $\theta$ | ejection angle of an electron in vacuum. |
| $\theta'$ | ejection angle of an electron in a solid |
| $\theta_c$ | critical angle of an electron for total reflection below the material surface |
| $\theta_n$, $\phi_n$ | angles of an electron in the coordinate system moving with the electron |
| $\vartheta$, $\varphi$ | scattering angle and azimuthal angle of an electron in the coordinate system fixed with the target material |
| $\sigma_{el}$ | total cross section for elastic collision |
| $\tau(E)$ | screening parameter in the screened Rutherford cross section |
| $\tau(E,\omega)$ | probability of an energy loss of an electron per unit distance in a solid |
| $\omega$ | energy transfer from an electron to material electron in inelastic collision |

## References

[1] M. Rösler, W. Brauer, J. Devooght, J.-C. Dehases, A. Dubus, M. Vailler and J.-P. Ganachaud, Particle Induced Electron Emission I (Springer, Berlin, 1991).

[2] D. Hasselkamp, H. Rothard, K.-O. Groeneveld, J. Kemmler, P. Varga and H. Winter, Particle Induced Electron Emission II (Springer, Berlin, 1992)

[3] J. Schou, Electron emission from Solids, in: Physical Processes of the Interaction of Fusion Plasmas with Solids, ed. W.O. Hofer and J. Roth (Academic, San Diego, 1996).

[4] W. Eckstein, Computer Simulation of Ion-Solid Interactions (Springer, Berlin, 1991).

[5] T. Koshikawa and R. Shimizu, J. Phys. D: Appl. Phys. **7** (1974) 1303.

[6] J.P. Ganachaud and M. Cailler, Surf. Sci. **83** (1979) 498.

[7] Z.J. Ding and R. Shimizu, Surf. Sci. **197** (1988) 539.

[8] M. Kotera, R. Ijichi, T. Suga and D.B. Wittry, Jpn. J. Appl. Phys. **29** (1990) 2277.

[9] A. Dubus, J.-C. Dehaes, J.P. Ganachaud, J.P. Hafni and M. Cailler, Phys. Rev. B **47** (1993) 11056.

[10] K. Ohya and J. Kawata, Scan. Microsc. **9** (1995) 331.

[11] J.-Ch. Kuhr and H.-J. Fitting, Phys. Stat. Sol. (a) **172** (1999) 433.

[12] K. Nishimura, J. Kawata and K. Ohya, Nucl. Instrum. and Meth. Phys. Res. B **164-165** (2000) 903.

[13] K. Murata and D.F. Kyser, Adv. Electronics Electron Phys. **69** (1987) 175.

[14] A. Jablonski, Surf. Interf Anal. **14** (1989) 659.

[15] H.-J. Fitting and J. Reinhardt, phys. stat. sol. (a) **88** (1985) 245.

[16] J. Lindhard, K. Dan. Vidensk. Selsk. Mat. Fys. Medd. **28** (1954) 8.

[17] R.H. Ritchie, Phys. Rev. **114** (1959) 644.

[18] D.R. Penn, Phys. Rev. B **35** (1987) 482.

[19] J.C. Ashley, J. Electron Spectrosc. Relat. Phenom. **46** (1988) 199.

[20] E.D. Palik (ed.), Handbook of Optical Constants of Solids (Academic, Boston, 1985).

[21] E.D. Palik (ed.), Handbook of Optical Constants of Solids II (Academic, Boston, 1991).

[22] K.-H. Hellwege, J.I. Olsen (Eds.), Electron states and Fermi surfaces of elements, Numerical Data and Functional Relationships in Science and Technology, Vol. 13, (Springer, Berlin, 1984) p.26.

[23] D.R. Lide (Ed.), Handbook of Chemistry and Physics (CRC, Boca Raton, 1993) p. 12 105.

[24] R. Shimizu and Z.-J. Ding, Rep. Prog. Phys. (1992) 487.

# Recent Issues of NIFS-DATA Series

NIFS-DATA-59 Yu. V. Ralchenko, R. K. Janev, T. Kato, D.V. Fursa, I. Bray and F.J. de Heer
Cross Section Database for Collision Processes of Helium Atom with Charged Particles.
I. Electron Impact Processes: Oct. 2000

NIFS-DATA-60 U.I. Safronova, C. Namba, W.R. Johnson, M.S. Safronova,
Relativistic Many-Body Calculations of Energies for n = 3 States in Aluminiumlike Ions: Jan. 2001

NIFS-DATA-61 U.I. Safronova, C. Namba, I. Murakami, W.R. Johnson and M.S. Safronova,
E1,E2, M1, and M2 Transitions in the Neon Isoelectronic Sequence: Jan. 2001

NIFS-DATA-62 R. K. Janev, Yu.V. Ralchenko, T. Kenmotsu,
Unified Analytic Formula for Physical Sputtering Yield at Normal Ion Incidence: Apr. 2001

NIFS-DATA-63 Y. Itikawa,
Bibliography on Electron Collisions with Molecules: Rotational and Vibrational Excitations, 1980-2000 Apr. 2001

NIFS-DATA-64 R.K. Janev, J.G. Wang and T.Kato,
Cross Sections and Rate Coefficients for Charge Exchange Reactions of Protons with Hydrocarbon Molecules: May 2001

NIFS-DATA-65 T. Kenmotsu, Y. Yamamura, T. Ono and T. Kawamura,
A New Formula of the Energy Spectrum of Sputtered Atoms from a Target Material Bombarded with Light Ions at Normal Incidence: May 2001

NIFS-DATA-66 I. Murakami, U. I. Safronova and T. Kato,
Dielectronic Recombination Rate Coefficients to Excited States of Be-like Oxygen: May 2001

NIFS-DATA-67 N. Matsunami, E. Hatanaka, J. Kondoh, H. Hosaka, K. Tsumori, H. Sakaue and H. Tawara,
Secondary Charged Particle Emission from Proton Conductive Oxides by Ion Impact; July 2001

NIFS-DATA-68 R.K. Janev, J.G. Wang, I. Murakami and T. Kato,
Cross Sections and Rate Coefficients for Electron-Impact Ionization of Hydrocarbon Molecules: Oct. 2001

NIFS-DATA-69 S. Zou, T. Kato, I. Murakami,
Charge Exchange Recombination Spectroscopy of Li Ⅲ Ions for Fusion Plasma Diagnostics: Oct. 2001

NIFS-DATA-70 I. Murakami, T. Kato, A. Igarashi, M. Imai, Y. Itikawa, D. Kato, M. Kimura, T. Kusakabe, K. Moribayashi, T. Morishita, K. Motohashi, L. Pichl
AMDIS and CHART update ( I ): Oct. 2002

NIFS-DATA-71 S. Zou, L. Pichl, M. Kimura and T. Kato
Total, Partial and Differential Ionization Cross Sections in Proton-hydrogen Collisions at Low Energy: Jan. 2003

NIFS-DATA-72 M. Hayashi
Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Argon –: Jan. 2003

NIFS-DATA-73 J. Horacek, K. Houfek, M. Cizek, I. Murakami and T. Kato
Rate Coefficients for Low-Energy Electron Dissociative Attachment to Molecular Hydrogen: Feb. 2003

NIFS-DATA-74 M. Hayashi
Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Carbon Dioxide –: Apr. 2003

NIFS-DATA-75 X. Ma, H.P. Liu, Z.H. Yang, Y.D. Wang, X.M. Chen, Z.Y. Liu, I. Murakami and C. Namba
Cross-section Data Measured at Low Impact Energies for $Ar^{q+}$ Ions on Argon and Neon Targets. Apr. 2003

NIFS-DATA-76 M. Hayashi
Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Sulphur Hexafluoride –; May 2003

NIFS-DATA-77 M. Hayashi
Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Nitrogen Molecule – ; June 2003

NIFS-DATA-78 A. Iwamae, T. Fujimoto, H. Zhang, D. P. Kilcrease, G. Csanak and K.A. Berrington
Population Alignment Collisional Radiative Model for Helium-like Carbon:Polarization of Emission Lines and Anisotropy of the Electron Velocity Distribution Function in Plasmas: Aug. 2003

NIFS-DATA-79 M. Hayashi
Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Xenon – ; Sep. 2003

NIFS-DATA-80 M. Hayashi
Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Halogen Molecules – ; Dec. 2003

NIFS-DATA-81 M. Hayashi
Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Water vapour – : Dec. 2003

NIFS-DATA-82 Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Hydrogen molecules – : Feb. 2004

NIFS-DATA-83 Bibliography of Electron and Photon Cross Sections with Atoms and Molecules Published in the 20th Century – Hydrogen Halide Molecules – : Mar. 2004

NIFS-DATA-84 ELECTRAN - Monte Carlo Program of Secondary Electron Emission from Monoatomic Solids under the Impact of 0.1 - 10 keV Electrons: Mar. 2004