Forschungszentrum Jülich

Institut für Plasmaphysik
EURATOM Association

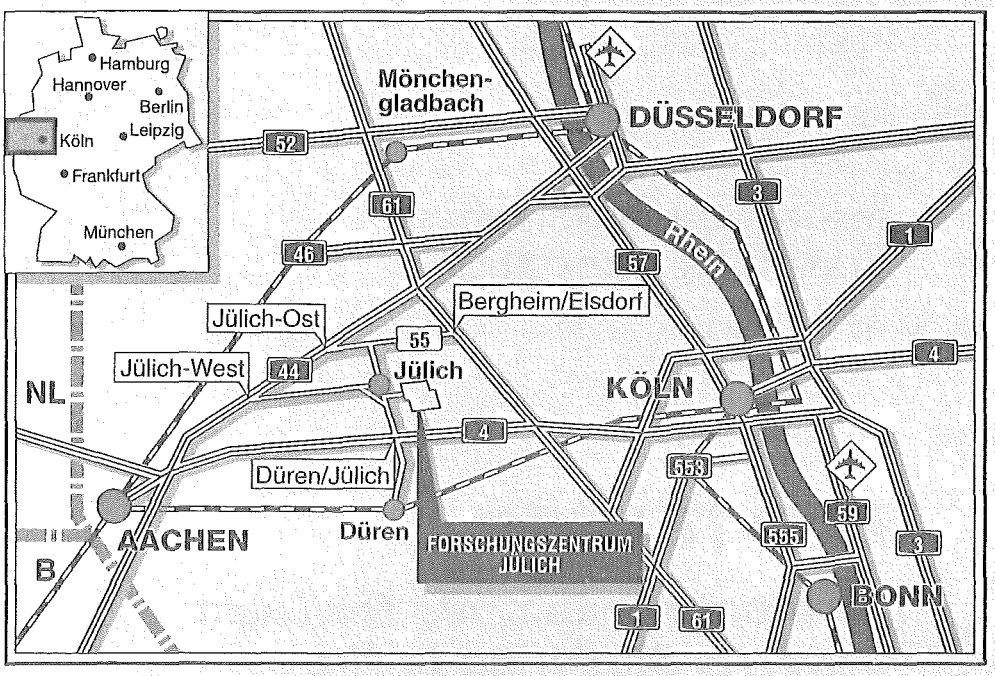
ERO-TEXTOR

3D-MonteCarlo Code for Local Impurity-Modeling in the Scrape-Off-Layer of TEXTOR

Version 2.0

U. Kögler J. Winter

JÜL-3361



Berichte des Forschungszentrums Jülich ; 3361
ISSN 0944-2952
Institut für Plasmaphysik Jül-3361
EURATOM Association

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek
D-52425 Jülich · Bundesrepublik Deutschland
☎ 02461/61-6102 · Telefax: 02461/61-6103 · e-mail: zb-publikation@fz-juelich.de

ERO-TEXTOR

3D-MonteCarlo Code for Local Impurity-Modeling in the Scrape-Off-Layer of TEXTOR

Version 2.0

U. Kögler[1] *J. Winter*

[1] Teil der geplanten Dissertation des Ulrich Kögler, Universität Düsseldorf
Part of the Ph. D. thesis of Ulrich Kögler, University Düsseldorf, in preparation

Abstract

The ERO-TEXTOR code is described in detail. The code solves the kinetic equations of impurities in the scrape-off layer of a tokamak plasma in the vicinity of material surfaces like limiters or divertors. A relaxation time ansatz in the traced impurity limit is chosen, taking the gyromotion of the particles into account. Since the background plasma is slightly non-maxwellian at the plasma edge higher order corrections (*thermal forces*) to the relaxation time ansatz are also considered. Background plasma parameters are calculated from a *simple* plasma model, i.e. the one dimensional continuity and momentum equations are used to derive the local electron density, the local flow velocity and the pre-sheath and sheath electric fields. Since these calculations are not done in a selfconsistent way, the measured values of electron density and temperature are used as basic input to derive the dependency of these quantities. The regarded magnetic topology is still straight and uniform.

Also detailed account is given to the plasma surface interaction and the erosion/deposition processes. A linear differential equation model for multi species impact on a material surface has been developed and is used in a discrete time step approximation.
External databases include the ionization rates for atomic species, molecular processes of methane and silane molecules and the sputtering and reflection yields, which are taken from binary collision calculation codes (e.g. TRIM) or from semi-empirical fits (e.g. the Bodhansky and Yamamura fits).

Contents

List of Figures

Chapter 1

Introduction

ERO-TEXTOR (Erosion and Redeposition in TEXTOR) is a local Monte-Carlo impurity transport code in a given background plasma. It was originally based on the ERO-code [34], [35], [36], [37], [5], [49] but has been substantially improved with respect to the background plasma, atomic, molecular and solid state physics which is taken into account. Particles originating from a limiter/divertor-plate are followed in the plasma in a local tracing box until they leave the box or strike the material surface. The code is locally 3-D (toroidal and poloidal bending is not taken into account). The code solves the equation of motion for the impurity ions in a magnetic and electrostatic field (sub-gyro-motion) and ion motion along $\vec{B}$ is assumed to be classical, while cross-field motion has to be specified by an anomalous cross-field diffusion coefficient $D_\perp$.

The plasma background conditions must be provided to ERO-TEXTOR as input, including local electron density n_e and temperature T_e, ion temperature $T_{i,b}$, plasma species (H,D,T) and plasma impurities (He,C,O) and the SOL geometry (i.e. flow conditions, LCFS-position (Last Closed Flux Surface) relative to the limiter/divertor-plate and Connection Length L_c).

Output from ERO-TEXTOR for each time step includes:

- Material Transport Information
  - Gross erosion, gross deposition and net transport pattern on the material surface.
  - Amount of particles lost from the tracing box.
  - Average impact energy and charge state of the traced species hitting the material surface.
- Spectroscopic Information
  - 2D spatial distributions (line of sight density $\frac{1}{cm^2}$ and line radiation $\frac{photons}{cm^2 ssr}$) of up to five different species (element, charge state) in the tracing box.

The basic physics of impurity production and motion as assumed by ERO-TEXTOR are described in Chapter 2. Also the different SOL and limiter/divertor geometries which can be examined are found there. For running ERO-TEXTOR different Include-Files are needed, such as ionization cross section, sputtering and reflection databases. These are described in Chapter 3 as well as the special launch file for each run. Chapter 4 describes the formatted output files created by ERO-TEXTOR, while Chapter 5 deals with more technical details of the code, such as the provided routines and their purpose.

Chapter 2

Basic Physics of ERO-TEXTOR

In this chapter the basic equations of motion of the traced particles due to plasma flow, magnetic and electrostatic forces, are presented (section 1). The second part introduces the relevant parameters of the background plasma, the flow conditions of the background plasma outside the LCFS and the magnetic field configuration. The possible limiter geometry and the calculation of the sheath potential can be found there, too. The third part deals with particle launch options (as sputtered particles or due to external sources) and the interaction between the background plasma and the material surface.

2.1 The Fokker-Planck Equation for a One Particle Distribution Function with Coulomb Interaction

For the description of the behavior of a *test particle* in a given background plasma the concept of the one particle distribution function is used, where the particle itself is described by a probability function $f((\vec{x},\vec{v},t)$ in the phase space with the normalizing condition:

$$\int f(\vec{x},\vec{v},t)d^3xd^3v = 1 \tag{2.1}$$

From the statistical point of view it is clear that this one particle distribution function has to satisfy a Boltzmann-equation, which can be written as (omitting the phase space coordinates in f)[32]:

$$\frac{\partial f}{\partial t} + \vec{v}\cdot\frac{\partial f}{\partial \vec{x}} + \frac{\vec{F}}{m}\frac{\partial f}{\partial \vec{v}} = \left(\frac{\delta f}{\delta t}\right)_{coll} \tag{2.2}$$

where the collision term on the right side describes the collisions of the *particle* with all other particles (each species of particles denoted by a subscript j and density n_{0j}), i.e.:

$$\left(\frac{\delta f}{\delta t}\right)_{coll} = \sum_j n_{0j}\int d\vec{v}_j \int d\Omega \left(\frac{d\sigma}{d\Omega}\right)(f'f'_j - ff_j)\|\vec{v}-\vec{v_j}\| \tag{2.3}$$

where the unprimed quantities are the distribution function before the collision and the primed ones after the collision respectively. $d\Omega$ is the space angle element and the differential cross section is given by those for Coulomb collisions which are naturally the dominating collisions in a fully ionized plasma:

$$\left(\frac{d\sigma}{d\Omega}\right) = \left(\frac{Z_j^2 Z^2 e^4}{(4\pi\epsilon_0)^2\mu^2\|\vec{v}-\vec{v}_j\|^4}\right)\frac{1}{\sin^4(\theta/2)} \tag{2.4}$$

with the charges of the collision partners Ze , Z_je and $\mu = \frac{mm_j}{m+m_j}$ the reduced mass. θ is the scattering angle in the center-of-mass coordinates and ϵ_0 the electric field constant.

Since the Coulomb force has infinite range and therefore small angle scattering dominates in a plasma, the collision term can be developed up to first order:

$$\vec{v}' = \vec{v} + \triangle\vec{v}\,, \tag{2.5}$$
$$\vec{w} = \vec{v} - \vec{v}_j\,, \tag{2.6}$$
$$w = \|\vec{v} - \vec{v}_j\|\,, \tag{2.7}$$
$$\vec{w}' = \vec{w} + \triangle\vec{w} \qquad \text{and finally} \tag{2.8}$$
$$\triangle\vec{v} = \frac{\mu}{m}\triangle\vec{w} \tag{2.9}$$

so that the Coulomb collision term becomes for small angle scattering:

$$\begin{aligned}\left(\frac{\delta f}{\delta t}\right)_{coll} = & \sum_j n_{0j}\int w d\vec{v}_j \int d\Omega \left(\frac{d\sigma_{\to j}}{d\Omega}\right) \\ & \left[-\frac{m}{m_j} f\triangle\vec{v}\frac{\partial f_j}{\partial\vec{v}_j}\right. \\ & +\triangle\vec{v} f_j\frac{\partial f}{\partial\vec{v}} + \frac{m^2}{m_j^2}\frac{f}{2}\triangle\vec{v}\triangle\vec{v}\frac{\partial^2 f_j}{\partial\vec{v}_j\partial\vec{v}_j} \\ & \left.-\frac{m}{m_j}\triangle\vec{v}\triangle\vec{v}\frac{\partial f}{\partial\vec{v}}\frac{\partial f_j}{\partial\vec{v}_j} + \frac{f_j}{2}\triangle\vec{v}\triangle\vec{v}\frac{\partial^2 f}{\partial\vec{v}\partial\vec{v}}\right]\end{aligned} \tag{2.10}$$

It is practical to consider first the angular integration $\int d\Omega$ in eqn.(2.10) which can be easily

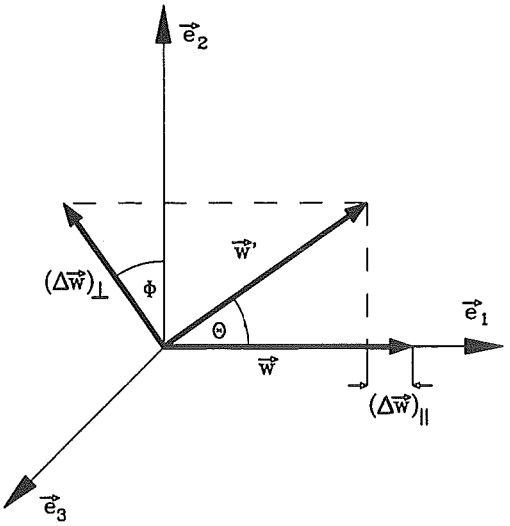


Figure 2.1: Rotational effect of a collision on the difference velocity vector $\vec{w}$ from [32].

done when an appropriate coordinate system is chosen, where the $\hat{e}_1$-axis is along $\vec{w}$. The effect of a collision is to rotate the vector $\vec{w}$ into another vector $\vec{w}' = \vec{w} + \triangle\vec{w}$, leaving the absolute value of $|\vec{w}| = |\vec{w}'|$ the same. In the chosen coordinate system the rotation of $\vec{w}$ is given by:

$$\triangle\vec{w} = w\begin{pmatrix} -2\sin^2\left(\frac{\theta}{2}\right) \\ \sin\theta\cos\phi \\ \sin\theta\sin\phi \end{pmatrix} \tag{2.11}$$

where the angles are shown in the fig. 2.1.
One is now interested to express the coefficients in eqn.(2.10) with the help of quantities $\{\triangle\vec{w}\}$ and $\{\triangle\vec{w}\triangle\vec{w}\}$ defined by the following relations:

$$\{\triangle\vec{w}\} = \int w d\Omega \frac{d\sigma_{\rightarrow j}}{d\Omega}\triangle\vec{w} \tag{2.12}$$

$$\{\triangle\vec{w}\triangle\vec{w}\} = \int w d\Omega \frac{d\sigma_{\rightarrow j}}{d\Omega}\triangle\vec{w}\triangle\vec{w} \tag{2.13}$$

which have in the chosen coordinate system the convenient form:

$$\{\triangle\vec{w}\} = \begin{pmatrix} \{\triangle w_{\|}\} \\ 0 \\ 0 \end{pmatrix} \tag{2.14}$$

and

$$\{\triangle\vec{w}\triangle\vec{w}\} = \begin{pmatrix} \{\triangle w_{\|}^2\} & 0 & 0 \\ 0 & \frac{1}{2}\{\triangle w_{\perp}^2\} & 0 \\ 0 & 0 & \frac{1}{2}\{\triangle w_{\perp}^2\} \end{pmatrix} \tag{2.15}$$

and the vector / tensor elements:

$$\{\triangle w_{\|}\} = \int \frac{Z_j^2 Z^2 e^4}{4\pi\epsilon_0^2\mu^2 w^2} \frac{\sin\theta \sin^2(\theta/2) d\theta}{\sin^4(\theta/2)} \tag{2.16}$$

$$\{\triangle w_{\|}^2\} = \int \frac{Z_j^2 Z^2 e^4 \sin\theta d\theta}{2\pi\epsilon_0^2\mu^2 w} \tag{2.17}$$

$$\{\triangle w_{\perp}^2\} = \int \frac{Z_j^2 Z^2 e^4}{8\pi\epsilon_0^2\mu^2 w} \frac{\sin\theta \sin^2\theta d\theta}{\sin^4(\theta/2)} \tag{2.18}$$

unfortunately the integrals in eqns.(2.16),(2.18) show a logarithmic divergence for small angle scattering $\theta \rightarrow 0$ and therefore a cutoff parameter (the Coulomb logarithm $\Lambda = \int_0^{\theta_{max}} d\theta \frac{\sin\theta}{\sin\frac{\theta}{2}}$) has to be invented. For a plasma this intuitively seems physical due to the screening of the Coulomb force, which has not been considered in these calculations yet. Since $\{\triangle w\}$ is a vector and $\{\triangle w \triangle w\}$ is a tensor it is possible to find a coordinate independent representation, so that after integration over θ, respecting the logarithmic divergence and neglecting small terms in $\{w_{\|}\}^2$ (since $\{w_{\|}\}^2 \overset{\theta_{min}\rightarrow 0}{\longrightarrow} 0 \ll \{w_{\perp}\}^2, \{\triangle w\} \cdot \{\triangle w\}$), these quantities can be written as:

$$\{\triangle\vec{w}\} = A\frac{\vec{w}}{w} \tag{2.19}$$

$$\{\triangle\vec{w}\triangle\vec{w}\} = A\left(\frac{\overleftrightarrow{1}\,^{\dagger} w^2 - \vec{w}\vec{w}}{w^3}\right) \tag{2.20}$$

and the proportionality factor

$$A = \frac{Z_j^2 Z^2 e^4}{4\pi\epsilon_0^2 m^2}\Lambda \tag{2.21}$$

[†] the symbol $\overleftrightarrow{1}$ denotes the unit tensor.

so that in terms of $\{\triangle w\}$ and $\{\triangle w \triangle w\}$ eqn.(2.10) can be written as:

$$\frac{1}{A}\left(\frac{\delta f}{\delta t}\right)_{coll} = \sum_j n_{0j} \int d\vec{v}_j \left[\{\triangle w\} \cdot \left(\frac{\mu}{m} f_j \frac{\partial f}{\partial \vec{v}} - f \frac{\partial f_j}{\partial \vec{v}_j} \frac{\mu}{m}\right) + \{\triangle w \triangle w\} \left[\frac{\mu^2}{2m_j^2} f \frac{\partial^2 f_j}{\partial \vec{v}_j \partial \vec{v}_j} - \frac{\mu^2}{m^2 m_j^2} f \frac{\partial^2 f_j}{\partial \vec{v}} \frac{\partial f}{\partial \vec{v}_j} + \frac{\mu^2}{2m^2} f_j \frac{\partial^2 f}{\partial \vec{v} \partial \vec{v}}\right]\right] \tag{2.22}$$

Integrating by parts and with the help of $\frac{\overleftrightarrow{1} w^2 - \vec{w}\vec{w}}{w^3} = \frac{\partial^2 w}{\partial \vec{v} \partial \vec{v}}$ one gets:

$$\frac{1}{A}\left(\frac{\delta f}{\delta t}\right)_{coll} = \sum_j n_{0j} \int d\vec{v}_j \left[\left(\frac{m}{\mu} f_j \frac{\partial f}{\partial \vec{v}} - \frac{m^2}{m_j \mu} f f_j \frac{\partial}{\partial \vec{v}}\right) \frac{\vec{w}}{w^3} - \left(\frac{m^2 - m_j^2}{2m_j^2} f f_j \frac{\partial^2}{\partial \vec{v} \partial \vec{v}} - \frac{m + m_j}{m_j} f_j \frac{\partial f}{\partial \vec{v}} \frac{\partial}{\partial \vec{v}}\right) \left(\frac{\overleftrightarrow{1}\, w^2 - \vec{w}\vec{w}}{w^3}\right) - \frac{\partial^2}{\partial \vec{v} \partial \vec{v}} \left(f \frac{\partial^2 w f_j}{\partial \vec{v} \partial \vec{v}}\right)\right] \tag{2.23}$$

with the following relations :

$$\frac{\partial}{\partial \vec{v}} \left(\frac{\overleftrightarrow{1}}{w} - \frac{\vec{w}\vec{w}}{w^3}\right) = -2\frac{\vec{w}}{w^3} \tag{2.24}$$

$$\frac{\partial}{\partial \vec{v} \partial \vec{v}} \left(\frac{\overleftrightarrow{1}}{w} - \frac{\vec{w}\vec{w}}{w^3}\right) = -2\frac{\partial}{\partial \vec{v}} \frac{\vec{w}}{w^3} \tag{2.25}$$

and collecting terms the final form of Coulomb collision term is:

$$\frac{1}{A}\left(\frac{\delta f}{\delta t}\right)_{coll} = -\frac{\partial}{\partial \vec{v}} \cdot \left(f \frac{\partial H}{\partial \vec{v}}\right) + \frac{\partial^2}{2 \partial \vec{v} \partial \vec{v}} \cdot \left(f \frac{\partial^2 G}{\partial \vec{v} \partial \vec{v}}\right) \tag{2.26}$$

$$\text{with} \quad H = \sum_j n_{0,j} \left(\frac{m + m_j}{m_j}\right) \int d\vec{v}_j \frac{f_j}{w} \tag{2.27}$$

$$G = \sum_j n_{0,j} \left(\frac{m + m_j}{m_j}\right) \int d\vec{v}_j w f_j \tag{2.28}$$

2.2 Relaxation Times

To make use of the Fokker-Planck-equation derived in sec. 2.1 it is necessary to specify explicitly the distribution functions f_j. Since even then the collision term for all distribution functions f_j is not specified the *traced impurity approximation* is made, i.e. the disturbance caused by the observed distribution function (which will be called *impurity* from now on) on all other distribution functions is negligible. Furthermore to be able to integrate the collision term, the assumption is made that the distribution functions f_j are of the simple Maxwellian type:

$$f_j = \left(\frac{m_j}{2\pi k_B T_j}\right)^{3/2} e^{-\frac{m_j (\vec{v} - \vec{v}_{j,0})^2}{2 k_B T_j}} \tag{2.29}$$

Inserting the distribution function into eqn. (2.26) the collision term then becomes [61],[47]:

$$\frac{1}{A}\left(\frac{\delta f}{\delta t}\right)_{coll} = \left[\frac{\partial}{\partial \vec{v}}\left(\frac{\vec{w}}{w}\{\triangle w_{||}\}\right) + \frac{1}{2}\frac{\partial^2}{\partial \vec{v}\partial \vec{v}}\left(\frac{1}{2}\left[\overleftrightarrow{1} - \frac{\vec{w}\vec{w}}{w^2}\right]\{\triangle w_{\perp}^2\} + \frac{\vec{w}\vec{w}}{w^2}\{\triangle w_{||}^2\}\right)\right] f \tag{2.30}$$

again the coefficients $\{\triangle w_{||}\}$, $\{\triangle w_{||}^2\}$ and $\{\triangle w_{\perp}^2\}$ are the components of the drift vector and the diffusion tensor respectively in the "$\vec{w} = \vec{v} - \vec{v}_0$"-coordinate system and can readily be written as:

$$\{\triangle w_{||}\} = -w\left(1 + \frac{m}{m_j}\right)\frac{\mu\left(\frac{w^2 m_j}{2k_B T_j}\right)}{\tau^{\alpha\beta}} = -\frac{w}{\tau_s} \tag{2.31}$$

$$\{\triangle w_{||}^2\} = w^2 \frac{\mu\left(\frac{w^2 m_j}{k_B T_j}\right)}{\frac{w^2 m_j}{2k_B T_j}\tau^{\alpha\beta}} = \frac{w^2}{\tau_E} \tag{2.32}$$

$$\{\triangle w_{\perp}^2\} = 2 \cdot w^2 \frac{\left(1 - \frac{k_B T_j}{w^2 m_j}\right)\mu\left(\frac{w^2 m_j}{2k_B T_j}\right) + \mu'\left(\frac{w^2 m_j}{2k_B T_j}\right)}{\tau^{\alpha\beta}} = \frac{w^2}{\tau_d} \tag{2.33}$$

where the function $\mu(x)$ is defined by:

$$\mu(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-\tilde{x}}\sqrt{\tilde{x}}\,d\tilde{x} \tag{2.34}$$

$$\mu'(x) = \frac{2}{\sqrt{\pi}}e^{-x}\sqrt{x} \tag{2.35}$$

and

$$\tau^{\alpha\beta} = \frac{w^3}{n_{0,j}A} \tag{2.36}$$

and the last equality in eqns.(2.31,2.32,2.33) gives the three classical relaxation (*Spitzer*) times. These are the slowing down time τ_s, the energy exchange time τ_E and the deflection time τ_d and can be understood as the characteristic time needed by a *beam* of impurities with the distribution function $f_t(\vec{x}, \vec{v}) = \delta(\vec{x}, \vec{v} - \vec{u})$ injected into the maxwellian plasma, to assume the velocity direction of the background plasma:

$$\tau_s = -\frac{u}{\frac{\partial u}{\partial t}} \quad , \tag{2.37}$$

to get an isotropic velocity distribution as the background plasma:

$$\tau_s = \frac{u^2}{\frac{\partial u_{\perp}}{\partial t}} \tag{2.38}$$

and to get a maxwellian energy distribution with the mean energy of the background plasma

$$\tau_E = -\frac{E}{\frac{\partial E}{\partial t}} = -\frac{u^2}{\frac{\partial u^2}{\partial t}} \tag{2.39}$$

$$\text{with:} \quad E = \frac{mu^2}{2} \tag{2.40}$$

All these times depend on the charge state of the impurity ion. Thus first of all at each time step it is calculated first whether the charge state has changed. Recombination is not taken into account yet. Due to the fact that the ions are not thermalized even when leaving the simulation this is in accordance with physical requirements.

2.3 Higher Order Corrections - Thermal Forces

One essential condition to derive the relaxation times in sec. 2.2 is that the integration is taken over a maxwell distribution. But due to finite temperature gradients this condition does not hold for the SOL plasma and therefore higher order corrections have to be considered. These are the thermal forces, which can be of the order of the dynamic friction forces [9],[39],[11]:

$$F_{\text{thermal}} = \alpha \frac{dT_e}{dx} + \beta \frac{dT_i}{dx} \tag{2.41}$$

where the coefficients can be written as:

$$\alpha = 0.71 Z^2 \tag{2.42}$$

$$\beta = -3 \frac{1 - \mu - 5\sqrt{2} Z^2 (1.1\mu^{5/2} - 0.35\mu^{3/2})}{2.6 - 2\mu + 5.4\mu^2} \tag{2.43}$$

where Z is again the charge of the impurity and $\mu = \frac{m_i}{m_i + m_B}$, m_i is the mass of the impurity ion and m_B is the mass of the background plasma ions.

2.4 Equations of motion

All particles launched from a material surface by ERO-TEXTOR are neutrals with an initial velocity $\vec{v}_i$. They are followed collision less through the background plasma (with its local n_e and T_e). For each time step (typically around $10^{-9} s$) it is controlled if the neutral is ionized. A random number ξ uniform in $]0, 1[$ is drawn and compared to the value

$$P_1 = 1 - e^{\frac{\triangle t}{t_{ion}}} \tag{2.44}$$

where

$$\begin{aligned} t_{ion} &= \frac{1}{< \sigma(T_e) v > n_{e,\vec{x}_0}} \quad \text{the ionization time} \\ \triangle t &= \text{time} \quad step \end{aligned} \tag{2.45}$$

If $\xi \geq P_1$ the neutral is ionized, leading to an exponential distribution in ionization time.

Thus the equations of motion for neutral particles are:

$$\begin{aligned} \frac{d\vec{v}}{dt} &= 0 \\ \vec{v}_{new} &= \vec{v}_{Start} \\ \vec{x}_{new} &= \vec{x}_{old} + N \cdot \vec{v}_{start} \triangle t \end{aligned} \tag{2.46}$$

while N is the number of time steps for which $\xi \leq P_1$ and $\vec{v}_{start}$ is the launching velocity of the particle.

If the neutral is ionized, a plasma particle will be produced. Now it is submitted to magnetic and electrostatic forces as well as the thermal forces described above and its classical behavior

is governed by the Spitzer time constants [‡]:

$$\begin{aligned}\triangle\vec{v} &= \frac{1}{m}\left(Z_I e(\vec{E}+\vec{v}\times\vec{B})+\alpha\frac{dT_e}{dx}+\beta\frac{dT_i}{dx}\right)\Delta t+\left(\frac{w}{\tau_s}\right)\Delta t\hat{\vec{w}}_{||} \\ &\pm \hat{\vec{w}}_{||}\sqrt{\left(\frac{w^2}{\tau_E}\right)\Delta t}\pm\hat{\vec{w}}_{\perp,1}\sqrt{\frac{1}{2}\left(\frac{w^2}{\tau_d}\right)\Delta t}\pm\hat{\vec{w}}_{\perp,2}\sqrt{\frac{1}{2}\left(\frac{w^2}{\tau_d}\right)\Delta t} \qquad (2.49)\end{aligned}$$

$$\vec{v}_{new} = \vec{v}_{old}+\triangle\vec{v} \qquad (2.50)$$

$$\vec{x}_{new} = \vec{x}_{old}+\vec{v}_{new}\triangle t+\sqrt{D_\perp\triangle t}\cdot\vec{e}_\perp \qquad (2.51)$$

where the indices *new* and *old* denote the actual and last computational step. As already described in sec. 2.1 $\vec{w} = \vec{v}_B - \vec{v}$ denotes the velocity of the traced particle relative to the background plasma flow velocity. $D_\perp[\frac{mm^2}{s}]$ is the user defined (anomalous) cross-field diffusion. Thus the energy of the particle becomes:

$$E_{new} = \frac{m\left(\vec{v}_{old}+\triangle\vec{v}\right)^2}{2} \qquad (2.52)$$

The tracing of a particle ends when it leaves the user specified tracing box or strikes the limiter surface and is not reflected. For Reflection the following model is used:
The particle of species I strikes the limiter surface of material L with an energy E_{in} under angle θ_{in} relative to the surface normal. From the database the probability of reflection is calculated as function of incidence energy, angle and material combination:

$$R = R(E_{in},\theta_{in},I\rightarrow L) \qquad \in[0,1] \qquad (2.53)$$

then a random number r uniform in $[0,1]$ is drawn and compared to R. If $r \leq R$ then the particle will be reflected and will be launched again as neutral with an energy distribution as described in sec. 2.6.3 , otherwise the simulation for this particle will end.

2.5 Background Plasma and Scrape Off Layer Geometry

The main assumption concerning the plasma modeled by the code is that there is no influence of the impurities on the background plasma. Therefore plasma parameters do not change during a specific run and especially for strong impurity sources this should be kept in mind.

[‡]In general it has to be shown the evolution of the probability $f(\vec{x},t|\vec{x}',t')$ for a small time interval $\triangle t$ for the variable $\vec{x}'$ can be written as [47]:

$$\vec{x}'(t+\triangle t) = \vec{x}'(t)+\vec{A}(\vec{x}'(t),t)\triangle t+\overleftrightarrow{B}(\vec{x}'(t),t)dW(t) \qquad (2.47)$$

where the second term on the right hand side of eqn.(2.47) is the drift component and the third term is the diffusion component of the process, with the defining relation for the diffusive part:

$$\overleftrightarrow{D}(\vec{x}'(t),t) = \overleftrightarrow{B}(\vec{x}'(t),t)\cdot\overleftrightarrow{B}^T(\vec{x}'(t),t) \qquad (2.48)$$

and dW(t) is the stochastic increment, which is achieved with $dW(t) = \sqrt{\triangle t}\gamma_G$, where γ_G is a gaussian random number with mean 0 and variance 1. Since this leads to numerical instabilities near the material surface for large γ_G's when practical time steps of $\triangle t \approx 10^{-9}-10^{-8}s$ are used the gaussian random number is approximated by ± 1.

2.5.1 Geometric Parameters

The plasma volume simulated in the code can be freely defined by the user, but from the equations of motion (eqn.(2.49)) and the fact that yet no recombination but only ionization is considered it should be clear that especially the transport times $\tau_{transport}$ in the plasma volume have to be small compared to the thermalization time τ_{therm} yielding the following simple estimate for the length ($l_{||} \parallel \vec{B}$) of the plasma cube

$$\tau_{transport} \approx \frac{l_{||}}{f \cdot \bar{c}_s} \ll \tau_{therm} \tag{2.54}$$

where $\bar{c}_s$ is the mean ion acoustic speed and f is the factor for the mean velocity along the field lines and τ_{therm} is a Spitzer time similar to those derived above:

$$\tau_{therm} = \frac{m_I T_{i,B} \sqrt{\frac{T_{i,B}}{m_B}}}{1.4 \cdot 10^5 n_e Z_{eff}^2 Z_I^2 \Lambda} \tag{2.55}$$

An estimate for the parameter f in eqn.(2.54) is given by:

$$f = \frac{1}{\bar{c}_s} \left(\frac{1}{3\tau_s} \int_0^{3\tau_s} \left[\int \frac{\bar{c}_s}{\tau_s} \cdot e^{-t/\tau_s} dt \right]_{v(0)=0} dt \right) \approx 0.35 \tag{2.56}$$

with τ_s the slowing down time of eqn.(2.31) and the assumption that for a plasma with $T_e \geq 20eV$ the initial velocity even of energetic sputtered particles can be neglected. Thus the length of the plasma cube along the field lines should not be longer than:

$$l_{||} \ll \tau_{therm} \cdot f \cdot \bar{c}_s \approx 0.35 \tau_{therm} \cdot \bar{c}_s \tag{2.57}$$

No such strict limitations for the maximum length for the directions perpendicular to the magnetic field are given but the minimal extension of the plasma cube perpendicular to the magnetic field lines is given by the sum of the ionization length in this direction and the crossfield transport:

$$l_\perp \gg l_{ion,\perp} + \sqrt{D_\perp \cdot \tau_{transport}} \tag{2.58}$$

Due to the limitations of the code to a small-scale plasma volume the magnetic field is assumed to be uniform in strength and without bending, curvature or shear:

$$\vec{B}(\vec{x}) = B_0 \cdot \vec{b} \qquad ||\vec{b}|| = 1 \tag{2.59}$$

For the scrape off layer there are two different scenarios as shown in figs. (2.2) and (2.3). In the first scenario the test limiter itself is not the last closed flux surface (LCFS) defining element, but withdrawn into the SOL and therefore above the tip of the limiter still a plasma flow to the LCFS defining element is present. The other possibility is that the test limiter defines the LCFS and the core plasma (which is assumed to be flow free) begins right at the limiter tip.

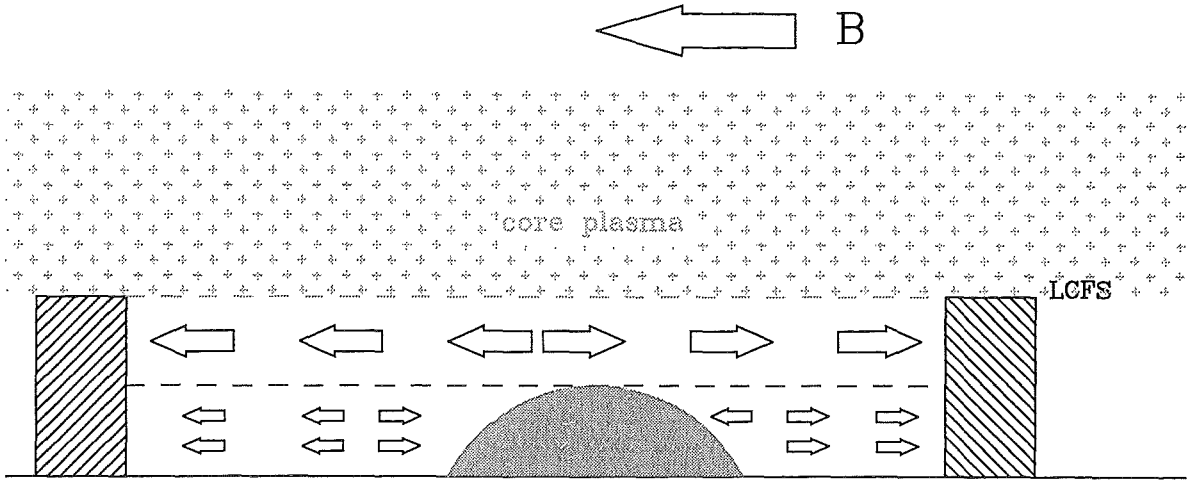


Figure 2.2: Geometry and flow pattern (arrows) in the scrape of layer when the test limiter (sphere shape) is withdrawn into the SOL and any other limiter is the LCFS defining element.

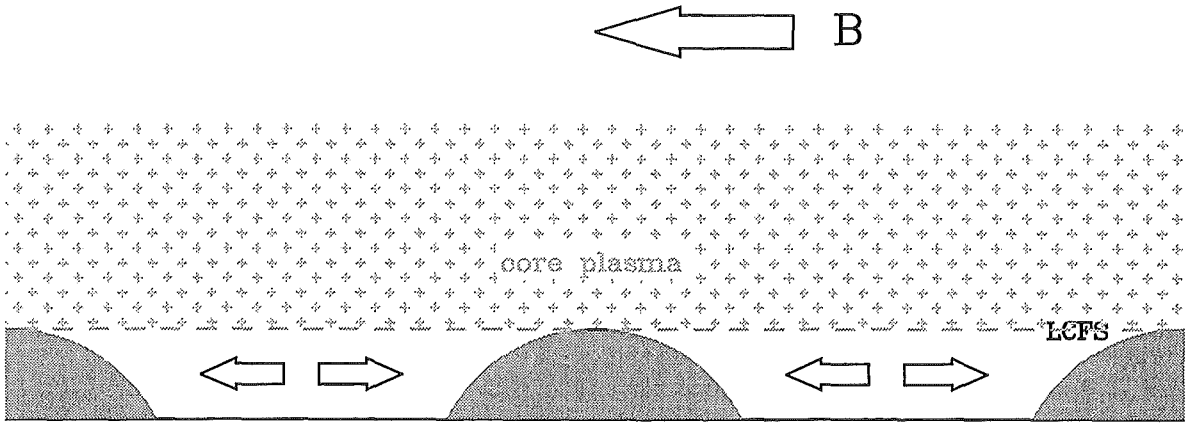


Figure 2.3: Geometry and flow pattern (arrows) in the scrape of layer when the test limiter (sphere shape) itself is the LCFS defining element. The figure shows the SOL double fold, i.e. the limiter in the middle is the same as the half limiters at the edges.

2.5.2 Plasma Parameters

Since ERO-TEXTOR is a traced impurity transport and surface interaction code, there is no selfconsistent treatment of the background plasma in the code itself. Therefore a set of background plasma parameters like the background plasma flow velocity $c_S(\vec{x})$, the electron density $n_e(\vec{x})$, the electron and ion temperature $(T_e(\vec{x})\ , T_i(\vec{x}))$ and the electric fields $\vec{E}(\vec{x})$ have to be specified. In principle these parameters can come from a fluid code, like B2[8], EB2[2] or SAFE[46] or use the results of a more simple model. The ERO-TEXTOR standard uses the results of the simple 1D sol model [57],[58]. Fig. 2.4 shows the principles of such a simple plasma model.

The *simple plasma* model assumes a constant flux $\Gamma_\perp(s) = \Gamma_\perp$ of plasma particles from the confined plasma into the scarpe-off-layer (SOL). The SOL is defined by two limiting surfaces with a connection length L_c along the magnetic field lines. Due to the sink action of the limiting surfaces a flow parallel to the magnetic field lines is created, with flow velocity $c_S(s)$ and flux

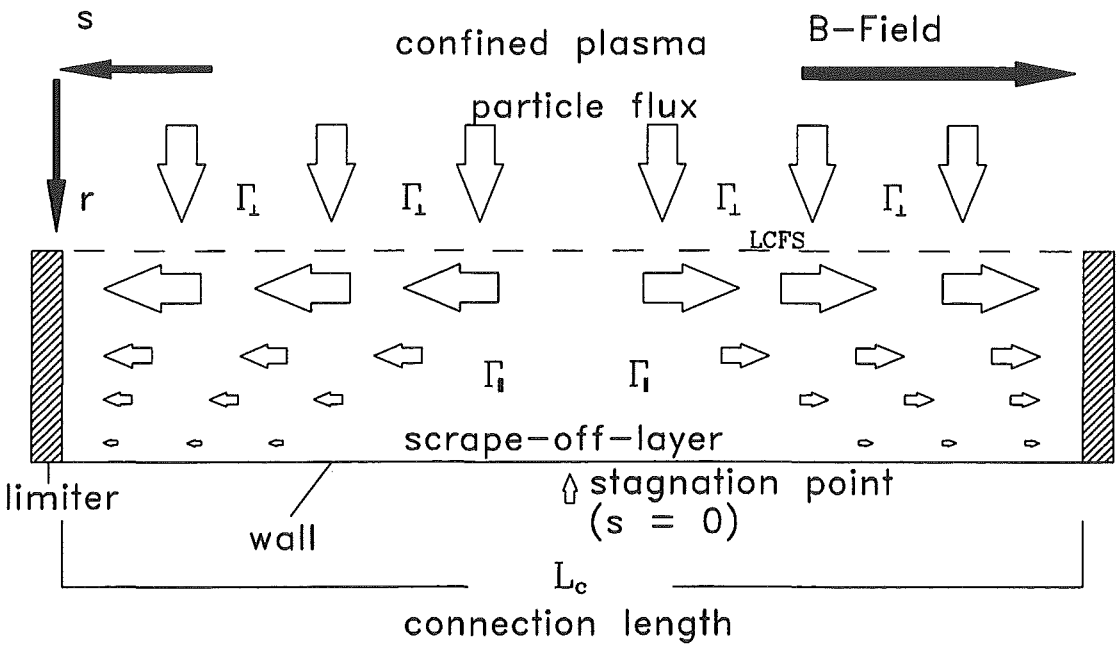


Figure 2.4: The simple one dimensional plasma used to derive the local electron density n_e and background plasma flow c_S.

density $\Gamma_{||}(r,s)$. The stagnation point at a distance $\frac{L_c}{2}$ from each surface is the point where the flow velocity $c_S(0) \equiv 0$. Due to the conservation of particles the parallel flux density decreases with increasing distance from the LCFS along the r-axis and due to the conservation of momentum the flow velocity increases with decreasing distance from the limiting surface.

In this section the plasma parameters, namely electron density, the background plasma velocity and the pre-sheath electric fields are deduced from the *simple plasma* model. For the dependence of the electron density perpendicular to the magnetic field one gets from the continuity equation:

$$\frac{d}{dr} D_\perp \frac{dn_e}{dr} = \frac{d}{ds}\Gamma_{||} = \frac{d}{ds}\left(n_e c_{S,||}\right) \tag{2.60}$$

and, with the assumptions of constant electron density along one flux tube ($n(s) = \text{const.}$) and a constant background plasma velocity ($\frac{dc_S}{ds} = \frac{2 \cdot c_{S,0}}{L_c}$, where $c_{S,0}$ is the ion acoustic speed) eqn.(2.60) can be transformed into:

$$\frac{1}{n}\frac{d^2 n}{dr^2} = \frac{2 \cdot c_{S,0}}{D_\perp L_c} \tag{2.61}$$

This equation is solved by a simple exponential function:

$$n(r) = n_0 \cdot e^{\frac{-r}{\lambda_{ne}}} \tag{2.62}$$

$$\text{with: } \lambda = \sqrt{\frac{D_\perp L_c}{2 \cdot c_{S,0}}} \tag{2.63}$$

From the one dimensional momentum equation the parallel plasma velocity and the parallel dependence of the electron density as well as the pre-sheath parallel electric field can be deduced.

For this the assumption is made that the perpendicular transport across the magnetic field lines into the SOL is independent of the location along s:

$$Q = \int_0^{L_c/2} \Gamma_\perp(s) ds = \Gamma_\perp \int_0^{L_c/2} ds = \Gamma_\perp \frac{L_c}{2} \tag{2.64}$$

Then the ion momentum equation can be written as:

$$m_I n c_S \frac{dc_S}{dx} + \frac{dp_I}{dx} - enE_{||} = Q_C \tag{2.65}$$

with $p_I = nk_BT_I$ the ion pressure, $E_{||} = -\frac{d\Phi}{dx}$ the pre-sheath electric field and $Q_C \equiv C$ the momentum source. Assuming a maxwellian distribution for the electrons one gets:

$$enE_{||} = -k_BT_e \frac{dn}{dx} \tag{2.66}$$

To close the momentum equation the assumption is made that the ions are isothermal and therefore the pressure gradient can be written as:

$$\frac{dp_I}{dx} = k_BT_i \frac{dn}{dx} \tag{2.67}$$

Thus one gets by combining eqns.(2.65),(2.66),(2.67) into eqn.(2.65), respecting the definition of the ion acoustic speed $c_{S,0} = \sqrt{\frac{k_B(T_i+T_e)}{m_I}}$, the following equation:

$$nc_S \frac{dc_S}{dx} + c_{S,0}^2 \frac{dn}{dx} = C \tag{2.68}$$

$$\overset{integration}{\Rightarrow} \quad nc_S^2 + c_{S,0}^2 n = C\frac{L_c}{2} \tag{2.69}$$

$$\overset{\Gamma_{||}=nc_S}{\Rightarrow} \quad \Gamma_{||} \cdot (c_S + \frac{c_{S,0}^2}{c_S}) = C' \tag{2.70}$$

$$\tag{2.71}$$

The parallel flux $\Gamma_{||}(s) = \int_0^s \Gamma_\perp d\tilde{s} = \Gamma_\perp \cdot s$ is given by the cumulated source strength:

$$(\Gamma_\perp s) \cdot (c_S + \frac{c_{S,0}^2}{c_S}) = \text{const} \tag{2.72}$$

Taking the Bohm-criterion into account ($c_S||_{\text{sheath edge}} \equiv c_{S,0}$) this yields the momentum equation at two different locations:

$$\Gamma_\perp s(c_S + \frac{c_{S,0}^2}{c_S})\Big|_{s<\frac{L_C}{2}} = \Gamma_\perp \frac{L_c}{2}(c_{S,0} + \frac{c_{S,0}^2}{c_{S,0}})\Big|_{\text{sheath edge}} \tag{2.73}$$

Finally solving this equation for c_S the local flow velocity along s is given by:

$$c_S(s) = c_{S,0} \cdot \left(\frac{L_c}{2s} - \sqrt{\left(\frac{L_c}{2s}\right)^2 - 1} \right) \tag{2.74}$$

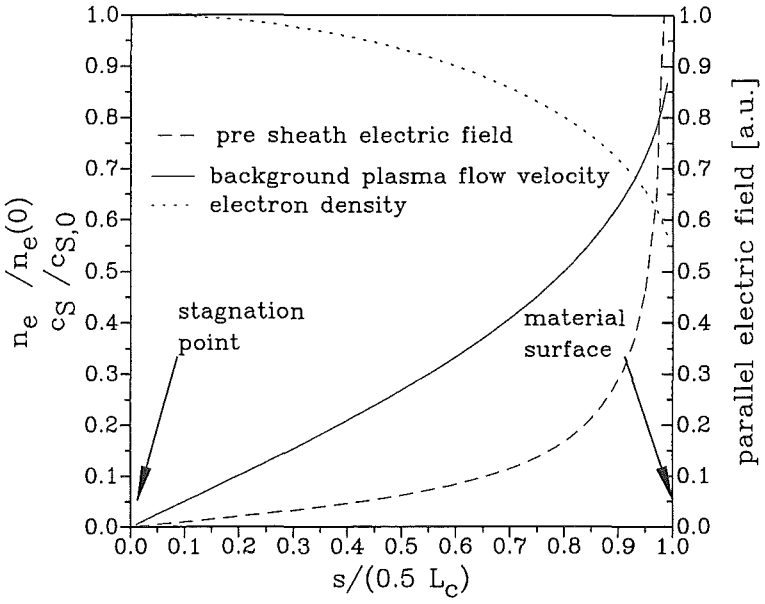


Figure 2.5: Dependence of local plasma parameters from the distance of the stagnation point as derived from the 1d model. The connection length between the material surfaces is L_C, thus the distance from stagnation point to the plasma-sheath edge is $\frac{L_c}{2}$ and the abscisa is normalized on this distance. The plasma-sheath edge is at $\frac{L_c}{2}$.

Since the *perpendicular* source strength and the local velocity $c_S(s)$ are known, the local parallel flux can be calculated

$$\Gamma_{\parallel}(s) = \int_0^s \frac{\Gamma_{\perp,tot}}{L_c/2} d\tilde{s} = \Gamma_{\perp,tot} \frac{s}{L_c/2} \equiv n(s) c_S(s) \tag{2.75}$$

$$\Rightarrow n(s) = \frac{\Gamma_{\perp,tot} \frac{s}{L_c/2}}{c_S(s)} \tag{2.76}$$

$$\Rightarrow n(s) = \frac{\Gamma_{\perp,tot}}{c_{S,0}} \frac{\frac{s}{L_c/2}}{\frac{L_c}{2s} - \sqrt{\left(\frac{L_c}{2s}\right)^2 - 1}} \tag{2.77}$$

$$\Rightarrow n(s) = \frac{\Gamma_{\perp,tot}}{c_{S,0}} \frac{\frac{s^2}{L_c/2}}{\frac{L_c}{2} - \sqrt{\left(\frac{L_c}{2}\right)^2 - s^2}} \tag{2.78}$$

for the limiting cases one gets:

$$\frac{s^2/\frac{L_c}{2}}{\frac{L_c}{2} - \sqrt{\left(\frac{L_c}{2}\right)^2 - s^2}} = \begin{cases} 2 & \text{for } s \to 0 \\ 1 & \text{for } s \to \frac{L_c}{2} \end{cases} \tag{2.79}$$

Respecting the boundary condition that $n_e(r, s = 0) \equiv n_{e,LCFS} \cdot e^{-\frac{r}{\lambda_{n_e}}}$, the density dependence as derived from a one dimensional model is given by:

$$n_e(r, s) = n_{e,LCFS} \frac{s^2/L_c}{\frac{L_c}{2} - \sqrt{\left(\frac{L_c}{2}\right)^2 - s^2}} \cdot e^{-\frac{r}{\lambda_{n_e}}} \tag{2.80}$$

In fig. 2.5 the results of the preceding section are summarized. It shows the local values of the electron density and the background plasma flow velocity as well as the pre-sheath parallel electric field which will be derived hereafter.

Pre-Sheath Electric Field

Since the force equation for the electrons parallel to the magnetic field can be written in an isothermal approximation as [60]:

$$\frac{dp_{e,\|(s)}}{ds} = k_B T_e \frac{dn_{e,\|}(s)}{ds} = -en_e(s)E_{\|}(s) \tag{2.81}$$

it is clear that a gradient in the electron density as described by eqn.(2.80) causes an electric field.

$$E_{\|}(s) = -\frac{k_B T_e}{e}\frac{1}{n_e(s)}\frac{dn_{e,\|}(s)}{ds} \tag{2.82}$$

with the help of eqn.(2.80) the gradient can be calculated and thus the pre-sheath electric field due to the gradient in electron density is:

$$E_{\|}(s) = \frac{k_B T_e}{e}\left(\frac{L_c}{2s\sqrt{\left(\frac{L_c}{2}\right)^2 - s^2}} - \frac{1}{s}\right) \tag{2.83}$$

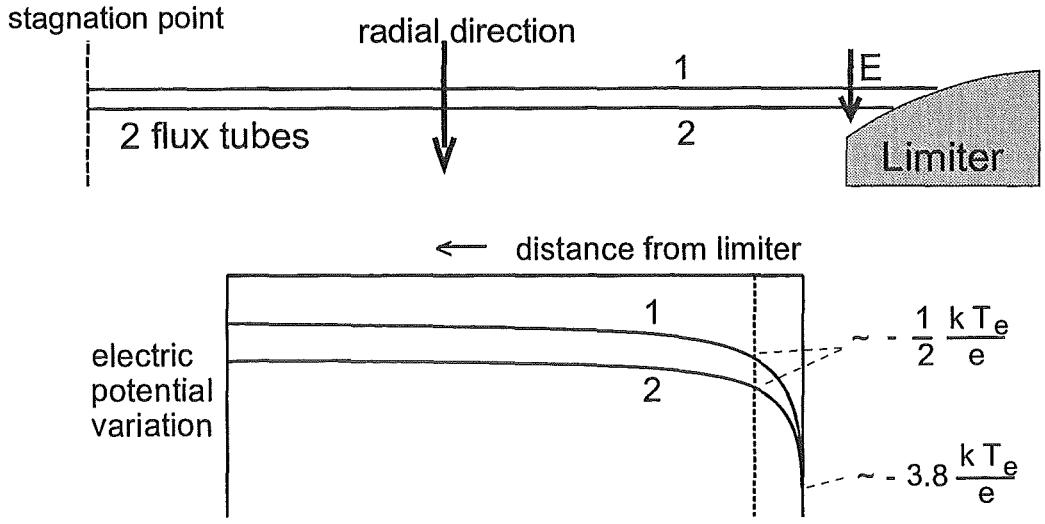


Figure 2.6: Schematic plot of the pre-sheath potential of two different flux tubes. Due to the fact that a radial temperature gradient is present a radial gradient in the in the pre-sheath potential is also generated

The boundary condition at the plasma-sheath entrance is that the ions have to have Mach number of one (ion acoustic speed, i.e. the Bohm criterion has to be satisfied). It can be shown that this condition given by:

$$\Phi(s=0) - \Phi(s=\epsilon|_{\text{sheath entrance}} \approx \frac{L_c}{2}) = -\int_{\epsilon\approx\frac{L_c}{2}}^{0} E_{\|}(\tilde{s})d\tilde{s} \stackrel{!}{\equiv} \ln 2\frac{k_B T_e}{e} \tag{2.84}$$

at the sheath entrance is satisfied.

In addition to this parallel electric field a *radial* electric field is caused by the fact that two flux tubes located at two different radial positions $\triangle r = r_2 - r_1$ have two different electron temperatures. With

$$\phi_{r_1}(r_1, s) = \int_0^s E_{||} d\tilde{s} = \frac{k_B T_e(r_1)}{e} \cdot f(s) \tag{2.85}$$

$$\phi_{r_2}(r_2, s) = \int_0^s E_{||} d\tilde{s} = \frac{k_B T_e(r_2)}{e} \cdot f(s) \tag{2.86}$$

$$\text{with} \qquad f(s) = \ln\left(L_c + \sqrt{L_c^2 - 4s^2}\right) - \ln(2L_c)$$

a gradient in the electron temperature gives rise to a radial electric field, depending also on the distance from the limiter surface (ref. fig. 2.7):

$$E_{\text{radial}}(r, s) = -\lim_{\triangle \to 0} \frac{\Phi(r_2, s) - \Phi(r_1, s)}{\triangle r} = -\frac{d\Phi(r)}{dr} = -\frac{k_B}{e}\frac{dT_e(r)}{dr} \cdot f(s) \tag{2.87}$$

This field is also evaluated in the code.

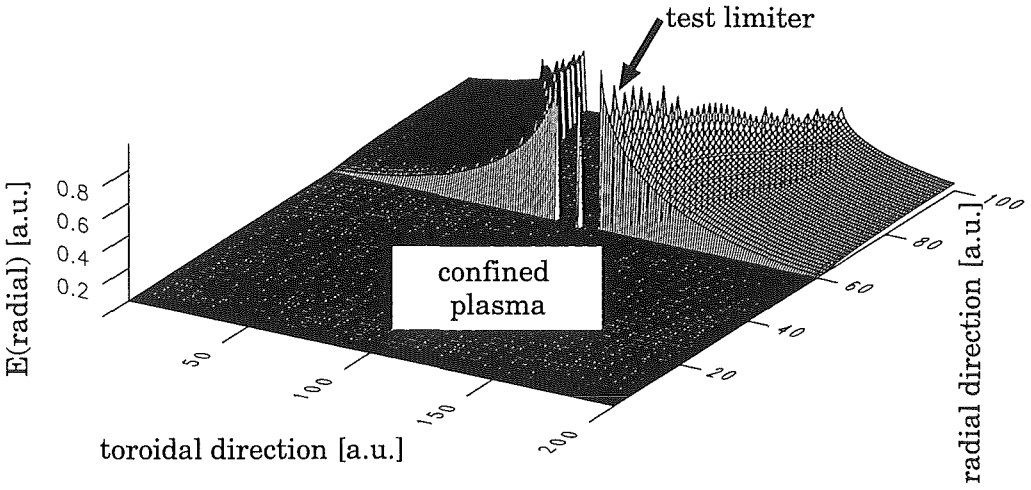


Figure 2.7: Schematic two dimensional plot of the absolute value of the radial electric field in the vicinity of a spherical test limiter. Since the potential along the flux tubes increases (ref. fig. 2.6), the radial electric field is decreasing as the distance from limiter along $\vec{B}$ increases. In the confined plasma this field vanishes. (The irregularities at the peaking of the radial electric field at the limiter surface are due to the finite grid spacing in the calculations).

Sheath Potential Drop

The potential drop in the sheath is given by [56]:

$$\Phi = -\frac{k_B T_e}{2e} \ln\left[2\pi(\frac{m_e}{m_i})(1 + \frac{T_i}{T_e})\frac{1}{(1-\gamma)^2}\right] \tag{2.88}$$

where T_e , T_i and m_e , m_i are the electron and ion temperature and mass, respectively and γ is the secondary electron emission coefficient of the material surface. While the sheath potential

drop depends only on the local values of the temperatures, the typical length of this potential drop, the Debye length:

$$\lambda_{Debye} = \sqrt{\frac{\epsilon_0 k T_e}{e^2 n_e}} \tag{2.89}$$

also depends on the local value of n_e.

For surfaces which are not normal to the magnetic field the sheath separates into the electric sheath (with a typical length of the potential drop of the Debye length) and the magnetic pre-sheath [13] (with a typical length of the potential drop of the (fuel) ion gyro radius R_D). The sum of the potential drops in these sheaths remains the same ($\approx 3.8 k_B T_e$) but the relative potential drop in the magnetic sheath increases with increasing angle of incidence. This is implemented in ERO-TEXTOR by using a calculation done by Brooks et al. [10], shown in fig. 2.8. The

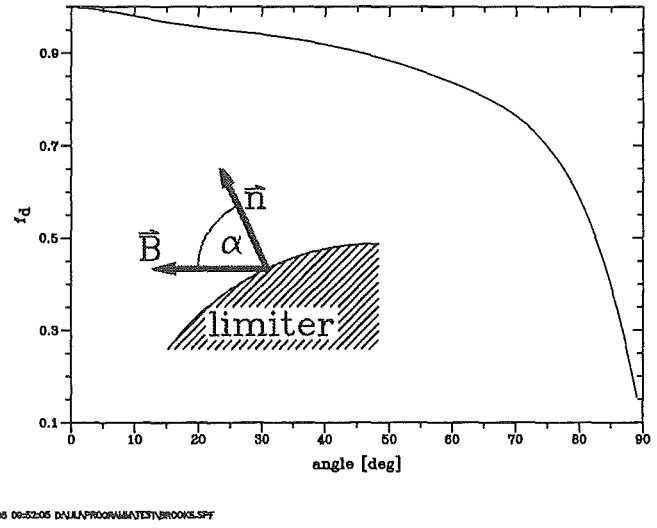


Figure 2.8: The parameter f_d as a function of angle between surface normal and magnetic field vector

potential drop in the sheath and the magnetic pre-sheath is then given by:

$$\Phi(u) = \Phi_S \cdot exp\left(-\frac{u}{2\lambda_d}\right) + \Phi_{MPS} \cdot exp\left(-\frac{u}{R_D}\right) \tag{2.90}$$

$$\text{where } \Phi_S = \Phi_0 \cdot f_d(\alpha) \tag{2.91}$$

$$\Phi_{MPS} = \Phi_0 \cdot (1 - f_d(\alpha)) \tag{2.92}$$

u – normal distance from the surface

Electron and Ion Temperatures

From conservation of energy one gets for the heat flux:

$$\nabla(q_{||} + q_{\perp}) + Q_{rad} = 0 \tag{2.93}$$

where $q_{||}$ and $q_{\perp}$ are the parallel and perpendicular heat fluxes.
For the SOL plasma it is reasonable to assume [29],[58] that the heat flux transported from the

bulk plasma $q_\perp$ to the SOL is conducted away via electron parallel conduction $q_{||} \approx \kappa T^{5/2} \nabla_{||} T$. The heat flux to the limiter/target plate is then given by (see also eqn.(2.147)):

$$q_{||}\Big|_{limiter} = \delta n_0 c_{s,0} k_B T_0 \sim \delta n_0 k_B T_0^{3/2} \tag{2.94}$$

where the subscript "0" stands for the conditions at the limiter/target plate.
For the further calculations the simplifying assumption is made that the radiated energy is negligible, so that the approximation $q_\perp = -q_{||}$ holds.

Since the perpendicular heat transport is not easily calculated from first principles the cross field heat transport is assumed to be diffusive, proportional to local density- and temperature gradients. With the simplifying assumptions that $\chi_i \approx \chi_e$ and $T_e = T_i = T$ the cross field transport equation can be written as the sum of a conduction and a convection term :

$$q_\perp = q_\perp^i + q_\perp^e = -\chi_\perp n \nabla_\perp (T_i + T_e) - \frac{5}{2}(T_i + T_e) D_\perp \nabla_\perp n \tag{2.95}$$

$$= -2\chi_\perp n \nabla_\perp T - 2 \cdot \frac{5}{2} T D_\perp \nabla_\perp n \tag{2.96}$$

a standard assumption then is to set $D_\perp \sim \chi_\perp$ with the proportionality written as $D_\perp \approx \frac{2}{5}\beta\chi_\perp$, so that the perpendicular heat flux can be written as

$$q_\perp = -2\chi_\perp n \nabla_\perp T - 2\chi_\perp \beta T \nabla_\perp n \tag{2.97}$$

and with the assumptions that

$$\begin{aligned} T = T(r) \quad &\Rightarrow \quad \nabla_\perp = \frac{d}{dr} \\ \frac{d}{dr} n(r) \quad &= \quad -\frac{1}{\lambda_{n_e}} \cdot n_0 \qquad \text{ref. eqn.(2.62)} \\ \chi(r) \quad &= \quad \chi \end{aligned}$$

the linear differential equation to be solved is given by:

$$q_\perp = -2\chi_\perp n_0 \frac{d}{dr} T(r) + 2\chi_\perp \frac{\beta}{\lambda_{n_e}} n_0 T(r) \tag{2.98}$$

$$\Rightarrow \frac{q_\perp}{2\chi_\perp n_0} = -\frac{d}{dr} T(r) + \frac{\beta}{\lambda_{n_e}} T(r) \tag{2.99}$$

and the solution of this is of the form:

$$T(r) = \frac{\lambda_{n_e} q_\perp}{2\beta\chi n_0} + C e^{-\frac{\beta r}{\lambda_{n_e}}} \tag{2.100}$$

According to LaBombard et al. [29] the value of β is of the order of unity, i.e. $\beta \approx 1$, so that in this quite rough estimation the exponential decay length of the temperature profiles is given by:

$$\lambda_T \approx \lambda_{n_e} \tag{2.101}$$

where typically $\lambda_T \geq \lambda_{n_e}$ from experimental results. Therefore the radial temperature profile is modeled with a simple exponential function and the assumption that electron and ion temperature are equal:

$$T_e(r) = T_{e,0} \cdot e^{-\frac{r}{\lambda_{T_e}}} \tag{2.102}$$

$$T_i(r) = T_{i,0} \cdot e^{-\frac{r}{\lambda_{T_i}}} \tag{2.103}$$

$$\text{with} \quad T_e(r) = T_i(r) \qquad \lambda_{T_e} = \lambda_{T_i} \tag{2.104}$$

The parallel behavior of the ion and electron temperatures is governed by the parallel heat conductivity

$$q_\alpha = -\chi_\alpha \nabla_{||} T_\alpha \qquad \alpha = i, e \tag{2.105}$$

with the parallel conductivity given by [14]:

$$\chi_e = 2.9 \cdot 10^{-19} n_e v_{Te} \lambda_{Te} = \kappa_{e,0} T_e^{5/2} \tag{2.106}$$

$$\chi_i = 5.1 \cdot 10^{-19} n_i v_{Ti} \lambda_{Ti} = \kappa_{i,0} \cdot T_i^{5/2} \tag{2.107}$$

where $\kappa_{e,0} \approx 1.8 \cdot 10^3 \frac{W}{meV^{-7/2}}$ and $\kappa_{i,0} \approx 75 \cdot \frac{W}{meV^{-7/2}}$.
For an estimate of $\nabla_{||} T_i(s) \equiv \frac{d}{ds} T_i(s)$ the same assumption as in the previous paragraph is made, i.e. all the power flux density at the limiter/target is carried by parallel electron heat conduction and $T_e = T_i$:

$$\kappa_0 T_0^{5/2} (\frac{d}{ds} T)_0 = q_0 \overset{eqn.(2.94)}{\equiv} \delta n_0 c_{S,0} k_B T_0 \tag{2.108}$$

again the subscript "0" stands for the conditions at the target plate itself. While due to the high parallel electron conductivity the assumption is made that the electron temperature is constant along $\vec{B}$, so that integration of eqn.(2.108) gives for the ion temperature at the distance s from the limiter [58]:

$$T_{e,i}(s) = \left(1 + \frac{7 q_0 s}{2 \kappa_0 T_{e,i,0}^{7/2}}\right)^{2/7} T_{e,i,0} \tag{2.109}$$

Eqn.(2.109) has been derived by Stangeby for the situation in a divertor plasma, where the perpendicular heat and particles fluxes out of the core plasma take place far away from the observed target plate and in the chosen ansatz κ_e is the guiding parameter.
The situation in a limiter define SOL is somewhat different, since even very near to the limiter surface there is still a perpendicular heat and particle flux from the core plasma into the SOL. To account for a temperature gradient one has either to rely on measurements, to take the temperature gradients from a magneto-hydrodynamical model or to modify the eqn.(2.109) for this different situation.
Unfortunately there exist no measurements of the temperature gradients near the limiter surface for the TEXTOR-SOL. B2-Eirene calculations yield ion temperature gradients $\frac{dT_i}{ds} \approx 20 \frac{eV}{m}$, which seem to be unrealisticly high, due to limited validity of the fluid approximation near the material surface. Therefore the ansatz to modify eqn.(2.109) to get a crude estimate of the temperature gradients is chosen.
For this we assume due to the high parallel conductivity of the electrons, that the electron temperature is constant along the flux tubes, whereas for the ion temperature eqn.(2.109) is solved. This yields temperature gradients $\frac{dT_i}{ds} \approx 0.5 \frac{eV}{m}$, which in comparsion to the fluid calculations are very low. Therefore an additional factor f_i is introduced to adjust for the specific situation. In this model the temperature gradients in parallel direction are then given by:

$$\nabla_{||} T_e(s) \approx 0 \tag{2.110}$$

$$\nabla_{||} T_i(s) = \frac{q_0}{\kappa_0 \left(1 + (\frac{7 f_i q_0 s}{2 \kappa_0 T_i})^{5/7}\right)} \tag{2.111}$$

2.6 Particle Generation and Plasma Surface Interaction

In principle ERO-TEXTOR supplies three possibilities of particle generation:

- particles sputtered by plasma ions, i.e. either sputtering by the background plasma or sputtering by a simulation particle,
- particles leaving the surface with an maxwellian energy distribution (chemical sputtering, sublimation) and
- particles supplied by an external source.

2.6.1 Sputtered Particles

Sputtering by Simulation particles

The physical sputtering yield due to ion bombardment on a material surface is calculated by ERO-TEXTOR using the revised *Bodhansky formula* [21]. For an ion with an impact energy E_0 and normal incidence $\alpha_0 = 0^o$ this gives a sputtering yield according:

$$Y(E_0, \alpha_0) = Q s_n^{KrC}(\epsilon) \left(1 - \left(\frac{E_{th}}{E_0}\right)^{\frac{2}{3}}\right)\left(1 - \frac{E_{th}}{E_o}\right)^2 \quad \left[\frac{\text{atoms}}{\text{ion}}\right] \tag{2.112}$$

Q – fit parameter

$$s_n^{KrC}(\epsilon) = \frac{0.5 \ln(1 + 1.2288\epsilon)}{\epsilon + 0.1728\sqrt{\epsilon} + 0.008\epsilon^{0.1504}}$$

nuclear stopping cross section based on the Kr-C potential

$$\epsilon = \frac{E_0}{E_{TF}}$$ – reduced energy

E_{th} – threshold energy

The angular dependence ($\alpha \neq 0^o$) of the sputtering yield is calculated according to *Yamamura et al.* [65]:

$$Y(E_0, \alpha) = Y(E_0, \alpha_0 = 0^o) \cdot (\cos\alpha)^{-f} e^{(f[1 - \frac{1}{\cos\alpha}]\cos\alpha_{opt})} \tag{2.113}$$

$$f = \sqrt{E_s}(0.94 - 1.33 \cdot 10^{-3}\frac{M_2}{M_1})$$

$$\alpha_{opt} = \frac{\pi}{2} - a_L n^{1/3}\left(2\epsilon\sqrt{\frac{E_s}{\gamma E_0}}\right)^{-1/2}$$

E_s – surface binding energy

M_1, M_2 – projectile and target mass

Z_1, Z_2 – projectile and target nuclear charge

$$a_L = \frac{0.4685}{\sqrt{Z_1^{2/3} + Z_2^{2/3}}}[\mathring{A}]$$ – *Lindhard* screening length

n – density of target material $\left[\frac{\text{atoms}}{\mathring{A}^3}\right]$

$$\gamma = \frac{4M_1M_2}{(M_1 + M_2)^2}$$ – energy transfer factor

Background Plasma Sputtering

While for a simulation particle the sputter yield can be calculated with the knowledge of impact energy and incidence angle according to *Bodhansky* and *Yamamura*, there are no "background plasma"-particles in ERO-TEXTOR which can be traced by the code. Thus the erosion yield on the material surface due to background plasma sputtering is calculated using the following assumptions:

- background plasma ions are maxwellian distributed in energy at $k_B T_e$ far away from the limiter and
- they are accelerated due to the sheath potential of $\approx 3k_B T_e$, satisfying the Bohm condition at the plasma-sheath edge.
- fuel and impurity ions possess an effective charge state Z_I (e.g. for the TEXTOR-SOL an effective charge state of $Z_I = 3$ for carbon ions is assumed).

Therefore the twice averaged sputtering yield $\bar{Y}_{I\to M}$, i.e. the yield averaged over the distributions of energy and angle of incidence is only a function of the electron temperature T_e and the charge state Z_I of the fuel/impurity ions, given by [1]:

$$\bar{Y}_{I\to M}(T_e, Z_I) = \frac{\int f_d Y_{I\to M}(E,\alpha) v_{\|} dv}{\int f_d v_{\|} dv_{\|}} \tag{2.114}$$

$$= \frac{\int f_d(E,\alpha) Y_{I\to M}(E,\alpha) \cos\alpha \; dE d(\cos\alpha)}{\int f_d(E,\alpha) \cos\alpha \, dE d(\cos\alpha)} \tag{2.115}$$

where

$Y_{I\to M}(E,\alpha)$ – sputtering yield according eqn. (2.113)

$f_d(E,\alpha)$ – maxwellian distribution function accelerated in the sheath potential of $\approx \frac{3k_B T_e}{e}$

Building the Sputter Distribution

After calculating the local sputtering yield $[\frac{\text{atoms}}{\text{ion surface-element}}]$ ERO-TEXTOR creates the sputtering distribution. This is done by building an energy and angular distribution for the sputtered atoms. For the energy distribution of the sputtered atoms a Thompsonian is assumed:

$$p(E) = \frac{E}{(E+E_s)^3} \qquad \text{where } E_s \text{ - surface binding energy} \tag{2.116}$$

The Thompsonian has an cut-off at the maximum energy which can be transfered from the incident ion to the released atom:

$$E_{max} = E_0 \cdot \gamma = E_0 \cdot \frac{4M_1 M_2}{(M_1+M_2)^2} \tag{2.117}$$

(Up to now E_{max} is set to be $4k_B T_e$ to reduce calculation time when not distinguishing between particles sputtered by different impurities).
Thus for each individual newly produced particle a random number ξ_E uniform in the interval:

$$\left]0 \;\; ; \int_0^{E_{max}} p(\tilde{E}) d\tilde{E}\right] = \left]0 \;\; ; \frac{1}{2E_s} - \frac{E_{max}+E_s/2}{(E_{max}+E_s)^2}\right] \tag{2.118}$$

is drawn and numerical inversion of eqn.(2.116) yields the starting energy of the particle.
The angular distribution of the sputtered atoms is assumed to be uniform in the azimuthal plane and cosine in the polar angle relative to the surface normal. To calculate the azimuthal angle a random number ξ_ϕ uniform in $[0, 2\pi[$ is drawn, while for the polar angle a random number ξ_θ uniform in $[0, 1]$ is drawn and the inversion:

$$\theta = \arcsin \xi_\theta \tag{2.119}$$

yields the desired cosine distribution.

2.6.2 Thermal Particles

Chemical Erosion for Carbon Substrates

In ERO-TEXTOR the chemical erosion of carbon is assumed to be dominated by the formation of hydrocarbons, especially to be dominated by the formation of methane (although it is well-known from laboratory experiments that a non-negligible fraction of the chemically eroded carbon, leaves the surface as higher hydrocarbons like C_2H_x or clusters, see e.g. [25], [62], [43]). In laboratory experiments it could be shown that the chemical erosion is mainly a function of the substrate temperature and has a maximum at $\approx 600^oC$. Furthermore the yield weekly depends on the energy of the incident particle and the particle flux to the surface. The erosion yield of methane due to irradiation of hydrogen is therefore described in ERO-TEXTOR by the following formula [48],[26]:

$$Y_{chem}(T, E_0, \Gamma) = \frac{6 \cdot 10^{19} e^{-1eV/k_BT}}{1 \cdot 10^{15} + 3 \cdot 10^{27} e^{-2eV/k_BT}} \cdot \left(200 Q S_n + 10^3 Y_{phys}(E_0) \left(\frac{\Gamma}{10^{16}} \right)^{-0.1} \right) \tag{2.120}$$

where Q, S_n and $Y(E_0)$ are taken from physical sputtering. Since the formation yield of methane can not exceed 0.25, a limiting condition has also to be respected:

$$Y'_{chem} = \frac{Y_{chem}}{1 + \frac{Y_{chem}}{0.25}} \tag{2.121}$$

and the broadening of the temperature dependence at low energies ($E_{in} < 100eV, T_{surface} < 1173^oC$) is taken into account in a somewhat crude form, by the additional condition that $Y''_{chem} = max(Y'_{chem}, Y_{low})$, where for the different hydrogen isotopes (H,D,T), the value of Y_{low} is 0.008, 0.04 and 0.1 respectively.

Yet a lot of open questions remain about the erosion yield under the high flux conditions of a fusion experiment. Under similar conditions erosion yields in between $\approx 2\% - 8\%$ [43], [44] have been measured. Therefore the code provides also the possibility to fix the erosion yield to a certain percentage of the incoming hydrogen flux $Y_{chem} = f_{hardwired} \Gamma_{H,D,T}$.

Building the Maxwellian Energy Distribution

As in sec. 2.6.1 the angular distribution is chosen to be cosine in the polar angle and uniform in the azimuthal plane. When assuming an maxwellian energy distribution:

$$p(E) \sim e^{-\frac{E}{E_0}} \tag{2.122}$$

the normalized generating probability function is given by:

$$P(E) = 1 - (\frac{E}{E_0} + 1) \cdot e^{-\frac{E}{E_0}} \tag{2.123}$$

Thus for each indivual newly produced particle a random number ξ_{Max} uniform in $]0;1[$ is drawn and numerical inversion of eqn.(2.123) yields the starting energy of the particle.

2.6.3 Reflected Particles

Reflection of Simulation Particles

There exist some analytical attempts to model the reflection of energetic ions on a smooth surface, as e.g. the single collision approximation [63] or to use empirical fit formulae (e.g. [59]). Since the validation of this approximations is still limited (for the single collision approximation the range of validity is limited to energies of the incident ions of $E \geq 100keV$), the ERO-TEXTOR code uses a probability approach extracted from the output of the Monte Carlo code TRIM. The TRIM code uses a binary collision approximation for the scattering of the ions in the bulk material, which is assumed to be amorphous. Energetic losses are due to elastic collisions with target atoms and nonlocal inelastic collisions with electrons. For detailed information on the computational procedure we refer to a review of Eckstein [19] or the book on Computer Simulation of Ion-Solid Interactions [20].
The procedure to extract the data is described in sec. 3.4.

Reflection of Background Plasma Particles

-not yet implemented-

With the same assumptions as for background plasma sputtering (see eqn.(2.114)) the reflection coefficient of background plasma (fuel and) impurities ions can be expressed as a twice averaged reflection coefficient only depending on the local electron temperature T_e and the effective charge state Z_I of the incident ions:

$$\bar{R}_{I\to M}(T_e, Z_I) = \frac{\int f_d R_{I\to M}(E,\alpha) v_{\|} dv}{\int f_d v_{\|} dv_{\|}} \tag{2.124}$$

$$= \frac{\int f_d(E,\alpha) R_{I\to M}(E,\alpha) \cos\alpha \; dE d(\cos\alpha)}{\int f_d(E,\alpha) \cos\alpha \, dE d(\cos\alpha)} \tag{2.125}$$

where

$R_{I\to M}(E,\alpha)$ – reflection coefficient of ions of species I incident on the target M

$f_d(E,\alpha)$ – maxwellian distribution function accelerated in the sheath potential of $\approx \frac{3k_B T_e}{e}$

2.6.4 External sources

In addition to sec. 2.6.1 and sec. 2.6.2 particles from external sources can be supplied by the code. In addition to the thompsonian and maxwellian energy distributions a well defined initial energy can also be chosen for these particles. For the angle distribution (which in this case do not refer to the surface normal but relative to the z-axis of the global coordinate system) it is possible to select between single valued, uniform, cosine, or the angle distribution, which results from the molecular flow of a gas through a cylindrical tube with a length to radius ratio, which can be freely chosen.

Angular Distribution under Molecular Flow Conditions

The condition for free molecular flow is that the free path length of a particle (molecule or atom) is large compared to the characteristic length of the vacuum vessel. Then the flow is determined by the fact that collisions between particles are very seldom and particle-wall collisions dominate the whole flow conditions. This flow condition is adequately described by the dimensionless Knudsen number, which is defined as the ratio mentioned before [64].

$$\kappa = \frac{\lambda}{d} \quad \text{Knudsen number,} \tag{2.126}$$

where $\lambda = \frac{1}{\sqrt{2}n\sigma}$ the free path length [16] and d the characteristic length of the vessel. From the Knudsen number it can be derived whether the conditions of laminar flow or free molecular flow apply.

$$\kappa > 0.5 \quad \text{free molecular flow} \tag{2.127}$$

$$\kappa < 0.01 \quad \text{laminar flow} \tag{2.128}$$

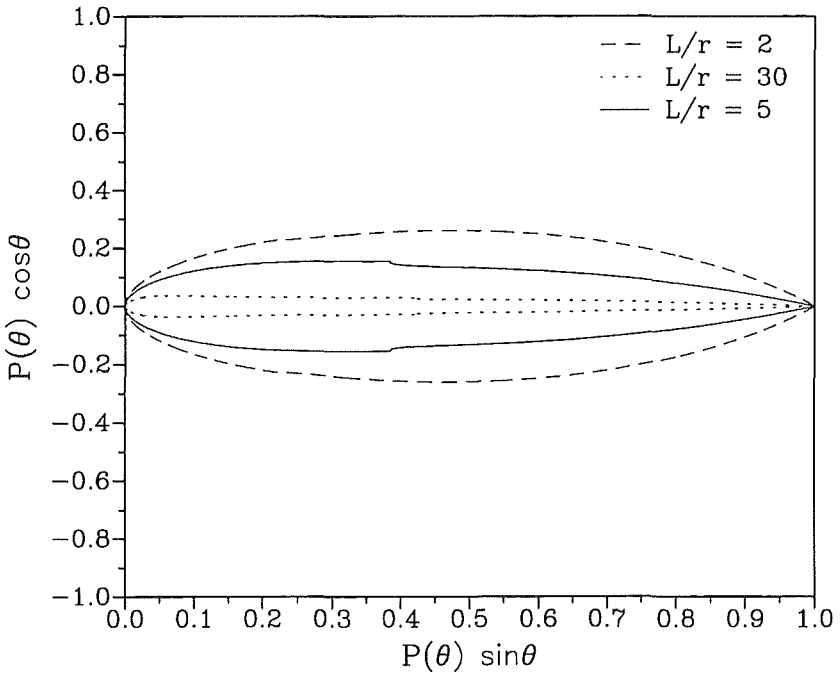


Figure 2.9: Angle distribution of particles transmitted through a cylindrical tube with $\frac{L}{r}$-ratio of 2, 5 and 30

Laminar flow is determined by the high collisionality of the particles and therefore the flow is determined by diffusion. In between $0.5 < \kappa < 0.01$ neither molecular flow nor laminar flow are fully developed and therefore it is theoretically hard to describe. Usually this parameter range can be interpolated from both sides.
Let us consider the gas flow through a cylindrical tube with radius $r = 1mm$ and a gas flow of $S = 5 \cdot 10^{17} \frac{particles}{s} \Rightarrow \Gamma = 1.6 \cdot 10^{19} \frac{particles}{cm^2 s}$ for example. For particles with a thermal velocity

of $v_{thermal} = 5.5 \cdot 10^5 \frac{mm}{s}$ and a cross section for molecular collisions of $\sigma = 4 \cdot 10^{-15} cm^2$ this yields the following parameters:

$$n = 2.9 \cdot 10^{14} \frac{1}{cm^{-3}} \quad \Rightarrow \quad \lambda = 0.61 \quad \Rightarrow \quad \kappa = 0.61$$

In this case this is well described under molecular flow conditions.
Under molecular flow conditions the probability distribution $P(p(\theta))$ of the polar angle θ of the flow of a gas through a cylindrical tube with radius r and length L and large vacuum vessels at each side of the tube can be described by the following formulae [17],[15]:

$$\begin{aligned} P(p \leq 1) &= \left(1 - \frac{2}{\pi}(1-a)\left[\arcsin(p) - p\sqrt{1-p^2}\right]\right. \\ &\quad \left. + \frac{4}{3p\pi}(1-2a)\left[1 - \sqrt{1-p^2}\right]\right)\cos(\theta) \end{aligned} \tag{2.129}$$

$$P(p \geq 1) = a + \frac{4(1-2a)}{3p\pi}\cos(\theta) \tag{2.130}$$

$$\begin{aligned} \text{with} \quad p &= \frac{L}{2r}\tan(\theta), \\ a &= \frac{[u\sqrt{u^2+1} - u^2] - [v\sqrt{v^2+1} - v^2]}{\frac{u(2v^2+1)-v}{\sqrt{v^2+1}} \frac{v(2u^2+1)-u}{\sqrt{u^2+1}}}, \\ u &= \frac{\sqrt{7}L}{3L + 2\sqrt{7}r} \quad \text{and} \\ v &= \frac{L}{2r} - v \end{aligned}$$

The formula is obtained by integrating over the angular distribution of the particles directly transmitted through the tube and the angular distribution of the particles reflected from the walls of the tube. The assumption is made that these particles are reflected with a cosine distribution from the walls. Fig. 2.9 shows the angular distribution of the transmitted particles for the length to radius ratios of $\frac{L}{r} = 2, 5$ and 30.

2.6.5 Plasma Surface Interaction

In sec. 2.6.1 and sec. 2.6.2 the principles of plasma surface interaction have already been discussed, namely that a plasma particle (fuel ion or impurity) strikes the material surface and when it is not reflected it is implanted into the surface. In addition it also may release other particles from the bulk material by virtue of chemical or physical erosion. In the preceding sections this was presented regarding the generation of new plasma particles and their specific energy and angle distributions. In this section the focus lais on the change of the surface properties itself caused by its irradiation with the ion flux of the plasma. These fluxes of different ionic species can either stick on the surface or erode material which then will be transported through the SOL plasma and will be redeposited with a certain probability. Accounting for all these fluxes, one obtains the change of the areal density of a material I at the location (x, y) according:

$$\frac{n_I(x,y)}{dt} = \Gamma_I^{Deposition}(x,y) + \Gamma_I^{Erosion}(x,y) + \Gamma_I^{Redeposition}(x,y) \tag{2.131}$$

The deposition flux is given by the flux of plasma particles I with a finite sticking probability S_I ($S_I = 1 - R_I$, R - reflection probability) on the surface

$$\begin{aligned} \Gamma_I^{Deposition}(x,y) &= \Gamma_I(x,y)(1-R_I) \\ &= f_I\Gamma_e(x,y)(1-R_I) \\ \text{with:} \quad \Gamma_I(x,y) &= f_I(x,y)\Gamma_e(x,y) \\ \Gamma_e &= \Gamma_\perp \cdot \cos(\alpha) \qquad \text{electron flux to the surface} \end{aligned} \tag{2.132}$$

f_I is the relative concentration of species I in the plasma and α the angle between surface normal and magnetic field direction. Since the fluxes to surfaces nearly parallel to the magnetic field lines are still about $\approx 10-20\%$ [54] of the fluxes to surfaces perpendicular to the magnetic field direction, the additional condition $\cos(\alpha) > 0.1$ is imposed.
The erosion flux of the surface species I is given by the sum of the different plasma particle fluxes Γ_P, the relative areal concentration c_I of the species I in the surface and the erosion yield $Y_{P\to I}$ of the species P impinging on the surface:

$$\begin{aligned} \Gamma_I^{Erosion}(x,y) &= \sum_P \Gamma_P(x,y)c_I(x,y)Y_{P\to I} \\ &= \sum_P f_P(x,y)\Gamma_e(x,y)c_I(x,y)Y_{P\to I} \end{aligned} \tag{2.133}$$

Finally the redeposition flux of specimen $\Gamma_I(x,y)$ at the location (x,y) is given by the flux of particles I eroded at any other location (x',y') on the limiter and transported through the plasma to this area. Since the transport through the plasma can not be described analytically, a transport operator $T((x',y') \to (x,y))$ is used to symbolize this transport:

$$\Gamma_I^{Redeposition}(x,y) = \int\int_{dS} \Gamma_I^{Erosion}(x',y')T((x',y') \to (x,y))(1-R_I)dx'dy' \tag{2.134}$$

$$\begin{aligned} &= \int\int_{dS} \sum_P (\Gamma_P(x',y')c_I(x',y')Y_{P\to I}) \times \\ &\quad \times T((x',y') \to (x,y))(1-R_I)dx'dy' \end{aligned} \tag{2.135}$$

The problem is then formally solved by simply integrating eqn.(2.131) or as done by ERO-TEXTOR by explicitly solving the equation for a discrete time interval and then adding up the changes:

$$n_I(x,y,t) = n_I(x,y,t_0) + \int_{t_0}^{t_1} \frac{dn_I(x,y,\tilde{t})}{dt} d\tilde{t} \tag{2.136}$$

$$\overset{\text{ERO-TEXTOR}}{\approx} n_I(x,y,t_0) + \sum_l \frac{dn_I(x,y,l\triangle t)}{dt} \triangle t \tag{2.137}$$

One problem remains: In principle for each newly arriving particle the actual surface concentration of all elements has to be known. Due to the finite range of the impinging ions in the bulk material the areal concentration of interest is of a depth Λ in between the depth range of the impinging ion and the depth range from which a sputtered particle can be released. Since the dependence of the sputter or the reflection yield on the surface may also be a nonlinear function of the areal concentration, it is suitable to define a formal relative areal concentration:

$$\hat{c}_I \equiv c_I(f(\Lambda)) \tag{2.138}$$

This formal relative areal concentration $\hat{c}_I$ of course can be quite different from the depth averaged areal concentration of a species

$$< c_I >= \frac{1}{\Lambda_{max}} \int_{\tilde{\Lambda}=0}^{\tilde{\Lambda}=\Lambda_{max}} c_I(\tilde{\Lambda})\, d\tilde{\Lambda} \tag{2.139}$$

and furthermore can also be different from the simple surface concentration $c_I(\Lambda)$ down to the relevant depth.

Attempts have been made [38] to describe this complicated behavior in a more general manner as described by eqn.(2.138). In this case Naujoks and Eckstein developed a model where the erosion yield is a function of a coverage parameter:

$$\hat{Y}_{P\rightarrow I} = \begin{cases} Y_{P\rightarrow I} \frac{n^P(t)}{n_0^P \Lambda^P} & \text{if } n^P(t) > \Lambda^P \\ Y_{P\rightarrow I} & \text{if } n^P(t) \leq \Lambda^P \end{cases} \tag{2.140}$$

where the last term $\frac{n^P(t)}{n_0^P \Lambda^P}$ represents this coverage parameter.

The ERO-TEXTOR code treats this coverage under the linear assumption that the erosion flux changes due the changes in the areal concentration only. The sputtering yield itself remains constant:

$$\hat{\Gamma}_{P\rightarrow I}^{Erosion} = c(\Lambda) \cdot Y_{P\rightarrow I} \tag{2.141}$$

The typical time step in a calculation is of the order $\triangle t \approx \frac{x}{10} sec.$ Thus the erosion and redeposition changes the gross thickness of the surface layer by typically:

$$\begin{aligned} \triangle d_{\text{surface layer}}(\triangle t) &= f_P \Gamma_e \triangle t \frac{1}{\text{areal density}} \\ \triangle d_{\text{surface layer}}(0.1s) &\approx 0.02 \cdot 10^{18} cm^{-2} s^{-1} \frac{1}{5 \cdot 10^{15} nm^{-1} cm^{-2}} \cdot 0.1s \\ \triangle d_{\text{surface layer}}(0.1s) &\approx \text{x nm} \end{aligned}$$

the parameter Λ can not be the physical relevant implantation depth of some ten $\mathring{A}$ and therefore $c_I(\Lambda)$ must a somewhat depth averaged effective areal density:

$$c_I(\Lambda) = \frac{1}{\Lambda_{max}} \int_{\tilde{\Lambda}=0}^{\tilde{\Lambda}=\Lambda_{max}} c_I(\tilde{\Lambda})\, d\tilde{\Lambda} \qquad \Lambda_{max} \approx x \cdot 10\, nm \ll d_{\text{surface layer}} \tag{2.142}$$

For each specific calculation the value of Λ can be chosen independent of any other parameter and has to be adjusted to be in accord with reasonable physical requirements.

Power Deposition on the Surface

To calculate the chemical erosion, the sublimation or the radiation enhanced sublimation (RES) the local temperature of the limiter surface has to be known. In ERO-TEXTOR this is done for the semi infinite material surface approximation. Then the heating of the limiter surface can be described by heat conduction equation [4]:

$$\vec{\nabla}\left(\kappa \vec{\nabla} T\right) + P - \rho c \frac{\partial T}{\partial t} = 0 \tag{2.143}$$

where the surface is given by $x = f(y,z)$ and the power pulse deposited on this surface by $P = P(x = f(y,z))$. With the assumption that κ (thermal conductivity) and c (specific heat) are independent of the temperature one gets from eqn.(2.143):

$$\frac{\partial^2 T(x,t)}{\partial x^2} + \frac{P}{\kappa} - \frac{\rho c}{\kappa}\frac{\partial T(x,t)}{\partial t} = 0 \tag{2.144}$$

which can be solved analytically by:

$$T(x,t) = T(0,0) + \left(\frac{2P}{\sqrt{\pi\rho c\kappa}}\sqrt{t}\right)\cdot\left(e^{-u^2} - 2u\int_u^\infty e^{-\xi^2}d\xi\right) \tag{2.145}$$

where $u = \frac{x}{2\sqrt{t}}\sqrt{\frac{\rho c}{\kappa}}$ is a length related to the material and pulse time. Finally for the temporal evolution of the surface temperature one gets from eqn.(2.145), neglecting the thermal diffusion into the bulk:

$$T(0,t) = T(0,0) + \left(\frac{2P}{\sqrt{\pi\rho c\kappa}}\sqrt{t}\right) \tag{2.146}$$

The power flux from the plasma deposited on the limiter surface is

$$P(n_e,T_e) = \Gamma_e \delta k_B T_e \qquad \text{or in practial units:} \tag{2.147}$$

$$P(n_e,T_e)[\frac{W}{cm^2}] = 1.603\cdot 10^{-19}\delta n_e[\frac{1}{\mathrm{cm}^3}]T_e^{3/2}[\mathrm{eV}]\sqrt{\frac{1.93\cdot 10^{12}}{m_{plasma}[\mathrm{amu}]}} \tag{2.148}$$

where $\delta \approx 7-8$ [56] is the energy transmission coefficient of the sheath - the proportionality factor relating power flux to (particle flux)$\times k_B T_e$.

Chapter 3

Formatted Input-Files

For each Run ERO-TEXTOR requires five specific formatted input file systems, which are:

1. Configuration File (ref. section 3.1)
2. Rate Coefficients Files (ref. section 3.2)
3. Sputtering Data Files (ref. section 3.3)
4. Reflection Data Files (ref. section 3.4)
5. Parameter File for the Individual Run (ref. section 3.5)

In this chapter these five different input file systems are described.
WARNING:
Since ERO-TEXTOR never checks for man made errors, please take care that all KEYWORDS are correctly spelled. On the following pages all keywords are written in typewriter letters. After all KEYWORDS there must be one blank and then the line feed. The specification of the keyword has to be in the next line.

3.1 Configuration File

Most important before starting with ERO-TEXTOR, the configuration file

`MERLIN.CFG`

has to be in the same folder as the code itself or from where the code should be started. In this configuration file ERO-TEXTOR requires the pathes to the graphics screen drivers, the ionization cross-section, the reflection database and the sputter data.

`DISCLAIM`

Here ERO-TEXTOR expects the full path and file name (`/usr/local/disclaim.dat`) to the copyright and disclaim file. In addition to the programmers disclaim notes the file contains latest update notices not yet reported in this manual.

`DRIVER`

Only the visual demonstration version of ERO-TEXTOR (MERLIN.EXE) needs the path to the graphics screen drivers. For the scientific code ERO-TEXTOR no path to drivers has to be specified, but the keyword must be found in the configuration file.

`CROSS_SECTION`

Here ERO-TEXTOR expects only the path to the ionization cross-sections. For the standard nomenclature of the data files (ref. section 3.2 and tab. 3.1).

`RESULTS`

Without any use at that time

`REFLECTION_DATA`

Expecting the path and file name (e.g. `/usr/local/reflection.dat`) of the reflection database (ref. section 3.4).

`DATA`

Followed by the path and file name (e.g. `/usr/local/sput.kor`) of the data for calculating the sputter yield according to *Bodhansky* and *Yamamura* (ref. section 3.3.1).

`SPUTTER_INTEGRAL`

Here ERO-TEXTOR expects the path and file name (e.g. `/usr/local/i_sput.dat`) of the data for 'calculating' the sputter yield of the background plasma according to *Abramov et al.* (ref.section 3.3.2).

`EMISSION`

Here ERO-TEXTOR expects only the path to the emission rate files.

3.2 Ionization Rates

In this section the formate of the files containing cross sections due to *electron impact* and the calculation procedure to obtain the individual cross section is explained. The first paragraph deals with cross sections for atomic species (e.g cross sections for carbon $[C]$ or tungsten $[W]$), while the second paragraph describes molecular species (e.g. silane $[SiH_4]$).

3.2.1 Ionization Rates For Atomic Species

ERO-TEXTOR already provides the ability to handle a variety of atomic species (see tab. 3.1 for a full list of these species) of interest in fusion plasmas. For these species cross sections as published in [6],[30],[27] are provided with the ERO-TEXTOR-package. For each element all ionization cross sections are contained in an individual file, which must obey a general formate. If the user wants to do the simulations with other cross sections only the cross section file has to be changed (but still keeping the general formate). The situation becomes more complicated when a simulation should be done with an element which is not supported yet, because then the code has to be recompiled (for this situation please refer to section 5.1). In this section only the standard formate of cross section files is described! As explained in sec. 3.1 the configuration file contains only the path where ERO-TEXTOR searches the files for ionization cross sections. For each element the cross section file name follows the standard ERO-TEXTOR nomenclature (ref. tab. 3.1 for element names) followed by `.dat` (e.g the cross sections for carbon must be in a file named `_c.dat`).

Calculating the rate coefficients $< \sigma \cdot v >$

As described in [6] the rate coefficients are calculated according to the following formula:
For a electron temperature range $\frac{I}{10} \leq kT_e \leq 10 \cdot I$, where I is the ionization potential

$$< \sigma \cdot v > (T_e) = e^{\frac{-I}{kT_e}} \sqrt{\frac{kT_e}{I}} \sum_{n=0}^{m} a_n \left(\log_{10} \left(\frac{kT_e}{I} \right) \right)^n \tag{3.1}$$

and for the electron temperature range $kT_e \geq 10 \cdot I$

$$< \sigma \cdot v > (T_e) = \sqrt{\frac{I}{kT_e}} \left[\alpha \ln \left(\frac{kT_e}{I} \right) + \sum_{i=0}^{k} \beta_i \left(\frac{kT_e}{I} \right)^{-i} \right] \tag{3.2}$$

File Formate

According eqns.(3.1),(3.2) ERO-TEXTOR demands the values for the ionization potential I, the coefficients a_n $(n = 0, 1, \ldots, 5)$, the value for the coefficient α and for the coefficients β_i $(i = 0, 1, 2)$. In addition information on the number of available rate coefficients has to be provided to ERO-TEXTOR.

Keywords	Description
2	Number of Ionization states
0	Coefficients of $X^0 \rightarrow X^{1+}$ follow:
I	keyword for ionization potential
11.26	ionization potential
a	keyword, refer eqn.(3.1)
5.9848e-08	a_0
1.1903e-08	a_1
-3.0140e-08	a_2
-1.3693e-08	a_3
8.3748e-09	a_4
4.0150e-09	a_5
alpha	keyword, refer eqn.(3.2)
3.7442e-07	α
beta	keyword, refer eqn.(3.2)
-6.5825e-07	β_0
2.0521e-06	β_1
-4.4694e-06	β_2
1	Coefficients of $X^{1+} \rightarrow X^{2+}$ follow:
I	keyword for ionization potential
24.38	ionization potential
a	keyword, refer eqn.(3.1)
2.8395e-08	a_0
-1.6698e-08	a_1
-2.3557e-09	a_2
3.2161e-10	a_3
9.6016e-10	a_4
5.2713e-10	a_5
alpha	keyword, refer eqn.(3.2)
6.0150e-08	α
beta	keyword, refer eqn.(3.2)
-4.0217e-08	β_0
-2.7908e-08	β_1
5.5499e-07	β_2

Error handling:
If ERO-TEXTOR cannot find the necessary coefficients (e.g. only coefficients for the ionization up to charge state $q = 3$ are provided but the simulation demands the cross section for $X^{3+} \rightarrow X^{4+}$) or the calculation of the rate coefficient yields $< \sigma \cdot v >= 0$, then ERO-TEXTOR will put:

$$t_{ion}(X^{q+} \rightarrow X^{(q+1)+}) = \frac{1}{< \sigma \cdot v > n_e} \equiv 1 \quad [sec],$$

which is usually equivalent to infinity.

3.2.2 Ionization Rates For Molecular Species

Methane Molecules

For methane molecules CH_4 ERO-TEXTOR provides the ionization rates as published by Ehrhardt et al. [22]. Fig. 3.1 shows the possible processes for electron impact, whereas for-

mula (3.3) show the possible proton-impact processes.

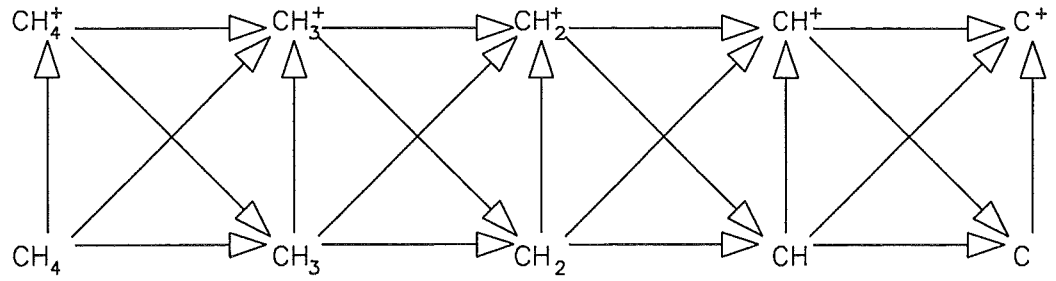


Figure 3.1: Possible electron-impact processes for the methane molecule. The hydrogen released in these processes is not included.

$$p + CH_X \rightarrow CH_X^+ + H \tag{3.3}$$

In contrast to atomic ionization processes (see eqn.(2.44)) there exist several possible processes competing for one specific molecule. Therefore to compute whether a process and which process happens the following procedure is used:
First a random number ξ_c (uniform in $]0;1[$) is drawn and compared to the value of

$$\frac{1}{\tau_c} = \sum_{all\ processes} \frac{1}{n_e < \sigma \cdot v >_{electron}} + \sum_{all\ processes} \frac{1}{n_p < \sigma \cdot v >_{proton}} \tag{3.4}$$

if $\xi_c < \tau_c$ then a process will occur and it has to be decided which process will win. Thus the rates for all possible processes are normalized to the sum:

$$\tau_i = \frac{1}{\tau_c} \cdot n_{e,p} < \sigma \cdot v >_{e,p} + \tau_{i-1} \tag{3.5}$$

Then a second random number ξ_{which} (uniform in $[0;1[$) is drawn and the process for which the equation:

$$\tau_{i-1} \leq \xi_{which} \leq \tau_i \tag{3.6}$$

holds is the "winner".
As described in [22] the rates due to electron-impact processes are fitted by the following equation:

$$\ln < \sigma \cdot v >= \sum_{n=0}^{8} b_n (\ln T_e)^n \tag{3.7}$$

whereas the rates due to proton-impact processes are fitted by a double logarithmic polynomial:

$$\ln < \sigma \cdot v >= \sum_{i=0}^{8} \sum_{j=0}^{8} a_{ij} (\ln E_{proton})^i (\ln T_e)^j \tag{3.8}$$

All this rates will not be loaded during runtime, but they are already provided with the routine `void prozesses()` (see sec. 5.4).

Silane Molecules

Since there exists no published rate coefficient scheme for silane like for the methane molecule a new rate coefficient scheme was developed based on that of Fantz et al [23],[50]. The total dissociation cross sections were measured by Perrin et al. [40]. As in the work of Ehrhardt and Langer the fundamental cross sections for dissociative ionization are taken from the work of Chatam et al. [12]. Furthermore an isotope effect has been detected by Perrin et al. [41] of about $10 - 20\%$ higher ionization cross sections for deuterated silane SiD_4. Fig. 3.2 shows the electron induced reactions of the silane molecule. Since the SiH_4^+-ion has not been detected yet and it is quite certain [24],[33] that it predissociates into SiH_2^+ the reaction chain is different from that of methane. The released hydrogen atoms / molecules are not included in the figure.

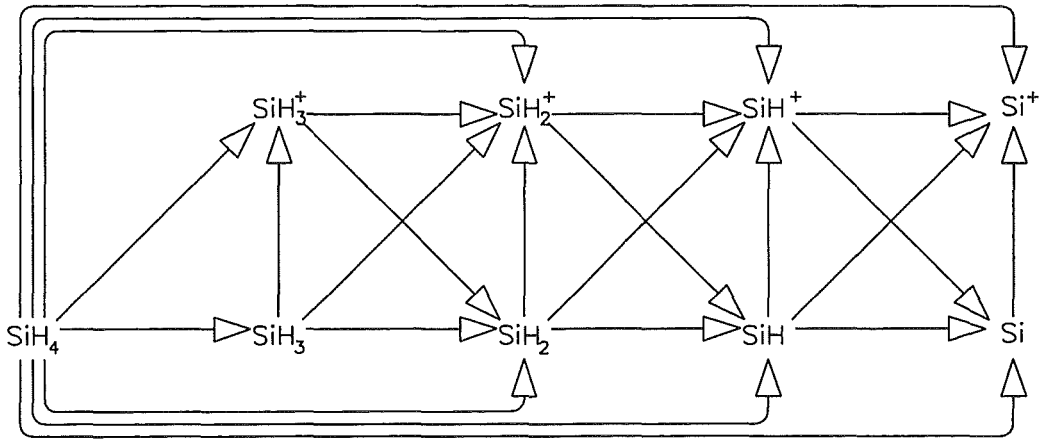


Figure 3.2: Possible electron-impact processes for the silane molecule from [23].

The rate coefficients were fitted according to Ehrhardt and Langer:

$$< \sigma \cdot v >= b \cdot \exp\left(\sum_{n=0}^{8} b_n (\ln T_e)^n\right) \tag{3.9}$$

In Appendix B the values for the fitting formulae and the graphical presentation of the calculated rate coefficients can be found. The rate coefficient system will be discussed in detail elsewhere [28].

3.3 Sputtering Data

3.3.1 Data for Simulation Particles

For simulation particles the individual sputter yield is calculated as a function of energy E_{in} and angle θ_{in} according to the *Bodhansky*- and *Yamamura*-Formula (eqn.(2.112), (2.113)). The necessary (formatted) input data are listed in the following table (see also [21]):

Keyword/Data	Description
	Projectile – Target combination according
	standard ERO-TEXTOR nomenclature
_hli	first 2 characters represent the projectile and second the target
1.67000	E_s - Target Surface Binding energy (in this case of Li) [eV]
6.88000	mass ratio = $\frac{M_{target}}{M_{projectile}}$
1.00000	$Z_{projectile}$
3.00000	Z_{target}
1.00790	mass projectile $M_{projectile}$
185.000	E_{TF}[eV]
6.22000	E_{th} threshold energy [eV]
0.10000	Q [$\frac{atoms}{ions}$]
0.04636	density of target material [$\frac{atoms}{\mathring{A}^3}$]
-------	data followed by blank line !!
_dli	next projectile-target combination
1.67000	E_s - Target Surface Binding energy (in this case of Li) [eV]
3.45000	mass ratio = $\frac{M_{target}}{M_{projectile}}$
1.00000	$Z_{projectile}$
...	

3.3.2 Data for the background plasma

For background plasma sputtering ERO-TEXTOR needs the pre-calculated integral yield Y ($[\frac{atoms}{ion}]$) of the relevant impurity I (projectile) – bulk material B (target) combination according eqn.(2.114), which is a function of charge state q and electron temperature T_e: Path and file name have to be specified in the configuration file and ERO-TEXTOR searches in this file (only one file for all projectile–target combinations) for the string IBq, where I and B obey the standard ERO-TEXTOR nomenclature (see tab. 3.1). The following table shows a part of the formatted input file:

Keyword/Data		Description
		Projectile–Target combination followed by charge state
_c_w4		of projectile; in this case $C^{4+} \rightarrow W$
1.000000	6.598122e-16	followed by data:
5.000000	6.301306e-03	first column : electron temperature T_e
10.000000	3.779705e-02	second column : Sputter Yield $Y(T_e)[\frac{atoms}{ion}]$
15.000000	6.401654e-02	(in this case $Y(T_e) = Y_{C^{4+}\rightarrow W}(T_e)$)
20.000000	8.791359e-02	
25.000000	1.068258e-01	electron temperature T_e has not necessarily to be
30.000000	1.239871e-01	incremented as in this example,
35.000000	1.384109e-01	but ERO-TEXTOR needs 15 (T_e, $Y(T_e)$) toubles
40.000000	1.513649e-01	
45.000000	1.630899e-01	
50.000000	1.737422e-01	
75.000000	2.152887e-01	
100.000000	2.450095e-01	
150.000000	2.858854e-01	
200.000000	3.133727e-01	followed by
_cpt1		next Projectile–Target combination
1.000000	1.585326e-15	(in this case $C^{1+} \rightarrow Pt$)
5.000000	5.401861e-04	...
10.000000	8.794207e-03	
15.000000	2.374422e-02	
20.000000	3.975607e-02	
25.000000	5.493518e-02	
...		

In Appendix C the sputtering yields for the most relevant elements in fusion (namely Lithium, Boron, Beryllium, Carbon, Silicon, Iron, Nickel, Copper, Molybdenum and Tungsten) due to impact of Hydrogen, Deuterium, Tritium, Helium, Carbon and Oxygen in the energy range of $T_e = 1 - 200eV$ calculated according eqn.(2.114) are presented. In addition the selfsputtering yields of these elements in the same energy range can be found there.

3.4 Reflection Data

Although there exist some approaches to describe the reflection of light ions on fusion relevant materials [3] with analytical formulae, the ERO-TX code uses the reflection data calculated with the TRIM code. The TRIM code calculates the reflection of incindent ions of distinct impact energy and angle and therefore a large set of data has to be read to describe this process.

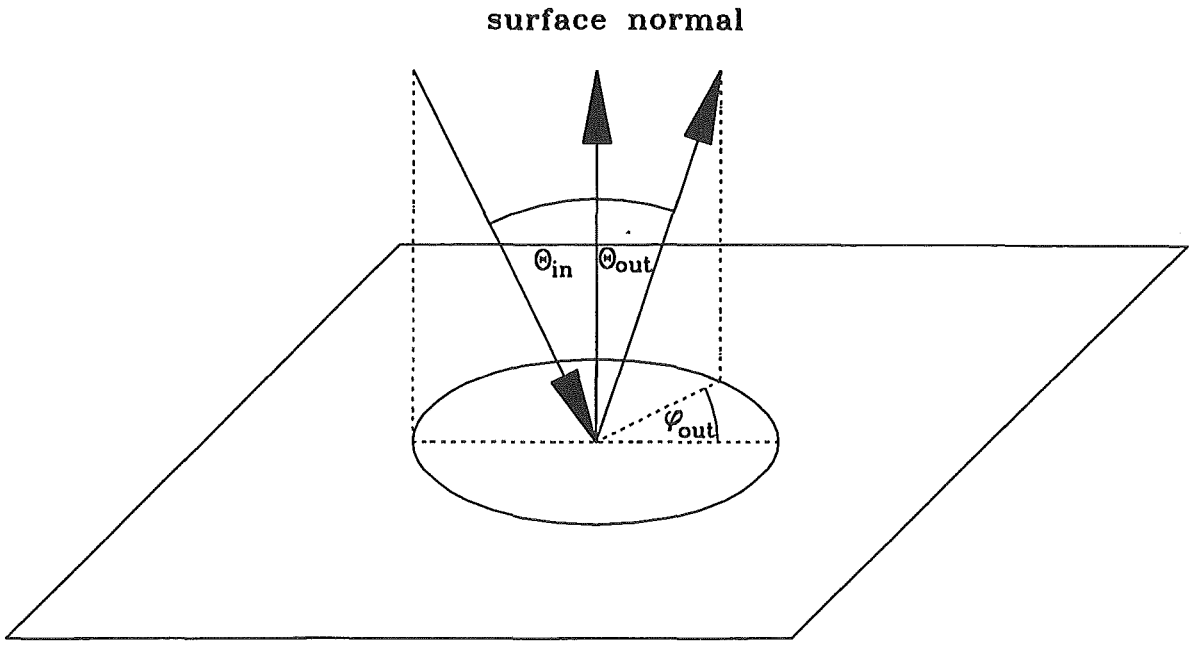


Figure 3.3: The reflection geometry. The surface normal and the direction of incidence define the plane of incidence which is the referential plane for ϕ_{out}.

According to Eckstein [18],[19] a set of 3 matrices is sufficient to describe the reflection probability for an incident particle with energy of incidence E_{in} and the angle to the surface normal θ_{in}. These matrices are the reflection probability for the energy E_{out} of the reflected particle, its polar angle θ_{out} and its azimuthal angle ϕ_{out} each described by its values x_l for the cumulative probabilities $F(\xi) = (\int_0^x f(x)dx)^{-1}$ of 0.1, 0.3, 0.5, 0.7, 0.9, where $f(x)$ is the probability distribution of the specific variable ($x = E, \theta, \phi$)(for the geometry ref. fig(3.3)). Therefore the first matrix (a 1×5 matrix) gives the energy of reflected particles for the five mentioned cumulative probabilities $E(\xi_i)$. The second matrix (a 5×5 matrix) gives the cosine of the polar angle for each of the energy distributions and the cumulative probability functions: $\theta(E(\xi_i), \xi_j)$. Finally the last matrix (a $5 \times 5 \times 5$ matrix) gives the azimuthal angle of the reflected particle for each energy and each polar angle distribution and the cumulative probability function $\theta(E(\xi_i), \theta(\xi_j), \xi_k)$.

Each of these sets describes the reflection probability of one well-defined set of energy of incidence E_{in} and angle of incidence θ_{in}. To cover the whole parameter range relevant to the conditions at a limiter / divertor surface a number with different energies E_{in} and angles θ_{in} have to be provided to the code. For the standard code these are 84 different conditions of incidence (twelve energies $E_{in} = 1, 2, 5, 10, 20, 50, \ldots, 2 \cdot 10^3, 5 \cdot 10^3$ eV and seven angles $\theta_{in} = 0°, 30°, 45°, 60°, 70°, 80°, 85°$).

#next:energy,angle,backscattered,reflection coefficient,125-matrix of data				
2.000000e+01	*E_{in} energy of incident particle*			
8.000000e+01	*θ_{in} angle to surface of incident particle*			
5.115000e-02	*reflection yield*			
2.67877E+00	4.55522E+00	6.02263E+00	7.53862E+00	9.51911E+00
		empty line		
1.81526E-01	3.26172E-01	4.52497E-01	5.74997E-01	7.40569E-01
1.74020E-01	3.10293E-01	4.10812E-01	5.09556E-01	6.66830E-01
1.54706E-01	2.77501E-01	3.69626E-01	4.70190E-01	6.14851E-01
1.33451E-01	2.41643E-01	3.27422E-01	4.14027E-01	5.50204E-01
1.07820E-01	1.97216E-01	2.59270E-01	3.37261E-01	4.52496E-01
		empty line		
5.75808E-01	8.29427E-01	9.24492E-01	9.73070E-01	9.97033E-01
5.36295E-01	7.90592E-01	9.15234E-01	9.70466E-01	9.95214E-01
5.81162E-01	8.04515E-01	9.15759E-01	9.76323E-01	9.98271E-01
5.55628E-01	8.00283E-01	9.07271E-01	9.71484E-01	9.97678E-01
5.17713E-01	7.77263E-01	8.94983E-01	9.68284E-01	9.98087E-01
6.63699E-01	8.65566E-01	9.33046E-01	9.77695E-01	9.97334E-01
6.89913E-01	8.45810E-01	9.34429E-01	9.80412E-01	9.97666E-01
6.93771E-01	8.58610E-01	9.33808E-01	9.72887E-01	9.95684E-01
6.76320E-01	8.49495E-01	9.34202E-01	9.80105E-01	9.96835E-01
7.72961E-01	8.95987E-01	9.54578E-01	9.85446E-01	9.98321E-01
7.45364E-01	8.81586E-01	9.45934E-01	9.82843E-01	9.97068E-01
7.04904E-01	8.80835E-01	9.46526E-01	9.82907E-01	9.97176E-01
7.56732E-01	9.01553E-01	9.63240E-01	9.88047E-01	9.99027E-01
7.59460E-01	8.97484E-01	9.49299E-01	9.81760E-01	9.98969E-01
7.74468E-01	9.00324E-01	9.53271E-01	9.82670E-01	9.98978E-01
7.85319E-01	9.11117E-01	9.60385E-01	9.85610E-01	9.97999E-01
7.80133E-01	9.10497E-01	9.54568E-01	9.84362E-01	9.98344E-01
8.20946E-01	9.19789E-01	9.63784E-01	9.89308E-01	9.98763E-01
8.07238E-01	9.20962E-01	9.68040E-01	9.89506E-01	9.99251E-01
8.14715E-01	9.14643E-01	9.70179E-01	9.90570E-01	9.98960E-01
8.79323E-01	9.50826E-01	9.78811E-01	9.92090E-01	9.99078E-01
8.87811E-01	9.55746E-01	9.79952E-01	9.93411E-01	9.99443E-01
8.80297E-01	9.46979E-01	9.77359E-01	9.91480E-01	9.99296E-01
8.85295E-01	9.58218E-01	9.81377E-01	9.94948E-01	9.99358E-01
9.04408E-01	9.57319E-01	9.79117E-01	9.92574E-01	9.99093E-01
		empty line		
#next:energy,angle,backscattered,reflection coefficient,125-matrix of data				
2.000000e+01	*energy of incident particle*			
8.000000e+01	*angle to surface of incident particle*			
5.859000e-02	*reflection yield*			
...	...	...	...	...

3.5 Parameter File Composing Individual Run

For each individual run ERO-TEXTOR expects a parameter input file in the *same* folder as the code. The name of this file can be chosen freely because ERO-TEXTOR request the file name (e.g. for a UNIX-system the code-start can be of the form: `echo "param.ero" | ero.prg`). In this section all reasonable options for ERO-TEXTOR are presented. Due to the fact that this version of ERO-TEXTOR is still a moving target some obvious inconsistencies can be found and will hopefully be removed in the next version!! ERO-TEXTOR allows for the parameter file not only keywords and keyword-data, but also comments, which have to be appended to the following letter: `#␣`. The following listing is complete in keywords, but in ordering the user has the freedom to mix frankly, although for the purpose of comprehensive overview it is recommended to keep this ordering.

3.5.1 Data For Numeric Dialog

- `# numeric dialog`

  example for a comment line!

- `dt1`

  time step $\triangle t$ for solving the equations of motion for the specific particle.

- `dt2`

  time step near the limiter surface to avoid cross miscalculation due to large $d\vec{v}/dt$ caused by the sheath potential. In the next version this time will be dynamically coupled to the Debye-length λ_D.

- `collision_time`

  Time for the collision cycle (typically $\approx 10 \cdot$ `dt1`). In this cycle friction and collision forces are calculated.

- `time_out`

  After this time simulation will end for a particles, if it is not transported out of the tracing box or deposited on the limiter (typically $\approx 10^{-4}s$).

- `radial_box`

  radial length of the tracing box in millimeters (corresponding to the z-coordinate of the global coordinate system), measured from the tip of the limiter (see fig. 3.4).

- `poloidal_box`

  poloidal length of the tracing box in millimeters (corresponding to the y-coordinate of the global coordinate system). The symmetry point of the limiter is located at $\frac{poloidalbox}{2}$ (see fig. 3.4).

- `toroidal_box`

  toroidal length of the tracing box in millimeters (corresponding to the x-coordinate of the global coordinate system). The symmetry point of the limiter is located at $\frac{toroidalbox}{2}$ (see fig. 3.4).

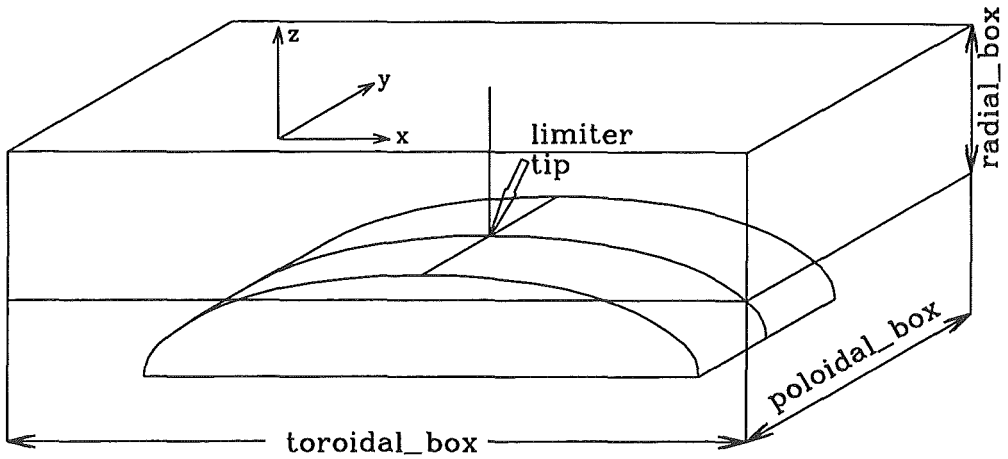


Figure 3.4: Visualization of the parameters of the tracing box. The parameter radial box is calculated from the tip of the limiter but the tracing box extends over the whole height of the limiter.

- `pathtoresult`

  Full path and first letters of filename (under a UNIX-system the user is free in designing the filename). ERO-TEXTOR will append to 'filename' the standard nomenclature output names of the form `abcde.ttt` (depending on the special demands of the run; for nomenclature of output files see chapter 4).
  Example: user defined input is `/usr/local/result/OUT` which will provide the output file for net transport in the first time step for molybdenum:
  `/usr/local/result/OUTOmo_T.T1`

- `job_class`

Input Number	Action
1	creates a chainjob for the LoadLeveler in the job class with 1 hour cpu time
10	creates a chainjob for the LoadLeveler in the job class with 10 hour cpu time

- `physical_erosion`

Input Number	Action
0	physical sputtering due to background plasma switched on
1	physical sputtering due to background plasma switched off

- `chemical_erosion`

Input Number	Action
0	chemical sputtering of carbon due to background hydrogen (methane formation) switched on
1	chemical sputtering of carbon due to background hydrogen (methane formation) switched off

- `hardwired_chemical_erosion`

Input Number	Action
< 0	yield of chemical sputtering according the Roth-formula eqn.(2.120) (function of surface temperature, physical sputtering yield and hydrogen flux)
> 0	hard wired chemical sputtering (function of hydrogen flux)

- `oxygen_erosion`

Input Number	Action
0	chemical sputtering of carbon due to background oxygen (carbonmonoxyde formation) switched on
1	chemical sputtering of carbon due to background oxygen (carbonmonoxyde formation) switched off

- `background_deposition`

Input Number	Action
0	deposition of background plasma carbon switched on
1	deposition of background plasma carbon switched off

- `particle_sputtering`

Input Number	Action
0	physical sputtering due to traced impurity switched on
1	physical sputtering due to traced impurity switched off

- `particle_reflection`

Input Number	Action
0	reflection of traced impurity switched on
1	reflection of traced impurity switched off

3.5.2 Background Plasma Data

- `#plasma dialog`
  `plasmaions`

  input must be an integer number 0,1 or 2, where 0 – Hydrogen (H), 1 – Deuterium (D) and 2 – Tritium (T), for the fuel plasma ions.

- `helium`

  Expecting an integer number $i \in [0, 100]$ for the percentage of helium impurities in the fuel plasma, e.g. if the fuel plasma ions are hydrogen ions and there are no other impurities specified, then `helium` $= 100 \cdot \frac{He_{atoms}}{H_{atoms}+He_{atoms}}$.

- `he_abfall`

  simple exponential decay length of impurity content of the background plasma according:

$$\Gamma_{helium}(z) = \Gamma_{helium}(z_{lcfs}) \cdot e^{-z/he_{abfall}}$$

  if `he_abfall` ≤ 0 then the helium content will increase from core plasma to SOL plasma. Please notice that the shape of the Γ_{helium} is independent of the chosen electron density form.

- `he_charge`

  the mean charge state of the helium ions in the SOL (default is set to `he_charge`= 2).

- `carbon`

  Expecting an integer number $i \in [0, 100]$ for the percentage of carbon impurities in the fuel plasma, e.g. if the fuel plasma ions are hydrogen ions and there are no other impurities specified then `carbon` $= 100 \cdot \frac{C_{atoms}}{H_{atoms}+C_{atoms}}$.

- `c_abfall`

  simple exponential decay length of impurity content of the background plasma according:

  $$\Gamma_{carbon}(z) = \Gamma_{carbon}(z_{lcfs}) \cdot e^{-z/c_{abfall}}$$

  if `c_abfall` ≤ 0 then the carbon content will increase from core plasma to SOL plasma. Please notice that the shape of the Γ_{carbon} is independent of the chosen electron density form.

- `c_charge`

  the mean charge state of the carbon ions in the SOL (default is set to `c_charge`= 4).

- `oxygen`

  Expecting an integer number $i \in [0, 100]$ for the percentage of oxygen impurities in the fuel plasma, e.g. if the fuel plasma ions are hydrogen ions and there are no other impurities specified then `oxygen` $= 100 \cdot \frac{O_{atoms}}{H_{atoms}+O_{atoms}}$.

- `o_abfall`

  simple exponential decay length of impurity content of the background plasma according:

  $$\Gamma_{oxygen}(z) = \Gamma_{oxygen}(z_{lcfs}) \cdot e^{-z/o_{abfall}}$$

  if `o_abfall` ≤ 0 then the oxygen content will increase from core plasma to SOL plasma. Please notice that the shape of the Γ_{oxygen} is independent of the chosen electron density form.

- `o_charge`

  the mean charge state of the oxygen ions in the SOL (default is set to `o_charge`= 5).

- `te`

  (background) electron temperature T_e at the LCFS in $[eV]$. If instead of a real number a string with the suffix "`.dat`" (e.g. "`te.dat`") is supplied, the code will expect a column of real numbers in a file with that name, each defining the electron temperature for one subsequent time step. The code reads the first number and deletes it from the table.

- `te_abf`

  e-folding decay length $[mm]$ of electron temperature from the LCFS according

  $$T_e(z) = T_e(z = z_{LCFS}) \cdot e^{-z/te_{abf}}$$

  If `te_abf`= 0 then $T_e = const.$

- `ti`

  (background) ion temperature T_i at the LCFS in $[eV]$. If instead of a real number a string with the suffix ".dat" (e.g. "ti.dat") is supplied, the code will expect a column of real numbers in a file with that name, each defining the ion temperature for one subsequent time step. The code reads the first number and deletes it from the table.

- `ti_abf`

  e-folding decay length $[mm]$ of ion temperature from the LCFS according

  $$T_i(z) = T_i(z = z_{LCFS}) \cdot e^{-z/ti_{abf}}$$

  If `ti_abf`$= 0$ then $T_i = const.$

- `ne`

  electron density n_e at the LCFS in $[\frac{1}{cm^3}]$. If instead of a real number a string with the suffix ".dat" (e.g. "ne.dat") is supplied, the code will expect a column of real numbers in a file with that name, each defining the electron density for one subsequent time step. The code reads the first number and deletes it from the table.

- `ne_abf`

  radial e-folding decay length $[mm]$ of electron density from the LCFS according (far away from the test limiter):

  $$n_e(z, x = \infty) = n_e(z = z_{LCFS}, x = \infty) \cdot e^{-z/ne_{abf}}$$

  If `ne_abf`$= 0$ then $n_e = const.$ If `ne_abf`< 0 then

  $$\begin{aligned} n_e(z, x = \infty) &= n_e(z = z_{LCFS}, x = \infty) \cdot e^{-z/\lambda_{ne}} \\ \lambda_{ne} &= \sqrt{\frac{D_\perp \cdot L_c}{c_s}} \end{aligned}$$

- `ne_factor`

  if `ne_abf` is set ≤ 0, i.e. the SOL decay length of the electron density is calculated according $\lambda_{ne} = \sqrt{\frac{D_\perp \cdot L_c}{c_s}}$, this factor can be used to scale λ_{ne}.

- `bfield_on`

Input Number	Action
0	With Magnetic Field
1	No Magnetic Field

- `bfield`

  Magnetic Induction Strength $\|\vec{B}_0\|$ in [Tesla].

- `pol_angle_of_bfield`

  azimuthal angle ϕ_B between x-axis and unit vector along the magnetic field lines.

  $$\phi_B = \angle(\vec{e}_x, \frac{\vec{B}}{\|\vec{B}\|}) \quad \in]0, 180[\quad \text{unit: } [^\circ]$$

- `tor_angle_of_bfield`

  polar angle ϑ_B between z-axis and unit vector along the magnetic field lines.

$$\vartheta_B = \angle(\vec{e}_z, \frac{\vec{B}}{\|\vec{B}\|}) \quad \in]0, 90[\quad \text{unit: } [^\circ]$$

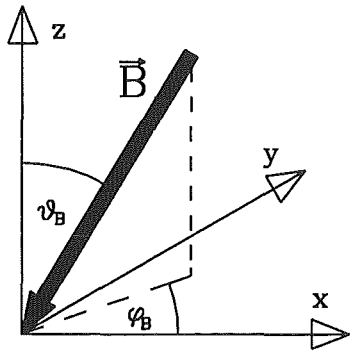


Figure 3.5: Orientation of the magnetic field vector $\vec{B}$ relative to the global coordinate system.

This item, `pol_angle_of_bfield` and `bfield` define the magnetic field vector according:

$$\vec{B} = \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} = \begin{pmatrix} -\|\vec{B_0}\| \cdot (\cos(\phi_B)\sin(\vartheta_B)) \\ -\|\vec{B_0}\| \cdot (\sin(\phi_B)\sin(\vartheta_B)) \\ -\|\vec{B_0}\| \cdot \cos(\vartheta_B) \end{pmatrix}$$

The magnetic field vector directs towards the origin (see fig. 3.5)!

- `lcfs_ontip`

Input Number	Action
0	LCFS is on the tip of the limiter
1	LCFS is NOT on the tip of the limiter (symmetry point) → `elec_of_lcfs` must be specified with an input number ≥ 0

- `elev_of_lcfs`

  If `lcfs_ontip` = 1, this is the elevation in [mm] of the LCFS over the tip of the limiter (see fig. 3.6).

- `cs_abf0`

  e-folding length for background plasma velocity decay in the "negative" "private" region[1] of the limiter according:

$$c_s(\vec{x}) = c_{s,0} \cdot \left(\frac{cs_{abf0}}{cs_{abf0} - x} - \sqrt{(\frac{cs_{abf0}}{cs_{abf0} - x})^2 - 1} \right)$$

[1]The "private" region is the part of the SOL for which the magnetic field lines cross the limiter surface, i.e. $\exists\ \lambda_S$ for which the equation $\vec{x}_{particle} + \lambda_S \cdot \frac{\vec{B}}{\|\vec{B}\|} = \vec{x}_{Limitersurface}$ holds. If the limiter is the LCFS defining element then the entire SOL (as modeled by ERO-TEXTOR) will be the private region.
"negative" "private" region is the part of the private region for which $\lambda_S < 0$ (see fig. 3.6).

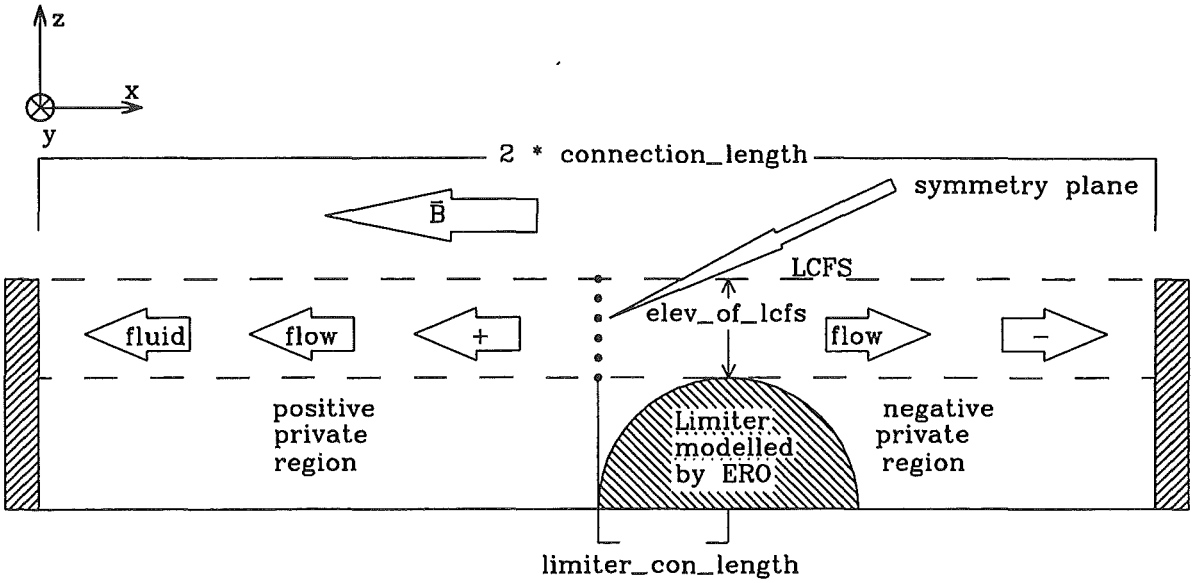


Figure 3.6: Visualization of SOL parameters for the situation when the modeled limiter is not the LCFS-defining element in the scrape-off-layer. Due to the fact that the limiter modeled by ERO-TEXTOR is on the right side of the symmetry (stagnation) plane the parameter "sign of con length" has to be specified as "-". dx is given by the distance of the particle from the limiter surface along $\vec{B}$, i.e $dx = (\vec{x}_{part} - \vec{x}_{surf}) \cdot \vec{B}/\|\vec{B}\|$.

- `cs_abf1`

  e-folding length for background plasma velocity decay in the "positive" "private" region[2] of the limiter according:

$$c_s(\vec{x}) = c_{s,0} \cdot \left(\frac{cs_{abf1}}{cs_{abf1} - x} - \sqrt{(\frac{cs_{abf1}}{cs_{abf1} - x})^2 - 1} \right)$$

- `connection_length`

  Distance of the LCFS-defining element from the symmetry point, i.e. the distance d of the LCFS-defining element from itself along the magnetic field lines is $d = 2 \cdot$`connection_length` (see fig. 3.6).

- `sign_of_con_length`

Input Number	Action
0	"+" fluid flow parallel to $\vec{B}$
1	"−" fluid flow parallel to $-\vec{B}$
2	"±" fluid flow ambivalent; limiter tip is the symmetry point

- `limiter_con_length`

  Distance of the limiter from the symmetry point of the LCFS (see fig. 3.6).

[2] "positive" "private" region is the part of the private region for which $\lambda_S > 0$ (see fig. 3.6).

- `sheat_on`

Input Number	Action
0	Calculation with sheath potential
1	No sheath potential is calculated

- `potential_drop`

Input Number	Action
0	total potential in the (magnetic) sheath: $\Phi = -\frac{3k_B T_e}{e}$
1	total potential in the (magnetic) sheath: $\Phi = -\frac{k_B T_e}{2e} \ln\left[2\pi(\frac{m_e}{m_i})(1+\frac{T_i}{T_e})\frac{1}{(1-\gamma)^2}\right]$ [56] γ is the secondary electron emission coefficient

- `sheat_factor`

  Enhancement factor for the sheath potential

- `presheath_factor`

  Enhancement factor for the pre-sheath radial electric field

- `diffusion_on`

Input Number	Action
0	With cross-field diffusion as specified in `perpendicular diffusion` $D_\perp = D(perp)$
1	No cross-field diffusion
2	With cross-field diffusion in an Alcator like scaling: $D_\perp = D(perp)\frac{n_{lcfs}}{n_e(z)}$
3	With cross-field diffusion in the Alcator scaling: $D_\perp = 5 \cdot 10^{12} \cdot \frac{1}{n_e}[\frac{1}{cm^2 s}]$

- `collision_for_diffusion`

Input Number	Action
0	Ion collision frequency is calculated according to plasma parameters; $\nu_i = 6.81 \cdot 10^{-8}\sqrt{\frac{Z_B}{m_I^2+m_I Z_B}}\frac{Z_B^2 \Lambda n_e}{T_e^{3/2}}$ $[s^{-1}]$
any other real number	user specified ion collision frequency $[s^{-1}]$

- `perpendicular_diffusion`

  (anomalous) cross-field diffusion coefficient $D(perp)$ in $[\frac{mm^2}{s}]$

3.5.3 Limiter/Divertor Geometry

- `# limiter geometry`
  `poloidal_radius`

  Accepts only integer values!

`poloidal_radius` R_p	`toroidal_radius` R_t	Limiter shape
$R_p = 0$	$R_t = 0$	plain surface
$R_p = 0$	$R_t \neq 0$	cylinder along y-axis
$R_p \neq 0$	$R_t = 0$	cylinder along x-axis
$R_p = R_t$	$R_p, R_t \neq 0$	sphere
$R_p > R_t$	$R_p, R_t \neq 0$	"poloidal" torus (y-axis = symmetry axis)
$R_p < R_t$	$R_p, R_t \neq 0$	"toroidal" torus (x-axis = symmetry axis)

- `poloidal_length`

  Width of the limiter (along y-axis). The limiter extends from $y = -\frac{1}{2}$`poloidal_length` to $y = +\frac{1}{2}$`poloidal_length`. Accepts only integer values.

- `toroidal_radius`

  ref. "`poloidal_radius`"

- `toroidal_length`

  Length of the limiter (along x-axis). The limiter extends from $x = -\frac{1}{2}$`toroidal_length` to $x = +\frac{1}{2}$`toroidal_length`. Accepts only integer values.

- `toroidal_angle`

  If a plane limiter is specified this angle α will be the angle of the surface normal $\vec{n}$ with the x-axis $\vec{e}_x$ according:

$$\alpha = \arccos\left(\left(\begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}\vec{n}\right)\cdot\vec{e}_x\right)$$

- `surface_temperatur`

  starting temperature of the limiter in degree Celsius (if necessary $\triangle T = 273.16K$ are added)

- `secondary_electron_emission`

  secondary electron emission coefficient of the material $\in [0, 1]$

- `heat_conductance`

  (temperature independent and isotropic) heat conductance of the limiter material in $[\frac{Kcm^2}{Ws^{1/2}}]$.

- `sheath_transmission`

  sheath energy transmission factor (typically $\approx 7 - 8$) [55]

- `interaction_depth`

  thickness of the surface layer in [nm], which is included in the calculation of the local concentration of different specimen (please mind that this is typically not related to the penetration depth of energetic ions into the bulk material, because of the fact that the product $d(\Lambda) = \Gamma \cdot \triangle t \rho^{-1}[nm]$ (ρ in $[\frac{\text{atoms}}{cm^3}]$) is much larger than the relevant penetration length).

3.5.4 External Source

- `# number of particles`
  `loops`

  number of particles for the external source per time step. If external source is a moving source (see below) then this will be the number of particles per source step and per time step.
  If instead of a real number a string with the suffix ".`dat`" (e.g. "`loop.dat`") is supplied, the code will expect a column of real numbers in a file with that name, each defining the number of particles for one subsequent time step. The code reads the first number and deletes it from the table.

- `# element choice for external source`
  `element_choice1`

  see tab. 3.1.

- `element_choice2`

  see tab. 3.1.

- `# external source data`
  `source_on`

Input Number	Action
0	External source is included in RUN
1	No external source

- `which_distribution`

  Energy distribution of the launched source particles. E_{mean} = `mean_energy` (ref. next item).

Input Number	Action
0	Particle of External Source are monoenergetic in energy according $E_{Particle} = E_{mean}$
1	Particle of External Source are maxwellian in energy according $P(E_{Particle}) \sim e^{-\frac{E_{Particle}}{E_{mean}}}$
2	Not Defined
3	Particle of External Source are thompsonian in energy according $P(E_{Particle}) \sim \frac{E_{mean}}{(E_{Particle}+E_{mean})^3}$

Element	ERO-TEXTOR-symbol	ERO-TEXTOR-Number	
		element_choice1	element_choice2
Hydrogen	_h	0	23
Deuterium	_d	1	23
Helium	he	2	23
Lithium	li	3	23
Beryllium	be	4	23
Boron	_b	5	23
Carbon	_c	6	23
Nitrogen	_n	7	23
Oxygen	_o	8	23
Fluor	_f	9	23
Neon	ne	10	23
Sodium	na	11	23
Magnesium	mg	12	23
Aluminum	al	13	23
Silicon	si	14	23
Phosphorus	_p	15	23
Sulfur	_s	16	23
Chlorine	cl	17	23
Argon	ar	18	23
Potassium	_k	19	23
Calcium	ca	20	23
Scandium	sc	21	23
Titanium	ti	22	23
Vanadium	_v	23	0
Manganese	mn	23	1
Iron	fe	23	2
Cobalt	co	23	3
Nickel	ni	23	4
Copper	cu	23	5
Zinc	zn	23	6
Gallium	ga	23	7
Germanium	ge	23	8
Bromine	br	23	9
Krypton	kr	23	10
Rubidium	rb	23	11
Strontium	sr	23	12
Zirconium	zr	23	13
Molybdenum	mo	23	14
Antimony	sb	23	15
Xenon	xe	23	16
Caesium	cs	23	17
Barium	ba	23	18
Tungsten	_w	23	19
Lead	pb	23	20
Bismuth	bi	23	21

Table 3.1: ERO-TEXTOR standard nomenclature for available Elements

- `mean_energy`

  Energy of monoenergetic particles (if 0 specified in `which_distribution`); mean energy of maxwellian particles (if 1 specified in `which_distribution`) or Surface Binding Energy of thompsonian particles (if 3 specified in `which_distribution`).

- `intervall_energy`

  This item belongs to Plasma-Surface-Interaction !!
  Number of particles launched from each surface cell when sputtering distribution is built, e.g. if there are $2 \cdot 10^3$ surface cells and `intervall_energy` is specified as 1000 then ERO-TEXTOR will have to trace 2 Million particles !! (Mind computational time!!)

- `polar_distribution`

  Polar angle distribution of launched particles, i.e. the angle θ between the z-axis and the unit-vector velocity- $\frac{\vec{v}_{Start}}{|\vec{v}_{Start}|}$ of the particles.

Input Number	Action
0	single angle; all particles start with the same angle θ
1	uniform distribution: θ uniform in $[0, \pi/2[$
2	cosine distribution: $P(\theta) \sim \cos(\theta)$
3	angle distribution according to the conditions of molecular flow through a cylindrical tube with ratio $\frac{L}{r}$, see below

- `polar_angle`

  If specified with 0 in `polar_distribution`, then all particles will start with $\theta =$ `polar_angle`. If uniform distribution is demanded, the `polar_angle` can be specified by any value. If cosine distribution is selected (`polar_distribution` = 2) then the probability of starting angle will be $P(\theta) \sim \cos($`polar_angle`$-\theta)$.

- `length_to_radius`

  length to radius ratio of cylindrical tube through which the gas is expanding under molecular flow conditions [17],[64].

- `azimut_distribution`

  Azimut angle distribution of launched particles, i.e. the angle ϕ between the x-axis and the unit-vector velocity- $\frac{\vec{v}_{Start}}{|\vec{v}_{Start}|}$ of the particles.

Input Number	Action
0	single angle; all particles start with the same angle *phi*
1	uniform distribution: ϕ uniform in $[0, 2\pi[$

- `azimut_angle`

  If specified with 0 in `azimut_distribution` then all particles will start with azimut angle $\phi =$`azimut_angle`, otherwise (`azimut_distribution` = 1) any angle between 0^o and 360^o will be satisfying.

- `source_rate`

  Source Rate in $\frac{particles}{s\, sourcepoint}$, i.e. if only one external source is specified then this will be the source rate, otherwise if, moving source (see below) is on and x source points are specified, then the total source rate S_{total} will be:

$$S_{total} = x \cdot \texttt{source rate}$$

If instead of a real number a string with the suffix ".dat" (e.g. "src.dat") is supplied, the code will expect a column of real numbers in a file with that name, each defining the source rate for one subsequent time step. The code reads the first number and deletes it from the table.

- source_location_y

  If just one external source point is specified (option: moving_source_on switched off), this will be the location of the source along the y-axis, otherwise it will be the center of the moving source along the y-axis.

- source_location_x

  If just one external source point is specified (option: moving_source_on switched off), this will be the location of the source along the x-axis, otherwise it will be the center of the moving source along the x-axis.

- moving_source_on

  Moving source means that several external source points separated by constant increments in x- and y-direction or in radial and angular direction centered around the point (source_location_x, source_location_y) can be specified. Each source point will be handled by ERO-TEXTOR as though only one external source is specified (and all source points are uniform in source rate, energy and angular distribution of particles and temporal behavior). All inputs for the spatial behavior of the source points are related to the x-y-plane and not to the surface of the limiter.

Input Number	Action
0	Moving source switched on
1	Moving source switched off

- moving_source_form

  There are two options for a moving source, either the source points are separated by constant increments in the x- and y-direction or they are separated by constant increments in radial and angular direction around the point source_location_x, source_location_y.

Input Number	Action
0	rectangular area of moving source (ERO-TEXTOR increments in x-/y-direction)
1	circular area of moving source (ERO-TEXTOR increments in radial and angular direction)

- moving_dx

  Defines the step width/increment which separates two source points according:

moving_source_form	Shape	Incrementing
0	rectangular	in x-direction (specify the dx wanted)
1	circular	in radial direction (specify the dr wanted)

- moving_dy

  Defines the step width/increment which separates two source points according:

moving_source_form	Shape	Incrementing
0	rectangular	in y-direction (specify the dy wanted)
1	circular	in angular direction (specify the $d\phi$ wanted)

- `intervall_dx`

  Defines the number of steps/source points in one direction according:

`moving_source_form`	Shape	Incrementing
0	rectangular	in x-direction
1	circular	in radial direction

- `intervall_dy`

  Defines the number of steps/source points in one direction according:

`moving_source_form`	Shape	Incrementing
0	rectangular	in y-direction
1	circular	in angular direction

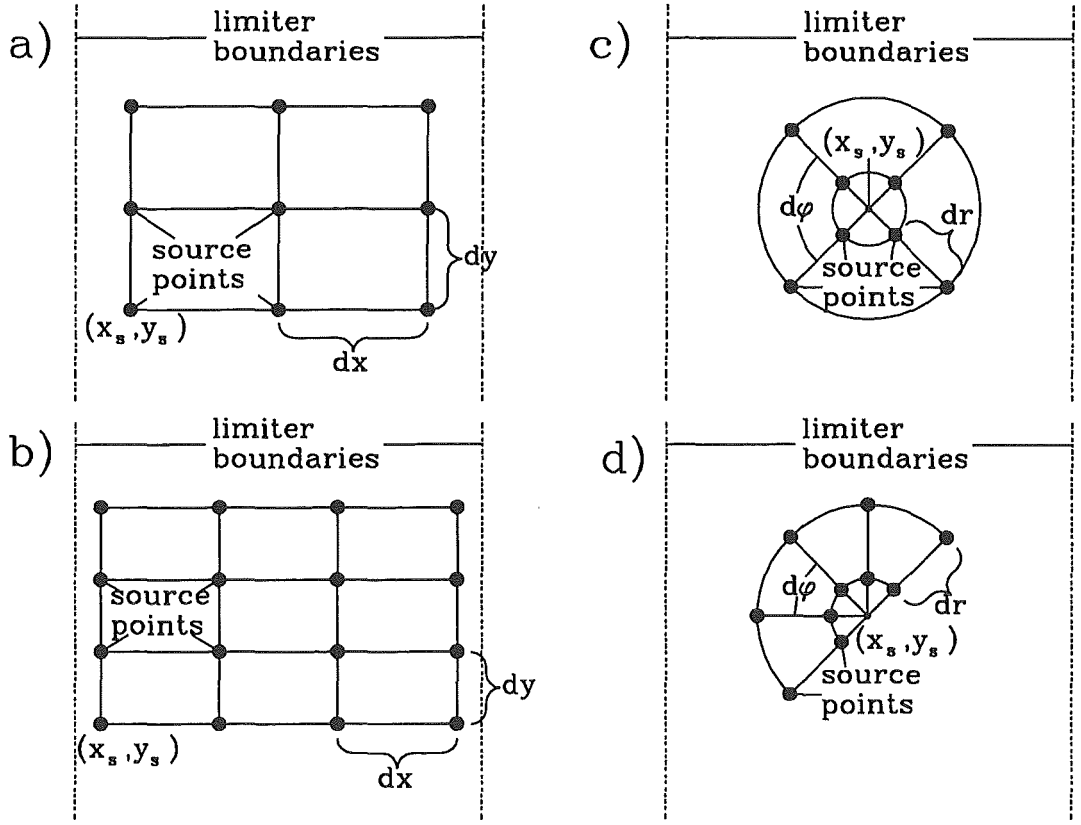


Figure 3.7: Different scenarios of "moving source": a) For rectangular moving source the central point is always the source point with lowest x- and y-values, all other source points obey the formula: $x_i = x_s + i * dx$; $y_j = y_s + j * dy$. In this case $N_x = N_y = 3$. b) $N_x = N_y = 4$. c) For circular moving source the central point is not a source point. All source points obey the formula $x_{ij} = (i+0.5)dr \cdot \cos((j+0.5)d\phi)$ $y_{ij} = (i+0.5)dr \cdot \sin((j+0.5)d\phi)$. In this case $N_r = 2$ $N_\phi = 4$, $d\phi = 90^o$. d)$N_r = 2$ $N_\phi = 4$, $d\phi = 45^o$.

Summarizing moving source:
Assuming that x_S, y_S is the central point of the moving source (as specified in `source_location_x`, `source_location_y`), dx, dy (respectively dr, $d\phi$) are the specified increments and N_x, N_y (respectively N_r, N_ϕ) are the defined number of steps, then the situation is as shown in fig. 3.7.

3.5.5 Chemical Data for External Source

- `# chemical data (1==off)`
  `chemie_on`

  ERO-TEXTOR provides the electron/proton impact ionization cross sections for molecules of the form XY_n ($X = C, Si$; $Y = H, D, T$; $n = 1, \ldots, 4$) according to Langer et al. [22] (methane) and appendix A (silane).

Input Number	Action
0	Source particles are molecules (chemistry on)
1	Source particles are atoms (chemistry off)

- `#parent_name == si oder _c  !! only methane & silane`
  `parent`

Input Symbol	Action
`_c`	"Parent"-atom in XY_n is Carbon (e.g. CH_4)
`si`	"Parent"-atom in XY_n is Silicon (e.g. SiH_4)

- `#child_name == h for hydrogen | d for deuterium | t for tritium`
  `child`

Input Symbol	Action
`h`	"Child"-atom in XY_n is Hydrogen (e.g. CH_4)
`d`	"Child"-atom in XY_n is Deuterium (e.g. CH_4)
`t`	"Child"-atom in XY_n is Tritium (e.g. CH_4)

  Attention: Not standard ERO-TEXTOR nomenclature (as defined in tab. 3.1) for hydrogen-like species (`h` $\neq$ `_h`).

- `child_number`

Input Number	Action
1	Number of "Child"-atoms in XY_n is $n = 1$ (e.g. CH_1)
2	Number of "Child"-atoms in XY_n is $n = 2$ (e.g. CH_2)
3	Number of "Child"-atoms in XY_n is $n = 3$ (e.g. CH_3)
4	Number of "Child"-atoms in XY_n is $n = 4$ (e.g. CH_4)

- `model_of_radical`

Input Symbol	Action
0	incoming radical reflected as XH_4 and a thompsonian energy distribution ($X = Si, C$ and $U_{\text{surface}} = U_{0,Si,C}$)
1	incoming radical reflected as it is and a thompsonian energy distribution ($X = Si, C$ and $U_{\text{surface}} = U_{0,Si,C}$)
2	incoming radical reflected as XH_4 and a maxwellian energy distribution ($X = Si, C$ and $E_{\text{mean}} = E_{\text{surface temperature}}$)
3	incoming radical reflected as it is and a maxwellian energy distribution ($X = Si, C$ and $E_{\text{mean}} = E_{\text{surface temperature}}$)
4	incoming radical reflected as XH_4 and the energy $E_{out} = E_{in}$ ($X = Si, C$)
5	incoming radical reflected as it is and the energy $E_{out} = E_{in}$ ($X = Si, C$)

- `reflection_of_radical`

  reflection coefficient ($R_M \in [0,1]$) of molecular radicals (like $CH_3, SiD_3, SiD_2, ...$).

- `model_of_ion`

Input Symbol	Action
0	incoming ion neutralized and reflected as XH_4 and a thompsonian energy distribution ($X = Si, C$ and $U_{\text{surface}} = U_{0,Si,C}$)
1	incoming ion neutralized and reflected as it is and a thompsonian energy distribution ($X = Si, C$ and $U_{\text{surface}} = U_{0,Si,C}$)
2	incoming ion neutralized and reflected as XH_4 and a maxwellian energy distribution ($X = Si, C$ and $E_{\text{mean}} = E_{\text{surface temperature}}$)
3	incoming ion neutralized and reflected as it is and a maxwellian energy distribution ($X = Si, C$ and $E_{\text{mean}} = E_{\text{surface temperature}}$)
4	incoming ion neutralized and reflected as XH_4 and the energy $E_{out} = E_{in}$ ($X = Si, C$)
5	incoming ion neutralized and reflected as it is and the energy $E_{out} = E_{in}$ ($X = Si, C$)

- `reflection_of_ion`

  reflection coefficient ($R_I \in [0,1]$) of molecular ions (like $CH_4^+, SiD_3^+, SiD_2^+, ...$).

- `ch4_model_of_radical`

  same as in `model_of_radical` but now for the particles chemically eroded by the background plasma.

- `ch4_reflection_of_radical`

  reflection coefficient ($R_M \in [0,1]$) of molecular radicals (like $CD_3, CD_2, ...$).

- `ch4_model_of_ion`

  same as in `model_of_ion` but now for the particles chemically eroded by the background plasma.

- `ch4_reflection_of_ion`

  reflection coefficient ($R_I \in [0,1]$) of molecular ions (like $CD_3^+, CD_2^+, ...$).

3.5.6 Marker Data

- `# markerdata 0=circular 1 = rectangular`
  `number_of_markers`

  With this option ERO-TEXTOR provides a possibility to have on a limiter of bulk material X "spots" of other materials (e.g. thin films evaporated onto the limiter). ERO-TEXTOR is able to handle two markers of different material (see table below)! In this section the markers have to be specified. If one marker is demanded the specification must be written after the keywords starting with `marker1`.
  Warning: This is only the specification of markers – elements to be traced by ERO-TEXTOR have to be specified in sec. 3.5.7.

Input Number	Action
0	No marker
1	One marker
2	Two markers

- `marker1_form`

  Shape of marker 1 in the projection to the x-y-plane and not on the limiter surface!

Input Number	Shape of Marker
0	circular in the plane projection to x-y-plane
1	rectangular in the plane projection to x-y-plane

  Warning: All surface densities (film thickness) of a material are bound to the *surface net cells* which have to be created by ERO-TEXTOR (and defined by the user as described in sec. 3.5.7). Therefore the actual form of a marker is also bound to the cells (see fig. 3.8).

- `marker1_dx`

  length of the marker along the x-axis, if rectangular marker is specified. If circular marker is specified this is the radius of the marker.

- `marker1_dy`

  length of the marker along the y-axis, if rectangular marker is specified. If circular marker is specified this is without use.

- `marker1_dz`

  Thickness of the marker (e.g. film thickness) [nm]

- `marker1_location_x`

  location of marker center along the x-axis.

- `marker1_location_y`

  location of marker center along the y-axis.

- `marker1_material`

  Marker material (ref. to the standard ERO-TEXTOR nomenclature of elements, tab. 3.1).

- `marker2_form`

  The following items refer to the second marker. Input is of the same form as marker 1.

- `marker2_dx`
- `marker2_dy`
- `marker2_dz`
- `marker2_location_x`
- `marker2_location_y`
- `marker2_material`

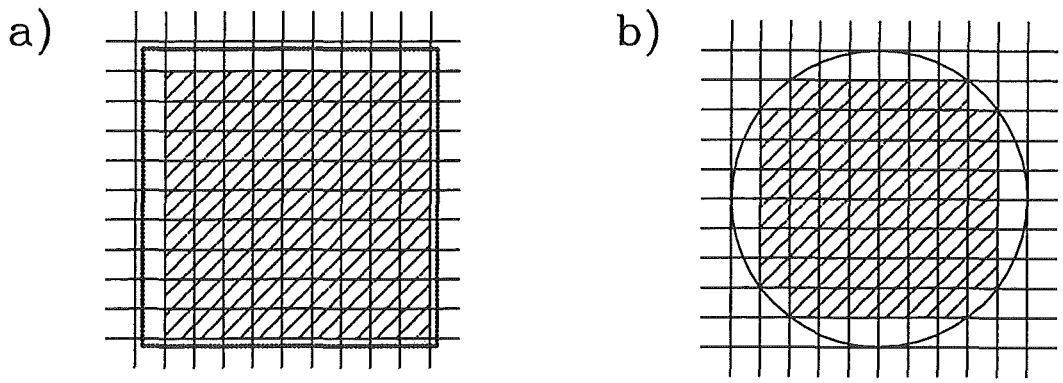


Figure 3.8: If the boundaries of a marker and the surface net cells (ref. sec. 3.5.7) do not coincide then ERO-TEXTOR will fill those surface net cells only with marker material, which are entirely covered by the marker area. Hatched cells are filled with marker material; blank cells do not belong to the marker. a) Situation if the marker boundaries and net cells are shifted. b) Situation if the marker is defined as circular.

3.5.7 Surface Net on Limiter/Divertor

- `# surface net`
  `s_poloidal_boundary_max`

  This section generates a surface net on the limiter, which is restricted to the extends of the limiter (the user may not specify extends larger than the limiter which will definitively lead to a program-crash)! All input data are related to the x-y-plane and not to the limiter surface! Basic idea of a user defined surface net on the limiter is the possibility of adapting cell size of the net to the specific run (species and therefore transport lengths) and of suppressing limiter edge effects.
  The following table gives the maximum/minimum input data:

	Item	Maximum/Minimum Input Value
y-coordinate:	`s_poloidal_boundary_max`	$+\frac{1}{2}$`poloidal_length`
	`s_poloidal_boundary_min`	$-\frac{1}{2}$`poloidal_length`
x-coordinate:	`s_toroidal_boundary_max`	$+\frac{1}{2}$`toroidal_length`
	`s_toroidal_boundary_min`	$-\frac{1}{2}$`toroidal_length`

  For these items ERO-TEXTOR expects real numbers!

- `s_poloidal_boundary_min`

- `s_toroidal_boundary_max`

- `s_toroidal_boundary_min`

- `s_cell_dpoloidal`

  Cell size of surface net along the y-axis in the projection to the x-y-plane (see fig. 3.9).

- `s_cell_dtoroidal`

  Cell size of surface net along the x-axis in the projection to the x-y-plane (see fig. 3.9) .

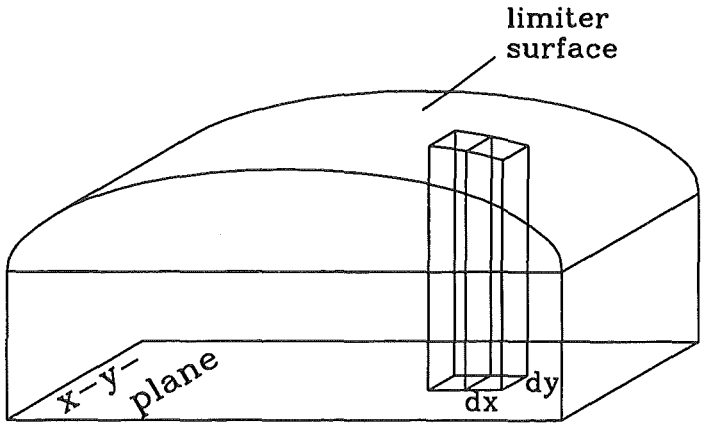


Figure 3.9: Definitions of cell size are made in the x-y-plane. ERO-TEXTOR then projects the equidistant plane cells onto the limiter surface (i.e. for curved limiters the cells on the limiter surface are not equal in area etc.). In this figure: s-cell-dtoroidal = dx and s-cell-dpoloidal = dy.

- `substrat_tracing_on`

Input Number	Action
0	the limiter bulk material will be traced, i.e. sputtering of this material will be considered in calculation
1	the limiter bulk material will not be traced

- `substrat`

  ERO-TEXTOR nomenclature for the bulk material of the limiter (ref. tab. 3.1).

- `element1_tracing`

  If no tracing in addition to bulk material is desired `element1_tracing` must be specified as 23. Otherwise (tracing of source particles or marker species), according ERO-TEXTOR nomenclature (ref. tab. 3.1), an element must be specified.

- `element2_tracing`

  If no tracing in addition to bulk material and `element1_tracing` is desired `element2_tracing` must be specified as 23. Otherwise (tracing of source particles or marker species), according ERO-TEXTOR nomenclature (ref. tab. 3.1), an element must be specified.

3.5.8 Volume Net for Spectroscopy

- `# net for spectroscopy`
  `spectroscopy_on`

  This section generates the volume grid for spectroscopic output. The base of the volume grid is restricted to the extends of the limiter surface (the user may not specify extends larger than the limiter which will definitively lead to a program-crash)! Basic idea of a user defined volume grid on the limiter is the possibility of adapting cell size of the net to the specific run (species and therefore transport lengths) and of suppressing limiter edge effects. The volume grid does not perfectly fit to curved limiter surfaces (see fig. 3.10) and

therefore always leaves a small space directly above the limiter surface unobserved.

Input Number	Action
0	Spectroscopy on
1	Spectroscopy off

- `emission_on`

Input Number	Action
0	Calculate emitted intensity on
1	Calculate emitted intensity off
2	Calculate emitted intensity on, but no ouput of calculated densities

- `v_poloidal_boundary_max`

The following table gives the maximum/minimum input data:

Item	Maximum/Minimum Input Value
length along the y-axis:	
`v_poloidal_boundary_max`	$+\frac{1}{2}$`poloidal_length`
`v_poloidal_boundary_min`	$-\frac{1}{2}$`poloidal_length`
length along the x-axis:	
`v_toroidal_boundary_max`	$+\frac{1}{2}$`toroidal_length`
`v_toroidal_boundary_min`	$-\frac{1}{2}$`toroidal_length`
length along the z-axis:	
`v_radial_boundary_max`	no limit
`v_radial_boundary_min`	must be ≥ 0

For these items ERO-TEXTOR expects real numbers!

- `v_poloidal_boundary_min`
- `v_toroidal_boundary_max`
- `v_toroidal_boundary_min`
- `v_radial_boundary_max`
- `v_radial_boundary_min`
- `v_cell_dpoloidal`

  Cell size of volume grid along the y-axis in the projection to the x-y-plane.

- `v_cell_dtoroidal`

  Cell size of volume grid along the x-axis in the projection to the x-y-plane.

- `v_cell_dradial`

  Cell size of volume grid along the z-axis in the projection to the x-y-plane.

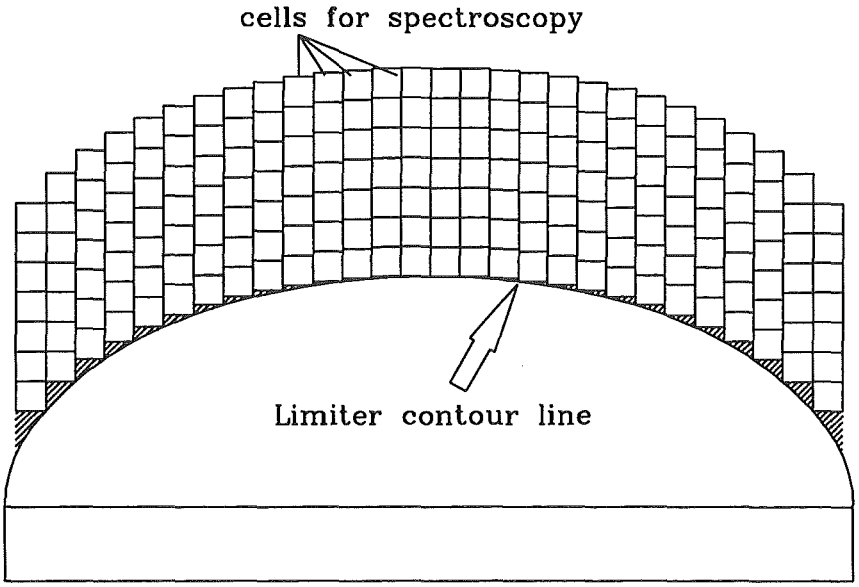


Figure 3.10: The spectroscopic net is made of cubes and therefore does not perfectly fit to curved limiter surfaces. ERO-TEXTOR calculates the net in a way that as few as possible space above the limiter is not contained in a cell. In this figure (a 2-dimensional plot of one row of cells) the hatched areas do not belong to a cell.

3.5.9 Species to be traced by Spectroscopy

ERO-TEXTOR provides the possibility of obtaining spectroscopic (i.e. density) information of up to five different species. Each specimen is defined by the element name and charge state. Item `element_anzahl` defines the number of specimen of which spectroscopic information has to be calculated. The following ten items (`elementX_name`, `elementX_charge`) define the name and charge state for each specimen.

- `# species (ion / neutrals) to be traced spectroscopically`
  `element_anzahl`

  Integer value $no = 0, 1, \ldots, 5$.

- `element1_name`

  Standard ERO-TEXTOR nomenclature for the element name (ref. tab. 3.1).

- `element1_charge`

  Integer value $q = 0, 1, 2, \ldots$ for the positive charge state.

- `element2_name`

- `element2_charge`

- `element3_name`

- `element3_charge`

- `element4_name`
- `element4_charge`
- `element5_name`
- `element5_charge`

3.5.10 Number of Time steps & Time per Step

- `# time grid`
  `total_time`

  Expects a real number for the total simulation time in [s].

- `time_per_step`

  Input is the time per step in [s]. All transport phenomena usually (i.e. the limiter size is of the order of 100 mm) have a time scale of μs. While a normal shot in TEXTOR is of the order of some seconds. Taking into account that multiple sputtering, transport and redeposition occur, a simulation can be divided into several time steps. If instead of a real number a string with the suffix "`.dat`" (e.g. "`time.dat`") is supplied, the code will expect a column of real numbers in a file with that name, each defining the time per step for one subsequent time step. The code reads the first number and deletes it from the table.

Chapter 4

Formatted Output Files

At the end of an individual run ERO-TEXTOR writes all parameters of the input file as the code "understood" them into the file `OUTERO.PRM`, where `OUT` is the string provided to ERO-TEXTOR in the input file, item `pathtoresult`. So it is easy to check whether the code received all parameters as the user wanted. In addition the calculation of the surface net and the spectroscopy grid may force ERO-TEXTOR to change the boundaries slightly. Then these changes can also be found in the `OUTERO.PRM` file.
After each time step ERO-TEXTOR provides a variety of Output-Files. All files obey the standard output nomenclature, which is described in this section and most of them are delivered as binary-files (binary files have the suffix `.BIN`, whereas ASCII files have the suffix `.T`) . All files start with the user defined output name, specified in the input file, item `pathtoresult` (in this section this will be called "`OUT`").
In principle there are two different types of output files:

- Surface Related Output Files
- Spectroscopy Related Output Files

For both types of files the ending (that what follows after the ".") obey the "time standard": If only one time step (see section 3.5.10 for input information) has to be calculated, all files will end with:

`.BIN`, `.T` respectively.

If more than one time step has to be calculated, the ending of all files will be in the nn^{th} time step:

`.BIN`nn, `.T`nn respectively.

(e.g. in the 7^{th} time step all files will end with `.BIN07`)

4.1 Surface Related Output Files

These files contain the information on (gross/net) erosion and deposition on the limiter surface in units of $[\frac{\text{atoms}}{\text{mm}^2}]$. As well as information on medium values like medium impact energy of particles hitting the surface etc. All surface related files start with `0` (abbreviation for german "Oberfläche" i.e. "surface") after the user defined output name ("`OUT`"). All files (except the file containing information on how many particles were lost from the limiter surface) are one dimensional tables following the structure:

"header"
value(x,y) of surface cell
$z(0,0)$
$z(0,1)$
...
$z(0,n)$
$z(1,0)$
$z(1,1)$
...
$z(n,n)$

Where each file starts with a header string of 120 byte containing some basic information on the specific run and the information on the grid used for the file:

```
All values in this file are in [atoms/mm^2].Element : "si"; time info:
step-No: "3"   step-time: "0.5"  total-time:  "60.5"000000...
```

(To fill the 120byte long string, it is filled with "000...". All values in quotation marks above are examples for illustration. The string begins with the ERO-TEXTOR nomenclature of the relevant specimen (here "si") and the charge of it (here "+2"). Next comes the time step (here "3") and the time per step (here "0.5" [s]). Finally the string ends with the information how much time in total will be calculated (here "60.5" [s]).)
The information on the grid used is organized in the following way:

value	size(bytes)
minimum x-value of the grid	8
minimum y-value of the grid	8
maximum x-value of the grid	8
maximum y-value of the grid	8
cell size dx	8
cell size dy	8
amount of cells along x	2
amount of cells along y	2
"BEGIN"-comment	5

The following table describes these files and gives for each an example, using tungsten (`_w`) as sample specimen and single time step mode (ending `.BIN`).

File Name	Description
OUTO_wNG.BIN	"NG" = NewGo; number of atoms of this specimen (_w) eroded from this surface cell in this time step
OUTO_wNC.BIN	"NC" = NewCome; number of atoms deposited on this surface cell in this time step
OUTO_w_R.BIN	"_R" = Resource; number of atoms remaining in this surface cell after this time step
OUTO_w_T.BIN	"_T" = Transport; number of atoms effectively transported away from/ to this surface cell in this time step
OUTO_wME.BIN	"ME" = Mean Energy; mean impact energy onto this surface cell of atoms of this specimen (_w)
OUTO_wMQ.BIN	"MQ" = Mean Charge state q; mean charge state of deposited atoms
OUTO_wML.T	"ML" = Lost particles; number of particles lost from the limiter surface net, where: lost in x_min $\in]-\infty, x_{min}]$, $\forall y$ lost in x_max $\in [x_{max}, \infty[$, $\forall y$ lost in y_min $\in]-\infty, y_{min}]$, $x_{min} < x < x_{max}$ lost in y_max $\in [y_{max}, \infty[$, $x_{min} < x < x_{max}$ x_{max}, y_{max}, x_{min}, y_{min} are the net boundaries lost in z_min $\in]-\infty, z_{min}]$ lost in z_max $\in [z_{max}, \infty[$ lost time_out particles lost after time_out seconds lost in errors particles lost due to calculation instabilities

4.1.1 Temporary Files

On most computer systems the batch facilities only allow for jobs with typically less than 10h cpu-time, which is of the order of the computational time for one time step. Therefore ERO-TEXTOR code saves after each time step the necessary surface information in a set of temporary files. These are like the surface related data files one-dimensional table, but without a header. The information to be stored in such files are the resources of each element (xxx_name_R.tmp), the depth profile of each element (xxx_film.tmp), the particles eroded by the simulation particles itself, which will be transported away in the next time step (xxx__newenergetics.tmp), the surface temperature reached in the time step before (xxx_surface_t.tmp) and the number of the time step before (time.tmp).

4.2 Spectroscopy Related Output Files

Files containing spectroscopic information (i.e. density or the emitted intensity of line radiation of a certain specimen, defined by element name and charge state) are line integrated along one coordinate axis or surface integrated along the distance from limiter surface. Up to now line integration is done as if the limiter would be transparent. These files are also one dimensional tables like the surface related output files. The standard notation is filename$\alpha\beta$q.ending, where α designates the integration axis, β is the standard ERO-TEXTOR nomenclature for atomic species (if molecular species are involved β will contain the full name of the molecule, e.g. SiD_4 then yields $\beta = sid4$) and q is the charge state.

File Name	Description
OUTX_w1.BIN	Density of this specimen (W^{1+}) line integrated along the x-axis in $[\frac{atoms}{cm^2}]$
OUTY_w1.BIN	Density of this specimen (W^{1+}) line integrated along the y-axis in $[\frac{atoms}{cm^2}]$
OUTZ_w1.BIN	Density of this specimen (W^{1+}) line integrated along the z-axis in $[\frac{atoms}{cm^2}]$
OUTD_w1.BIN	Density of this specimen (W^{1+}) as a function of distance from the limiter surface (surface integrated) $[\frac{atoms}{cm}]$

File Name	Description
OUTX_w1.BINSPEC	Emitted intensity of this specimen line integrated along the x-axis in $[\frac{photons}{cm^2 ssr}]$
OUTY_w1.BINSPEC	Emitted intensity of this specimen line integrated along the y-axis in $[\frac{photons}{cm^2 ssr}]$
OUTZ_w1.BINSPEC	Emitted intensity of this specimen line integrated along the z-axis in $[\frac{phtons}{cm^2 ssr}]$

Like the surface output files each spectroscopy-file starts with a header of 120 byte containing basic information on the specific run, the spectroscopy task (in german) and the information on the grid used for the file: :

```
Element : "si"  der Ladung "+2" Zeitinfo: step-No: "3"   step-time: "0.5"
total-time:  "60.5"000000...
```

(again the string is filled with "000...". All values in quotation marks above are examples for illustration. The string begins with the ERO-TEXTOR nomenclature of the relevant specimen (here "si") and the charge of it (here "+2"). Next comes the time step (here "3") and the time per step (here "0.5" [s]). Finally the string ends with the information how much time in total will be calculated (here "60.5" [s]).)
The information on the grid used is organized as described above.

Chapter 5

Data-Structures and ERO-TEXTOR-Routines

5.1 Building In New Elements

For elements not delivered with the ERO-TEXTOR-code the relevant data have to be supplied for the databases described in chapter 3 (i.e. the ionization rates, the sputtering data for the relevant projectile-target combination, the background plasma sputtering yield according eqn.(2.114) and the reflection database). In addition the two intrinsic ERO-TEXTOR-routines `void NoElem()` and `void ElemNo()` (see also below) responsible for element recognition must be changed.

Therefore in `void NoElem()` the following block must be inserted:

```
case number:{
              spezies->mass =  mass of the element
              spezies->z    =  atomic number of this element
              strcpy(spezies->name,"xx" )
              }
```

where `number` has to be an unused case number and `xx` must be a *two* letter abbreviation for the name of the element.
In the `void ElemNo()`-routine also the following block has to be inserted

```
if(strcmp(spezies->name,"xx")==0)
              {
              spezies->elem = number
              spezies->mass = mass of the element
              spezies->z    = atomic number of this element
              }
```

with the same data as in the `void NoElem()`-routine.

If some background plasma impurities are wanted be replaced by other ones, in the program kernel routine `move_ctrl()` the following blocks have to be changed (replace names, effective charge state and mass):

```
strcpy(plasma_ptr->impurities[1],"he");
strcpy(plasma_ptr->impurities[2],"_c");
strcpy(plasma_ptr->impurities[3],"_o");
plasma_ptr->imp_mass[1]=4.0026;
plasma_ptr->imp_mass[2]=12.011;
plasma_ptr->imp_mass[3]=15.994;
```

If additional background plasma impurities are required, the following scenario will hold:

- the variable `impur_max` (in `move_ctrl`) has to be incremented by 1.

- the variable `Material_max` (in `move_ctrl`) has to be incremented by 4.
- the variable `kombination[30][10]` (in `move_ctrl`) has to be incremented when `Material_max` > 30 according to the increment of this.
- in the block above additional names, effective charge states and masses have to be defined.
- in the following block an additional if-construct has to be inserted:

```
if(strcmp(read,plasma_ptr->impurities[0])==0)/* H,D,T*/
{
  strcat(kombination[j],"1");
  kombination[j][5]='\0';
}
if(strcmp(read,plasma_ptr->impurities[1])==0)/* He*/
{
  sprintf(qstr,"%d",plasma_ptr->imp_charge[1]);
  strcat(kombination[j],qstr);
  kombination[j][5]='\0';
}
if(strcmp(read,plasma_ptr->impurities[2])==0)/* C*/
{
  sprintf(qstr,"%d",plasma_ptr->imp_charge[2]);
  strcat(kombination[j],qstr);
  kombination[j][5]='\0';
}
if(strcmp(read,plasma_ptr->impurities[3])==0)/* O*/
{
  sprintf(qstr,"%d",plasma_ptr->imp_charge[3]);
  strcat(kombination[j],qstr);
  kombination[j][5]='\0';
}
```

5.2 Data-Structures

As far as possible in ERO-TEXTOR all important data are part of a data-structure and the data of one structure are grouped as comprehensive as possible. There are seven groups of data structures, which are presented in this section. In the first group, containing the data of the particle the important structure "ParticleStruct" is located, which contains all information on a specific (quasi-)particle available. The second group of structures are those, containing information on the background plasma and the relevant fields. The third group holds the material data, such as source information, sputter yields and ionization rates. The fourth group is dedicated to the limiter/divertor geometry and in the fifth group the properties of their surfaces are located. Group number six contains the spectroscopic information, whereas the seventh group is dedicated to the numeric treatment.

5.2.1 Particle-Structures

```
struct ParticleStruct
{
      double location[5];     space-coordinates of particle x,y,z and
                              two direction angles
                               location[0] = x-coordinate in space
                               location[1] = y-coordinate in space
                               location[2] = z-coordinate in space
                               location[3] = azimuth angle relative to x-axis
                               location[4] = polar angle relative to z-axis
      double dz;              distance of the particle from the limiter surface
      double time;            time passed since particle started
      double v[6];            velocity array
                               v[0] = scalar velocity
                               v[1],v[2],v[3]=x-,y-,z-component of
                               velocity vector
                               v[4] = azimuth angle relative to z-axis
                               v[5] = polar angle relative to z-axis
      double energy;          energy of particle
      double Ti;              temperature of the particle
      int    charge;          charge of particle
      float  mass;            mass of particle
      int    z;               atomic number
      char   elem_name[5];    name of the elements (SI,C,HE,H,D,T,...)
      double atoms;           # of atoms represented by this (quasi-) particle

      struct ParticleStruct *ptr_for; pointing to next ParticleStruct
      struct ParticleStruct *ptr_bak; pointing to last ParticleStruct
};
```

Solving the equations of motion using the Boris-integrator[7], i.e.:

$$\vec{v}_- = \vec{v}_{old} - \frac{q\vec{E}}{m}\frac{\triangle t}{2} \tag{5.1}$$

$$\vec{t} = \frac{q\vec{B}}{m}\frac{\triangle t}{2} \tag{5.2}$$

$$\vec{t}_k = \vec{v}_- \times \vec{t} \tag{5.3}$$

$$\vec{v}' = \vec{v}_- + \vec{t}_k \tag{5.4}$$

$$\vec{s} = \frac{2}{1+t^2}\vec{t} \tag{5.5}$$

$$\vec{s}' = \vec{v}' \times \vec{s} \tag{5.6}$$

$$\vec{v}_+ = \vec{v}_- + \vec{s}' \tag{5.7}$$

$$\vec{v}_{new} = \vec{v}_+ + \frac{q\vec{E}}{m}\frac{\triangle t}{2} \tag{5.8}$$

the following two structures are needed, when identifying $\vec{v}^-$ =vmin[4], $\vec{t}$ = T[4], $\vec{t}_k$ =TK[4], $\vec{v}'$ =VS[4], $\vec{s}$ =SS[4],$\vec{s}'$ =SK[4] and $\vec{v}^+$ =vplu[4].

```
struct BRotator
{
      double T[4];
      double TK[4];
      double VS[4];
      double SS[4];
      double SK[4];
};

struct V_PM
{
      double vmin[4];
      double vplu[4];
};
```

5.2.2 Plasma-Structures

The following four structures (Plasma, Magnetic, Electric and Diffusion) mainly contain the input data for the specific run.

```
struct Plasma
{
      int elem ;                  fuel ions (H = 0, D = 1, T = 2)
      char impurities[4][6];      chemical names (ref. ERO-TEXTOR nomenclature)
                                  of the plasma ions (fuel and impurities)
      double imp_percent[4];      percentage of plasma ions
                                  [0]= H,D,T; [1]=He; [2]=C; [3]=O;
      double sum_fiqi;
```

$$s_f = \sum_{all\ species} f_i \cdot q_i \tag{5.9}$$

s_f – sum-fiqi
f_i – imp-percent
q_i – imp-charge

```
      int  imp_charge[4];     charge of plasma ions (mean)
      int  imp_mass[4];       plasma ion mass

      float z_eff;            Z effective of the plasma
      float q_plasma;         mean charge of the plasma
      int impur_max;          amount of maximum impurities which are trace
                              (used in another routine than the
                              move_ctrl(..)-routine
};

struct nTdata
{
      float Te;               electron temperature at the lcfs
      float Te_abf;           decay length of Te from the lcfs
      float grad_Te[3];       local gradient of Te
                              (gradients are calculated in separate function)
      float Ti;               ion temperature at the lcfs
      float Ti_abf;           decay length of Ti from the lcfs
      float grad_Ti[3];       local gradient of Ti
      float ne;               electron density at the lcfs
      float ne_abf;           decay length of ne from the lcfs
                              (can also be negative!)
      float ne_fac;           factor for s.c.\ decay length of ne from lcfs
      float grad_ne[3];       local gradient of ne
      float z_lcfs;           non-local z-value of the LCFS
      int   diff_on;          for selfconsistent decay length of density
      float c_perp;           user defined perpendicular diffusion
};


struct Magnetic
{
      int on;                   magnetic field on/off
      float streng;             magnetic induction strength in Tesla
      float p_ang;              azimuth angle of B-field vector
      float t_ang;              polar angle of B-field vector
      int   lcfs;               0 = limiter itself is LCFS defining
                                1 = limiter is outside the LCFS
      float z_lcfs;             elevation of the LCFS over the limiter tip
} ;

struct Electric
{
      int on;                   sheath potential on/off
```

```
      float fac;                enhancement factor of the sheath potential
      float Rfac;               enhancement factor of the pre-sheath
                                electric field
};

struct Diffusion
{
      int on;                   diffusion on/off
      float c_frq;              user defined collision frequency
      float c_perp;             user defined diffusion coefficient
};

struct BField
{
      double B[3];              magnetic field vector
      double Br[3];             unit vector parallel B
      double dl;                local distance of the particle from the
                                surface along the magnetic field line
      double dz_B;              local distance of the particle from the
                                surface along the magnetic field line
                                (but used in a different routine)
      double d;
```

Defining the Hesse normal form of a plane through the limiter tip perpendicular to $\vec{B}$:

$$\vec{x} \cdot \frac{\vec{B}}{B} - d = 0 \tag{5.10}$$

Distance of the particle with space vector $\vec{x}_0$ from this plane:

$$dz_B = \vec{x}_0 \cdot \frac{\vec{B}}{B} - d \tag{5.11}$$

```
};

struct EField
{
      int    fehler;        if an error in the routine Efield() occurs,
                            this will be set equal TRUE
      double dz;            normal distance of the particle from
                            the limiter surface
      double E0;            field strength
      double E[3];          electric field vector
      double Er[3];         unit vector parallel E
      double Er_1[3];       not in use
      double Er_2[3];       not in use
      double nB_angle;      angle between surface normal and B-field vector
                            to calculate potential drop in the magnetic
                            sheath
};
```

```
struct Force
{
      double alpha;          coefficients for the thermal forces by
                             the gradient of Te
      double beta_i;         coefficients for the thermal forces by
                             the gradient of Ti
      double dv_thermal[3];velocity change due to the thermal forces
}
```

where the change in velocity is given by:

$$\frac{\triangle \vec{v}_{\|,thermal}}{\triangle t} = \frac{\alpha}{m}\vec{\nabla}_{\|}T_e + \frac{\beta_i}{m}\vec{\nabla}_{\|}T_i$$

```
struct PConst
{
      float fd;              Brooks-parameter
      float in_angle;        angle between surface normal and magnetic
                             field vector
      double Dl;             local Debye length
      double Et;
      double pot;            sheath factor
      double cs0;            local ion acoustic speed
      double cs_abf;
      double cs_abf0;        e-folding length of background plasma flow
      double cs_abf1;        e-folding length of background plasma flow
      double cs_abf2;        half of the connection length of the LCFS
                             defining element
      double cs;             local plasma flow velocity
      int    sign_Lc;        direction of plasma flow above limiter tip (if
                             limiter itself is not LCFS defining)
      double Lim_Lc;         distance from the limiter to the LCFS defining
                             element (along the magnetic field lines)
      double lane;           decay length of the electric field in the
                             magnetic pre-sheath
      double Lambda;         Coulomb-logarithm
      double spitzer_c;
```

local constant used for the calculation of the Spitzer times:

$$c_{Spitzer} = \frac{m \cdot \sqrt{\frac{T_e}{Z_{eff}}}}{6.8 \cdot 10^4 \cdot n_e q^2 \Lambda} \tag{5.12}$$

```
      double tstop;          Spitzer's relaxation time for dynamic friction
      double Dper;           cross-field diffusion coefficient (user defined)
      double v_diff;         difference velocity between background and
                             traced particles
      double w_stop;         = w/tau_stop stopping time
      double w_para;         = w^2/tau_E  energy exchange time
      double w_perp;         = w^2/tau_D  deflection time
};
```

5.2.3 Material-Structures

```
struct Spezies
{
      int    elem;              number of the element according
                                ERO-TEXTOR nomenclature
      char   name[6];           chemical sign
      float  mass;              mass of particles of this species
      int    z;                 atomic number
      double source_rate;       source rate of particles of this species
      int    dist;              energy distribution of particles of
                                external source
                                0 = monoenergetic
                                1 = maxwellian
                                2 = not in use
                                3 = thompsonian
      float  m_energ;           mean energy of particles
      int    intervall_e;
      int    intervall_p;
      int    intervall_a;
```

The product of intervall-e, intervall-p and intervall-a is the number of particles eroded from one surface cell.

```
      int    pol_distr;         polar angle distribution (in the reference
                                coordinate system) of particles
                                0 = one angle (ref. p_angle)
                                1 = uniform distributed
                                2 = cosine distributed
      float  p_angle;           polar angle
      int    azi_distr;         polar angle distribution (in the reference
                                coordinate system) of particles
                                0 = one angle (ref. a_angle)
                                1 = uniform distributed
      float  a_angle;           azimuthal angle
};

struct ChemConst
{
      int     chemie_on;        external source is a chemical source
                                0 = yes  ||  1 = no
      char    parent_name[5];   _C or Si
      char    child_name[5];    H,D or T
      double  child_factor;     factor for reduced silane reaction rates if
                                child == D,T, then child_factor=0.15 and in
                                the calculation the value:
                                rate = (1. - child_factor)*ne_rate is taken
      double  parent_mass;      mass of _C or Si
      double  child_mass;       mass of  H, D or  T
```

```
      int    radical_model;
      double radical_reflection;
      int    ion_model;
      double ion_reflection;
};
```

The next structure contains information on the actual status of the molecule, while dissociating, being ionized and so on.

```
struct Chemie
{
      int    number;          process number
      int    mass;            mass of molecule in number of hydrogenic
                              ligands
      int    charge;          charge
      double amu;             mass of the molecule (in units of amu)
      char   name[10];        name of the molecule
};
```

The following structure provides the database for sputtering data according *Bodhansky*- and *Yamamura*-formula (see eqns. (2.112), (2.113), see also sec. 3.3.1).

```
struct MaterialData
{
      char   kombination[6];     the projectile-traget combination
                                 (according ERO-TEXTOR-nomeclature)
      double data[10];           array of data, e.g.
                                 data[0] = surface binding energy
                                 data[1] = mass ratio
                                 ...
      struct MaterialData *next; pointing to the next data
};
```

The data calculated according to eqn.(2.114) are contained in an array of fifteen value pairs (T_e and the Yield $[\frac{atoms}{ion}]$) explained in sec. 3.3.2.

```
struct SputterData
{
      char   kombination[6];    background plasma specimen-target combination
      double yield[15][2];      data
      struct SputterData *next; pointing to the next data
};

struct ContentsOfReflection
{
      double yield;              the reflection yield for this energy
                                 and angle
      double energy[5];          cumulative probability distribution
                                 matrix for the energy
      double polar[5][5];        same for the polar angle (for each
```

```
                                energy level )
      double azimut[5][5][5];   same for the azimut angle (for each
                                energy and each polar angle level)
};

struct ReflectionData
{
      double yield;              if the particle is reflected, this will be
      double energy;             the yield and this will be the energy
      double polar;              and this will be the polar angle
      double azimut;             and this will be the azimut angle of the
                                 reflected particle
      char   kombination[10];    name of the projectile-target combination
      struct ContentsOfReflection  data[84];   the 84 x 125 matrices calculated
                                               by the TRIM code
      struct ReflectionData *next;             pointing to the next combination
};
```

The three preceding structures are combined to a ring and in searching for a specific projectile-target combination the code runs through this ring structure.

The following structure for the calculation of the ionization rates of atomic species is self explaining, when compared with eqns.(3.1) and (3.2).

```
struct Ion
{
      int charge;             charge of "actual" ionization state
      float I;                ionization potential
      float A;                data for I/10 <= kT <= 10I
      float B[6];                    -"-
      float a[6];                    -"-
      float alpha;            data for kT > 10I
      float beta[3];                 -"-
      struct Ion *forw;       pointing to the following structure
      struct Ion *backw;      pointing to the preceding structure
};
```

5.2.4 Limiter-Structures

This structure describes the geometry of the test limiter and where on this limiter the external sources are located.

```
struct Ort
{
      int    p_rad;             radius along the y-axis
      int    p_len;             length  along the y-axis
      int    t_rad;             radius along the x-axis
      int    t_len;             length  along the x-axis
      float  angle;             angle of plane limiter relative to x-axis
      float  s_pol;             y-coordinate of source (center)
      float  s_tor;             x-coordinate of source (center)
```

```
      int    move_source;          is source a moving source?
      int    move_coord;           rectangular / circular moving source
      float  dx_move;              increment for moving steps along the x-axis
      float  dy_move;              increment for moving steps along the y-axis
      int    interv_dx;            number of steps along the x-axis
      int    interv_dy;            number of steps along the y-axis
      double see_coeff;            secondary electron emission coefficient
                                   of the limiter material
      double ini_temperature;      starting value for the surface temperature
      double heatconductance;      = 2/Sqrt(Pi *rho *c *k) to calculate
                                   the heating of the surface from the
                                   power load and the time
      double sheath_transmission;sheath transmission factor approx 8
      double interaction_depth;  for film deposition - calculating the local
                                   concentration of different species (linearly)
                                   and depth profiling
};
```

5.2.5 Structures for Surface Data

For each cell on the limiter surface one structure "SurfaceCell" is allocated containing information about the geometry of and the local plasma parameter directly above this cell. Each SurfaceCell contains up to three (at this time) structures of the "Contents"-type, each representing one elemental specimen on the surface.

```
struct Contents
{
      char element[5];          name of the element
      double newgo;             how many atoms will be transported away in the
                                next time step (due to physical sputtering)
      double newnew;            how many atoms are transported to this cell in
                                this time step
      double newcome;           how many atoms are transported to this cell in
                                this time step
      double transport;         net transport onto this cell
                                (<0 = net erosion | >0 = net deposition)
      double resource;          resources of this element in this cell
      double energy;            mean energy of incident atoms/ions
      double charge;            mean charge state of incident atoms/ions
};

struct SurfaceCell
{
      double area;
      double edge[2];           edge with (xmin,ymin)
      double midth[3];          center coordinates of this cell
      double normal[3];         normal in the center
      double b_angle;           cosine of the angle (B-field,normal)
      double Te;                local electron temperature
```

```
      double Ti;                local ion temperature
      double ne;                local electron density
      double T_surf;            surface temperature
                                (i.e. substrate temperature)
      double AT_surf;           surface temperature
                                (after "timestep" seconds)
      double P;                 powerload on this surface cell in this
      double sub_ero;           timestep erosion from the bulk material the
                                the layer of thickness "interaction_depth"
                                (for concentration calculations)
      double layer[50][6];      for depth profiling each layer[j] contains
                                information on actual depth composition
      int    which[50][6];      number corresponding to specific name
      int    upmost;            which layer is upmost
      int    lowest;            which layer is lowest to be considered
      double cij[6];            local concentration ratio in the upmost
                                "interaction_depth" nm  (interaction volume)
      double energetic[6];      particles energetically released from
                                the limiter (physical sputtering)
      double newenergetic[6]; particles energetically released from
                                the limiter (physical sputtering) by
                                simulation particles
                                particles and transported away in the
                                next time step
      double thermic[6];        particles thermally released from the limiter
                                (i.e. chemical sputtering )
      double fluxtocell;        total flux of atoms to this cell in the
                                pre-previous time step

      struct Contents content[3];
      struct SurfaceCell *x_ptr;  pointing to the next cell in x direction
      struct SurfaceCell *y_ptr;  pointing to the next cell in y direction
};



struct LostParticle
{
      char   name[3][5];        name of species under investigation
      double x_max[3];          particles lost in xmax direction
      double y_max[3];          particles lost in ymax direction
      double x_min[3];          particles lost in xmin direction
      double y_min[3];          particles lost in ymin direction
      double z_min[7];          particles lost in zmin direction
      double z_max[7];          particles lost in zmax direction
      double time_out[7];       particles lost after time out signal
      double error[7];};        particles lost due to numerical instabilities
```

5.2.6 Structures for Spectroscopy

The pointer for spectroscopic data which is passed in ERO-TEXTOR from routine to routine is of the structure "VolumeAll"-type. This pointer refers to an array of pointers of the structure "Base"-type with each of them responsible for one column of volume cells above a rectangular mesh in the x-y-plane. Each "Base" then points to a column (array) of "VolumeCell"s, which itself points to an array of "atomps", where the spectroscopic data are stored. To calculate the emitted intensity of a certain impurity, the emission rates must be known. These are stored in the structure "Emission".

```
struct VolumeCell
{
      double *atomps;          array of data about the atomic/ionic specimen
                               in units of atoms/sec
      double z;                z-coordinate of the bottom of the cell
      double distance;         normal distance from the surface to the center
                               of this spectroscopic cell
};

struct Base
{
      double z_min;            minimum z-value of the column of spectroscopic
                               cells
      double z_max;            maximum z-value of the column of spectroscopic
                               cells
      double edge[2];          edge with  (xmin,ymin)
      struct VolumeCell **vcell; pointing to the different "VolumeCell"s
};

struct VolumeAll
{
      int    anz[4];           number of cells in the x-, y-, z-coordinate
                               direction [0,1,2] and number [3]
                               of spectroscopic elements under investigation
      double diff[3];          incremental length of a volume cell
      double rand[6];          boundary along the x- and y-axis of the block
                               of spectroscopic cells
      struct Base ***b;        pointing to the Base structure
      char   name[5][8];       name of the species under investigation
      int    charge[5];        charge state of the species under
                               investigation
      int    who_e;            variable for local manipulation
      int    who_q;            variable for local manipulation
};

struct Emission
{
      int    anz;              how many elements have to be traced
      int    found[5];         which of these were found in database
```

```
                                (0=not found  1=found 2= van Regemorter formula)
      char   snames[5][20];     spectroscopic names of these elements
      double ***ex;             array for database of emission rates
      double exmax[5][2];
};

struct TimeCell
{
      double total;             total time of simulation
      double step;              time per step
      int    anz_step;          number of steps
};
```

5.2.7 Handling-Structures

```
struct Path
{
      char CROSS_SECTION[200]; path to ionization rates data (atomic species)
      char RESULT[200];         without use
      char DATA[200];           path/file name where the Bodhansky-data
                                can be found
      char SPUTTER_INT[200];    path/file name where the background plasma
                                sputter yield can be found
      char REFLECTION[200];     path/file name where the reflection yields
                                can be found
      char EMISSION[200];       path/file name where the emission rates
                                can be found
};

struct Numeric
{
      float  t1;                incremental time for solving the equations
                                of motion
      float  t2;                incremental time near the surface
                                (in the sheath)
      float  coll_t;            incremental time for collisions
      float  t_out;             default time out
      float  box_rad;             size of tracing box along the z-axis
      float  box_radmin;        starting value along the z-axis
      float  box_pol;           size of tracing box along the y-axis
       float  box_tor;           size of tracing box along the x-axis
       float  hardwired;         value of the chemical erosion for hardwired
                                erosion yields $>=0 $[\%]  if the  Roth-formula
                                is used this value has to be $ <0$
      int    p_sputter;         secondary sputtering of simulation particles /
                                traced impurities on/off
      int    p_reflect;         reflection of simulation particles /
                                traced impurities on/off
     char pathtoresult[128]; path to / prefix of output files };
```

5.3 Flow Diagrams

The following four figures show the flow diagrams of the ERO-TEXTOR-code. After reading the input file the program kernel is started (fig. 5.1). The kernel then sets plasma and geometrical data, reads data from the databases and allocates the surface and spectroscopic volume net (fig. 5.2). Next the time loop is executed. In this loop the distinction between different origins (external source, physical sputtering) of particles is made and physical results output is organized (figs. 5.3 and 5.4). Finally the physical sputtering (analogous the external source loop) is executed, which - after setting the initial values of the particles (location, energy and angular values) - moves the neutral particle until it is ionized. The routine `part_go(...)` then solves the equations of motion for the ion until it leaves the simulation (by striking the surface of the limiter without being reflected or leaving the volume of observation) (figs. 5.5, 5.6 and 5.7).

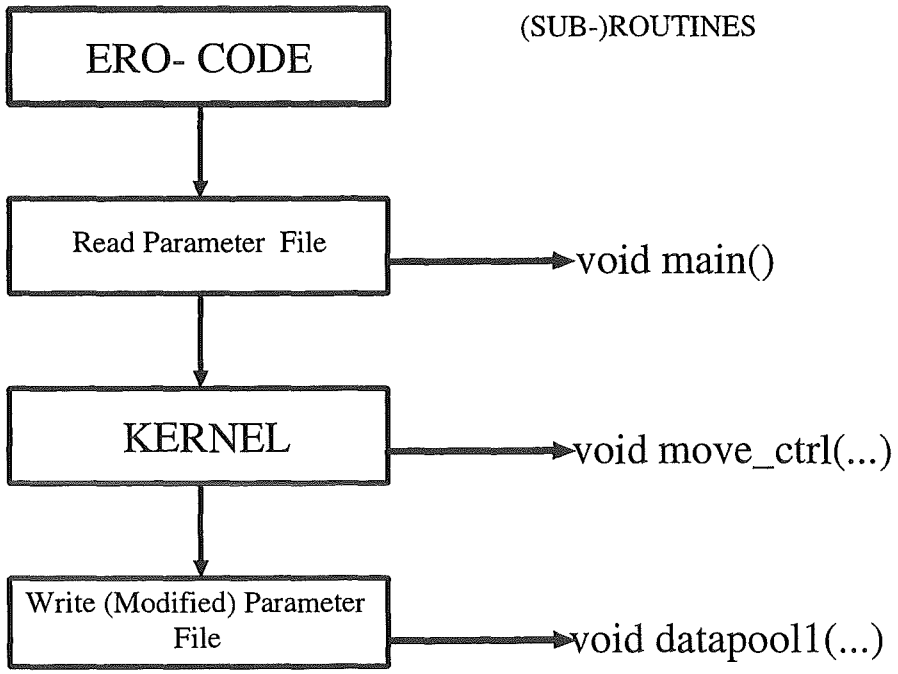


Figure 5.1: Major Program Loop

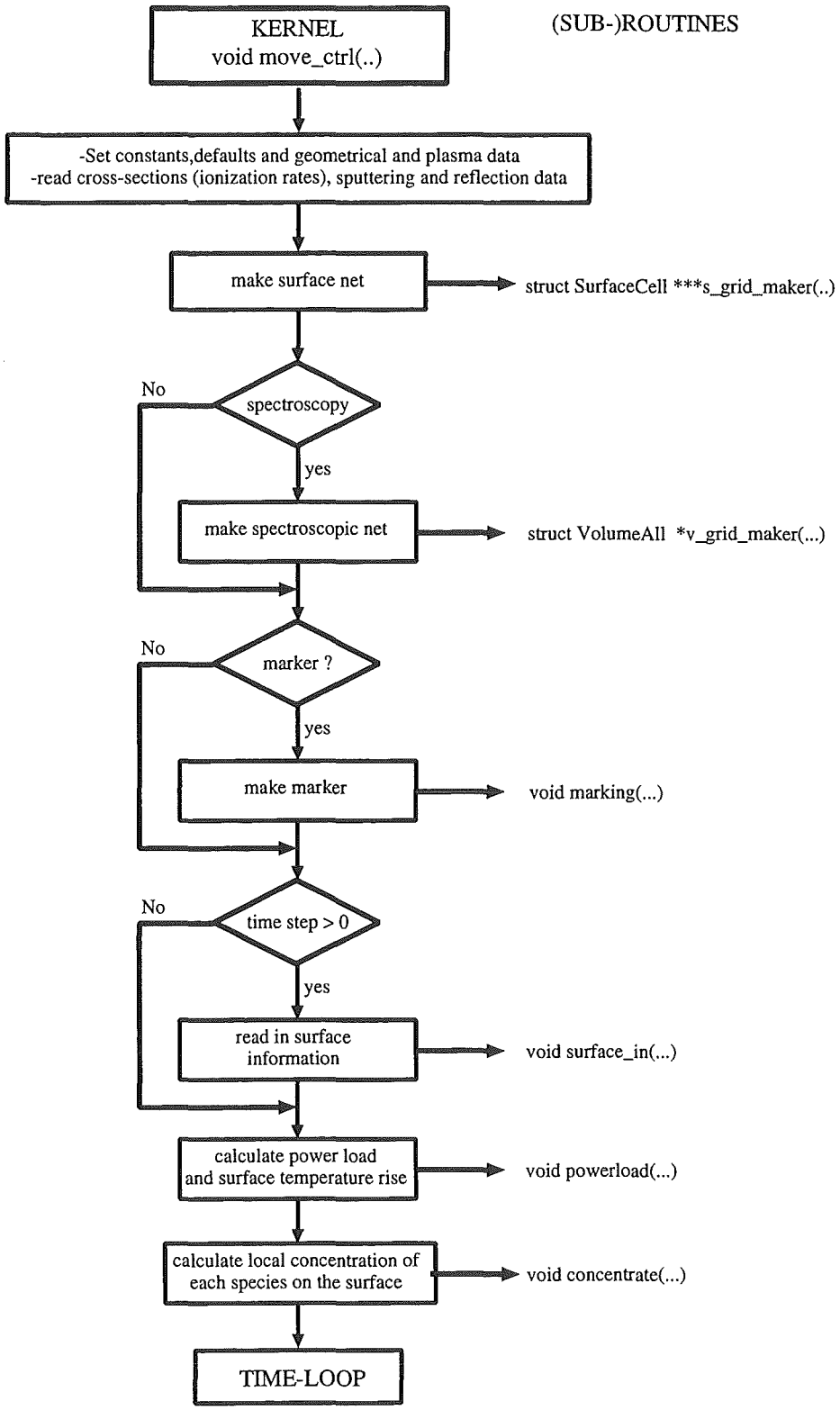


Figure 5.2: KERNEL-Routine move-ctrl. This routine organizes the plasma, geometrical data and partly the data from the databases. Then it handles control to the time loop, where for each time step the simulation is done.

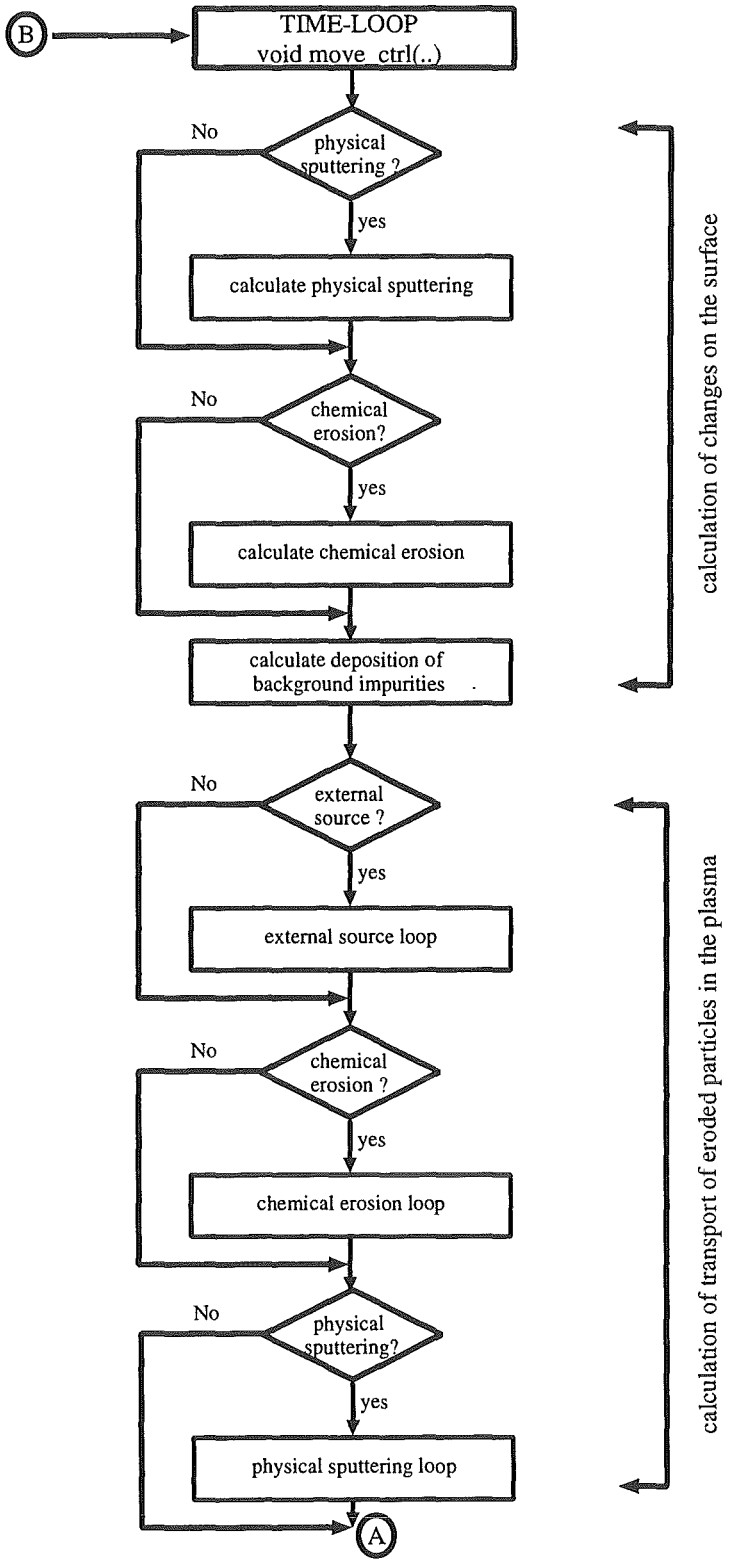


Figure 5.3: TIME-Loop. This loop is also part of the move-ctrl()-routine. After calculating the trajectories of the particles ("external source loop", "chemical erosion loop" and "physical sputtering loop"), the output of the results is done here.

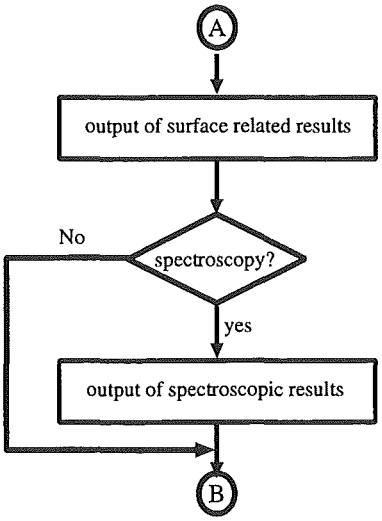


Figure 5.4: TIME-Loop (continued). This loop is also part of the move-ctrl()-routine. After calculating the trajectories of the particles ("external source loop", "chemical erosion loop" and "physical sputtering loop"), output of the results is done here.

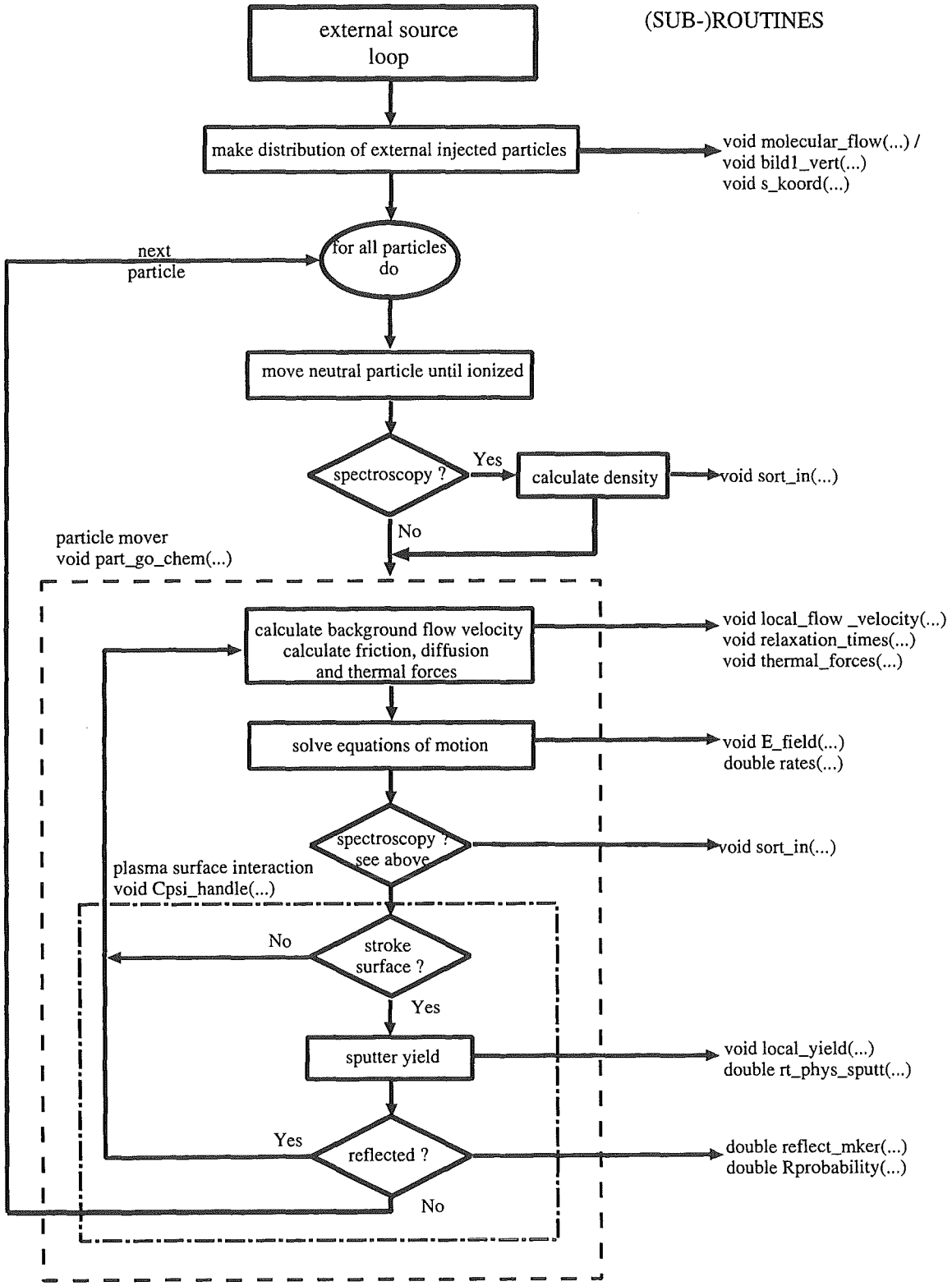


Figure 5.5: Particle Mover ("external source loop"). This routine calculates the initial values of each injected (quasi-)particle and solves the equations of motion.

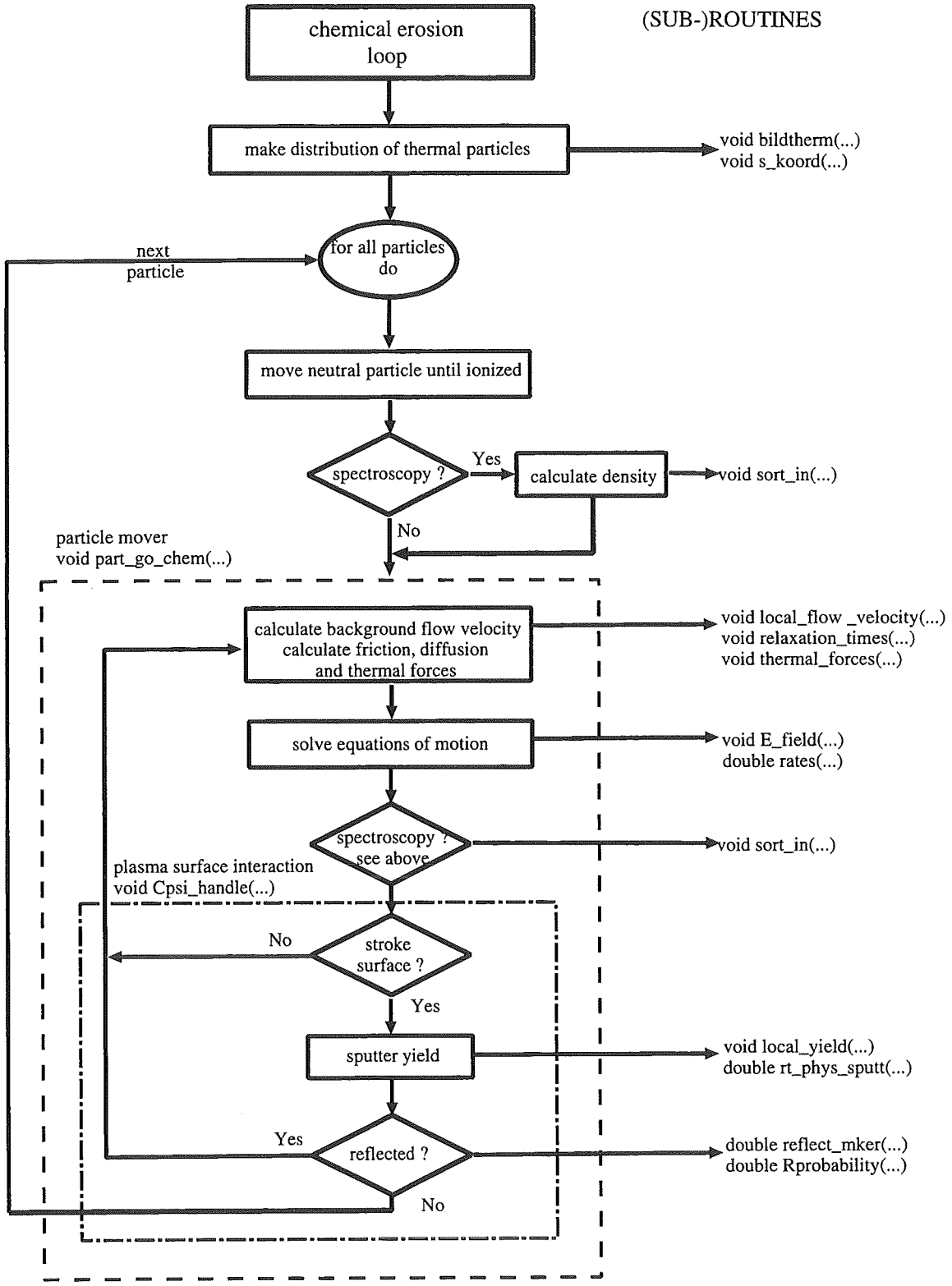


Figure 5.6: Particle Mover ("chemical erosion loop"). This routine calculates the initial values of each chemically eroded (quasi-)particle and solves the equations of motion.

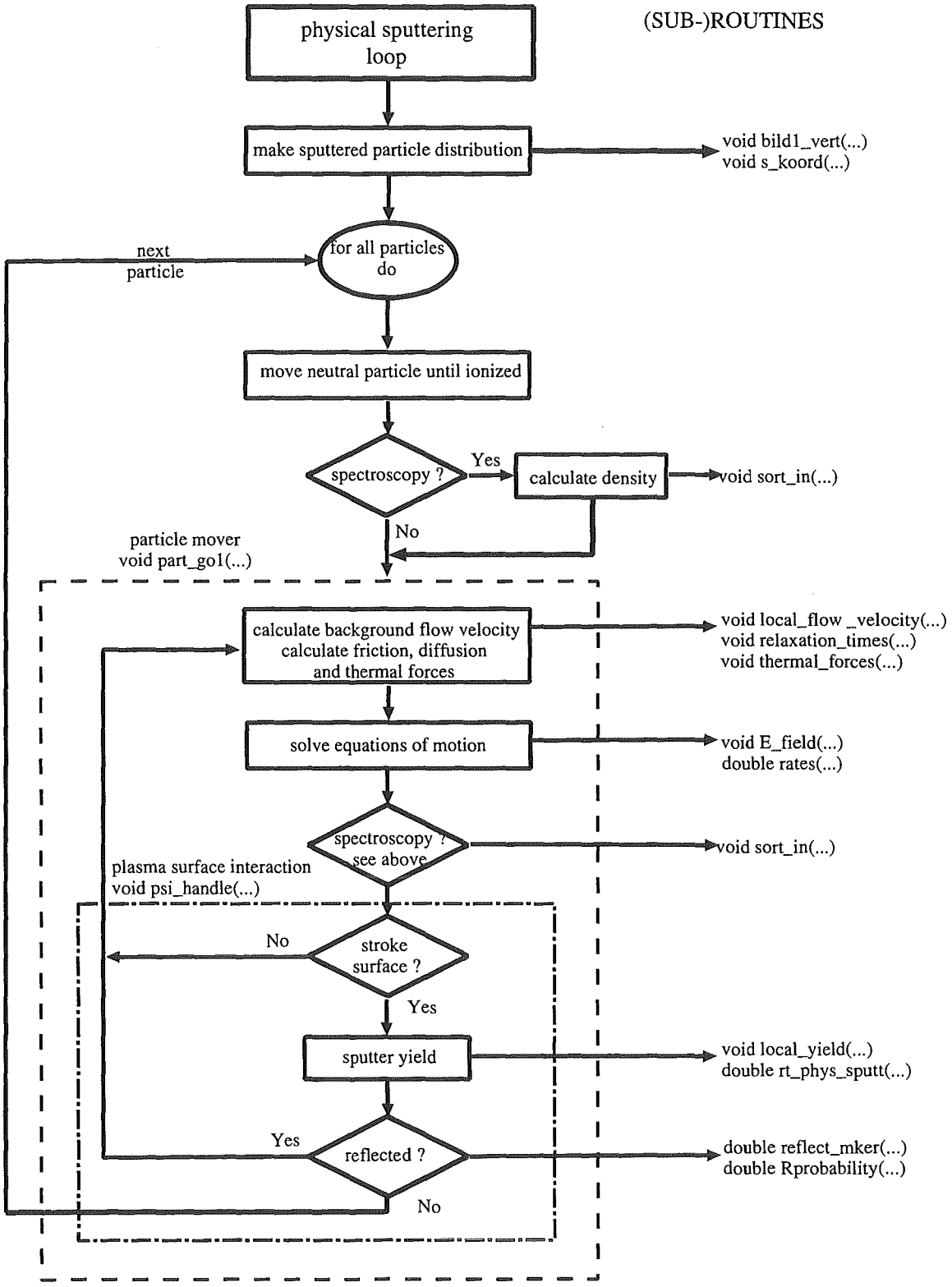


Figure 5.7: Particle Mover ("physical sputtering loop"). This routine calculates the initial values of each physically sputtered (quasi-)particle and solves the equations of motion.

5.4 ERO-TEXTOR-Routines

In this section the routines used to build the ERO-TEXTOR-code are briefly described. As can be seen from the flow diagrams (figs. 5.1, 5.2, 5.3, 5.5, 5.6 and 5.7) the major routines are the code kernel `move_ctrl()` and the routines solving the equations of motion `part_go1()` and `part_go_chem()`. A set of subroutines with one specific task complete the source code. As far as possible the layout for the routines was chosen as void. Therefore the table at each routine summarizes the variables or structures modified in this specific routine (to be written).

5.4.1 Major Routines

- Also not best programing technique input from the parameter file is read by "main()". Then the program kernel `move_ctrl()` is called and when done -physical data output ref. `move_ctrl()`- the output function "datapool1" writes the parameters to the `␣␣␣ERO.PRM` file.

```
void main();
```

- The program kernel of the code is the `move_ctrl`-routine. This routine is responsible for most of data loading, calculations of plasma and geometry data, handling and interaction of all other routines, best explained in figs. 5.1, 5.2 and 5.3.

```
void move_ctrl( struct LoopDialogData     *,
                struct NumericDialogData  *,
                struct PlasmaDialogData   *,
                struct ElementDialogData  *,
                struct LimiterDialogData  *,
                struct TeilchenDialogData *,
                struct MarkerDialogData   *,
                struct ChemieDialogData   *,
                struct SurfaceGridData    *,
                struct VolumenGridData    *,
                struct V_TraceData        *,
                struct TimeGridData       *);
```

- The integration of the equations of motion is done by the `part_go()`-routines. For physical sputtered and atomic external source particles this is done in `part_go1()`, whereas for methane-like molecules, due to the different ionization/dissociation scenario, this is evaluated in `part_go_chem()`.

```
void part_go1(  struct ParticleStruct  *,
                struct ParticleStruct  *,
                struct ParticleStruct  *,
                struct SurfaceCell     *,
                struct LostParticle    *,
                struct SurfaceGridData *,
                int                 *rand,
                double             *bound,
                struct MaterialData    *,
                struct ReflectionData  *,
                struct VolumeAll       *,
                int              spec_inf,
                struct Ion             *,
                struct Ion             *,
                struct Numeric         *,
                struct Plasma          *,
                struct nTdata          *,
                struct Magnetic        *,
                struct Electric        *,
                struct Diffusion       *,
                struct Ort             *,
                struct BRotator        *,
                struct V_PM            *,
                struct BField          *,
                struct EField          *,
                struct Force           *,
                struct PConst          *,
                int                   suf,
                int            which_elem,
                long int            *dum
              );
```

```
void part_go_chem( struct ParticleStruct       *,
                   struct ParticleStruct       *,
                   struct ParticleStruct       *,
                   struct SurfaceCell          *,
                   struct LostParticle         *,
                   struct SurfaceGridData      *,
                   int                      *rand,
                   double                  *bound,
                   struct MaterialData         *,
                   struct ReflectionData       *,
                   struct VolumeAll            *,
                   int                   spec_inf,
                   struct Chemie               *,
                   struct Chemie               *,
                   struct ChemConst            *,
                   double            **proz_tree,
                   double          **mass_charge,
                   double             **speicher,
                   double           ***speicherp,
                   double             *si_factor,
                   double          **siproz_tree,
                   double        **simass_charge,
                   double           **sispeicher,
                   double         ***sispeicherp,
                   struct Ion                  *,
                   struct Ion                  *,
                   struct Numeric              *,
                   struct Plasma               *,
                   struct nTdata               *,
                   struct Magnetic             *,
                   struct Electric             *,
                   struct Diffusion            *,
                   struct Ort                  *,
                   struct BRotator             *,
                   struct V_PM                 *,
                   struct BField               *,
                   struct EField               *,
                   struct Force                *,
                   struct PConst               *,
                   int                       suf,
                   int                 which_elem,
                   long int                  *dum
                  );
```

- The plasma surface interaction of the traced impurities is calculated in the `psi_handle`-routines. Due to the nature of different reflection models of the ions and molecular ions a different routine for the molecular ions `Cpsi_handle` is necessary.

```
int psi_handle(  struct ParticleStruct          *,
                 struct Numeric                 *,
                 struct EField                  *,
                 struct SurfaceCell             *,
                 struct LostParticle            *,
                 struct SurfaceGridData         *,
                 int                          *rand,
                 double                      *bound,
                 struct MaterialData            *,
                 struct ReflectionData          *,
                 int                    welch_elem,
                 long int                     *idum
              );
```

```
int Cpsi_handle( struct ParticleStruct          *,
                 struct Numeric                 *,
                 struct EField                  *,
                 struct SurfaceCell             *,
                 struct LostParticle            *,
                 struct SurfaceGridData         *,
                 int                          *rand,
                 double                      *bound,
                 struct MaterialData            *,
                 struct ReflectionData          *,
                 struct ChemConst               *,
                 struct Chemie                  *,
                 int                    welch_elem,
                 long int                     *idum
               );
```

- For each collision cycle the background plasma flow velocity has to be evaluated to determine the force due to dynamic friction and the diffusion problem has to be solved. For the user specified geometry this is done in the `local_flow_velocity`-routine. For toroidal geometry of the test limiter this is done by numerical iteration and therefore very slow, for all other geometries an analytic problem is solved.

```
int local_flow_velocity( struct ParticleStruct *,
                         struct Numeric        *,
                         struct Plasma         *,
                         struct nTdata         *,
                         struct Magnetic       *,
                         struct Ort            *,
                         struct BField         *,
                         struct PConst         *,
                         int                  suf,
                         double             w_frq6,
                         double            delta_t,
                         long int           *idum
                       );
```

- to solve the friction and diffusion problem the coefficients $\{\triangle w_{\|}\}$, $\{\triangle w_{\|}^2\}$ and $\{\triangle w_{\perp}^2\}$ have to be calculated, this is done in the `relaxation_times`-routine.

```
void relaxation_times( struct ParticleStruct *,
                       struct Plasma         *,
                       struct PConst         *,
                       struct nTdata         *,
                       struct Numeric        *,
                       struct Magnetic       *,
                       struct Ort            *,
                       struct BField         *,
                       int                  surf
                     );
```

- To calculate the local electron density, the local pre-sheath field and the gradient of the temperatures, the distance of the particle from the limiter surface has to be known. This is calculated in the `local_dl`-routine. For toroidal geometry of the test limiter this is done by numerical iteration and therefore very slow, for all other geometries an analytic problem is solved.

```
int local_dl( double                    x,
              double                    y,
              double                    z,
              struct Numeric            *,
              struct Magnetic           *,
              struct Ort                *,
              struct BField             *,
              struct PConst             *,
              int                      suf
            );
```

- the values of α and β and the thermal forces $\alpha\frac{dT_e}{ds}$ and $\beta\frac{dT_i}{ds}$ are calculated in the `thermal_force`-routine.

```
void thermal_force( struct ParticleStruct    *,
                    struct nTdata            *,
                    struct Numeric           *,
                    struct Plasma            *,
                    struct Magnetic          *,
                    struct Ort               *,
                    struct BField            *,
                    struct Force             *,
                    struct PConst            *,
                    int                     suf
                  );
```

- To simplify the change of the calculation of electron density and temperature and ion temperature, these calculations are done in separate routines, called `Te`, `Ti` and `ne`.

```
double Te( double           x,
           double           y,
           double           z,
           struct nTdata    *,
           struct Numeric   *,
           struct Magnetic  *,
           struct Ort       *,
           struct BField    *,
           struct PConst    *,
           int            suf
         );

double Ti( double           x,
           double           y,
           double           z,
           struct nTdata    *,
           struct Numeric   *,
           struct Magnetic  *,
           struct Ort       *,
           struct BField    *,
           struct PConst    *,
           int            suf
         );


double ne( double           x,
           double           y,
           double           z,
           struct nTdata    *,
           struct Numeric   *,
           struct Magnetic  *,
           struct Ort       *,
           struct BField    *,
           struct PConst    *,
           struct Plasma    *,
           int            suf
         );
```

- For the calculation of the thermal forces the gradients of electron temperature (up to now $\nabla T_e = 0$ and ion temperature have to be calculated.

```
void grad_Ti( double                    x,
              double                    y,
              double                    z,
              struct nTdata             *,
              struct Numeric            *,
              struct Plasma             *,
              struct Magnetic           *,
              struct Ort                *,
              struct BField             *,
              struct PConst             *,
              int                      suf
             );

void grad_Te( double                    x,
              double                    y,
              double                    z,
              struct nTdata             *,
              struct Numeric            *,
              struct Plasma             *,
              struct Magnetic           *,
              struct Ort                *,
              struct BField             *,
              struct PConst             *,
              int                      suf
             );
```

- For each time step the local electric field is computed in the `E_field`-routine

```
void E_field( struct ParticleStruct *,
              struct Numeric        *,
              struct Ort            *,
              struct Electric       *,
              struct BField         *,
              struct Magnetic       *,
              struct EField         *,
              struct Plasma         *,
              struct nTdata         *,
              struct PConst         *,
              int              decide);
```

5.4.2 Sub-Routines

- In `B_field` the input data for the magnetic field vector are converted in code usable data.

```
void B_field( struct Magnetic *,
              struct BField   *);
```

- The following two routines convert the names of elements to ERO-TEXTOR-numbers ("ElemNo") and vice versa ("NoElem") according to ERO-TEXTOR nomenclature (see tab. 3.1).

```
void NoElem(struct Spezies *);
```

```
void ElemNo(struct Spezies *);
```

- The following four routines are used in plasma chemistry of methane/ silane molecules (due to the fact that there are two different rate coefficient system all routines exist for methane and for silane, with the exception of the change_status-routine, which deals with both molecules). In each time step in change_status it is calculated whether the molecule is changed by any process (dissociation, ionization, etc.) or not. Therefore the two routines chem_rate for electron impact processes and chem_rate_prot for proton impact processes are evaluated. The prozesses-routine loads the necessary data.

```
int change_status( struct ParticleStruct *,
                   struct Chemie         *,
                   struct Chemie         *,
                   struct ChemConst      *,
                   struct Plasma         *,
                   struct nTdata         *,
                   struct Numeric        *,
                   struct Magnetic       *,
                   struct Ort            *,
                   struct BField         *,
                   struct PConst         *,
                   double      **proz_tree,
                   double    **mass_charge,
                   double       **speicher,
                   double     ***speicherp,
                   double        *sifactor,
                   double    **siproz_tree,
                   double  **simass_charge,
                   double     **sispeicher,
                   double   ***sispeicherp,
                   double                dt,
                   int                surf,
                   long int           *dum
                 );

double chem_rate( int nummer     ,
                  double Te_l    ,
                  double **speich);

double chem_rate_prot( int nummer       ,
                       double Te_l      ,
                       double Ep        ,
                       double ***speichp);

void prozesses( double **proz,
                double ***m_q,
                double   **sp,
                double ***spc);
```

```
double si_chem_rate( int        nummer,
                     double       Te_l,
                     double   **speich,
                     double *si_factor
                   );


double si_chem_rate_prot( int        nummer,
                          double       Te_l,
                          double         Ep,
                          double ***speichp,
                          double *si_factor
                        );


void si_prozesses( double      **proz,
                   double       **m_q,
                   double        **sp,
                   double      ***spc,
                   double *c_2_si_fac
                 );
```

- The impurity content of the SOL plasma is not constant, but can increase further out. Therefore the mean mass, charge and impurity percentage depend on the location of the particle in the plasma. The following routines calculate these parameters locally.

```
double fiqi( double              x,
             double              y,
             double              z,
             struct Plasma       *,
             struct nTdata       *
           );

double impurity_percent( double              x,
                         double              y,
                         double              z,
                         struct Plasma       *,
                         struct nTdata       *,
                         int                 which
                       );

double z_effective( double              x,
                    double              y,
                    double              z,
                    struct Plasma       *,
                    struct nTdata       *
                  );

double q_plasma( double              x,
                 double              y,
                 double              z,
                 struct Plasma       *,
                 struct nTdata       *
               );
```

- Before each run the actual surface concentration of each element has to be calculated in the concentrate-routine.

```
void concentrate( struct SurfaceCell          *,
                  struct Ort                  *,
                  struct SurfaceGridData      *,
                  int                        *rand,
                  double                     *bound
                );
```

- After each run the actual total flux of atoms to the surface has to be calculated in the flux_to_cell-routine.

```
void flux_to_cell( struct SurfaceCell         *,
                   struct Ort                 *,
                   struct SurfaceGridData     *,
                   int                       *rand,
                   double                    *bound
                 );
```

- And before each run the power load to the limiter surface and the rise of the surface temperature are calculated in the powerload-routine.

```
void powerload(  struct SurfaceCell     *,
                 struct Path            *,
                 struct Plasma          *,
                 struct Ort             *,
                 struct TimeCell        *,
                 struct SurfaceGridData *,
                 struct nTdata          *,
                 int                   *rand,
                 double               *bound,
                 double            max_angle,
                 int              time_cycle
               );
```

- If markers are specified in the parameter file, this routine will write the resources to the cells of the surface net.

```
void marking(struct MarkerDialogData *,
             struct SurfaceCell      *,
             struct SurfaceGridData  *,
             int                    *rand,
             struct MaterialData     *);
```

- `bild_vert1` is called to build the energy and angle distribution of physical sputtered particles (in the physical sputtering loop), as described in eqn.(2.116), whereas `bildtherm` is called to build the energy and angle distribution of thermal particles leaving the surface with a maxwellian energy distribution.

```
void bild_vert1(struct ParticleStruct *,
                struct ParticleStruct *,
                struct Ort            *,
                struct Spezies        *,
                double                E0,
                double                M1,
                double              atom,
                int                  suf,
                long int            *dum);
```

```
void bildtherm( struct ParticleStruct *,
                struct ParticleStruct *,
                struct Ort            *,
                struct Spezies        *,
                double                E0,
                double              atom,
                int                 loop,
                int                  suf,
                long int            *dum
              );
```

- A set of separate routines exist for the reflection of ions and molecules on the surface: the `reflect_mker`-routine builds the angle and energy dependence of the reflected ions, whereas the `Rprobability`-routine answers the question whether a particle is reflected or not. A special routine is used to read in the very large data files for the reflection data. For the reflection of the molecular particles, there are two routines: the `bild_molecule`-routine which creates the name of the reflected molecular particles and the `bild_reflection`-routine to bild the angle and energy dependence of them.

```
double reflect_mker( struct ParticleStruct          *,
                     struct SurfaceCell             *,
                     struct ReflectionData          *,
                     long int                      *dum
                   );


double Rprobability( struct ParticleStruct          *,
                     struct SurfaceCell             *,
                     struct ReflectionData          *,
                     double                       E_in,
                     double                       w_in,
                     long int                      *dum
                   );


void read_reflection( struct ReflectionData *refl_ptr,
                      struct Path           *pfad
                    );

void bild_molecule( struct ParticleStruct       *,
                    struct Chemie               *,
                    int                       which
                  );

void bild_reflection( struct ParticleStruct       *,
                      struct SurfaceCell          *,
                      double                     E0,
                      int                    Renergy,
                      long int                 *dum
                    );
```

- The s_koord-routine calculates the initial values of location for the particles (in the external source loop as well as in the physical sputtering loop).

```
void s_koord( struct ParticleStruct *,
              struct ParticleStruct *,
              struct Ort            *,
              int                   xi,
              int                   suf);
```

- For the external source loop the angle distribution is computed here:

```
void s_angle( struct ParticleStruct *,
              struct ParticleStruct *,
              struct Spezies        *,
              int                   xii,
              long int              *dum);
```

- The following routine calculates the angle distribution of particles released from a thin throat with length to radius L/r under molecular flow conditions:

```
void molecular_flow( double   **speicher,
                     int        anzahl,
                     int        which,
                     double     ratio,
                     long int   *idum
                   );
```

- For the external source loop the energy distribution is computed here:

```
void s_energy( struct ParticleStruct *,
               struct ParticleStruct *,
               struct Spezies        *,
               double              atome,
               int                 xiii,
               long int            *dum);
```

- This routine calculates (in each time step) the ionization rate for the specific particle, according eqns.(3.1) and (3.2).

```
double rates( struct Ion            *,
              struct Ion            *,
              struct ParticleStruct *,
              double                E);
```

- Called in external source loop or in the physical sputter loop this routine loads the data for calculating the rate coefficients (see eqns.(3.1) and (3.2)).

```
void l_ratecoeff( struct Ion            *,
                  struct Ion            *,
                  struct ParticleStruct *p,
                  int                max_q,
                  struct Path           *);
```

- Here the physical sputter yield $Y_{I \rightarrow T}(E_{in}, \theta_{in})[\frac{atoms}{ion}]$ for an incident ion I to a target material T is calculated: While `local_yield` is necessary for the data handling, the `rt_phys_sputt`-routine and the `phys_sputt`-routine evaluate the yield according the *Bodhansky*- and *Yamamura*-formula.

```
void local_yield( struct ParticleStruct          *wor_ptr,
                  struct SurfaceCell               *n_ptr,
                  struct SurfaceGridData     *sgrid_dialog,
                  struct MaterialData              *mater
                );
```

```
double rt_phys_sputt( double daten[10],
                      double        eo,
                      double     c_win);
```

```
double phys_sputt( double daten[10],
                   double        eo,
                   double      win);
```

- For carbon based substrates the following routines calculate the sputter yield due to chemical erosion by oxygen -chem_sputt()- and the sputter yield due to radiation enhanced sublimation (RES) -res_sput()- due to the ion flux to the surface.

```
double chem_sputt( char        code,
                   double daten[10],
                   double         T,
                   double        eo,
                   double     GAMMA);
```

-RES is not yet implemented -

```
double res_sputt( double daten[10],
                  double         T,
                  double        eo,
                  double     GAMMA);
```

- This subroutine allocates the cells of the surface net and fills them with basic information, such as coordinates, area and normal of the cell. Additionally the names of specimen are written to the cells. At the beginning of each time step the `surface_in`-routine reads in the actual surface conditions from a temporary file. At the end of each run the calculated changes of the surface conditions are written to result and temporary files by the `surface_out`-routine. On exit of the program `s_grid_free` frees the allocated memory.

```
struct SurfaceCell ***s_grid_maker( struct LimiterDialogData  *,
                                    struct PlasmaDialogData   *,
                                    struct Plasma             *,
                                    struct Magnetic           *,
                                    struct SurfaceGridData    *,
                                    int                  *s_rand,
                                    double              *s_bound);


void surface_out( struct SurfaceCell       *,
                  struct Path              *,
                  struct TimeCell          *,
                  struct LostParticle      *,
                  struct SurfaceGridData   *,
                  int                  *rand,
                  double              *bound,
                  int             time_cycle
                );


void surface_in(  struct SurfaceCell       *,
                  struct Path              *,
                  struct TimeCell          *,
                  struct LostParticle      *,
                  struct SurfaceGridData   *,
                  int                  *rand,
                  double              *bound,
                  int             time_cycle
                );


void   s_grid_free( struct SurfaceCell ***,
                    int      *s_rand      );
```

- As before in `v_grid_maker` the volume cells for the spectroscopic data are allocated and filled with basic information, such as coordinates of the cells and specimen names and ionization states for which the density has to be calculated. `volume_out` writes the spectroscopic data (density) to the output files. The `v_grid_free` frees the allocated memory on exit.
  In `load_emission` memory is allocates and the emission rates of the relevant species are loaded from the databases. The `free_emission` frees the allocated memory. `emission_out` writes the spectroscopic data (emitted intensity) to the output files.

```
struct VolumeAll *v_grid_maker( struct Ort              *,
                                struct VolumenGridData  *,
                                struct V_TraceData      *,
                                int               decide);


void volume_out( struct VolumeAll *,
                 struct Path      *,
                 struct TimeCell  *,
                 int           step);


void   v_grid_free( struct VolumeAll *);


void load_emission(struct Path         *,
                   struct Emission     *,
                   struct V_TraceData  *
                  );


void free_emission(struct Emission *);


void emission_out(struct VolumeAll  *,
                  struct Emission   *,
                  struct Path       *,
                  struct TimeCell   *,
                  struct nTdata     *,
                  struct Numeric    *,
                  struct Magnetic   *,
                  struct Ort        *,
                  struct BField     *,
                  struct PConst     *,
                  struct Plasma     *,
                  int             suf,
                  int            step
                  );
```

- This subroutine calculates the spectroscopic density $\frac{\text{particles}}{cm^3}$ of a specific ion/atom at its location.

```
void sort_in( struct ParticleStruct  *,
              struct VolumeAll       *,
              double           delta_t
            );
```

- This subroutine calculates the emitted line integrated intensity $\frac{\text{photons}}{cm^2\,ssr}$ of a specific ion/atom at.

```
double emission(struct Emission   *,
                double           Te,
                int           which
                );
```

- This routine writes the possibly modified parameters to the ␣␣␣ERO.PRM-file.

```
void datapool1( struct FileDialogData     *,
                struct FileDialogData     *,
                struct LoopDialogData     *,
                struct NumericDialogData  *,
                struct PlasmaDialogData   *,
                struct ElementDialogData  *,
                struct LimiterDialogData  *,
                struct TeilchenDialogData *,
                struct MarkerDialogData   *,
                struct ChemieDialogData   *,
                struct SurfaceGridData    *,
                struct VolumenGridData    *,
                struct V_TraceData        *,
                struct TimeGridData       *);

);
```

- The chain_submitter-routine creates the next batch-job for the IBM-LoadLeveler batch facility.

```
void chain_submitter( int actl_step,
                      int job_class
                    );
```

Appendix A

A Rate Coefficient System for Silane in the Energy Range of 0.5 - 100eV

The reaction rates for the electron impact reactions of the silane molecule have been calculated from the cross sections by integrating over the energy distribution of the electrons:

$$< \sigma v > (T_e) = \frac{\int_{E_{th}}^{\infty} \sigma(E)\sqrt{\frac{2E}{m_e}} f(E,T_e)dE}{\int_0^{\infty} f(E,T_e)dE} \tag{A.1}$$

where the integration starts at the threshold energy of the reaction E_{th}. To calculate the reaction rate the assumption is made that the molecular radicals / ions are at rest compared to the fast moving electrons. For the electrons an isotropic energy distribution in thermal equilibrium, i.e. an maxwellian electron energy distribution is assumed:

$$f(E,T_e)dE = \frac{2\sqrt{E}}{\sqrt{\pi}(k_B T_e)^{3/2}} e^{-\frac{E}{k_B T_e}} \tag{A.2}$$

Thus the reaction rate is given by [50]:

$$< \sigma v > = \sqrt{\frac{8k_B T_e}{\pi m_e}} y^2 \int_1^{\infty} \sigma(u) u e^{-yu} du \tag{A.3}$$

in partical units

$$= 6.69 \cdot 10^7 \sqrt{T_e} y^2 \int_1^{\infty} \sigma(u) u e^{-yu} du \tag{A.4}$$

with the electron energy in units of the threshold energy $u = \frac{E}{E_{th}}$ and $y = \frac{E_{th}}{k_B T_E}$. For the practical calculation the following units are used: T_e in [eV] and $\sigma(u)$ in $[cm^{-2}]$.

name	reaction	E_{th} [eV]	$\triangle E$ [eV] [1]	scaling[2]	data[3]
E 1	$SiH_4 + e \rightarrow SiH_3^+ + H + 2e$	12.2	12.2	-	[12]
E 2	$SiH_4 + e \rightarrow SiH_2^+ + 2H + 2e$	11.6	16.1	-	[12]
E 3	$SiH_4 + e \rightarrow SiH^+ + H_2 + H + 2e$	14.7	14.7	-	[12]
E 4	$SiH_4 + e \rightarrow Si^+ + H_2 + 2H + 2e$	13.3	18.4	-	[12]
E 5	$SiH_4 + e \rightarrow SiH_3 + H^+ + 2e$	24.5	24.5	-	[12]
E 6	$SiH_4 + e \rightarrow SiH_2 + H_2^+ + 2e$	24.5	24.5	-	[12]
D 1	$SiH_4 + e \rightarrow SiH_3 + H + e$	8.0	3.86	-	[40][12]
D 2	$SiH_4 + e \rightarrow SiH_2 + 2H + e$	8.0	6.9	-	[40][12]
D 3	$SiH_4 + e \rightarrow SiH + H_2 + H + e$	8.0	5.7	-	[40][12]
D 4	$SiH_4 + e \rightarrow Si + H_2 + 2H + e$	8.0	8.74	-	[40][12]
I 1	$SiH_3 + e \rightarrow SiH_3^+ + 2e$	11.6	8.0	$\frac{4}{5}\cdot$ DI[4]	[12]
E 7	$SiH_3 + e \rightarrow SiH_2^+ + H + 2e$	12.2	12.2	$\frac{1}{5}\cdot$ DI	-
E 8	$SiH_3 + e \rightarrow SiH_2 + H^+ + 2e$	24.5	24.5	$\frac{3}{4}\cdot$ E 5	-
D 5	$SiH_3 + e \rightarrow SiH_2 + H + e$	11.6	3.0	$\frac{3}{4}\cdot$ ND[5]	[12]
I 2	$SiH_2 + e \rightarrow SiH_2^+ + 2e$	11.6	9.4	$\frac{4}{5}\cdot$ DI	-
E 9	$SiH_2 + e \rightarrow SiH^+ + H + 2e$	12.2	12.5	$\frac{1}{5}\cdot$ DI	-
E 10	$SiH_2 + e \rightarrow SiH + H^+ + 2e$	24.5	24.5	$\frac{1}{2}\cdot$ E 5	-
D 6	$SiH_2 + e \rightarrow SiH + H + e$	11.6	3.3	$\frac{1}{2}\cdot$ ND	-
I 3	$SiH + e \rightarrow SiH^+ + 2e$	11.6	9.0	$\frac{4}{5}\cdot$ DI	-
E 11	$SiH + e \rightarrow Si^+ + H + 2e$	12.2	12.7	$\frac{1}{5}\cdot$ DI	-
E 12	$SiH + e \rightarrow Si + H^+ + 2e$	24.5	24.5	$\frac{1}{4}\cdot$ E 5	-
D 7	$SiH + e \rightarrow Si + H + e$	11.6	3.0	$\frac{1}{4}\cdot$ ND	-
R 1	$SiH_3^+ + e \rightarrow SiH_2 + H$	-	≈ 17	Langer[22]	-
A 1	$SiH_3^+ + e \rightarrow SiH_2^+ + H + e$	11.6	3.0	$\frac{3}{4}\cdot D5$	-
A 2	$SiH_3^+ + e \rightarrow SiH_2 + H^+ + e$	24.5	3.0	$\frac{3}{4}\cdot D5$	-
R 2	$SiH_2^+ + e \rightarrow SiH + H$	-	≈ 17	Langer[22]	-
A 3	$SiH_2^+ + e \rightarrow SiH^+ + H + e$	11.6	3.3	$\frac{1}{2}\cdot D6$	-
A 4	$SiH_2^+ + e \rightarrow SiH + H^+ + e$	24.5	3.3	$\frac{1}{2}\cdot D6$	-
R 3	$SiH^+ + e \rightarrow Si + H$	-	≈ 17	Langer[22]	-
A 5	$SiH^+ + e \rightarrow Si^+ + H + e$	11.6	3.0	$\frac{1}{2}\cdot D7$	-
A 6	$SiH^+ + e \rightarrow Si + H^+ + 2e$	24.5	3.0	$\frac{1}{2}\cdot D7$	-
proton impact reactions					
CX 1	$SiH_4 + p \rightarrow SiH_2^+ + 2H + H + 2e$	-	0.0	Langer[22]	-
CX 2	$SiH_3 + p \rightarrow SiH_3^+ + H + 2e$	-	0.0	Langer[22]	-
CX 3	$SiH_2 + p \rightarrow SiH_2^+ + H + 2e$	-	0.0	Langer[22]	-
CX 4	$SiH + p \rightarrow SiH^+ + H + 2e$	-	0.0	Langer[22]	-
CX 4	$Si + p \rightarrow Si^+ + H + 2e$	-	0.0	Langer[22]	-

[1] These values are taken from Schneider [50], except those for the two reaction types:
1) $SiH_x + e \rightarrow SiH_{x-n} + H_n^+ + 2e$ with $n = 1, 2$ and
2) $SiH_x + e \rightarrow SiH_{x-1} + H$.

[2] if no measured cross sections exist, the reaction rates will be scaled with this factor to the mentioned reaction.

[3] where the measured cross sections are taken from.

[4] dissociative ionization of the SiH_4-molecule.

[5] neutral dissociation of the SiH_4-molecule.

A.1 Fitting of the cross sections

For the fitting of the cross sections different fitting formulae have been used:

- $SiH_4 + e \rightarrow SiH_3^+ + H + 2e$
  Fit: Lotz-formula[6] with coefficients:
  $a_0 = 1.084112 \cdot 10^2$
  $a_1 = 6.071566 \cdot 10^{-1}$
  $a2 = 1.29847 \cdot 10^{-1}$

- $SiH_4 + e \rightarrow SiH_2^+ + H_2 + 2e$
  Fit: Lotz-formula with coefficients:
  $a_0 = 1.355031 \cdot 10^2$
  $a_1 = 6.44493 \cdot 10^{-1}$
  $a2 = 1.139226 \cdot 10^{-1}$

- $SiH_4 + e \rightarrow SiH^+ + H_2 + H + 2e$
  Fit: Splines

- $SiH_4 + e \rightarrow Si^+ + H_2 + 2H + 2e$
  Fit: Splines

- $SiH_4 + e \rightarrow SiH_3 + H^+ + 2e$
  Fit: Splines

- $SiH_4 + e \rightarrow SiH_2 + H_2^+ + 2e$
  Fit: Splines

- total dissociation of SiH_4
  Fit: Splines

- neutral dissociation of SiH_4
  Fit: Splines

- total dissociative ionization of SiH_4
  Fit: Splines

A.2 Fit Parameters of the reaction rates

- $SiH_4 + e \rightarrow SiH_3^+ + H + 2e$
  $b = 1.00$
  $b_0 = -2.98154 \cdot 10^{+1}$ $b_1 = +1.26599 \cdot 10^{+1}$ $b_2 = -6.10431 \cdot 10^{+0}$
  $b_3 = +2.01061 \cdot 10^{+0}$ $b_4 = -4.87348 \cdot 10^{-1}$ $b_5 = +8.81920 \cdot 10^{-2}$
  $b_6 = -1.12818 \cdot 10^{-2}$ $b_7 = +8.84541 \cdot 10^{-4}$ $b_8 = -3.11082 \cdot 10^{-5}$

[6] The fit with Lotz formula used for energies $E \leq 200eV$ was taken from [50] where the Lotz-fit [31] is given by:

$$\sigma = \frac{a_0}{E} \ln \frac{E}{E_{th}} \left[1 - a_1 \cdot e^{-a_2 \left(\frac{E}{E_{th}} - 1 \right)} \right]$$

for higher energies a spline fit was used.

- $SiH_4 + e \rightarrow SiH_2^+ + 2H + 2e$
  $b = 1.00$
  $b_0 = -2.90427 \cdot 10^{+1}$ $b_1 = +1.20595 \cdot 10^{+1}$ $b_2 = -5.80462 \cdot 10^{+0}$
  $b_3 = +1.91152 \cdot 10^{+0}$ $b_4 = -4.63135 \cdot 10^{-1}$ $b_5 = +8.37831 \cdot 10^{-2}$
  $b_6 = -1.06972 \cdot 10^{-2}$ $b_7 = +8.33755 \cdot 10^{-4}$ $b_8 = -2.90057 \cdot 10^{-5}$

- $SiH_4 + e \rightarrow SiH^+ + H_2 + H + 2e$
  $b = 1.00$
  $b_0 = -3.37217 \cdot 10^{+1}$ $b_1 = +1.52473 \cdot 10^{+1}$ $b_2 = -8.42467 \cdot 10^{+0}$
  $b_3 = +4.00281 \cdot 10^{+0}$ $b_4 = -1.52947 \cdot 10^{+0}$ $b_5 = +4.02745 \cdot 10^{-1}$
  $b_6 = -6.61927 \cdot 10^{-2}$ $b_7 = +6.08508 \cdot 10^{-3}$ $b_8 = -2.38863 \cdot 10^{-4}$

- $SiH_4 + e \rightarrow Si^+ + H_2 + 2H + 2e$
  $b = 1.00$
  $b_0 = -3.41568 \cdot 10^{+1}$ $b_1 = +1.39865 \cdot 10^{+1}$ $b_2 = -7.37975 \cdot 10^{+0}$
  $b_3 = +4.08995 \cdot 10^{+0}$ $b_4 = -1.96068 \cdot 10^{+0}$ $b_5 = +6.28791 \cdot 10^{-1}$
  $b_6 = -1.21818 \cdot 10^{-1}$ $b_7 = +1.28853 \cdot 10^{-2}$ $b_8 = -5.71620 \cdot 10^{-4}$

- $SiH_4 + e \rightarrow SiH_3 + H^+ + 2e$
  $b = 1.00$
  $b_0 = -4.53938 \cdot 10^{+1}$ $b_1 = +2.49633 \cdot 10^{+1}$ $b_2 = -1.26745 \cdot 10^{+1}$
  $b_3 = +4.71172 \cdot 10^{+0}$ $b_4 = -1.22843 \cdot 10^{+0}$ $b_5 = +2.05980 \cdot 10^{-1}$
  $b_6 = -2.03181 \cdot 10^{-2}$ $b_7 = +1.00192 \cdot 10^{-3}$ $b_8 = -1.50767 \cdot 10^{-5}$

- $SiH_4 + e \rightarrow SiH_2 + H_2^+ + 2e$
  $b = 1.00$
  $b_0 = -4.75861 \cdot 10^{+1}$ $b_1 = +2.70810 \cdot 10^{+1}$ $b_2 = -1.39828 \cdot 10^{+1}$
  $b_3 = +5.84433 \cdot 10^{+0}$ $b_4 = -2.06347 \cdot 10^{+0}$ $b_5 = +5.40212 \cdot 10^{-1}$
  $b_6 = -9.20076 \cdot 10^{-2}$ $b_7 = +8.91458 \cdot 10^{-3}$ $b_8 = -3.70806 \cdot 10^{-4}$

- $SiH_4 + e \rightarrow SiH_3 + H + e$
  $b = 0.35$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH_4 + e \rightarrow SiH_2 + 2H + e$
  $b = 0.50$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH_4 + e \rightarrow SiH + H_2 + H + e$
  $b = 0.10$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH_4 + e \rightarrow Si + H_2 + 2H + e$
  $b = 0.05$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH_3 + e \rightarrow SiH_3^+ + 2e$
  $b = 0.80$
  $b_0 = -2.75585 \cdot 10^{+1}$ $b_1 = +1.02656 \cdot 10^{+1}$ $b_2 = -4.04565 \cdot 10^{+0}$
  $b_3 = +9.18383 \cdot 10^{-1}$ $b_4 = -9.11069 \cdot 10^{-2}$ $b_5 = -8.53707 \cdot 10^{-3}$
  $b_6 = +3.63420 \cdot 10^{-3}$ $b_7 = -4.15887 \cdot 10^{-4}$ $b_8 = +1.76030 \cdot 10^{-5}$

- $SiH_3 + e \rightarrow SiH_2^+ + H + 2e$
  $b = 0.20$
  $b_0 = -2.75585 \cdot 10^{+1}$ $b_1 = +1.02656 \cdot 10^{+1}$ $b_2 = -4.04565 \cdot 10^{+0}$
  $b_3 = +9.18383 \cdot 10^{-1}$ $b_4 = -9.11069 \cdot 10^{-2}$ $b_5 = -8.53707 \cdot 10^{-3}$
  $b_6 = +3.63420 \cdot 10^{-3}$ $b_7 = -4.15887 \cdot 10^{-4}$ $b_8 = +1.76030 \cdot 10^{-5}$

- $SiH_3 + e \rightarrow SiH_2 + H^+ + 2e$
  $b = 0.75$
  $b_0 = -4.53938 \cdot 10^{+1}$ $b_1 = +2.49633 \cdot 10^{+1}$ $b_2 = -1.26745 \cdot 10^{+1}$
  $b_3 = +4.71172 \cdot 10^{+0}$ $b_4 = -1.22843 \cdot 10^{+0}$ $b_5 = +2.05980 \cdot 10^{-1}$
  $b_6 = -2.03181 \cdot 10^{-2}$ $b_7 = +1.00192 \cdot 10^{-3}$ $b_8 = -1.50767 \cdot 10^{-5}$

- $SiH_3 + e \rightarrow SiH_2 + H + e$
  $b = 0.75$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH_2 + e \rightarrow SiH_2^+ + 2e$
  $b = 0.80$
  $b_0 = -2.75585 \cdot 10^{+1}$ $b_1 = +1.02656 \cdot 10^{+1}$ $b_2 = -4.04565 \cdot 10^{+0}$
  $b_3 = +9.18383 \cdot 10^{-1}$ $b_4 = -9.11069 \cdot 10^{-2}$ $b_5 = -8.53707 \cdot 10^{-3}$
  $b_6 = +3.63420 \cdot 10^{-3}$ $b_7 = -4.15887 \cdot 10^{-4}$ $b_8 = +1.76030 \cdot 10^{-5}$

- $SiH_2 + e \rightarrow SiH^+ + H + 2e$
  $b = 0.20$
  $b_0 = -2.75585 \cdot 10^{+1}$ $b_1 = +1.02656 \cdot 10^{+1}$ $b_2 = -4.04565 \cdot 10^{+0}$
  $b_3 = +9.18383 \cdot 10^{-1}$ $b_4 = -9.11069 \cdot 10^{-2}$ $b_5 = -8.53707 \cdot 10^{-3}$
  $b_6 = +3.63420 \cdot 10^{-3}$ $b_7 = -4.15887 \cdot 10^{-4}$ $b_8 = +1.76030 \cdot 10^{-5}$

- $SiH_2 + e \rightarrow SiH + H^+ + 2e$
  $b = 0.50$
  $b_0 = -4.53938 \cdot 10^{+1}$ $b_1 = +2.49633 \cdot 10^{+1}$ $b_2 = -1.26745 \cdot 10^{+1}$
  $b_3 = +4.71172 \cdot 10^{+0}$ $b_4 = -1.22843 \cdot 10^{+0}$ $b_5 = +2.05980 \cdot 10^{-1}$
  $b_6 = -2.03181 \cdot 10^{-2}$ $b_7 = +1.00192 \cdot 10^{-3}$ $b_8 = -1.50767 \cdot 10^{-5}$

- $SiH_2 + e \rightarrow SiH + H + e$
  $b = 0.50$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH + e \rightarrow SiH^+ + 2e$
  $b = 0.80$
  $b_0 = -2.75585 \cdot 10^{+1}$ $b_1 = +1.02656 \cdot 10^{+1}$ $b_2 = -4.04565 \cdot 10^{+0}$
  $b_3 = +9.18383 \cdot 10^{-1}$ $b_4 = -9.11069 \cdot 10^{-2}$ $b_5 = -8.53707 \cdot 10^{-3}$
  $b_6 = +3.63420 \cdot 10^{-3}$ $b_7 = -4.15887 \cdot 10^{-4}$ $b_8 = +1.76030 \cdot 10^{-5}$

- $SiH + e \rightarrow Si^+ + H + 2e$
  $b = 0.20$
  $b_0 = -2.75585 \cdot 10^{+1}$ $b_1 = +1.02656 \cdot 10^{+1}$ $b_2 = -4.04565 \cdot 10^{+0}$
  $b_3 = +9.18383 \cdot 10^{-1}$ $b_4 = -9.11069 \cdot 10^{-2}$ $b_5 = -8.53707 \cdot 10^{-3}$
  $b_6 = +3.63420 \cdot 10^{-3}$ $b_7 = -4.15887 \cdot 10^{-4}$ $b_8 = +1.76030 \cdot 10^{-5}$

- $SiH + e \rightarrow Si + H^+ + 2e$
  $b = 0.25$
  $b_0 = -4.53938 \cdot 10^{+1}$ $b_1 = +2.49633 \cdot 10^{+1}$ $b_2 = -1.26745 \cdot 10^{+1}$
  $b_3 = +4.71172 \cdot 10^{+0}$ $b_4 = -1.22843 \cdot 10^{+0}$ $b_5 = +2.05980 \cdot 10^{-1}$
  $b_6 = -2.03181 \cdot 10^{-2}$ $b_7 = +1.00192 \cdot 10^{-3}$ $b_8 = -1.50767 \cdot 10^{-5}$

- $SiH + e \rightarrow Si + H + e$
  $b = 0.25$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH_3^+ + e \rightarrow SiH_2 + H$
  $b = 1.$
  $b_0 = -1.65188 \cdot 10^{+1}$ $b_1 = -8.84597 \cdot 10^{-1}$ $b_2 = -3.23417 \cdot 10^{-2}$
  $b_3 = +5.93140 \cdot 10^{-3}$ $b_4 = -6.06568 \cdot 10^{-4}$ $b_5 = -6.80725 \cdot 10^{-6}$
  $b_6 = +1.01312 \cdot 10^{-5}$ $b_7 = -1.07506 \cdot 10^{-6}$ $b_8 = +3.73853 \cdot 10^{-8}$

- $SiH_3^+ + e \rightarrow SiH_2^+ + H$
  $b = 0.495$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH_3^+ + e \rightarrow SiH_2 + H^+ + e$
  $b = 0.75$
  $b_0 = -4.53938 \cdot 10^{+1}$ $b_1 = +2.49633 \cdot 10^{+1}$ $b_2 = -1.26745 \cdot 10^{+1}$
  $b_3 = +4.71172 \cdot 10^{+0}$ $b_4 = -1.22843 \cdot 10^{+0}$ $b_5 = +2.05980 \cdot 10^{-1}$
  $b_6 = -2.03181 \cdot 10^{-2}$ $b_7 = +1.00192 \cdot 10^{-3}$ $b_8 = -1.50767 \cdot 10^{-5}$

- $SiH_2^+ + e \rightarrow SiH + H$
  $b = 1.$
  $b_0 = -1.61538 \cdot 10^{+1}$ $b_1 = -7.58647 \cdot 10^{-1}$ $b_2 = -1.13053 \cdot 10^{-1}$
  $b_3 = -6.96563 \cdot 10^{-4}$ $b_4 = +4.52860 \cdot 10^{-3}$ $b_5 = -6.55797 \cdot 10^{-4}$
  $b_6 = -1.97544 \cdot 10^{-5}$ $b_7 = +1.05815 \cdot 10^{-5}$ $b_8 = -6.04128 \cdot 10^{-7}$

- $SiH_2^+ + e \rightarrow SiH^+ + H + e$
  $b = 0.25$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH_2^+ + e \rightarrow SiH + H^+ + e$
  $b = 0.5$
  $b_0 = -4.53938 \cdot 10^{+1}$ $b_1 = +2.49633 \cdot 10^{+1}$ $b_2 = -1.26745 \cdot 10^{+1}$
  $b_3 = +4.71172 \cdot 10^{+0}$ $b_4 = -1.22843 \cdot 10^{+0}$ $b_5 = +2.05980 \cdot 10^{-1}$
  $b_6 = -2.03181 \cdot 10^{-2}$ $b_7 = +1.00192 \cdot 10^{-3}$ $b_8 = -1.50767 \cdot 10^{-5}$

- $SiH^+ + e \rightarrow Si + H$
  $b = 1.$
  $b_0 = -1.65019 \cdot 10^{+1}$ $b_1 = -7.57799 \cdot 10^{-1}$ $b_2 = -1.13737 \cdot 10^{-1}$
  $b_3 = -1.20805 \cdot 10^{-3}$ $b_4 = +4.85270 \cdot 10^{-3}$ $b_5 = -6.27751 \cdot 10^{-4}$
  $b_6 = -5.97824 \cdot 10^{-5}$ $b_7 = +1.78869 \cdot 10^{-5}$ $b_8 = -1.01245 \cdot 10^{-6}$

- $SiH^+ + e \rightarrow Si^+ + H + e$
  $b = 0.25$
  $b_0 = -2.53277 \cdot 10^{+1}$ $b_1 = +8.42838 \cdot 10^{+0}$ $b_2 = -3.37947 \cdot 10^{+0}$
  $b_3 = +5.83943 \cdot 10^{-1}$ $b_4 = +1.12597 \cdot 10^{-1}$ $b_5 = -6.96675 \cdot 10^{-2}$
  $b_6 = +9.89709 \cdot 10^{-3}$ $b_7 = -9.64827 \cdot 10^{-5}$ $b_8 = -5.19730 \cdot 10^{-5}$

- $SiH^+ + e \rightarrow Si + H^+ + 2e$
  $b = 0.25$
  $b_0 = -4.53938 \cdot 10^{+1}$ $b_1 = +2.49633 \cdot 10^{+1}$ $b_2 = -1.26745 \cdot 10^{+1}$
  $b_3 = +4.71172 \cdot 10^{+0}$ $b_4 = -1.22843 \cdot 10^{+0}$ $b_5 = +2.05980 \cdot 10^{-1}$
  $b_6 = -2.03181 \cdot 10^{-2}$ $b_7 = +1.00192 \cdot 10^{-3}$ $b_8 = -1.50767 \cdot 10^{-5}$

- $Si + e \rightarrow Si^+ + 2e$ [6]
  $b = 1.0$
  $b_0 = -2.38605 \cdot 10^{+1}$ $b_1 = +8.70021 \cdot 10^{+0}$ $b_2 = -4.37859 \cdot 10^{+0}$
  $b_3 = +1.46033 \cdot 10^{+0}$ $b_4 = -3.35497 \cdot 10^{-1}$ $b_5 = +5.13336 \cdot 10^{-2}$
  $b_6 = -4.94238 \cdot 10^{-3}$ $b_7 = +2.68509 \cdot 10^{-4}$ $b_8 = -6.22562 \cdot 10^{-6}$

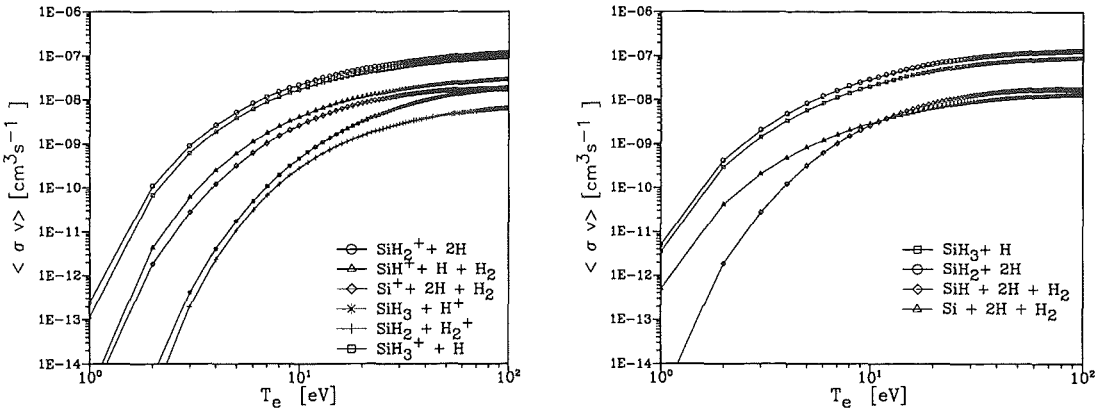
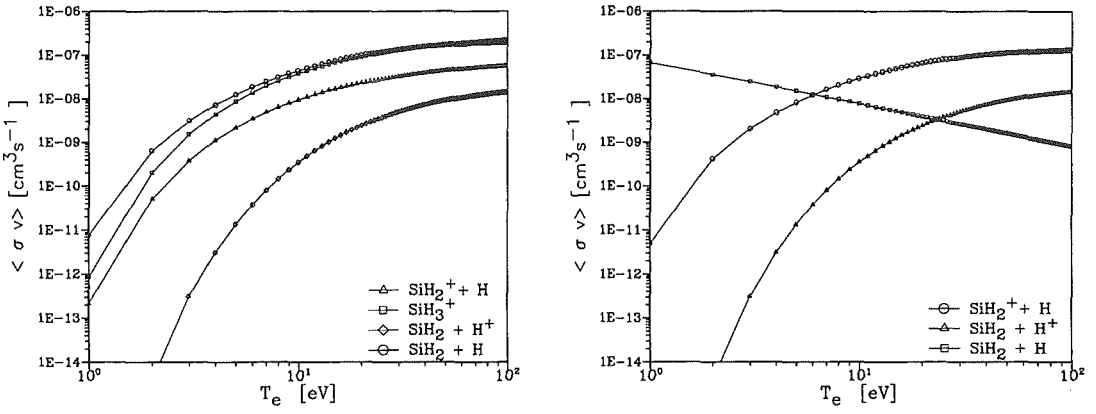


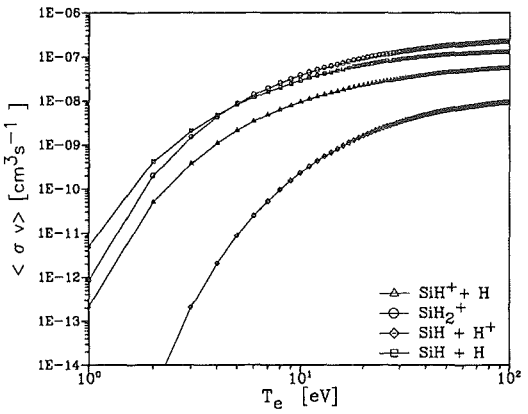
Figure A.1: Reaction rates of the SiD_4-molecule.



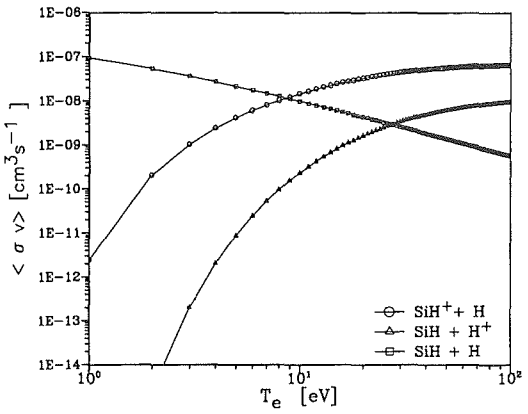
(a) Reaction rates of the SiD_3-molecule

(b) Reaction rates of the SiD_3^+-ion

Figure A.2: Reaction rates of the SiD_3-radicals.

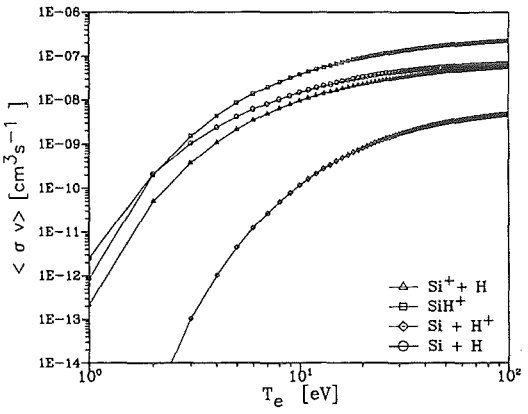


(a) Reaction rates of the SiD_2-molecule

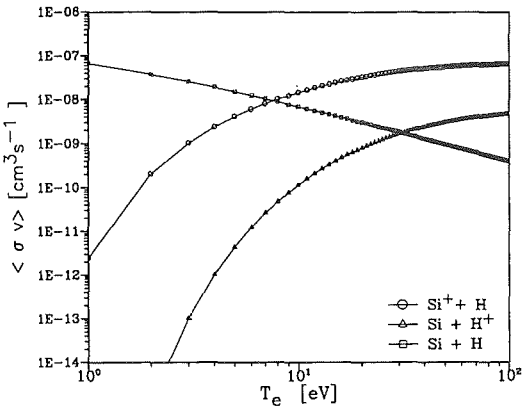


(b) Reaction rates of the SiD_2^+-ion

Figure A.3: Reaction rates of the SiD_2-radicals.



(a) Reaction rates of the SiD_1-molecule



(b) Reaction rates of the SiD_1^+-ion

Figure A.4: Reaction rates of the SiD_1-radicals.

Appendix B

Rate Coefficients for Molybdenum and Tungsten in the Energy Range of 0.5 - 200eV

B.1 Calculating the rate coefficients $< \sigma \cdot \mathbf{v} >$

As described in [6] the rate coefficients are calculated according to the following formula: For an electron temperature range $\frac{I}{10} \leq kT_e \leq 10 \cdot I$, where I is the ionization potential

$$< \sigma \cdot v > (T_e) = e^{\frac{-I}{kT_e}} \sqrt{\frac{kT_e}{I}} \sum_{n=0}^{m} a_n \left(\log_{10} \left(\frac{kT_e}{I} \right) \right)^n \tag{B.1}$$

and for the electron temperature range $kT_e \geq 10 \cdot I$

$$< \sigma \cdot v > (T_e) = \sqrt{\frac{I}{kT_e}} \left[\alpha \ln \left(\frac{kT_e}{I} \right) + \sum_{i=0}^{k} \beta_i \left(\frac{kT_e}{I} \right)^{-i} \right] \tag{B.2}$$

The rate coefficients have been calculated according to the following Lotz-formula [51]:

$$< \sigma \cdot v > = 10^{-8} \cdot 6m \left| \frac{Ry}{E} \right|^{3/2} \sqrt{\beta} e^{-\beta} f(\beta) [\frac{cm^3}{s}] \tag{B.3}$$

$$\beta = \frac{E}{T_e} \quad \text{and} \tag{B.4}$$

$$f(\beta) = -\beta e^{\beta} Ei(-\beta) \tag{B.5}$$

where for the coefficients the parameter m used, is the amount of electrons in the outer shell of the atom in the ground state:

charge state	0	1	2	3	4
Mo	1	5	4	3	2
W	2	1	4	3	2

B.2 Rate Coefficients for Molybdenum

charge	0	1	2	3	4
ionization potential $[eV]$	7.09	16.2	27.1	46.4	54.48
a_0	$+9.1341 \cdot 10^{-8}$	$+1.3278 \cdot 10^{-7}$	$+4.8647 \cdot 10^{-8}$	$+1.6335 \cdot 10^{-8}$	$+8.5993 \cdot 10^{-9}$
a_1	$-8.2145 \cdot 10^{-8}$	$-1.2023 \cdot 10^{-7}$	$-4.0111 \cdot 10^{-8}$	$-1.4407 \cdot 10^{-8}$	$-7.3307 \cdot 10^{-9}$
a_2	$-4.1762 \cdot 10^{-8}$	$-5.9924 \cdot 10^{-8}$	$-2.0261 \cdot 10^{-8}$	$-5.9594 \cdot 10^{-9}$	$-6.6144 \cdot 10^{-9}$
a_3	$+1.9845 \cdot 10^{-7}$	$+2.9179 \cdot 10^{-7}$	$+4.8329 \cdot 10^{-8}$	$+2.7174 \cdot 10^{-8}$	$+1.7474 \cdot 10^{-8}$
a_4	$-1.2112 \cdot 10^{-7}$	$-1.8257 \cdot 10^{-7}$	$+4.5822 \cdot 10^{-8}$	$-5.5024 \cdot 10^{-9}$	$+1.9344 \cdot 10^{-8}$
a_5	$+9.0759 \cdot 10^{-9}$	$+1.6345 \cdot 10^{-8}$	$-5.5194 \cdot 10^{-8}$	$-1.1885 \cdot 10^{-8}$	$-3.9772 \cdot 10^{-8}$
α	$+2.3867 \cdot 10^{-6}$	$+2.6145 \cdot 10^{-6}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$
β_0	$-7.6421 \cdot 10^{-6}$	$-7.6794 \cdot 10^{-6}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$
β_1	$+3.4467 \cdot 10^{-5}$	$+2.9060 \cdot 10^{-5}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$
β_2	$-8.122 \cdot 10^{-5}$	$-5.3634 \cdot 10^{-5}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$

B.3 Rate Coefficients for Tungsten

charge	0	1	2	3	4
ionization potential $[eV]$	8.0	15.08	25.43	39.29	53.15
a_0	$+1.5410 \cdot 10^{-7}$	$+2.9570 \cdot 10^{-8}$	$+5.335 \cdot 10^{-8}$	$+2.0919 \cdot 10^{-8}$	$+8.9193 \cdot 10^{-9}$
a_1	$-1.4091 \cdot 10^{-7}$	$-2.6783 \cdot 10^{-8}$	$-4.4638 \cdot 10^{-8}$	$-1.9055 \cdot 10^{-8}$	$-7.6145 \cdot 10^{-9}$
a_2	$-6.7428 \cdot 10^{-8}$	$-1.3382 \cdot 10^{-8}$	$-2.1863 \cdot 10^{-8}$	$-2.6973 \cdot 10^{-9}$	$-6.4909 \cdot 10^{-9}$
a_3	$+3.4182 \cdot 10^{-7}$	$+6.5236 \cdot 10^{-8}$	$+5.8908 \cdot 10^{-8}$	$+3.4011 \cdot 10^{-8}$	$+1.7524 \cdot 10^{-8}$
a_4	$-2.2023 \cdot 10^{-7}$	$-4.1061 \cdot 10^{-8}$	$+3.6855 \cdot 10^{-8}$	$-4.1255 \cdot 10^{-8}$	$+1.7849 \cdot 10^{-8}$
a_5	$+2.3284 \cdot 10^{-8}$	$+3.8345 \cdot 10^{-9}$	$-5.2445 \cdot 10^{-8}$	$+2.6448 \cdot 10^{-8}$	$-3.7134 \cdot 10^{-8}$
α	$+4.3672 \cdot 10^{-6}$	$+6.0248 \cdot 10^{-7}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$
β_0	$-1.4321 \cdot 10^{-5}$	$-1.7891 \cdot 10^{-6}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$
β_1	$+6.791 \cdot 10^{-5}$	$+6.9148 \cdot 10^{-6}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$
β_2	$-1.7063 \cdot 10^{-4}$	$-1.3160 \cdot 10^{-5}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$	$+1.0000 \cdot 10^{+0}$

Appendix C

Calculated Sputtering Yields

For all elements, for which the sputtering data have been published by Eckstein et al. [21], the sputtering yields of a maxwellian ion distribution, accelerated in the sheath of $\Phi_{\text{wall}} - \Phi_{\text{plasma edge}} = \frac{3k_B T_e}{e}$ have been calculated according eqn.(2.114) as a function of electron temperature and charge state of the incident ion. (Since the integration needs a large amount of cpu-time, it was cut off at an maximum energy of $E = 3.6keV$). The data for the most relevant elements in fusion (namely Lithium, Beryllium, Boron, Carbon, Silicon, Iron, Nickel, Copper, Molybdenum and Tungsten) are shown in this chapter.

- Lithium

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$1.847 \cdot 10^{-4}$	$1.523 \cdot 10^{-4}$	$6.958 \cdot 10^{-5}$	$3.668 \cdot 10^{-5}$	$5.452 \cdot 10^{-4}$
5.0	$1.575 \cdot 10^{-2}$	$2.245 \cdot 10^{-2}$	$2.464 \cdot 10^{-2}$	$3.395 \cdot 10^{-2}$	$6.182 \cdot 10^{-2}$
10.0	$2.554 \cdot 10^{-2}$	$3.863 \cdot 10^{-2}$	$4.671 \cdot 10^{-2}$	$7.558 \cdot 10^{-2}$	$1.011 \cdot 10^{-1}$
15.0	$2.916 \cdot 10^{-2}$	$4.521 \cdot 10^{-2}$	$5.645 \cdot 10^{-2}$	$9.888 \cdot 10^{-2}$	$1.184 \cdot 10^{-1}$
20.0	$3.052 \cdot 10^{-2}$	$4.811 \cdot 10^{-2}$	$6.105 \cdot 10^{-2}$	$1.126 \cdot 10^{-1}$	$1.269 \cdot 10^{-1}$
25.0	$3.095 \cdot 10^{-2}$	$4.925 \cdot 10^{-2}$	$6.334 \cdot 10^{-2}$	$1.215 \cdot 10^{-1}$	$1.309 \cdot 10^{-1}$
30.0	$3.086 \cdot 10^{-2}$	$4.951 \cdot 10^{-2}$	$6.420 \cdot 10^{-2}$	$1.274 \cdot 10^{-1}$	$1.330 \cdot 10^{-1}$
35.0	$3.051 \cdot 10^{-2}$	$4.931 \cdot 10^{-2}$	$6.441 \cdot 10^{-2}$	$1.312 \cdot 10^{-1}$	$1.335 \cdot 10^{-1}$
40.0	$3.006 \cdot 10^{-2}$	$4.877 \cdot 10^{-2}$	$6.405 \cdot 10^{-2}$	$1.339 \cdot 10^{-1}$	$1.332 \cdot 10^{-1}$
45.0	$2.953 \cdot 10^{-2}$	$4.810 \cdot 10^{-2}$	$6.346 \cdot 10^{-2}$	$1.355 \cdot 10^{-1}$	$1.324 \cdot 10^{-1}$
50.0	$2.898 \cdot 10^{-2}$	$4.733 \cdot 10^{-2}$	$6.268 \cdot 10^{-2}$	$1.363 \cdot 10^{-1}$	$1.314 \cdot 10^{-1}$
75.0	$2.609 \cdot 10^{-2}$	$4.320 \cdot 10^{-2}$	$5.803 \cdot 10^{-2}$	$1.359 \cdot 10^{-1}$	$1.236 \cdot 10^{-1}$
100.0	$2.365 \cdot 10^{-2}$	$3.945 \cdot 10^{-2}$	$5.339 \cdot 10^{-2}$	$1.318 \cdot 10^{-1}$	$1.153 \cdot 10^{-1}$
150.0	$1.996 \cdot 10^{-2}$	$3.368 \cdot 10^{-2}$	$4.606 \cdot 10^{-2}$	$1.218 \cdot 10^{-1}$	$1.013 \cdot 10^{-1}$
200.0	$1.737 \cdot 10^{-2}$	$2.950 \cdot 10^{-2}$	$4.059 \cdot 10^{-2}$	$1.123 \cdot 10^{-1}$	$9.044 \cdot 10^{-2}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$4.164 \cdot 10^{-8}$	$1.925 \cdot 10^{-6}$	$3.993 \cdot 10^{-5}$	$4.489 \cdot 10^{-4}$	$2.790 \cdot 10^{-3}$	$8.745 \cdot 10^{-3}$
5.0	$2.090 \cdot 10^{-2}$	$6.810 \cdot 10^{-2}$	$1.157 \cdot 10^{-1}$	$1.561 \cdot 10^{-1}$	$1.899 \cdot 10^{-1}$	$2.181 \cdot 10^{-1}$
10.0	$9.769 \cdot 10^{-2}$	$1.785 \cdot 10^{-1}$	$2.353 \cdot 10^{-1}$	$2.768 \cdot 10^{-1}$	$3.094 \cdot 10^{-1}$	$3.365 \cdot 10^{-1}$
15.0	$1.641 \cdot 10^{-1}$	$2.507 \cdot 10^{-1}$	$3.059 \cdot 10^{-1}$	$3.455 \cdot 10^{-1}$	$3.749 \cdot 10^{-1}$	$3.970 \cdot 10^{-1}$
20.0	$2.158 \cdot 10^{-1}$	$3.010 \cdot 10^{-1}$	$3.538 \cdot 10^{-1}$	$3.885 \cdot 10^{-1}$	$4.142 \cdot 10^{-1}$	$4.325 \cdot 10^{-1}$
25.0	$2.565 \cdot 10^{-1}$	$3.386 \cdot 10^{-1}$	$3.877 \cdot 10^{-1}$	$4.186 \cdot 10^{-1}$	$4.404 \cdot 10^{-1}$	$4.554 \cdot 10^{-1}$
30.0	$2.889 \cdot 10^{-1}$	$3.685 \cdot 10^{-1}$	$4.128 \cdot 10^{-1}$	$4.399 \cdot 10^{-1}$	$4.585 \cdot 10^{-1}$	$4.711 \cdot 10^{-1}$
35.0	$3.160 \cdot 10^{-1}$	$3.917 \cdot 10^{-1}$	$4.319 \cdot 10^{-1}$	$4.559 \cdot 10^{-1}$	$4.720 \cdot 10^{-1}$	$4.816 \cdot 10^{-1}$
40.0	$3.387 \cdot 10^{-1}$	$4.096 \cdot 10^{-1}$	$4.466 \cdot 10^{-1}$	$4.682 \cdot 10^{-1}$	$4.811 \cdot 10^{-1}$	$4.882 \cdot 10^{-1}$
45.0	$3.586 \cdot 10^{-1}$	$4.258 \cdot 10^{-1}$	$4.589 \cdot 10^{-1}$	$4.773 \cdot 10^{-1}$	$4.879 \cdot 10^{-1}$	$4.935 \cdot 10^{-1}$
50.0	$3.754 \cdot 10^{-1}$	$4.384 \cdot 10^{-1}$	$4.682 \cdot 10^{-1}$	$4.842 \cdot 10^{-1}$	$4.925 \cdot 10^{-1}$	$4.962 \cdot 10^{-1}$
75.0	$4.342 \cdot 10^{-1}$	$4.785 \cdot 10^{-1}$	$4.953 \cdot 10^{-1}$	$5.003 \cdot 10^{-1}$	$5.000 \cdot 10^{-1}$	$4.959 \cdot 10^{-1}$
100.0	$4.676 \cdot 10^{-1}$	$4.976 \cdot 10^{-1}$	$5.036 \cdot 10^{-1}$	$5.005 \cdot 10^{-1}$	$4.940 \cdot 10^{-1}$	$4.848 \cdot 10^{-1}$
150.0	$5.015 \cdot 10^{-1}$	$5.088 \cdot 10^{-1}$	$4.996 \cdot 10^{-1}$	$4.856 \cdot 10^{-1}$	$4.705 \cdot 10^{-1}$	$4.549 \cdot 10^{-1}$
200.0	$5.153 \cdot 10^{-1}$	$5.055 \cdot 10^{-1}$	$4.858 \cdot 10^{-1}$	$4.646 \cdot 10^{-1}$	$4.446 \cdot 10^{-1}$	$4.254 \cdot 10^{-1}$

T_e [eV]	O^+	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$1.982 \cdot 10^{-9}$	$1.260 \cdot 10^{-7}$	$3.573 \cdot 10^{-6}$	$5.707 \cdot 10^{-5}$	$5.476 \cdot 10^{-4}$	$3.106 \cdot 10^{-3}$
5.0	$1.415 \cdot 10^{-2}$	$5.540 \cdot 10^{-2}$	$1.013 \cdot 10^{-1}$	$1.497 \cdot 10^{-1}$	$1.906 \cdot 10^{-1}$	$2.261 \cdot 10^{-1}$
10.0	$8.672 \cdot 10^{-2}$	$1.765 \cdot 10^{-1}$	$2.467 \cdot 10^{-1}$	$2.972 \cdot 10^{-1}$	$3.432 \cdot 10^{-1}$	$3.786 \cdot 10^{-1}$
15.0	$1.590 \cdot 10^{-1}$	$2.651 \cdot 10^{-1}$	$3.371 \cdot 10^{-1}$	$3.901 \cdot 10^{-1}$	$4.312 \cdot 10^{-1}$	$4.630 \cdot 10^{-1}$
20.0	$2.201 \cdot 10^{-1}$	$3.279 \cdot 10^{-1}$	$4.007 \cdot 10^{-1}$	$4.507 \cdot 10^{-1}$	$4.891 \cdot 10^{-1}$	$5.178 \cdot 10^{-1}$
25.0	$2.695 \cdot 10^{-1}$	$3.796 \cdot 10^{-1}$	$4.483 \cdot 10^{-1}$	$4.954 \cdot 10^{-1}$	$5.302 \cdot 10^{-1}$	$5.567 \cdot 10^{-1}$
30.0	$3.116 \cdot 10^{-1}$	$4.196 \cdot 10^{-1}$	$4.843 \cdot 10^{-1}$	$5.285 \cdot 10^{-1}$	$5.613 \cdot 10^{-1}$	$5.846 \cdot 10^{-1}$
35.0	$3.464 \cdot 10^{-1}$	$4.521 \cdot 10^{-1}$	$5.146 \cdot 10^{-1}$	$5.555 \cdot 10^{-1}$	$5.849 \cdot 10^{-1}$	$6.058 \cdot 10^{-1}$
40.0	$3.761 \cdot 10^{-1}$	$4.797 \cdot 10^{-1}$	$5.382 \cdot 10^{-1}$	$5.765 \cdot 10^{-1}$	$6.034 \cdot 10^{-1}$	$6.222 \cdot 10^{-1}$
45.0	$4.038 \cdot 10^{-1}$	$5.025 \cdot 10^{-1}$	$5.595 \cdot 10^{-1}$	$5.942 \cdot 10^{-1}$	$6.185 \cdot 10^{-1}$	$6.350 \cdot 10^{-1}$
50.0	$4.264 \cdot 10^{-1}$	$5.230 \cdot 10^{-1}$	$5.760 \cdot 10^{-1}$	$6.084 \cdot 10^{-1}$	$6.305 \cdot 10^{-1}$	$6.451 \cdot 10^{-1}$
75.0	$5.116 \cdot 10^{-1}$	$5.918 \cdot 10^{-1}$	$6.312 \cdot 10^{-1}$	$6.528 \cdot 10^{-1}$	$6.647 \cdot 10^{-1}$	$6.703 \cdot 10^{-1}$
100.0	$5.651 \cdot 10^{-1}$	$6.307 \cdot 10^{-1}$	$6.594 \cdot 10^{-1}$	$6.715 \cdot 10^{-1}$	$6.760 \cdot 10^{-1}$	$6.748 \cdot 10^{-1}$
150.0	$6.287 \cdot 10^{-1}$	$6.703 \cdot 10^{-1}$	$6.803 \cdot 10^{-1}$	$6.786 \cdot 10^{-1}$	$6.719 \cdot 10^{-1}$	$6.614 \cdot 10^{-1}$
200.0	$6.628 \cdot 10^{-1}$	$6.852 \cdot 10^{-1}$	$6.816 \cdot 10^{-1}$	$6.700 \cdot 10^{-1}$	$6.551 \cdot 10^{-1}$	$6.385 \cdot 10^{-1}$

T_e [eV]	Li^+	Li^{2+}	Li^{3+}
1.0	$2.719 \cdot 10^{-6}$	$6.983 \cdot 10^{-5}$	$7.750 \cdot 10^{-4}$
5.0	$3.053 \cdot 10^{-2}$	$6.946 \cdot 10^{-2}$	$1.030 \cdot 10^{-1}$
10.0	$9.162 \cdot 10^{-2}$	$1.401 \cdot 10^{-1}$	$1.685 \cdot 10^{-1}$
15.0	$1.336 \cdot 10^{-1}$	$1.768 \cdot 10^{-1}$	$2.004 \cdot 10^{-1}$
20.0	$1.616 \cdot 10^{-1}$	$1.996 \cdot 10^{-1}$	$2.174 \cdot 10^{-1}$
25.0	$1.817 \cdot 10^{-1}$	$2.139 \cdot 10^{-1}$	$2.275 \cdot 10^{-1}$
30.0	$1.963 \cdot 10^{-1}$	$2.233 \cdot 10^{-1}$	$2.334 \cdot 10^{-1}$
35.0	$2.080 \cdot 10^{-1}$	$2.300 \cdot 10^{-1}$	$2.366 \cdot 10^{-1}$
40.0	$2.166 \cdot 10^{-1}$	$2.344 \cdot 10^{-1}$	$2.384 \cdot 10^{-1}$
45.0	$2.235 \cdot 10^{-1}$	$2.375 \cdot 10^{-1}$	$2.388 \cdot 10^{-1}$
50.0	$2.289 \cdot 10^{-1}$	$2.396 \cdot 10^{-1}$	$2.383 \cdot 10^{-1}$
75.0	$2.430 \cdot 10^{-1}$	$2.400 \cdot 10^{-1}$	$2.306 \cdot 10^{-1}$
100.0	$2.469 \cdot 10^{-1}$	$2.343 \cdot 10^{-1}$	$2.198 \cdot 10^{-1}$
150.0	$2.425 \cdot 10^{-1}$	$2.186 \cdot 10^{-1}$	$1.985 \cdot 10^{-1}$
200.0	$2.338 \cdot 10^{-1}$	$2.032 \cdot 10^{-1}$	$1.807 \cdot 10^{-1}$

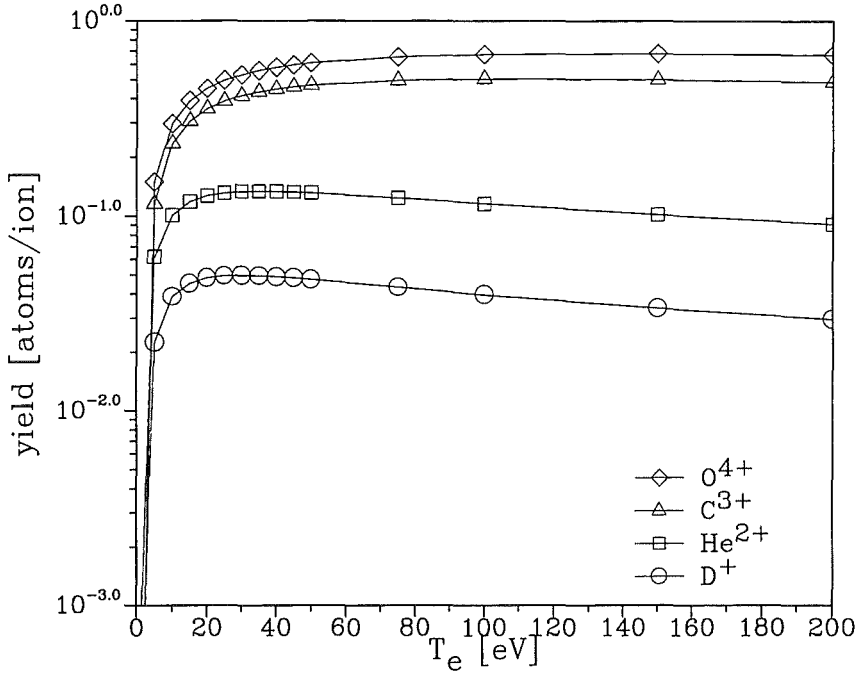


Figure C.1: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

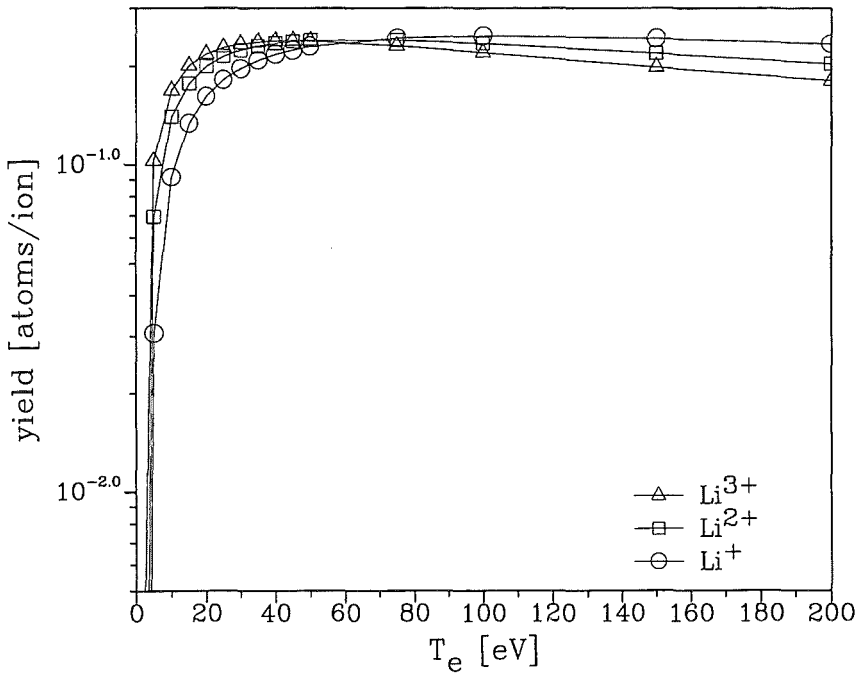


Figure C.2: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Beryllium

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$1.454 \cdot 10^{-7}$	$1.976 \cdot 10^{-7}$	$4.104 \cdot 10^{-8}$	$9.171 \cdot 10^{-9}$	$4.394 \cdot 10^{-7}$
5.0	$3.785 \cdot 10^{-3}$	$5.714 \cdot 10^{-3}$	$5.442 \cdot 10^{-3}$	$6.760 \cdot 10^{-3}$	$2.112 \cdot 10^{-2}$
10.0	$1.122 \cdot 10^{-2}$	$1.730 \cdot 10^{-2}$	$1.939 \cdot 10^{-2}$	$3.020 \cdot 10^{-2}$	$5.061 \cdot 10^{-2}$
15.0	$1.544 \cdot 10^{-2}$	$2.398 \cdot 10^{-2}$	$2.817 \cdot 10^{-2}$	$4.820 \cdot 10^{-2}$	$6.653 \cdot 10^{-2}$
20.0	$1.784 \cdot 10^{-2}$	$2.794 \cdot 10^{-2}$	$3.361 \cdot 10^{-2}$	$6.061 \cdot 10^{-2}$	$7.630 \cdot 10^{-2}$
25.0	$1.925 \cdot 10^{-2}$	$3.025 \cdot 10^{-2}$	$3.686 \cdot 10^{-2}$	$6.946 \cdot 10^{-2}$	$8.230 \cdot 10^{-2}$
30.0	$2.003 \cdot 10^{-2}$	$3.161 \cdot 10^{-2}$	$3.893 \cdot 10^{-2}$	$7.575 \cdot 10^{-2}$	$8.582 \cdot 10^{-2}$
35.0	$2.043 \cdot 10^{-2}$	$3.238 \cdot 10^{-2}$	$4.013 \cdot 10^{-2}$	$8.072 \cdot 10^{-2}$	$8.835 \cdot 10^{-2}$
40.0	$2.062 \cdot 10^{-2}$	$3.284 \cdot 10^{-2}$	$4.099 \cdot 10^{-2}$	$8.422 \cdot 10^{-2}$	$8.963 \cdot 10^{-2}$
45.0	$2.069 \cdot 10^{-2}$	$3.295 \cdot 10^{-2}$	$4.139 \cdot 10^{-2}$	$8.697 \cdot 10^{-2}$	$9.037 \cdot 10^{-2}$
50.0	$2.064 \cdot 10^{-2}$	$3.295 \cdot 10^{-2}$	$4.160 \cdot 10^{-2}$	$8.919 \cdot 10^{-2}$	$9.087 \cdot 10^{-2}$
75.0	$1.972 \cdot 10^{-2}$	$3.175 \cdot 10^{-2}$	$4.064 \cdot 10^{-2}$	$9.382 \cdot 10^{-2}$	$8.911 \cdot 10^{-2}$
100.0	$1.848 \cdot 10^{-2}$	$2.994 \cdot 10^{-2}$	$3.869 \cdot 10^{-2}$	$9.410 \cdot 10^{-2}$	$8.540 \cdot 10^{-2}$
150.0	$1.625 \cdot 10^{-2}$	$2.653 \cdot 10^{-2}$	$3.461 \cdot 10^{-2}$	$9.038 \cdot 10^{-2}$	$7.741 \cdot 10^{-2}$
200.0	$1.447 \cdot 10^{-2}$	$2.376 \cdot 10^{-2}$	$3.120 \cdot 10^{-2}$	$8.544 \cdot 10^{-2}$	$7.054 \cdot 10^{-2}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$1.190 \cdot 10^{-14}$	$1.835 \cdot 10^{-12}$	$1.174 \cdot 10^{-10}$	$4.316 \cdot 10^{-9}$	$1.035 \cdot 10^{-7}$	$1.712 \cdot 10^{-6}$
5.0	$1.805 \cdot 10^{-3}$	$1.162 \cdot 10^{-2}$	$3.155 \cdot 10^{-2}$	$5.678 \cdot 10^{-2}$	$7.861 \cdot 10^{-2}$	$9.747 \cdot 10^{-2}$
10.0	$2.514 \cdot 10^{-2}$	$7.082 \cdot 10^{-2}$	$1.110 \cdot 10^{-1}$	$1.434 \cdot 10^{-1}$	$1.690 \cdot 10^{-1}$	$1.912 \cdot 10^{-1}$
15.0	$6.168 \cdot 10^{-2}$	$1.225 \cdot 10^{-1}$	$1.674 \cdot 10^{-1}$	$1.987 \cdot 10^{-1}$	$2.250 \cdot 10^{-1}$	$2.447 \cdot 10^{-1}$
20.0	$9.631 \cdot 10^{-2}$	$1.626 \cdot 10^{-1}$	$2.056 \cdot 10^{-1}$	$2.368 \cdot 10^{-1}$	$2.603 \cdot 10^{-1}$	$2.778 \cdot 10^{-1}$
25.0	$1.259 \cdot 10^{-1}$	$1.927 \cdot 10^{-1}$	$2.355 \cdot 10^{-1}$	$2.642 \cdot 10^{-1}$	$2.846 \cdot 10^{-1}$	$2.996 \cdot 10^{-1}$
30.0	$1.513 \cdot 10^{-1}$	$2.178 \cdot 10^{-1}$	$2.578 \cdot 10^{-1}$	$2.844 \cdot 10^{-1}$	$3.022 \cdot 10^{-1}$	$3.156 \cdot 10^{-1}$
35.0	$1.732 \cdot 10^{-1}$	$2.377 \cdot 10^{-1}$	$2.756 \cdot 10^{-1}$	$2.997 \cdot 10^{-1}$	$3.159 \cdot 10^{-1}$	$3.269 \cdot 10^{-1}$
40.0	$1.921 \cdot 10^{-1}$	$2.549 \cdot 10^{-1}$	$2.903 \cdot 10^{-1}$	$3.118 \cdot 10^{-1}$	$3.259 \cdot 10^{-1}$	$3.355 \cdot 10^{-1}$
45.0	$2.083 \cdot 10^{-1}$	$2.690 \cdot 10^{-1}$	$3.016 \cdot 10^{-1}$	$3.214 \cdot 10^{-1}$	$3.339 \cdot 10^{-1}$	$3.418 \cdot 10^{-1}$
50.0	$2.226 \cdot 10^{-1}$	$2.815 \cdot 10^{-1}$	$3.111 \cdot 10^{-1}$	$3.290 \cdot 10^{-1}$	$3.399 \cdot 10^{-1}$	$3.465 \cdot 10^{-1}$
75.0	$2.750 \cdot 10^{-1}$	$3.208 \cdot 10^{-1}$	$3.413 \cdot 10^{-1}$	$3.509 \cdot 10^{-1}$	$3.551 \cdot 10^{-1}$	$3.554 \cdot 10^{-1}$
100.0	$3.075 \cdot 10^{-1}$	$3.424 \cdot 10^{-1}$	$3.547 \cdot 10^{-1}$	$3.580 \cdot 10^{-1}$	$3.570 \cdot 10^{-1}$	$3.536 \cdot 10^{-1}$
150.0	$3.432 \cdot 10^{-1}$	$3.611 \cdot 10^{-1}$	$3.608 \cdot 10^{-1}$	$3.556 \cdot 10^{-1}$	$3.480 \cdot 10^{-1}$	$3.391 \cdot 10^{-1}$
200.0	$3.613 \cdot 10^{-1}$	$3.657 \cdot 10^{-1}$	$3.572 \cdot 10^{-1}$	$3.457 \cdot 10^{-1}$	$3.337 \cdot 10^{-1}$	$3.216 \cdot 10^{-1}$

T_e [eV]	O^{+}	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$3.616 \cdot 10^{-17}$	$7.964 \cdot 10^{-15}$	$6.951 \cdot 10^{-13}$	$3.442 \cdot 10^{-11}$	$1.116 \cdot 10^{-9}$	$2.542 \cdot 10^{-8}$
5.0	$8.861 \cdot 10^{-4}$	$7.399 \cdot 10^{-3}$	$2.183 \cdot 10^{-2}$	$4.913 \cdot 10^{-2}$	$6.890 \cdot 10^{-2}$	$9.750 \cdot 10^{-2}$
10.0	$1.887 \cdot 10^{-2}$	$6.409 \cdot 10^{-2}$	$1.104 \cdot 10^{-1}$	$1.497 \cdot 10^{-1}$	$1.823 \cdot 10^{-1}$	$2.094 \cdot 10^{-1}$
15.0	$5.406 \cdot 10^{-2}$	$1.232 \cdot 10^{-1}$	$1.779 \cdot 10^{-1}$	$2.181 \cdot 10^{-1}$	$2.549 \cdot 10^{-1}$	$2.813 \cdot 10^{-1}$
20.0	$9.204 \cdot 10^{-2}$	$1.714 \cdot 10^{-1}$	$2.266 \cdot 10^{-1}$	$2.704 \cdot 10^{-1}$	$3.040 \cdot 10^{-1}$	$3.297 \cdot 10^{-1}$
25.0	$1.260 \cdot 10^{-1}$	$2.104 \cdot 10^{-1}$	$2.683 \cdot 10^{-1}$	$3.092 \cdot 10^{-1}$	$3.394 \cdot 10^{-1}$	$3.641 \cdot 10^{-1}$
30.0	$1.566 \cdot 10^{-1}$	$2.420 \cdot 10^{-1}$	$2.988 \cdot 10^{-1}$	$3.385 \cdot 10^{-1}$	$3.679 \cdot 10^{-1}$	$3.892 \cdot 10^{-1}$
35.0	$1.836 \cdot 10^{-1}$	$2.701 \cdot 10^{-1}$	$3.247 \cdot 10^{-1}$	$3.621 \cdot 10^{-1}$	$3.893 \cdot 10^{-1}$	$4.096 \cdot 10^{-1}$
40.0	$2.072 \cdot 10^{-1}$	$2.940 \cdot 10^{-1}$	$3.465 \cdot 10^{-1}$	$3.819 \cdot 10^{-1}$	$4.066 \cdot 10^{-1}$	$4.254 \cdot 10^{-1}$
45.0	$2.284 \cdot 10^{-1}$	$3.137 \cdot 10^{-1}$	$3.637 \cdot 10^{-1}$	$3.973 \cdot 10^{-1}$	$4.214 \cdot 10^{-1}$	$4.377 \cdot 10^{-1}$
50.0	$2.475 \cdot 10^{-1}$	$3.323 \cdot 10^{-1}$	$3.806 \cdot 10^{-1}$	$4.111 \cdot 10^{-1}$	$4.332 \cdot 10^{-1}$	$4.483 \cdot 10^{-1}$
75.0	$3.195 \cdot 10^{-1}$	$3.942 \cdot 10^{-1}$	$4.326 \cdot 10^{-1}$	$4.553 \cdot 10^{-1}$	$4.698 \cdot 10^{-1}$	$4.783 \cdot 10^{-1}$
100.0	$3.674 \cdot 10^{-1}$	$4.312 \cdot 10^{-1}$	$4.615 \cdot 10^{-1}$	$4.774 \cdot 10^{-1}$	$4.861 \cdot 10^{-1}$	$4.892 \cdot 10^{-1}$
150.0	$4.265 \cdot 10^{-1}$	$4.722 \cdot 10^{-1}$	$4.889 \cdot 10^{-1}$	$4.938 \cdot 10^{-1}$	$4.936 \cdot 10^{-1}$	$4.894 \cdot 10^{-1}$
200.0	$4.607 \cdot 10^{-1}$	$4.917 \cdot 10^{-1}$	$4.976 \cdot 10^{-1}$	$4.946 \cdot 10^{-1}$	$4.877 \cdot 10^{-1}$	$4.786 \cdot 10^{-1}$

T_e [eV]	Be^{+}	Be^{2+}	Be^{3+}	Be^{4+}
1.0	$1.336 \cdot 10^{-12}$	$1.468 \cdot 10^{-10}$	$6.955 \cdot 10^{-9}$	$1.900 \cdot 10^{-7}$
5.0	$3.001 \cdot 10^{-3}$	$1.516 \cdot 10^{-2}$	$3.675 \cdot 10^{-2}$	$5.249 \cdot 10^{-2}$
10.0	$2.889 \cdot 10^{-2}$	$6.661 \cdot 10^{-2}$	$9.970 \cdot 10^{-2}$	$1.226 \cdot 10^{-1}$
15.0	$6.158 \cdot 10^{-2}$	$1.083 \cdot 10^{-1}$	$1.383 \cdot 10^{-1}$	$1.588 \cdot 10^{-1}$
20.0	$8.827 \cdot 10^{-2}$	$1.361 \cdot 10^{-1}$	$1.635 \cdot 10^{-1}$	$1.829 \cdot 10^{-1}$
25.0	$1.113 \cdot 10^{-1}$	$1.569 \cdot 10^{-1}$	$1.824 \cdot 10^{-1}$	$1.976 \cdot 10^{-1}$
30.0	$1.298 \cdot 10^{-1}$	$1.719 \cdot 10^{-1}$	$1.950 \cdot 10^{-1}$	$2.084 \cdot 10^{-1}$
35.0	$1.446 \cdot 10^{-1}$	$1.849 \cdot 10^{-1}$	$2.050 \cdot 10^{-1}$	$2.158 \cdot 10^{-1}$
40.0	$1.575 \cdot 10^{-1}$	$1.947 \cdot 10^{-1}$	$2.120 \cdot 10^{-1}$	$2.213 \cdot 10^{-1}$
45.0	$1.680 \cdot 10^{-1}$	$2.022 \cdot 10^{-1}$	$2.179 \cdot 10^{-1}$	$2.245 \cdot 10^{-1}$
50.0	$1.774 \cdot 10^{-1}$	$2.085 \cdot 10^{-1}$	$2.220 \cdot 10^{-1}$	$2.273 \cdot 10^{-1}$
75.0	$2.082 \cdot 10^{-1}$	$2.274 \cdot 10^{-1}$	$2.319 \cdot 10^{-1}$	$2.308 \cdot 10^{-1}$
100.0	$2.245 \cdot 10^{-1}$	$2.341 \cdot 10^{-1}$	$2.324 \cdot 10^{-1}$	$2.271 \cdot 10^{-1}$
150.0	$2.393 \cdot 10^{-1}$	$2.348 \cdot 10^{-1}$	$2.248 \cdot 10^{-1}$	$2.141 \cdot 10^{-1}$
200.0	$2.430 \cdot 10^{-1}$	$2.292 \cdot 10^{-1}$	$2.142 \cdot 10^{-1}$	$2.004 \cdot 10^{-1}$

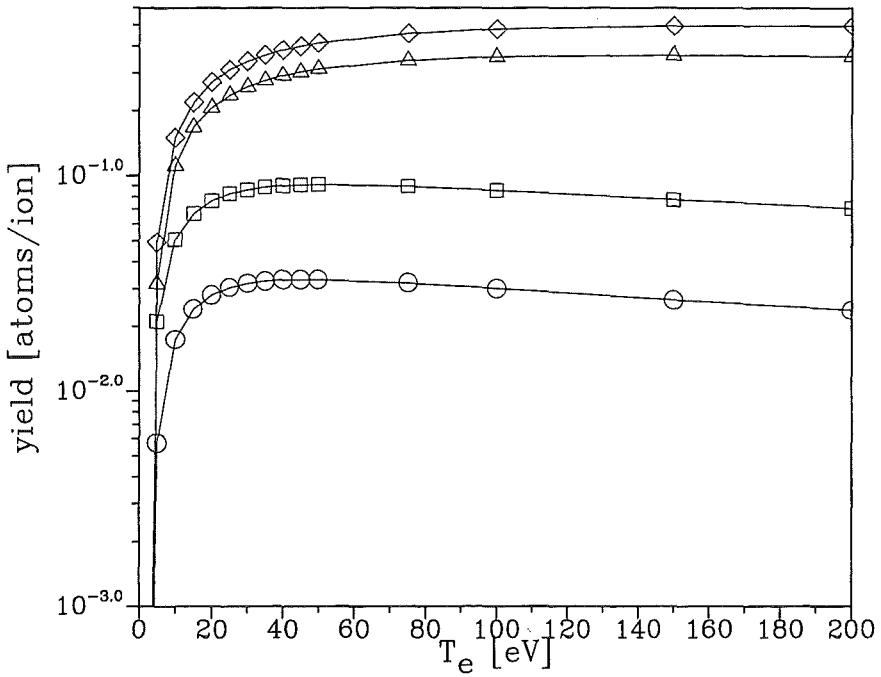


Figure C.3: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

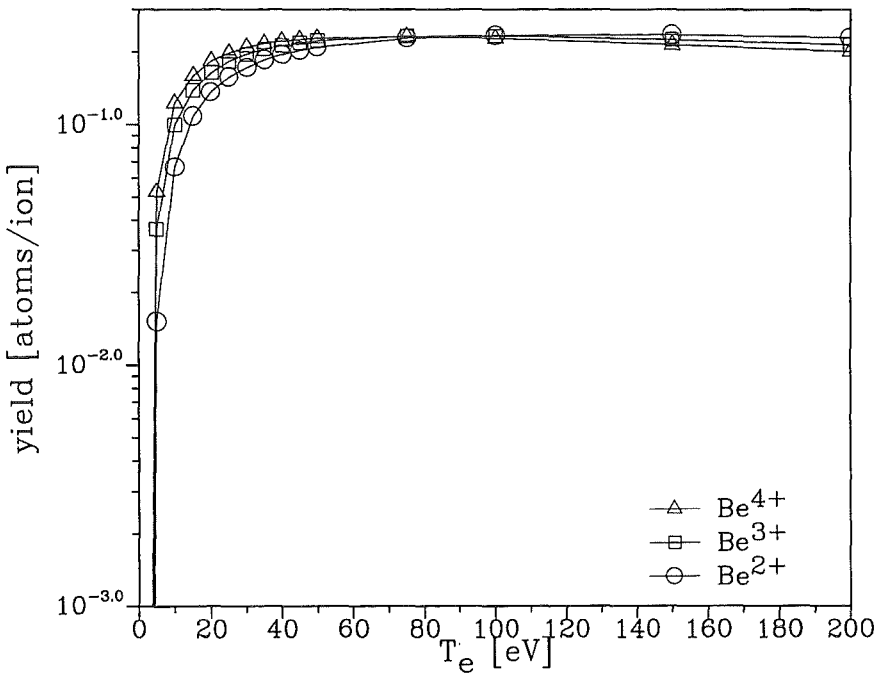


Figure C.4: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Boron

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$6.626 \cdot 10^{-13}$	$7.188 \cdot 10^{-12}$	$9.324 \cdot 10^{-13}$	$7.524 \cdot 10^{-14}$	$9.292 \cdot 10^{-12}$
5.0	$4.870 \cdot 10^{-4}$	$9.949 \cdot 10^{-4}$	$9.670 \cdot 10^{-4}$	$1.005 \cdot 10^{-3}$	$5.203 \cdot 10^{-3}$
10.0	$3.757 \cdot 10^{-3}$	$6.723 \cdot 10^{-3}$	$7.811 \cdot 10^{-3}$	$1.061 \cdot 10^{-2}$	$2.437 \cdot 10^{-2}$
15.0	$7.062 \cdot 10^{-3}$	$1.213 \cdot 10^{-2}$	$1.501 \cdot 10^{-2}$	$2.297 \cdot 10^{-2}$	$3.784 \cdot 10^{-2}$
20.0	$9.369 \cdot 10^{-3}$	$1.585 \cdot 10^{-2}$	$2.015 \cdot 10^{-2}$	$3.292 \cdot 10^{-2}$	$4.737 \cdot 10^{-2}$
25.0	$1.100 \cdot 10^{-2}$	$1.829 \cdot 10^{-2}$	$2.388 \cdot 10^{-2}$	$4.082 \cdot 10^{-2}$	$5.354 \cdot 10^{-2}$
30.0	$1.212 \cdot 10^{-2}$	$2.017 \cdot 10^{-2}$	$2.648 \cdot 10^{-2}$	$4.668 \cdot 10^{-2}$	$5.781 \cdot 10^{-2}$
35.0	$1.286 \cdot 10^{-2}$	$2.135 \cdot 10^{-2}$	$2.825 \cdot 10^{-2}$	$5.170 \cdot 10^{-2}$	$6.078 \cdot 10^{-2}$
40.0	$1.345 \cdot 10^{-2}$	$2.219 \cdot 10^{-2}$	$2.972 \cdot 10^{-2}$	$5.551 \cdot 10^{-2}$	$6.302 \cdot 10^{-2}$
45.0	$1.383 \cdot 10^{-2}$	$2.286 \cdot 10^{-2}$	$3.066 \cdot 10^{-2}$	$5.872 \cdot 10^{-2}$	$6.455 \cdot 10^{-2}$
50.0	$1.405 \cdot 10^{-2}$	$2.327 \cdot 10^{-2}$	$3.132 \cdot 10^{-2}$	$6.110 \cdot 10^{-2}$	$6.564 \cdot 10^{-2}$
75.0	$1.437 \cdot 10^{-2}$	$2.374 \cdot 10^{-2}$	$3.248 \cdot 10^{-2}$	$6.822 \cdot 10^{-2}$	$6.721 \cdot 10^{-2}$
100.0	$1.398 \cdot 10^{-2}$	$2.317 \cdot 10^{-2}$	$3.199 \cdot 10^{-2}$	$7.074 \cdot 10^{-2}$	$6.625 \cdot 10^{-2}$
150.0	$1.284 \cdot 10^{-2}$	$2.131 \cdot 10^{-2}$	$2.974 \cdot 10^{-2}$	$7.061 \cdot 10^{-2}$	$6.190 \cdot 10^{-2}$
200.0	$1.174 \cdot 10^{-2}$	$1.952 \cdot 10^{-2}$	$2.741 \cdot 10^{-2}$	$6.848 \cdot 10^{-2}$	$5.754 \cdot 10^{-2}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$2.731 \cdot 10^{-23}$	$1.202 \cdot 10^{-20}$	$1.891 \cdot 10^{-18}$	$1.624 \cdot 10^{-16}$	$9.040 \cdot 10^{-15}$	$3.560 \cdot 10^{-13}$
5.0	$9.025 \cdot 10^{-5}$	$1.249 \cdot 10^{-3}$	$6.431 \cdot 10^{-3}$	$1.325 \cdot 10^{-2}$	$2.886 \cdot 10^{-2}$	$4.495 \cdot 10^{-2}$
10.0	$5.369 \cdot 10^{-3}$	$2.361 \cdot 10^{-2}$	$5.150 \cdot 10^{-2}$	$7.636 \cdot 10^{-2}$	$9.289 \cdot 10^{-2}$	$1.146 \cdot 10^{-1}$
15.0	$2.097 \cdot 10^{-2}$	$5.965 \cdot 10^{-2}$	$9.419 \cdot 10^{-2}$	$1.212 \cdot 10^{-1}$	$1.429 \cdot 10^{-1}$	$1.602 \cdot 10^{-1}$
20.0	$4.165 \cdot 10^{-2}$	$9.065 \cdot 10^{-2}$	$1.271 \cdot 10^{-1}$	$1.533 \cdot 10^{-1}$	$1.750 \cdot 10^{-1}$	$1.909 \cdot 10^{-1}$
25.0	$6.304 \cdot 10^{-2}$	$1.144 \cdot 10^{-1}$	$1.529 \cdot 10^{-1}$	$1.787 \cdot 10^{-1}$	$1.983 \cdot 10^{-1}$	$2.123 \cdot 10^{-1}$
30.0	$8.232 \cdot 10^{-2}$	$1.378 \cdot 10^{-1}$	$1.731 \cdot 10^{-1}$	$1.975 \cdot 10^{-1}$	$2.149 \cdot 10^{-1}$	$2.286 \cdot 10^{-1}$
35.0	$9.930 \cdot 10^{-2}$	$1.559 \cdot 10^{-1}$	$1.893 \cdot 10^{-1}$	$2.128 \cdot 10^{-1}$	$2.288 \cdot 10^{-1}$	$2.407 \cdot 10^{-1}$
40.0	$1.157 \cdot 10^{-1}$	$1.704 \cdot 10^{-1}$	$2.034 \cdot 10^{-1}$	$2.251 \cdot 10^{-1}$	$2.396 \cdot 10^{-1}$	$2.500 \cdot 10^{-1}$
45.0	$1.301 \cdot 10^{-1}$	$1.840 \cdot 10^{-1}$	$2.151 \cdot 10^{-1}$	$2.345 \cdot 10^{-1}$	$2.482 \cdot 10^{-1}$	$2.572 \cdot 10^{-1}$
50.0	$1.418 \cdot 10^{-1}$	$1.955 \cdot 10^{-1}$	$2.242 \cdot 10^{-1}$	$2.428 \cdot 10^{-1}$	$2.547 \cdot 10^{-1}$	$2.629 \cdot 10^{-1}$
75.0	$1.907 \cdot 10^{-1}$	$2.345 \cdot 10^{-1}$	$2.566 \cdot 10^{-1}$	$2.681 \cdot 10^{-1}$	$2.744 \cdot 10^{-1}$	$2.773 \cdot 10^{-1}$
100.0	$2.219 \cdot 10^{-1}$	$2.574 \cdot 10^{-1}$	$2.727 \cdot 10^{-1}$	$2.788 \cdot 10^{-1}$	$2.811 \cdot 10^{-1}$	$2.806 \cdot 10^{-1}$
150.0	$2.591 \cdot 10^{-1}$	$2.806 \cdot 10^{-1}$	$2.850 \cdot 10^{-1}$	$2.839 \cdot 10^{-1}$	$2.801 \cdot 10^{-1}$	$2.750 \cdot 10^{-1}$
200.0	$2.797 \cdot 10^{-1}$	$2.895 \cdot 10^{-1}$	$2.867 \cdot 10^{-1}$	$2.804 \cdot 10^{-1}$	$2.728 \cdot 10^{-1}$	$2.645 \cdot 10^{-1}$

T_e [eV]	O^{+}	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$2.582 \cdot 10^{-27}$	$1.719 \cdot 10^{-24}$	$3.804 \cdot 10^{-22}$	$4.456 \cdot 10^{-20}$	$3.337 \cdot 10^{-18}$	$1.759 \cdot 10^{-16}$
5.0	$2.490 \cdot 10^{-5}$	$4.672 \cdot 10^{-4}$	$3.508 \cdot 10^{-3}$	$1.057 \cdot 10^{-2}$	$1.710 \cdot 10^{-2}$	$3.595 \cdot 10^{-2}$
10.0	$3.258 \cdot 10^{-3}$	$1.783 \cdot 10^{-2}$	$4.557 \cdot 10^{-2}$	$6.811 \cdot 10^{-2}$	$9.833 \cdot 10^{-2}$	$1.189 \cdot 10^{-1}$
15.0	$1.574 \cdot 10^{-2}$	$5.438 \cdot 10^{-2}$	$9.356 \cdot 10^{-2}$	$1.260 \cdot 10^{-1}$	$1.526 \cdot 10^{-1}$	$1.773 \cdot 10^{-1}$
20.0	$3.526 \cdot 10^{-2}$	$8.749 \cdot 10^{-2}$	$1.329 \cdot 10^{-1}$	$1.704 \cdot 10^{-1}$	$1.983 \cdot 10^{-1}$	$2.210 \cdot 10^{-1}$
25.0	$5.714 \cdot 10^{-2}$	$1.202 \cdot 10^{-1}$	$1.650 \cdot 10^{-1}$	$2.029 \cdot 10^{-1}$	$2.291 \cdot 10^{-1}$	$2.523 \cdot 10^{-1}$
30.0	$7.924 \cdot 10^{-2}$	$1.460 \cdot 10^{-1}$	$1.936 \cdot 10^{-1}$	$2.292 \cdot 10^{-1}$	$2.558 \cdot 10^{-1}$	$2.765 \cdot 10^{-1}$
35.0	$9.958 \cdot 10^{-2}$	$1.706 \cdot 10^{-1}$	$2.174 \cdot 10^{-1}$	$2.506 \cdot 10^{-1}$	$2.756 \cdot 10^{-1}$	$2.951 \cdot 10^{-1}$
40.0	$1.176 \cdot 10^{-1}$	$1.906 \cdot 10^{-1}$	$2.367 \cdot 10^{-1}$	$2.687 \cdot 10^{-1}$	$2.929 \cdot 10^{-1}$	$3.106 \cdot 10^{-1}$
45.0	$1.354 \cdot 10^{-1}$	$2.075 \cdot 10^{-1}$	$2.536 \cdot 10^{-1}$	$2.843 \cdot 10^{-1}$	$3.064 \cdot 10^{-1}$	$3.234 \cdot 10^{-1}$
50.0	$1.515 \cdot 10^{-1}$	$2.240 \cdot 10^{-1}$	$2.678 \cdot 10^{-1}$	$2.973 \cdot 10^{-1}$	$3.184 \cdot 10^{-1}$	$3.339 \cdot 10^{-1}$
75.0	$2.133 \cdot 10^{-1}$	$2.812 \cdot 10^{-1}$	$3.178 \cdot 10^{-1}$	$3.413 \cdot 10^{-1}$	$3.558 \cdot 10^{-1}$	$3.663 \cdot 10^{-1}$
100.0	$2.576 \cdot 10^{-1}$	$3.174 \cdot 10^{-1}$	$3.480 \cdot 10^{-1}$	$3.648 \cdot 10^{-1}$	$3.754 \cdot 10^{-1}$	$3.810 \cdot 10^{-1}$
150.0	$3.139 \cdot 10^{-1}$	$3.598 \cdot 10^{-1}$	$3.785 \cdot 10^{-1}$	$3.866 \cdot 10^{-1}$	$3.894 \cdot 10^{-1}$	$3.891 \cdot 10^{-1}$
200.0	$3.486 \cdot 10^{-1}$	$3.816 \cdot 10^{-1}$	$3.917 \cdot 10^{-1}$	$3.931 \cdot 10^{-1}$	$3.906 \cdot 10^{-1}$	$3.855 \cdot 10^{-1}$

T_e [eV]	B^{+}	B^{2+}	B^{3+}	B^{4+}	B^{5+}
1.0	$5.105 \cdot 10^{-22}$	$1.945 \cdot 10^{-19}$	$2.715 \cdot 10^{-17}$	$2.089 \cdot 10^{-15}$	$1.046 \cdot 10^{-13}$
5.0	$1.297 \cdot 10^{-4}$	$1.604 \cdot 10^{-3}$	$7.195 \cdot 10^{-3}$	$1.427 \cdot 10^{-2}$	$3.037 \cdot 10^{-2}$
10.0	$6.039 \cdot 10^{-3}$	$2.457 \cdot 10^{-2}$	$5.000 \cdot 10^{-2}$	$7.254 \cdot 10^{-2}$	$9.095 \cdot 10^{-2}$
15.0	$2.189 \cdot 10^{-2}$	$5.779 \cdot 10^{-2}$	$8.741 \cdot 10^{-2}$	$1.102 \cdot 10^{-1}$	$1.290 \cdot 10^{-1}$
20.0	$4.159 \cdot 10^{-2}$	$8.564 \cdot 10^{-2}$	$1.158 \cdot 10^{-1}$	$1.400 \cdot 10^{-1}$	$1.562 \cdot 10^{-1}$
25.0	$6.130 \cdot 10^{-2}$	$1.078 \cdot 10^{-1}$	$1.376 \cdot 10^{-1}$	$1.595 \cdot 10^{-1}$	$1.754 \cdot 10^{-1}$
30.0	$7.869 \cdot 10^{-2}$	$1.256 \cdot 10^{-1}$	$1.553 \cdot 10^{-1}$	$1.755 \cdot 10^{-1}$	$1.890 \cdot 10^{-1}$
35.0	$9.396 \cdot 10^{-2}$	$1.413 \cdot 10^{-1}$	$1.692 \cdot 10^{-1}$	$1.869 \cdot 10^{-1}$	$1.995 \cdot 10^{-1}$
40.0	$1.082 \cdot 10^{-1}$	$1.544 \cdot 10^{-1}$	$1.804 \cdot 10^{-1}$	$1.965 \cdot 10^{-1}$	$2.072 \cdot 10^{-1}$
45.0	$1.202 \cdot 10^{-1}$	$1.646 \cdot 10^{-1}$	$1.892 \cdot 10^{-1}$	$2.038 \cdot 10^{-1}$	$2.131 \cdot 10^{-1}$
50.0	$1.311 \cdot 10^{-1}$	$1.739 \cdot 10^{-1}$	$1.966 \cdot 10^{-1}$	$2.098 \cdot 10^{-1}$	$2.180 \cdot 10^{-1}$
75.0	$1.709 \cdot 10^{-1}$	$2.049 \cdot 10^{-1}$	$2.197 \cdot 10^{-1}$	$2.267 \cdot 10^{-1}$	$2.298 \cdot 10^{-1}$
100.0	$1.961 \cdot 10^{-1}$	$2.216 \cdot 10^{-1}$	$2.302 \cdot 10^{-1}$	$2.325 \cdot 10^{-1}$	$2.318 \cdot 10^{-1}$
150.0	$2.248 \cdot 10^{-1}$	$2.362 \cdot 10^{-1}$	$2.360 \cdot 10^{-1}$	$2.318 \cdot 10^{-1}$	$2.261 \cdot 10^{-1}$
200.0	$2.392 \cdot 10^{-1}$	$2.403 \cdot 10^{-1}$	$2.338 \cdot 10^{-1}$	$2.251 \cdot 10^{-1}$	$2.166 \cdot 10^{-1}$

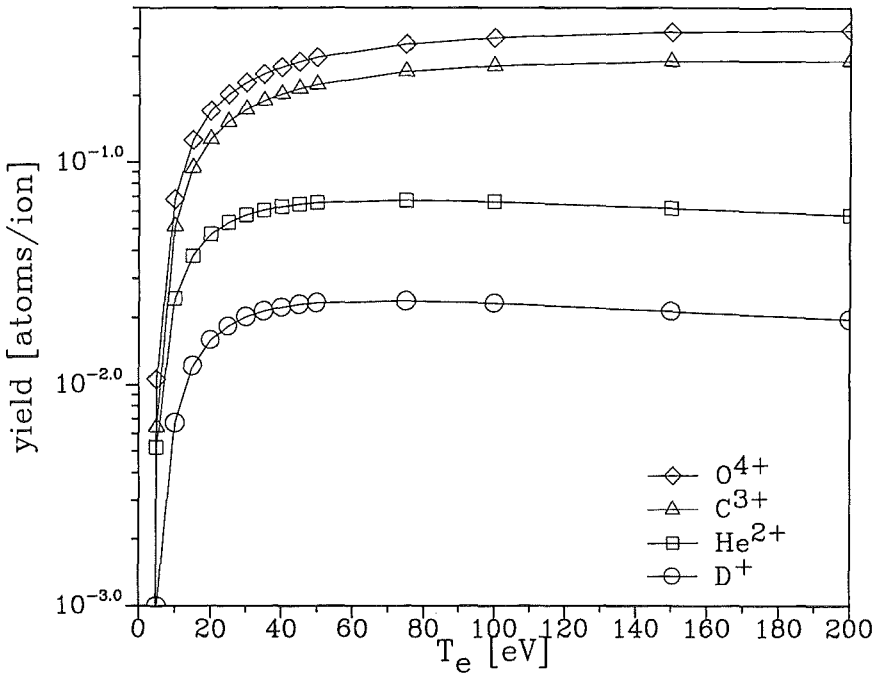


Figure C.5: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

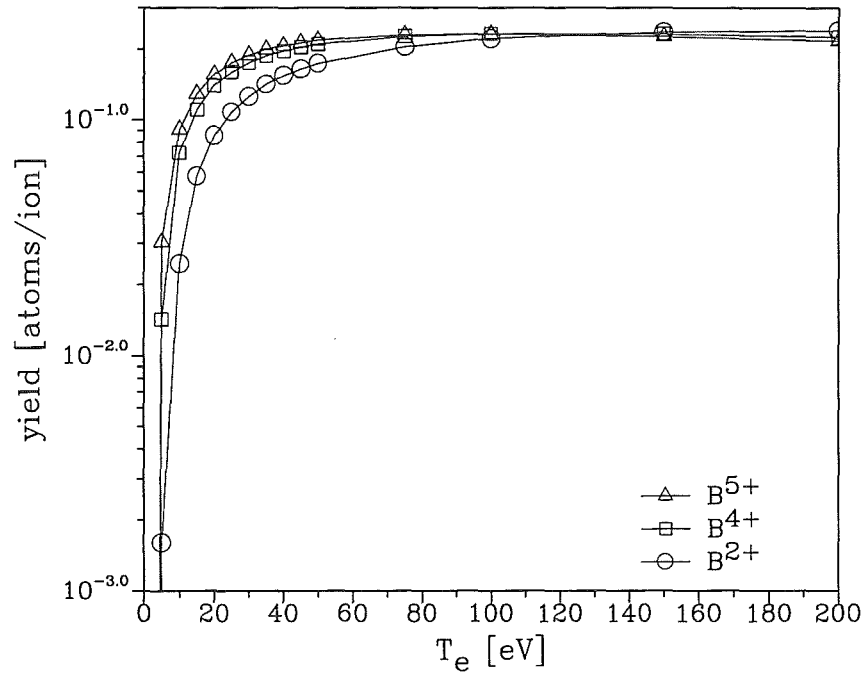


Figure C.6: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Carbon

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$2.807 \cdot 10^{-17}$	$3.672 \cdot 10^{-15}$	$4.328 \cdot 10^{-16}$	$2.298 \cdot 10^{-17}$	$4.653 \cdot 10^{-15}$
5.0	$1.237 \cdot 10^{-4}$	$3.530 \cdot 10^{-4}$	$3.132 \cdot 10^{-4}$	$3.157 \cdot 10^{-4}$	$2.303 \cdot 10^{-3}$
10.0	$1.968 \cdot 10^{-3}$	$4.138 \cdot 10^{-3}$	$4.363 \cdot 10^{-3}$	$5.901 \cdot 10^{-3}$	$1.713 \cdot 10^{-2}$
15.0	$4.789 \cdot 10^{-3}$	$9.126 \cdot 10^{-3}$	$1.020 \cdot 10^{-2}$	$1.557 \cdot 10^{-2}$	$3.022 \cdot 10^{-2}$
20.0	$7.269 \cdot 10^{-3}$	$1.309 \cdot 10^{-2}$	$1.517 \cdot 10^{-2}$	$2.499 \cdot 10^{-2}$	$3.948 \cdot 10^{-2}$
25.0	$9.080 \cdot 10^{-3}$	$1.605 \cdot 10^{-2}$	$1.882 \cdot 10^{-2}$	$3.251 \cdot 10^{-2}$	$4.606 \cdot 10^{-2}$
30.0	$1.053 \cdot 10^{-2}$	$1.825 \cdot 10^{-2}$	$2.178 \cdot 10^{-2}$	$3.902 \cdot 10^{-2}$	$5.091 \cdot 10^{-2}$
35.0	$1.155 \cdot 10^{-2}$	$1.991 \cdot 10^{-2}$	$2.388 \cdot 10^{-2}$	$4.406 \cdot 10^{-2}$	$5.447 \cdot 10^{-2}$
40.0	$1.238 \cdot 10^{-2}$	$2.109 \cdot 10^{-2}$	$2.556 \cdot 10^{-2}$	$4.833 \cdot 10^{-2}$	$5.714 \cdot 10^{-2}$
45.0	$1.298 \cdot 10^{-2}$	$2.202 \cdot 10^{-2}$	$2.679 \cdot 10^{-2}$	$5.184 \cdot 10^{-2}$	$5.942 \cdot 10^{-2}$
50.0	$1.346 \cdot 10^{-2}$	$2.271 \cdot 10^{-2}$	$2.783 \cdot 10^{-2}$	$5.467 \cdot 10^{-2}$	$6.099 \cdot 10^{-2}$
75.0	$1.449 \cdot 10^{-2}$	$2.424 \cdot 10^{-2}$	$3.004 \cdot 10^{-2}$	$6.363 \cdot 10^{-2}$	$6.462 \cdot 10^{-2}$
100.0	$1.452 \cdot 10^{-2}$	$2.423 \cdot 10^{-2}$	$3.035 \cdot 10^{-2}$	$6.764 \cdot 10^{-2}$	$6.485 \cdot 10^{-2}$
150.0	$1.380 \cdot 10^{-2}$	$2.302 \cdot 10^{-2}$	$2.908 \cdot 10^{-2}$	$6.963 \cdot 10^{-2}$	$6.222 \cdot 10^{-2}$
200.0	$1.288 \cdot 10^{-2}$	$2.147 \cdot 10^{-2}$	$2.727 \cdot 10^{-2}$	$6.869 \cdot 10^{-2}$	$5.860 \cdot 10^{-2}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$5.766 \cdot 10^{-29}$	$4.408 \cdot 10^{-26}$	$1.096 \cdot 10^{-23}$	$1.429 \cdot 10^{-21}$	$1.184 \cdot 10^{-19}$	$6.891 \cdot 10^{-18}$
5.0	$1.193 \cdot 10^{-5}$	$2.437 \cdot 10^{-4}$	$2.051 \cdot 10^{-3}$	$7.233 \cdot 10^{-3}$	$1.118 \cdot 10^{-2}$	$2.374 \cdot 10^{-2}$
10.0	$2.129 \cdot 10^{-3}$	$1.224 \cdot 10^{-2}$	$3.201 \cdot 10^{-2}$	$5.107 \cdot 10^{-2}$	$7.233 \cdot 10^{-2}$	$8.351 \cdot 10^{-2}$
15.0	$1.102 \cdot 10^{-2}$	$3.919 \cdot 10^{-2}$	$6.671 \cdot 10^{-2}$	$9.069 \cdot 10^{-2}$	$1.134 \cdot 10^{-1}$	$1.314 \cdot 10^{-1}$
20.0	$2.555 \cdot 10^{-2}$	$6.524 \cdot 10^{-2}$	$9.694 \cdot 10^{-2}$	$1.235 \cdot 10^{-1}$	$1.457 \cdot 10^{-1}$	$1.620 \cdot 10^{-1}$
25.0	$4.219 \cdot 10^{-2}$	$8.948 \cdot 10^{-2}$	$1.234 \cdot 10^{-1}$	$1.493 \cdot 10^{-1}$	$1.686 \cdot 10^{-1}$	$1.831 \cdot 10^{-1}$
30.0	$5.929 \cdot 10^{-2}$	$1.080 \cdot 10^{-1}$	$1.440 \cdot 10^{-1}$	$1.683 \cdot 10^{-1}$	$1.858 \cdot 10^{-1}$	$1.998 \cdot 10^{-1}$
35.0	$7.474 \cdot 10^{-2}$	$1.265 \cdot 10^{-1}$	$1.600 \cdot 10^{-1}$	$1.833 \cdot 10^{-1}$	$2.004 \cdot 10^{-1}$	$2.127 \cdot 10^{-1}$
40.0	$8.926 \cdot 10^{-2}$	$1.408 \cdot 10^{-1}$	$1.744 \cdot 10^{-1}$	$1.960 \cdot 10^{-1}$	$2.116 \cdot 10^{-1}$	$2.229 \cdot 10^{-1}$
45.0	$1.022 \cdot 10^{-1}$	$1.549 \cdot 10^{-1}$	$1.860 \cdot 10^{-1}$	$2.063 \cdot 10^{-1}$	$2.211 \cdot 10^{-1}$	$2.311 \cdot 10^{-1}$
50.0	$1.149 \cdot 10^{-1}$	$1.663 \cdot 10^{-1}$	$1.955 \cdot 10^{-1}$	$2.153 \cdot 10^{-1}$	$2.285 \cdot 10^{-1}$	$2.375 \cdot 10^{-1}$
75.0	$1.619 \cdot 10^{-1}$	$2.065 \cdot 10^{-1}$	$2.303 \cdot 10^{-1}$	$2.432 \cdot 10^{-1}$	$2.516 \cdot 10^{-1}$	$2.561 \cdot 10^{-1}$
100.0	$1.940 \cdot 10^{-1}$	$2.317 \cdot 10^{-1}$	$2.485 \cdot 10^{-1}$	$2.573 \cdot 10^{-1}$	$2.613 \cdot 10^{-1}$	$2.626 \cdot 10^{-1}$
150.0	$2.337 \cdot 10^{-1}$	$2.579 \cdot 10^{-1}$	$2.657 \cdot 10^{-1}$	$2.670 \cdot 10^{-1}$	$2.653 \cdot 10^{-1}$	$2.619 \cdot 10^{-1}$
200.0	$2.572 \cdot 10^{-1}$	$2.704 \cdot 10^{-1}$	$2.709 \cdot 10^{-1}$	$2.667 \cdot 10^{-1}$	$2.614 \cdot 10^{-1}$	$2.549 \cdot 10^{-1}$

T_e [eV]	O^{+}	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$6.194 \cdot 10^{-34}$	$7.393 \cdot 10^{-31}$	$2.641 \cdot 10^{-28}$	$4.758 \cdot 10^{-26}$	$5.343 \cdot 10^{-24}$	$4.169 \cdot 10^{-22}$
5.0	$2.204 \cdot 10^{-6}$	$6.117 \cdot 10^{-5}$	$7.319 \cdot 10^{-4}$	$4.149 \cdot 10^{-3}$	$1.002 \cdot 10^{-2}$	$1.296 \cdot 10^{-2}$
10.0	$1.128 \cdot 10^{-3}$	$8.391 \cdot 10^{-3}$	$2.308 \cdot 10^{-2}$	$4.835 \cdot 10^{-2}$	$6.285 \cdot 10^{-2}$	$9.052 \cdot 10^{-2}$
15.0	$7.579 \cdot 10^{-3}$	$3.162 \cdot 10^{-2}$	$6.542 \cdot 10^{-2}$	$9.620 \cdot 10^{-2}$	$1.219 \cdot 10^{-1}$	$1.408 \cdot 10^{-1}$
20.0	$2.012 \cdot 10^{-2}$	$6.213 \cdot 10^{-2}$	$1.023 \cdot 10^{-1}$	$1.355 \cdot 10^{-1}$	$1.627 \cdot 10^{-1}$	$1.851 \cdot 10^{-1}$
25.0	$3.639 \cdot 10^{-2}$	$8.749 \cdot 10^{-2}$	$1.326 \cdot 10^{-1}$	$1.671 \cdot 10^{-1}$	$1.938 \cdot 10^{-1}$	$2.145 \cdot 10^{-1}$
30.0	$5.393 \cdot 10^{-2}$	$1.142 \cdot 10^{-1}$	$1.573 \cdot 10^{-1}$	$1.931 \cdot 10^{-1}$	$2.181 \cdot 10^{-1}$	$2.398 \cdot 10^{-1}$
35.0	$7.190 \cdot 10^{-2}$	$1.344 \cdot 10^{-1}$	$1.816 \cdot 10^{-1}$	$2.146 \cdot 10^{-1}$	$2.389 \cdot 10^{-1}$	$2.588 \cdot 10^{-1}$
40.0	$8.845 \cdot 10^{-2}$	$1.553 \cdot 10^{-1}$	$2.006 \cdot 10^{-1}$	$2.326 \cdot 10^{-1}$	$2.568 \cdot 10^{-1}$	$2.750 \cdot 10^{-1}$
45.0	$1.041 \cdot 10^{-1}$	$1.714 \cdot 10^{-1}$	$2.163 \cdot 10^{-1}$	$2.481 \cdot 10^{-1}$	$2.707 \cdot 10^{-1}$	$2.881 \cdot 10^{-1}$
50.0	$1.182 \cdot 10^{-1}$	$1.879 \cdot 10^{-1}$	$2.308 \cdot 10^{-1}$	$2.615 \cdot 10^{-1}$	$2.830 \cdot 10^{-1}$	$2.993 \cdot 10^{-1}$
75.0	$1.786 \cdot 10^{-1}$	$2.449 \cdot 10^{-1}$	$2.829 \cdot 10^{-1}$	$3.073 \cdot 10^{-1}$	$3.241 \cdot 10^{-1}$	$3.352 \cdot 10^{-1}$
100.0	$2.221 \cdot 10^{-1}$	$2.831 \cdot 10^{-1}$	$3.146 \cdot 10^{-1}$	$3.335 \cdot 10^{-1}$	$3.458 \cdot 10^{-1}$	$3.532 \cdot 10^{-1}$
150.0	$2.799 \cdot 10^{-1}$	$3.278 \cdot 10^{-1}$	$3.492 \cdot 10^{-1}$	$3.600 \cdot 10^{-1}$	$3.654 \cdot 10^{-1}$	$3.669 \cdot 10^{-1}$
200.0	$3.172 \cdot 10^{-1}$	$3.530 \cdot 10^{-1}$	$3.662 \cdot 10^{-1}$	$3.704 \cdot 10^{-1}$	$3.705 \cdot 10^{-1}$	$3.676 \cdot 10^{-1}$

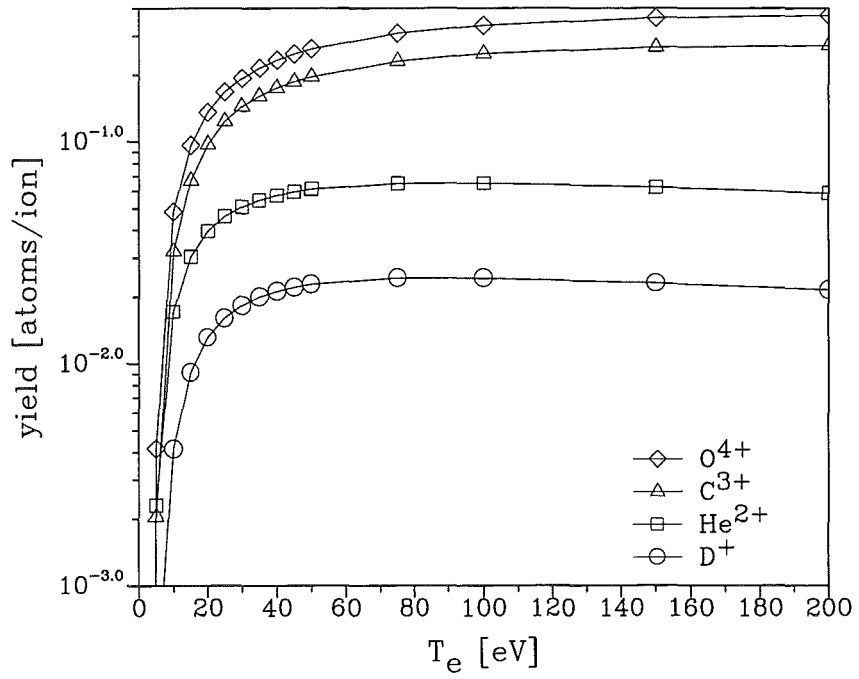


Figure C.7: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^{+}, $He^{2}+$, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

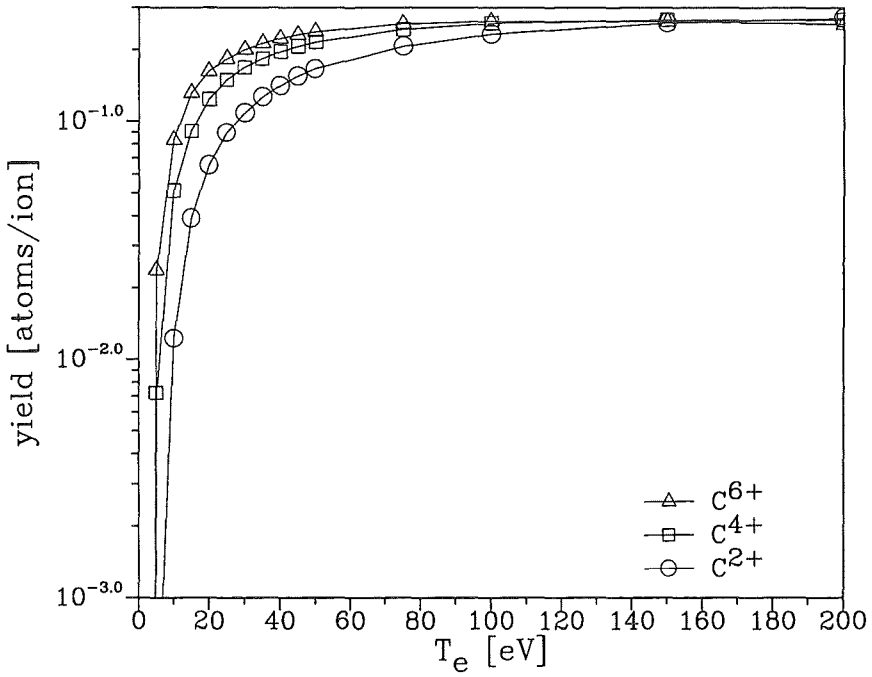


Figure C.8: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Silicon

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$3.625 \cdot 10^{-19}$	$1.149 \cdot 10^{-11}$	$5.502 \cdot 10^{-10}$	$1.887 \cdot 10^{-9}$	$1.020 \cdot 10^{-7}$
5.0	$7.381 \cdot 10^{-5}$	$1.139 \cdot 10^{-3}$	$2.530 \cdot 10^{-3}$	$4.445 \cdot 10^{-3}$	$1.522 \cdot 10^{-2}$
10.0	$1.701 \cdot 10^{-3}$	$7.970 \cdot 10^{-3}$	$1.363 \cdot 10^{-2}$	$2.359 \cdot 10^{-2}$	$4.230 \cdot 10^{-2}$
15.0	$4.814 \cdot 10^{-3}$	$1.495 \cdot 10^{-2}$	$2.316 \cdot 10^{-2}$	$4.068 \cdot 10^{-2}$	$6.094 \cdot 10^{-2}$
20.0	$8.001 \cdot 10^{-3}$	$2.032 \cdot 10^{-2}$	$3.047 \cdot 10^{-2}$	$5.414 \cdot 10^{-2}$	$7.352 \cdot 10^{-2}$
25.0	$1.063 \cdot 10^{-2}$	$2.426 \cdot 10^{-2}$	$3.580 \cdot 10^{-2}$	$6.478 \cdot 10^{-2}$	$8.259 \cdot 10^{-2}$
30.0	$1.280 \cdot 10^{-2}$	$2.751 \cdot 10^{-2}$	$3.998 \cdot 10^{-2}$	$7.323 \cdot 10^{-2}$	$8.952 \cdot 10^{-2}$
35.0	$1.464 \cdot 10^{-2}$	$3.000 \cdot 10^{-2}$	$4.321 \cdot 10^{-2}$	$8.022 \cdot 10^{-2}$	$9.511 \cdot 10^{-2}$
40.0	$1.605 \cdot 10^{-2}$	$3.199 \cdot 10^{-2}$	$4.571 \cdot 10^{-2}$	$8.595 \cdot 10^{-2}$	$9.931 \cdot 10^{-2}$
45.0	$1.737 \cdot 10^{-2}$	$3.358 \cdot 10^{-2}$	$4.785 \cdot 10^{-2}$	$9.078 \cdot 10^{-2}$	$1.027 \cdot 10^{-1}$
50.0	$1.838 \cdot 10^{-2}$	$3.491 \cdot 10^{-2}$	$4.946 \cdot 10^{-2}$	$9.497 \cdot 10^{-2}$	$1.057 \cdot 10^{-1}$
75.0	$2.160 \cdot 10^{-2}$	$3.871 \cdot 10^{-2}$	$5.435 \cdot 10^{-2}$	$1.089 \cdot 10^{-1}$	$1.140 \cdot 10^{-1}$
100.0	$2.311 \cdot 10^{-2}$	$4.022 \cdot 10^{-2}$	$5.615 \cdot 10^{-2}$	$1.165 \cdot 10^{-1}$	$1.173 \cdot 10^{-1}$
150.0	$2.394 \cdot 10^{-2}$	$4.057 \cdot 10^{-2}$	$5.642 \cdot 10^{-2}$	$1.233 \cdot 10^{-1}$	$1.180 \cdot 10^{-1}$
200.0	$2.369 \cdot 10^{-2}$	$3.959 \cdot 10^{-2}$	$5.502 \cdot 10^{-2}$	$1.251 \cdot 10^{-1}$	$1.156 \cdot 10^{-1}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$1.149 \cdot 10^{-11}$	$1.081 \cdot 10^{-9}$	$4.437 \cdot 10^{-8}$	$1.047 \cdot 10^{-6}$	$1.567 \cdot 10^{-5}$	$1.533 \cdot 10^{-4}$
5.0	$4.327 \cdot 10^{-3}$	$2.065 \cdot 10^{-2}$	$4.788 \cdot 10^{-2}$	$7.417 \cdot 10^{-2}$	$9.550 \cdot 10^{-2}$	$1.207 \cdot 10^{-1}$
10.0	$3.845 \cdot 10^{-2}$	$9.085 \cdot 10^{-2}$	$1.351 \cdot 10^{-1}$	$1.696 \cdot 10^{-1}$	$2.013 \cdot 10^{-1}$	$2.261 \cdot 10^{-1}$
15.0	$8.149 \cdot 10^{-2}$	$1.484 \cdot 10^{-1}$	$1.975 \cdot 10^{-1}$	$2.354 \cdot 10^{-1}$	$2.657 \cdot 10^{-1}$	$2.904 \cdot 10^{-1}$
20.0	$1.205 \cdot 10^{-1}$	$1.928 \cdot 10^{-1}$	$2.441 \cdot 10^{-1}$	$2.817 \cdot 10^{-1}$	$3.109 \cdot 10^{-1}$	$3.344 \cdot 10^{-1}$
25.0	$1.537 \cdot 10^{-1}$	$2.299 \cdot 10^{-1}$	$2.802 \cdot 10^{-1}$	$3.168 \cdot 10^{-1}$	$3.451 \cdot 10^{-1}$	$3.673 \cdot 10^{-1}$
30.0	$1.831 \cdot 10^{-1}$	$2.600 \cdot 10^{-1}$	$3.094 \cdot 10^{-1}$	$3.444 \cdot 10^{-1}$	$3.718 \cdot 10^{-1}$	$3.927 \cdot 10^{-1}$
35.0	$2.086 \cdot 10^{-1}$	$2.853 \cdot 10^{-1}$	$3.337 \cdot 10^{-1}$	$3.677 \cdot 10^{-1}$	$3.931 \cdot 10^{-1}$	$4.129 \cdot 10^{-1}$
40.0	$2.309 \cdot 10^{-1}$	$3.074 \cdot 10^{-1}$	$3.538 \cdot 10^{-1}$	$3.866 \cdot 10^{-1}$	$4.110 \cdot 10^{-1}$	$4.296 \cdot 10^{-1}$
45.0	$2.508 \cdot 10^{-1}$	$3.260 \cdot 10^{-1}$	$3.715 \cdot 10^{-1}$	$4.028 \cdot 10^{-1}$	$4.262 \cdot 10^{-1}$	$4.433 \cdot 10^{-1}$
50.0	$2.692 \cdot 10^{-1}$	$3.432 \cdot 10^{-1}$	$3.870 \cdot 10^{-1}$	$4.167 \cdot 10^{-1}$	$4.388 \cdot 10^{-1}$	$4.549 \cdot 10^{-1}$
75.0	$3.381 \cdot 10^{-1}$	$4.042 \cdot 10^{-1}$	$4.412 \cdot 10^{-1}$	$4.650 \cdot 10^{-1}$	$4.809 \cdot 10^{-1}$	$4.921 \cdot 10^{-1}$
100.0	$3.857 \cdot 10^{-1}$	$4.431 \cdot 10^{-1}$	$4.741 \cdot 10^{-1}$	$4.921 \cdot 10^{-1}$	$5.038 \cdot 10^{-1}$	$5.106 \cdot 10^{-1}$
150.0	$4.469 \cdot 10^{-1}$	$4.904 \cdot 10^{-1}$	$5.097 \cdot 10^{-1}$	$5.191 \cdot 10^{-1}$	$5.233 \cdot 10^{-1}$	$5.236 \cdot 10^{-1}$
200.0	$4.852 \cdot 10^{-1}$	$5.159 \cdot 10^{-1}$	$5.267 \cdot 10^{-1}$	$5.291 \cdot 10^{-1}$	$5.273 \cdot 10^{-1}$	$5.228 \cdot 10^{-1}$

T_e [eV]	O^+	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$3.251 \cdot 10^{-13}$	$4.042 \cdot 10^{-11}$	$2.133 \cdot 10^{-9}$	$6.480 \cdot 10^{-8}$	$1.273 \cdot 10^{-6}$	$1.695 \cdot 10^{-5}$
5.0	$2.997 \cdot 10^{-3}$	$1.709 \cdot 10^{-2}$	$4.554 \cdot 10^{-2}$	$7.105 \cdot 10^{-2}$	$1.048 \cdot 10^{-1}$	$1.278 \cdot 10^{-1}$
10.0	$3.506 \cdot 10^{-2}$	$9.255 \cdot 10^{-2}$	$1.457 \cdot 10^{-1}$	$1.941 \cdot 10^{-1}$	$2.332 \cdot 10^{-1}$	$2.661 \cdot 10^{-1}$
15.0	$8.289 \cdot 10^{-2}$	$1.622 \cdot 10^{-1}$	$2.268 \cdot 10^{-1}$	$2.777 \cdot 10^{-1}$	$3.194 \cdot 10^{-1}$	$3.532 \cdot 10^{-1}$
20.0	$1.277 \cdot 10^{-1}$	$2.208 \cdot 10^{-1}$	$2.884 \cdot 10^{-1}$	$3.385 \cdot 10^{-1}$	$3.818 \cdot 10^{-1}$	$4.154 \cdot 10^{-1}$
25.0	$1.691 \cdot 10^{-1}$	$2.683 \cdot 10^{-1}$	$3.375 \cdot 10^{-1}$	$3.890 \cdot 10^{-1}$	$4.301 \cdot 10^{-1}$	$4.629 \cdot 10^{-1}$
30.0	$2.046 \cdot 10^{-1}$	$3.082 \cdot 10^{-1}$	$3.770 \cdot 10^{-1}$	$4.285 \cdot 10^{-1}$	$4.686 \cdot 10^{-1}$	$5.006 \cdot 10^{-1}$
35.0	$2.375 \cdot 10^{-1}$	$3.425 \cdot 10^{-1}$	$4.113 \cdot 10^{-1}$	$4.615 \cdot 10^{-1}$	$5.005 \cdot 10^{-1}$	$5.313 \cdot 10^{-1}$
40.0	$2.665 \cdot 10^{-1}$	$3.709 \cdot 10^{-1}$	$4.406 \cdot 10^{-1}$	$4.895 \cdot 10^{-1}$	$5.274 \cdot 10^{-1}$	$5.569 \cdot 10^{-1}$
45.0	$2.929 \cdot 10^{-1}$	$3.982 \cdot 10^{-1}$	$4.662 \cdot 10^{-1}$	$5.142 \cdot 10^{-1}$	$5.506 \cdot 10^{-1}$	$5.791 \cdot 10^{-1}$
50.0	$3.165 \cdot 10^{-1}$	$4.224 \cdot 10^{-1}$	$4.884 \cdot 10^{-1}$	$5.351 \cdot 10^{-1}$	$5.709 \cdot 10^{-1}$	$5.979 \cdot 10^{-1}$
75.0	$4.104 \cdot 10^{-1}$	$5.108 \cdot 10^{-1}$	$5.710 \cdot 10^{-1}$	$6.116 \cdot 10^{-1}$	$6.411 \cdot 10^{-1}$	$6.628 \cdot 10^{-1}$
100.0	$4.769 \cdot 10^{-1}$	$5.705 \cdot 10^{-1}$	$6.242 \cdot 10^{-1}$	$6.582 \cdot 10^{-1}$	$6.836 \cdot 10^{-1}$	$7.006 \cdot 10^{-1}$
150.0	$5.671 \cdot 10^{-1}$	$6.468 \cdot 10^{-1}$	$6.881 \cdot 10^{-1}$	$7.134 \cdot 10^{-1}$	$7.292 \cdot 10^{-1}$	$7.381 \cdot 10^{-1}$
200.0	$6.275 \cdot 10^{-1}$	$6.925 \cdot 10^{-1}$	$7.250 \cdot 10^{-1}$	$7.411 \cdot 10^{-1}$	$7.498 \cdot 10^{-1}$	$7.524 \cdot 10^{-1}$

T_e [eV]	Si^+	Si^{2+}	Si^{3+}	Si^{4+}	Si^{5+}	Si^{6+}
1.0	$1.376 \cdot 10^{-17}$	$3.231 \cdot 10^{-15}$	$2.980 \cdot 10^{-13}$	$1.555 \cdot 10^{-11}$	$5.317 \cdot 10^{-10}$	$1.279 \cdot 10^{-8}$
5.0	$9.213 \cdot 10^{-4}$	$8.138 \cdot 10^{-3}$	$2.560 \cdot 10^{-2}$	$5.988 \cdot 10^{-2}$	$9.027 \cdot 10^{-2}$	$1.260 \cdot 10^{-1}$
10.0	$2.212 \cdot 10^{-2}$	$8.076 \cdot 10^{-2}$	$1.465 \cdot 10^{-1}$	$2.057 \cdot 10^{-1}$	$2.583 \cdot 10^{-1}$	$3.099 \cdot 10^{-1}$
15.0	$6.761 \cdot 10^{-2}$	$1.663 \cdot 10^{-1}$	$2.544 \cdot 10^{-1}$	$3.267 \cdot 10^{-1}$	$3.946 \cdot 10^{-1}$	$4.497 \cdot 10^{-1}$
20.0	$1.204 \cdot 10^{-1}$	$2.420 \cdot 10^{-1}$	$3.420 \cdot 10^{-1}$	$4.255 \cdot 10^{-1}$	$4.949 \cdot 10^{-1}$	$5.535 \cdot 10^{-1}$
25.0	$1.711 \cdot 10^{-1}$	$3.086 \cdot 10^{-1}$	$4.199 \cdot 10^{-1}$	$5.055 \cdot 10^{-1}$	$5.771 \cdot 10^{-1}$	$6.356 \cdot 10^{-1}$
30.0	$2.189 \cdot 10^{-1}$	$3.698 \cdot 10^{-1}$	$4.845 \cdot 10^{-1}$	$5.721 \cdot 10^{-1}$	$6.441 \cdot 10^{-1}$	$7.056 \cdot 10^{-1}$
35.0	$2.633 \cdot 10^{-1}$	$4.240 \cdot 10^{-1}$	$5.405 \cdot 10^{-1}$	$6.300 \cdot 10^{-1}$	$7.040 \cdot 10^{-1}$	$7.644 \cdot 10^{-1}$
40.0	$3.027 \cdot 10^{-1}$	$4.709 \cdot 10^{-1}$	$5.896 \cdot 10^{-1}$	$6.813 \cdot 10^{-1}$	$7.548 \cdot 10^{-1}$	$8.142 \cdot 10^{-1}$
45.0	$3.415 \cdot 10^{-1}$	$5.149 \cdot 10^{-1}$	$6.355 \cdot 10^{-1}$	$7.262 \cdot 10^{-1}$	$7.997 \cdot 10^{-1}$	$8.596 \cdot 10^{-1}$
50.0	$3.754 \cdot 10^{-1}$	$5.537 \cdot 10^{-1}$	$6.743 \cdot 10^{-1}$	$7.668 \cdot 10^{-1}$	$8.394 \cdot 10^{-1}$	$8.987 \cdot 10^{-1}$
75.0	$5.221 \cdot 10^{-1}$	$7.085 \cdot 10^{-1}$	$8.318 \cdot 10^{-1}$	$9.210 \cdot 10^{-1}$	$9.914 \cdot 10^{-1}$	$1.046 \cdot 10^{0}$
100.0	$6.322 \cdot 10^{-1}$	$8.216 \cdot 10^{-1}$	$9.416 \cdot 10^{-1}$	$1.027 \cdot 10^{0}$	$1.093 \cdot 10^{0}$	$1.144 \cdot 10^{0}$
150.0	$7.938 \cdot 10^{-1}$	$9.788 \cdot 10^{-1}$	$1.091 \cdot 10^{0}$	$1.167 \cdot 10^{0}$	$1.225 \cdot 10^{0}$	$1.268 \cdot 10^{0}$
200.0	$9.099 \cdot 10^{-1}$	$1.086 \cdot 10^{0}$	$1.189 \cdot 10^{0}$	$1.257 \cdot 10^{0}$	$1.307 \cdot 10^{0}$	$1.342 \cdot 10^{0}$

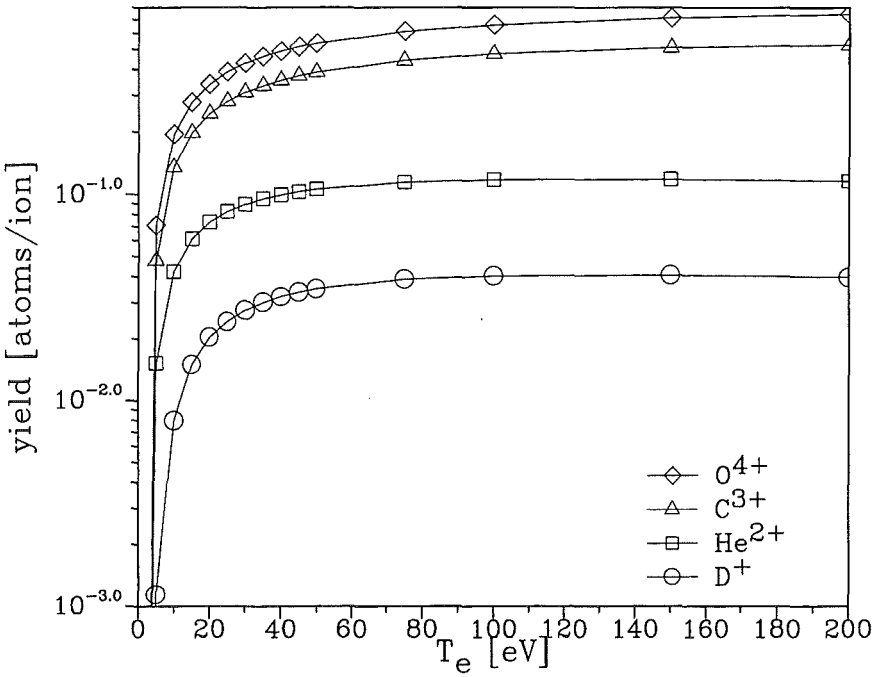


Figure C.9: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

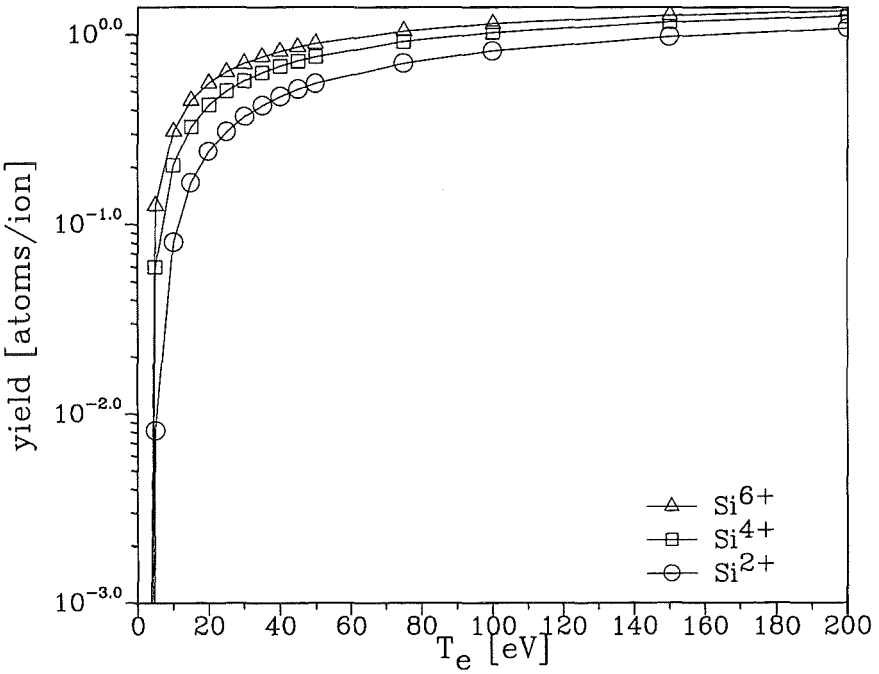


Figure C.10: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Iron

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$8.635 \cdot 10^{-35}$	$2.097 \cdot 10^{-17}$	$1.212 \cdot 10^{-12}$	$1.340 \cdot 10^{-10}$	$9.196 \cdot 10^{-9}$
5.0	$2.836 \cdot 10^{-7}$	$1.629 \cdot 10^{-4}$	$9.108 \cdot 10^{-4}$	$2.474 \cdot 10^{-3}$	$9.693 \cdot 10^{-3}$
10.0	$1.401 \cdot 10^{-4}$	$2.975 \cdot 10^{-3}$	$7.903 \cdot 10^{-3}$	$1.636 \cdot 10^{-2}$	$3.310 \cdot 10^{-2}$
15.0	$8.933 \cdot 10^{-4}$	$7.937 \cdot 10^{-3}$	$1.621 \cdot 10^{-2}$	$3.088 \cdot 10^{-2}$	$5.005 \cdot 10^{-2}$
20.0	$2.273 \cdot 10^{-3}$	$1.285 \cdot 10^{-2}$	$2.308 \cdot 10^{-2}$	$4.303 \cdot 10^{-2}$	$6.289 \cdot 10^{-2}$
25.0	$3.977 \cdot 10^{-3}$	$1.700 \cdot 10^{-2}$	$2.875 \cdot 10^{-2}$	$5.309 \cdot 10^{-2}$	$7.248 \cdot 10^{-2}$
30.0	$5.718 \cdot 10^{-3}$	$2.063 \cdot 10^{-2}$	$3.334 \cdot 10^{-2}$	$6.128 \cdot 10^{-2}$	$8.019 \cdot 10^{-2}$
35.0	$7.443 \cdot 10^{-3}$	$2.362 \cdot 10^{-2}$	$3.705 \cdot 10^{-2}$	$6.834 \cdot 10^{-2}$	$8.646 \cdot 10^{-2}$
40.0	$8.961 \cdot 10^{-3}$	$2.620 \cdot 10^{-2}$	$4.036 \cdot 10^{-2}$	$7.460 \cdot 10^{-2}$	$9.177 \cdot 10^{-2}$
45.0	$1.034 \cdot 10^{-2}$	$2.842 \cdot 10^{-2}$	$4.308 \cdot 10^{-2}$	$7.984 \cdot 10^{-2}$	$9.622 \cdot 10^{-2}$
50.0	$1.151 \cdot 10^{-2}$	$3.033 \cdot 10^{-2}$	$4.527 \cdot 10^{-2}$	$8.441 \cdot 10^{-2}$	$9.985 \cdot 10^{-2}$
75.0	$1.622 \cdot 10^{-2}$	$3.693 \cdot 10^{-2}$	$5.322 \cdot 10^{-2}$	$1.012 \cdot 10^{-1}$	$1.130 \cdot 10^{-1}$
100.0	$1.910 \cdot 10^{-2}$	$4.058 \cdot 10^{-2}$	$5.750 \cdot 10^{-2}$	$1.117 \cdot 10^{-1}$	$1.201 \cdot 10^{-1}$
150.0	$2.217 \cdot 10^{-2}$	$4.424 \cdot 10^{-2}$	$6.155 \cdot 10^{-2}$	$1.240 \cdot 10^{-1}$	$1.271 \cdot 10^{-1}$
200.0	$2.363 \cdot 10^{-2}$	$4.552 \cdot 10^{-2}$	$6.284 \cdot 10^{-2}$	$1.304 \cdot 10^{-1}$	$1.295 \cdot 10^{-1}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$8.588 \cdot 10^{-9}$	$4.407 \cdot 10^{-7}$	$1.009 \cdot 10^{-5}$	$1.268 \cdot 10^{-4}$	$9.075 \cdot 10^{-4}$	$3.470 \cdot 10^{-3}$
5.0	$9.555 \cdot 10^{-3}$	$3.467 \cdot 10^{-2}$	$6.271 \cdot 10^{-2}$	$8.811 \cdot 10^{-2}$	$1.106 \cdot 10^{-1}$	$1.320 \cdot 10^{-1}$
10.0	$5.186 \cdot 10^{-2}$	$1.038 \cdot 10^{-1}$	$1.460 \cdot 10^{-1}$	$1.814 \cdot 10^{-1}$	$2.105 \cdot 10^{-1}$	$2.353 \cdot 10^{-1}$
15.0	$9.417 \cdot 10^{-2}$	$1.581 \cdot 10^{-1}$	$2.061 \cdot 10^{-1}$	$2.438 \cdot 10^{-1}$	$2.739 \cdot 10^{-1}$	$2.994 \cdot 10^{-1}$
20.0	$1.305 \cdot 10^{-1}$	$2.016 \cdot 10^{-1}$	$2.516 \cdot 10^{-1}$	$2.891 \cdot 10^{-1}$	$3.202 \cdot 10^{-1}$	$3.455 \cdot 10^{-1}$
25.0	$1.616 \cdot 10^{-1}$	$2.366 \cdot 10^{-1}$	$2.873 \cdot 10^{-1}$	$3.255 \cdot 10^{-1}$	$3.561 \cdot 10^{-1}$	$3.810 \cdot 10^{-1}$
30.0	$1.891 \cdot 10^{-1}$	$2.660 \cdot 10^{-1}$	$3.171 \cdot 10^{-1}$	$3.551 \cdot 10^{-1}$	$3.853 \cdot 10^{-1}$	$4.092 \cdot 10^{-1}$
35.0	$2.132 \cdot 10^{-1}$	$2.906 \cdot 10^{-1}$	$3.421 \cdot 10^{-1}$	$3.800 \cdot 10^{-1}$	$4.094 \cdot 10^{-1}$	$4.326 \cdot 10^{-1}$
40.0	$2.349 \cdot 10^{-1}$	$3.129 \cdot 10^{-1}$	$3.640 \cdot 10^{-1}$	$4.007 \cdot 10^{-1}$	$4.298 \cdot 10^{-1}$	$4.524 \cdot 10^{-1}$
45.0	$2.540 \cdot 10^{-1}$	$3.327 \cdot 10^{-1}$	$3.830 \cdot 10^{-1}$	$4.194 \cdot 10^{-1}$	$4.474 \cdot 10^{-1}$	$4.693 \cdot 10^{-1}$
50.0	$2.716 \cdot 10^{-1}$	$3.501 \cdot 10^{-1}$	$4.001 \cdot 10^{-1}$	$4.356 \cdot 10^{-1}$	$4.631 \cdot 10^{-1}$	$4.841 \cdot 10^{-1}$
75.0	$3.406 \cdot 10^{-1}$	$4.167 \cdot 10^{-1}$	$4.627 \cdot 10^{-1}$	$4.945 \cdot 10^{-1}$	$5.182 \cdot 10^{-1}$	$5.357 \cdot 10^{-1}$
100.0	$3.901 \cdot 10^{-1}$	$4.620 \cdot 10^{-1}$	$5.040 \cdot 10^{-1}$	$5.319 \cdot 10^{-1}$	$5.522 \cdot 10^{-1}$	$5.667 \cdot 10^{-1}$
150.0	$4.582 \cdot 10^{-1}$	$5.212 \cdot 10^{-1}$	$5.553 \cdot 10^{-1}$	$5.766 \cdot 10^{-1}$	$5.908 \cdot 10^{-1}$	$5.997 \cdot 10^{-1}$
200.0	$5.045 \cdot 10^{-1}$	$5.582 \cdot 10^{-1}$	$5.856 \cdot 10^{-1}$	$6.008 \cdot 10^{-1}$	$6.100 \cdot 10^{-1}$	$6.142 \cdot 10^{-1}$

T_e [eV]	O^{+}	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$2.505 \cdot 10^{-9}$	$1.480 \cdot 10^{-7}$	$3.907 \cdot 10^{-6}$	$5.765 \cdot 10^{-5}$	$5.032 \cdot 10^{-4}$	$2.522 \cdot 10^{-3}$
5.0	$9.499 \cdot 10^{-3}$	$3.729 \cdot 10^{-2}$	$6.835 \cdot 10^{-2}$	$1.013 \cdot 10^{-1}$	$1.331 \cdot 10^{-1}$	$1.605 \cdot 10^{-1}$
10.0	$5.788 \cdot 10^{-2}$	$1.217 \cdot 10^{-1}$	$1.771 \cdot 10^{-1}$	$2.227 \cdot 10^{-1}$	$2.607 \cdot 10^{-1}$	$2.947 \cdot 10^{-1}$
15.0	$1.085 \cdot 10^{-1}$	$1.919 \cdot 10^{-1}$	$2.541 \cdot 10^{-1}$	$3.054 \cdot 10^{-1}$	$3.473 \cdot 10^{-1}$	$3.827 \cdot 10^{-1}$
20.0	$1.546 \cdot 10^{-1}$	$2.478 \cdot 10^{-1}$	$3.154 \cdot 10^{-1}$	$3.674 \cdot 10^{-1}$	$4.118 \cdot 10^{-1}$	$4.479 \cdot 10^{-1}$
25.0	$1.949 \cdot 10^{-1}$	$2.939 \cdot 10^{-1}$	$3.643 \cdot 10^{-1}$	$4.186 \cdot 10^{-1}$	$4.625 \cdot 10^{-1}$	$4.988 \cdot 10^{-1}$
30.0	$2.297 \cdot 10^{-1}$	$3.337 \cdot 10^{-1}$	$4.054 \cdot 10^{-1}$	$4.604 \cdot 10^{-1}$	$5.045 \cdot 10^{-1}$	$5.402 \cdot 10^{-1}$
35.0	$2.612 \cdot 10^{-1}$	$3.680 \cdot 10^{-1}$	$4.409 \cdot 10^{-1}$	$4.956 \cdot 10^{-1}$	$5.394 \cdot 10^{-1}$	$5.751 \cdot 10^{-1}$
40.0	$2.892 \cdot 10^{-1}$	$3.978 \cdot 10^{-1}$	$4.721 \cdot 10^{-1}$	$5.262 \cdot 10^{-1}$	$5.701 \cdot 10^{-1}$	$6.047 \cdot 10^{-1}$
45.0	$3.142 \cdot 10^{-1}$	$4.253 \cdot 10^{-1}$	$4.992 \cdot 10^{-1}$	$5.536 \cdot 10^{-1}$	$5.965 \cdot 10^{-1}$	$6.305 \cdot 10^{-1}$
50.0	$3.382 \cdot 10^{-1}$	$4.501 \cdot 10^{-1}$	$5.239 \cdot 10^{-1}$	$5.777 \cdot 10^{-1}$	$6.200 \cdot 10^{-1}$	$6.535 \cdot 10^{-1}$
75.0	$4.321 \cdot 10^{-1}$	$5.456 \cdot 10^{-1}$	$6.169 \cdot 10^{-1}$	$6.675 \cdot 10^{-1}$	$7.067 \cdot 10^{-1}$	$7.367 \cdot 10^{-1}$
100.0	$5.012 \cdot 10^{-1}$	$6.127 \cdot 10^{-1}$	$6.804 \cdot 10^{-1}$	$7.276 \cdot 10^{-1}$	$7.631 \cdot 10^{-1}$	$7.899 \cdot 10^{-1}$
150.0	$5.999 \cdot 10^{-1}$	$7.037 \cdot 10^{-1}$	$7.641 \cdot 10^{-1}$	$8.042 \cdot 10^{-1}$	$8.326 \cdot 10^{-1}$	$8.528 \cdot 10^{-1}$
200.0	$6.689 \cdot 10^{-1}$	$7.643 \cdot 10^{-1}$	$8.172 \cdot 10^{-1}$	$8.498 \cdot 10^{-1}$	$8.725 \cdot 10^{-1}$	$8.869 \cdot 10^{-1}$

T_e [eV]	Fe^{+}	Fe^{2+}	Fe^{3+}	Fe^{4+}	Fe^{5+}	Fe^{6+}
1.0	$6.949 \cdot 10^{-16}$	$1.104 \cdot 10^{-13}$	$8.080 \cdot 10^{-12}$	$3.473 \cdot 10^{-10}$	$9.899 \cdot 10^{-9}$	$1.985 \cdot 10^{-7}$
5.0	$2.953 \cdot 10^{-3}$	$1.676 \cdot 10^{-2}$	$4.492 \cdot 10^{-2}$	$9.465 \cdot 10^{-2}$	$1.275 \cdot 10^{-1}$	$1.881 \cdot 10^{-1}$
10.0	$4.767 \cdot 10^{-2}$	$1.322 \cdot 10^{-1}$	$2.231 \cdot 10^{-1}$	$3.091 \cdot 10^{-1}$	$3.886 \cdot 10^{-1}$	$4.614 \cdot 10^{-1}$
15.0	$1.257 \cdot 10^{-1}$	$2.582 \cdot 10^{-1}$	$3.770 \cdot 10^{-1}$	$4.887 \cdot 10^{-1}$	$5.839 \cdot 10^{-1}$	$6.729 \cdot 10^{-1}$
20.0	$2.070 \cdot 10^{-1}$	$3.710 \cdot 10^{-1}$	$5.142 \cdot 10^{-1}$	$6.378 \cdot 10^{-1}$	$7.471 \cdot 10^{-1}$	$8.443 \cdot 10^{-1}$
25.0	$2.810 \cdot 10^{-1}$	$4.709 \cdot 10^{-1}$	$6.293 \cdot 10^{-1}$	$7.651 \cdot 10^{-1}$	$8.844 \cdot 10^{-1}$	$9.883 \cdot 10^{-1}$
30.0	$3.515 \cdot 10^{-1}$	$5.606 \cdot 10^{-1}$	$7.317 \cdot 10^{-1}$	$8.770 \cdot 10^{-1}$	$1.003 \cdot 10^{0}$	$1.114 \cdot 10^{0}$
35.0	$4.146 \cdot 10^{-1}$	$6.415 \cdot 10^{-1}$	$8.241 \cdot 10^{-1}$	$9.760 \cdot 10^{-1}$	$1.110 \cdot 10^{0}$	$1.225 \cdot 10^{0}$
40.0	$4.744 \cdot 10^{-1}$	$7.152 \cdot 10^{-1}$	$9.072 \cdot 10^{-1}$	$1.067 \cdot 10^{0}$	$1.205 \cdot 10^{0}$	$1.326 \cdot 10^{0}$
45.0	$5.287 \cdot 10^{-1}$	$7.827 \cdot 10^{-1}$	$9.841 \cdot 10^{-1}$	$1.150 \cdot 10^{0}$	$1.294 \cdot 10^{0}$	$1.416 \cdot 10^{0}$
50.0	$5.807 \cdot 10^{-1}$	$8.459 \cdot 10^{-1}$	$1.053 \cdot 10^{0}$	$1.225 \cdot 10^{0}$	$1.373 \cdot 10^{0}$	$1.499 \cdot 10^{0}$
75.0	$7.985 \cdot 10^{-1}$	$1.109 \cdot 10^{0}$	$1.348 \cdot 10^{0}$	$1.537 \cdot 10^{0}$	$1.697 \cdot 10^{0}$	$1.833 \cdot 10^{0}$
100.0	$9.712 \cdot 10^{-1}$	$1.316 \cdot 10^{0}$	$1.571 \cdot 10^{0}$	$1.773 \cdot 10^{0}$	$1.941 \cdot 10^{0}$	$2.080 \cdot 10^{0}$
150.0	$1.243 \cdot 10^{0}$	$1.632 \cdot 10^{0}$	$1.909 \cdot 10^{0}$	$2.122 \cdot 10^{0}$	$2.296 \cdot 10^{0}$	$2.440 \cdot 10^{0}$
200.0	$1.455 \cdot 10^{0}$	$1.872 \cdot 10^{0}$	$2.160 \cdot 10^{0}$	$2.378 \cdot 10^{0}$	$2.553 \cdot 10^{0}$	$2.696 \cdot 10^{0}$

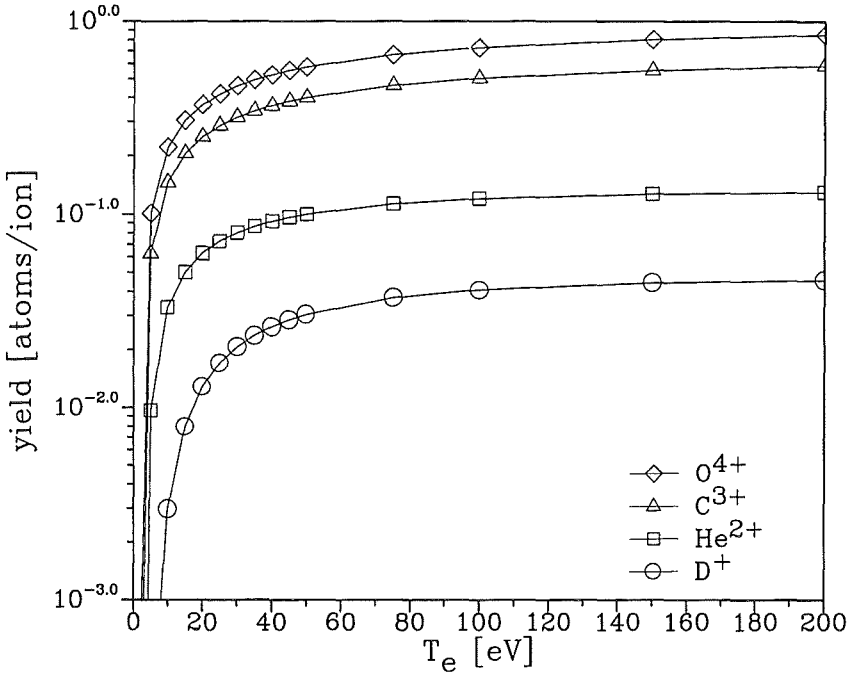


Figure C.11: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

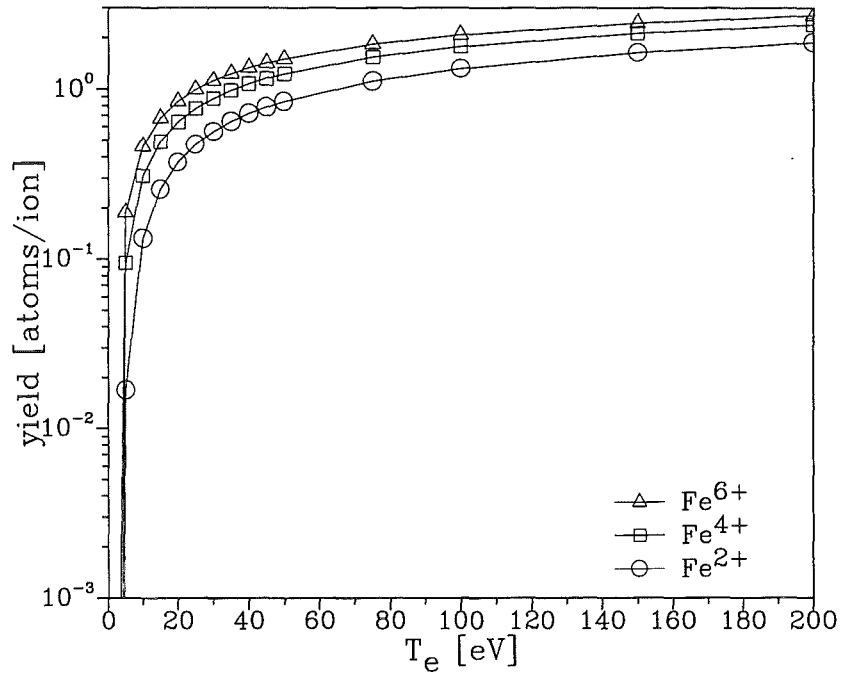


Figure C.12: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Nickel

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$5.305 \cdot 10^{-38}$	$8.254 \cdot 10^{-19}$	$1.690 \cdot 10^{-13}$	$3.335 \cdot 10^{-11}$	$2.577 \cdot 10^{-9}$
5.0	$8.512 \cdot 10^{-8}$	$1.019 \cdot 10^{-4}$	$6.765 \cdot 10^{-4}$	$1.966 \cdot 10^{-3}$	$8.138 \cdot 10^{-3}$
10.0	$8.487 \cdot 10^{-5}$	$2.349 \cdot 10^{-3}$	$6.773 \cdot 10^{-3}$	$1.443 \cdot 10^{-2}$	$3.059 \cdot 10^{-2}$
15.0	$6.431 \cdot 10^{-4}$	$6.779 \cdot 10^{-3}$	$1.469 \cdot 10^{-2}$	$2.839 \cdot 10^{-2}$	$4.719 \cdot 10^{-2}$
20.0	$1.773 \cdot 10^{-3}$	$1.147 \cdot 10^{-2}$	$2.129 \cdot 10^{-2}$	$4.021 \cdot 10^{-2}$	$5.980 \cdot 10^{-2}$
25.0	$3.262 \cdot 10^{-3}$	$1.554 \cdot 10^{-2}$	$2.698 \cdot 10^{-2}$	$4.995 \cdot 10^{-2}$	$6.961 \cdot 10^{-2}$
30.0	$4.876 \cdot 10^{-3}$	$1.904 \cdot 10^{-2}$	$3.157 \cdot 10^{-2}$	$5.824 \cdot 10^{-2}$	$7.729 \cdot 10^{-2}$
35.0	$6.483 \cdot 10^{-3}$	$2.209 \cdot 10^{-2}$	$3.543 \cdot 10^{-2}$	$6.532 \cdot 10^{-2}$	$8.372 \cdot 10^{-2}$
40.0	$7.992 \cdot 10^{-3}$	$2.461 \cdot 10^{-2}$	$3.862 \cdot 10^{-2}$	$7.134 \cdot 10^{-2}$	$8.898 \cdot 10^{-2}$
45.0	$9.338 \cdot 10^{-3}$	$2.694 \cdot 10^{-2}$	$4.135 \cdot 10^{-2}$	$7.659 \cdot 10^{-2}$	$9.360 \cdot 10^{-2}$
50.0	$1.056 \cdot 10^{-2}$	$2.886 \cdot 10^{-2}$	$4.376 \cdot 10^{-2}$	$8.149 \cdot 10^{-2}$	$9.738 \cdot 10^{-2}$
75.0	$1.526 \cdot 10^{-2}$	$3.572 \cdot 10^{-2}$	$5.205 \cdot 10^{-2}$	$9.850 \cdot 10^{-2}$	$1.110 \cdot 10^{-1}$
100.0	$1.827 \cdot 10^{-2}$	$3.974 \cdot 10^{-2}$	$5.660 \cdot 10^{-2}$	$1.093 \cdot 10^{-1}$	$1.186 \cdot 10^{-1}$
150.0	$2.167 \cdot 10^{-2}$	$4.381 \cdot 10^{-2}$	$6.120 \cdot 10^{-2}$	$1.223 \cdot 10^{-1}$	$1.264 \cdot 10^{-1}$
200.0	$2.331 \cdot 10^{-2}$	$4.542 \cdot 10^{-2}$	$6.282 \cdot 10^{-2}$	$1.291 \cdot 10^{-1}$	$1.294 \cdot 10^{-1}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$5.866 \cdot 10^{-9}$	$3.120 \cdot 10^{-7}$	$7.405 \cdot 10^{-6}$	$9.692 \cdot 10^{-5}$	$7.297 \cdot 10^{-4}$	$2.994 \cdot 10^{-3}$
5.0	$8.754 \cdot 10^{-3}$	$3.246 \cdot 10^{-2}$	$5.912 \cdot 10^{-2}$	$8.299 \cdot 10^{-2}$	$1.056 \cdot 10^{-1}$	$1.271 \cdot 10^{-1}$
10.0	$4.894 \cdot 10^{-2}$	$9.858 \cdot 10^{-2}$	$1.408 \cdot 10^{-1}$	$1.751 \cdot 10^{-1}$	$2.039 \cdot 10^{-1}$	$2.281 \cdot 10^{-1}$
15.0	$8.973 \cdot 10^{-2}$	$1.525 \cdot 10^{-1}$	$1.994 \cdot 10^{-1}$	$2.362 \cdot 10^{-1}$	$2.664 \cdot 10^{-1}$	$2.918 \cdot 10^{-1}$
20.0	$1.247 \cdot 10^{-1}$	$1.946 \cdot 10^{-1}$	$2.439 \cdot 10^{-1}$	$2.819 \cdot 10^{-1}$	$3.125 \cdot 10^{-1}$	$3.379 \cdot 10^{-1}$
25.0	$1.554 \cdot 10^{-1}$	$2.291 \cdot 10^{-1}$	$2.795 \cdot 10^{-1}$	$3.177 \cdot 10^{-1}$	$3.485 \cdot 10^{-1}$	$3.733 \cdot 10^{-1}$
30.0	$1.824 \cdot 10^{-1}$	$2.580 \cdot 10^{-1}$	$3.092 \cdot 10^{-1}$	$3.475 \cdot 10^{-1}$	$3.776 \cdot 10^{-1}$	$4.019 \cdot 10^{-1}$
35.0	$2.062 \cdot 10^{-1}$	$2.833 \cdot 10^{-1}$	$3.344 \cdot 10^{-1}$	$3.722 \cdot 10^{-1}$	$4.018 \cdot 10^{-1}$	$4.257 \cdot 10^{-1}$
40.0	$2.271 \cdot 10^{-1}$	$3.052 \cdot 10^{-1}$	$3.563 \cdot 10^{-1}$	$3.931 \cdot 10^{-1}$	$4.224 \cdot 10^{-1}$	$4.455 \cdot 10^{-1}$
45.0	$2.462 \cdot 10^{-1}$	$3.245 \cdot 10^{-1}$	$3.749 \cdot 10^{-1}$	$4.119 \cdot 10^{-1}$	$4.404 \cdot 10^{-1}$	$4.627 \cdot 10^{-1}$
50.0	$2.634 \cdot 10^{-1}$	$3.416 \cdot 10^{-1}$	$3.922 \cdot 10^{-1}$	$4.281 \cdot 10^{-1}$	$4.562 \cdot 10^{-1}$	$4.778 \cdot 10^{-1}$
75.0	$3.319 \cdot 10^{-1}$	$4.087 \cdot 10^{-1}$	$4.556 \cdot 10^{-1}$	$4.881 \cdot 10^{-1}$	$5.126 \cdot 10^{-1}$	$5.311 \cdot 10^{-1}$
100.0	$3.809 \cdot 10^{-1}$	$4.542 \cdot 10^{-1}$	$4.976 \cdot 10^{-1}$	$5.269 \cdot 10^{-1}$	$5.481 \cdot 10^{-1}$	$5.635 \cdot 10^{-1}$
150.0	$4.501 \cdot 10^{-1}$	$5.148 \cdot 10^{-1}$	$5.509 \cdot 10^{-1}$	$5.735 \cdot 10^{-1}$	$5.890 \cdot 10^{-1}$	$5.990 \cdot 10^{-1}$
200.0	$4.966 \cdot 10^{-1}$	$5.535 \cdot 10^{-1}$	$5.827 \cdot 10^{-1}$	$5.995 \cdot 10^{-1}$	$6.102 \cdot 10^{-1}$	$6.157 \cdot 10^{-1}$

T_e [eV]	O^+	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$1.949 \cdot 10^{-9}$	$1.175 \cdot 10^{-7}$	$3.164 \cdot 10^{-6}$	$4.774 \cdot 10^{-5}$	$4.283 \cdot 10^{-4}$	$2.227 \cdot 10^{-3}$
5.0	$8.927 \cdot 10^{-3}$	$3.535 \cdot 10^{-2}$	$6.497 \cdot 10^{-2}$	$9.788 \cdot 10^{-2}$	$1.285 \cdot 10^{-1}$	$1.550 \cdot 10^{-1}$
10.0	$5.530 \cdot 10^{-2}$	$1.174 \cdot 10^{-1}$	$1.710 \cdot 10^{-1}$	$2.155 \cdot 10^{-1}$	$2.533 \cdot 10^{-1}$	$2.868 \cdot 10^{-1}$
15.0	$1.043 \cdot 10^{-1}$	$1.855 \cdot 10^{-1}$	$2.461 \cdot 10^{-1}$	$2.973 \cdot 10^{-1}$	$3.382 \cdot 10^{-1}$	$3.741 \cdot 10^{-1}$
20.0	$1.491 \cdot 10^{-1}$	$2.401 \cdot 10^{-1}$	$3.071 \cdot 10^{-1}$	$3.594 \cdot 10^{-1}$	$4.023 \cdot 10^{-1}$	$4.386 \cdot 10^{-1}$
25.0	$1.881 \cdot 10^{-1}$	$2.856 \cdot 10^{-1}$	$3.550 \cdot 10^{-1}$	$4.090 \cdot 10^{-1}$	$4.533 \cdot 10^{-1}$	$4.892 \cdot 10^{-1}$
30.0	$2.225 \cdot 10^{-1}$	$3.249 \cdot 10^{-1}$	$3.965 \cdot 10^{-1}$	$4.510 \cdot 10^{-1}$	$4.947 \cdot 10^{-1}$	$5.310 \cdot 10^{-1}$
35.0	$2.531 \cdot 10^{-1}$	$3.588 \cdot 10^{-1}$	$4.312 \cdot 10^{-1}$	$4.866 \cdot 10^{-1}$	$5.303 \cdot 10^{-1}$	$5.659 \cdot 10^{-1}$
40.0	$2.807 \cdot 10^{-1}$	$3.890 \cdot 10^{-1}$	$4.624 \cdot 10^{-1}$	$5.170 \cdot 10^{-1}$	$5.607 \cdot 10^{-1}$	$5.959 \cdot 10^{-1}$
45.0	$3.052 \cdot 10^{-1}$	$4.161 \cdot 10^{-1}$	$4.891 \cdot 10^{-1}$	$5.441 \cdot 10^{-1}$	$5.873 \cdot 10^{-1}$	$6.221 \cdot 10^{-1}$
50.0	$3.287 \cdot 10^{-1}$	$4.398 \cdot 10^{-1}$	$5.138 \cdot 10^{-1}$	$5.683 \cdot 10^{-1}$	$6.112 \cdot 10^{-1}$	$6.452 \cdot 10^{-1}$
75.0	$4.216 \cdot 10^{-1}$	$5.353 \cdot 10^{-1}$	$6.075 \cdot 10^{-1}$	$6.591 \cdot 10^{-1}$	$6.991 \cdot 10^{-1}$	$7.301 \cdot 10^{-1}$
100.0	$4.901 \cdot 10^{-1}$	$6.028 \cdot 10^{-1}$	$6.720 \cdot 10^{-1}$	$7.204 \cdot 10^{-1}$	$7.569 \cdot 10^{-1}$	$7.847 \cdot 10^{-1}$
150.0	$5.891 \cdot 10^{-1}$	$6.952 \cdot 10^{-1}$	$7.576 \cdot 10^{-1}$	$7.991 \cdot 10^{-1}$	$8.292 \cdot 10^{-1}$	$8.507 \cdot 10^{-1}$
200.0	$6.586 \cdot 10^{-1}$	$7.569 \cdot 10^{-1}$	$8.121 \cdot 10^{-1}$	$8.470 \cdot 10^{-1}$	$8.715 \cdot 10^{-1}$	$8.876 \cdot 10^{-1}$

T_e [eV]	Ni^+	Ni^{2+}	Ni^{3+}	Ni^{4+}	Ni^{5+}	Ni^{6+}
1.0	$2.573 \cdot 10^{-16}$	$4.067 \cdot 10^{-14}$	$3.108 \cdot 10^{-12}$	$1.401 \cdot 10^{-10}$	$4.197 \cdot 10^{-9}$	$8.879 \cdot 10^{-8}$
5.0	$2.836 \cdot 10^{-3}$	$1.574 \cdot 10^{-2}$	$4.238 \cdot 10^{-2}$	$9.155 \cdot 10^{-2}$	$1.255 \cdot 10^{-1}$	$1.847 \cdot 10^{-1}$
10.0	$4.838 \cdot 10^{-2}$	$1.315 \cdot 10^{-1}$	$2.223 \cdot 10^{-1}$	$3.080 \cdot 10^{-1}$	$3.873 \cdot 10^{-1}$	$4.600 \cdot 10^{-1}$
15.0	$1.294 \cdot 10^{-1}$	$2.603 \cdot 10^{-1}$	$3.798 \cdot 10^{-1}$	$4.890 \cdot 10^{-1}$	$5.874 \cdot 10^{-1}$	$6.738 \cdot 10^{-1}$
20.0	$2.139 \cdot 10^{-1}$	$3.756 \cdot 10^{-1}$	$5.168 \cdot 10^{-1}$	$6.395 \cdot 10^{-1}$	$7.498 \cdot 10^{-1}$	$8.518 \cdot 10^{-1}$
25.0	$2.910 \cdot 10^{-1}$	$4.772 \cdot 10^{-1}$	$6.359 \cdot 10^{-1}$	$7.697 \cdot 10^{-1}$	$8.954 \cdot 10^{-1}$	$1.002 \cdot 10^{0}$
30.0	$3.636 \cdot 10^{-1}$	$5.679 \cdot 10^{-1}$	$7.388 \cdot 10^{-1}$	$8.877 \cdot 10^{-1}$	$1.018 \cdot 10^{0}$	$1.133 \cdot 10^{0}$
35.0	$4.284 \cdot 10^{-1}$	$6.494 \cdot 10^{-1}$	$8.357 \cdot 10^{-1}$	$9.918 \cdot 10^{-1}$	$1.128 \cdot 10^{0}$	$1.249 \cdot 10^{0}$
40.0	$4.897 \cdot 10^{-1}$	$7.247 \cdot 10^{-1}$	$9.207 \cdot 10^{-1}$	$1.084 \cdot 10^{0}$	$1.228 \cdot 10^{0}$	$1.352 \cdot 10^{0}$
45.0	$5.452 \cdot 10^{-1}$	$7.953 \cdot 10^{-1}$	$9.999 \cdot 10^{-1}$	$1.171 \cdot 10^{0}$	$1.319 \cdot 10^{0}$	$1.447 \cdot 10^{0}$
50.0	$5.985 \cdot 10^{-1}$	$8.594 \cdot 10^{-1}$	$1.074 \cdot 10^{0}$	$1.249 \cdot 10^{0}$	$1.402 \cdot 10^{0}$	$1.533 \cdot 10^{0}$
75.0	$8.205 \cdot 10^{-1}$	$1.134 \cdot 10^{0}$	$1.377 \cdot 10^{0}$	$1.573 \cdot 10^{0}$	$1.742 \cdot 10^{0}$	$1.886 \cdot 10^{0}$
100.0	$9.967 \cdot 10^{-1}$	$1.347 \cdot 10^{0}$	$1.610 \cdot 10^{0}$	$1.823 \cdot 10^{0}$	$2.000 \cdot 10^{0}$	$2.151 \cdot 10^{0}$
150.0	$1.277 \cdot 10^{0}$	$1.676 \cdot 10^{0}$	$1.968 \cdot 10^{0}$	$2.195 \cdot 10^{0}$	$2.383 \cdot 10^{0}$	$2.538 \cdot 10^{0}$
200.0	$1.495 \cdot 10^{0}$	$1.929 \cdot 10^{0}$	$2.235 \cdot 10^{0}$	$2.470 \cdot 10^{0}$	$2.662 \cdot 10^{0}$	$2.818 \cdot 10^{0}$

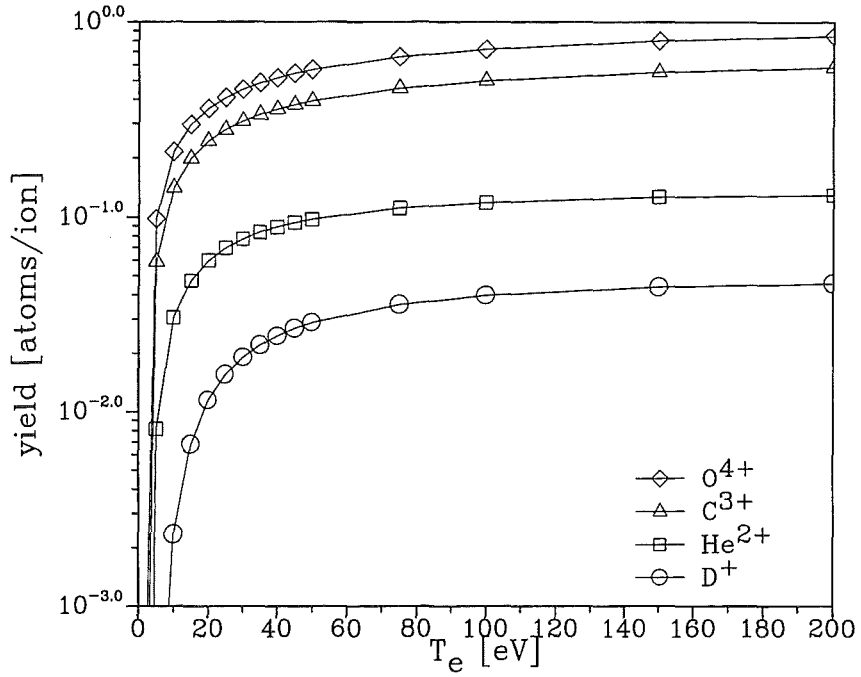


Figure C.13: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

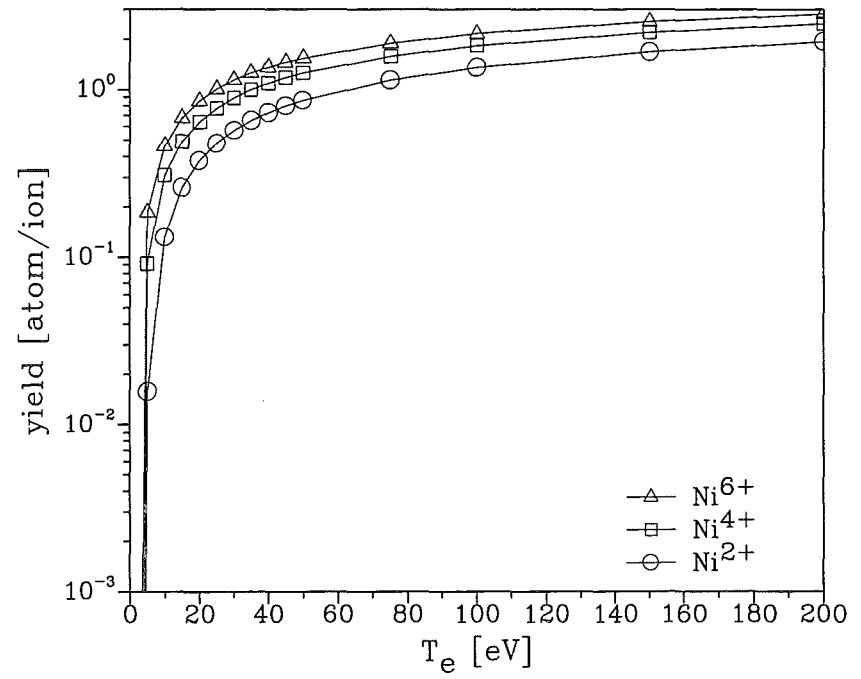


Figure C.14: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Copper

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$2.564 \cdot 10^{-32}$	$7.440 \cdot 10^{-16}$	$2.738 \cdot 10^{-11}$	$2.582 \cdot 10^{-9}$	$1.349 \cdot 10^{-7}$
5.0	$7.351 \cdot 10^{-7}$	$2.846 \cdot 10^{-4}$	$1.535 \cdot 10^{-3}$	$4.016 \cdot 10^{-3}$	$1.402 \cdot 10^{-2}$
10.0	$2.157 \cdot 10^{-4}$	$4.114 \cdot 10^{-3}$	$1.079 \cdot 10^{-2}$	$2.143 \cdot 10^{-2}$	$3.993 \cdot 10^{-2}$
15.0	$1.221 \cdot 10^{-3}$	$1.013 \cdot 10^{-2}$	$2.064 \cdot 10^{-2}$	$3.774 \cdot 10^{-2}$	$5.917 \cdot 10^{-2}$
20.0	$2.936 \cdot 10^{-3}$	$1.576 \cdot 10^{-2}$	$2.863 \cdot 10^{-2}$	$5.099 \cdot 10^{-2}$	$7.303 \cdot 10^{-2}$
25.0	$4.958 \cdot 10^{-3}$	$2.046 \cdot 10^{-2}$	$3.497 \cdot 10^{-2}$	$6.197 \cdot 10^{-2}$	$8.374 \cdot 10^{-2}$
30.0	$6.996 \cdot 10^{-3}$	$2.457 \cdot 10^{-2}$	$4.017 \cdot 10^{-2}$	$7.114 \cdot 10^{-2}$	$9.206 \cdot 10^{-2}$
35.0	$8.905 \cdot 10^{-3}$	$2.790 \cdot 10^{-2}$	$4.453 \cdot 10^{-2}$	$7.881 \cdot 10^{-2}$	$9.912 \cdot 10^{-2}$
40.0	$1.060 \cdot 10^{-2}$	$3.081 \cdot 10^{-2}$	$4.811 \cdot 10^{-2}$	$8.548 \cdot 10^{-2}$	$1.050 \cdot 10^{-1}$
45.0	$1.211 \cdot 10^{-2}$	$3.326 \cdot 10^{-2}$	$5.117 \cdot 10^{-2}$	$9.132 \cdot 10^{-2}$	$1.100 \cdot 10^{-1}$
50.0	$1.353 \cdot 10^{-2}$	$3.544 \cdot 10^{-2}$	$5.396 \cdot 10^{-2}$	$9.642 \cdot 10^{-2}$	$1.142 \cdot 10^{-1}$
75.0	$1.873 \cdot 10^{-2}$	$4.285 \cdot 10^{-2}$	$6.291 \cdot 10^{-2}$	$1.149 \cdot 10^{-1}$	$1.288 \cdot 10^{-1}$
100.0	$2.190 \cdot 10^{-2}$	$4.716 \cdot 10^{-2}$	$6.802 \cdot 10^{-2}$	$1.267 \cdot 10^{-1}$	$1.372 \cdot 10^{-1}$
150.0	$2.547 \cdot 10^{-2}$	$5.147 \cdot 10^{-2}$	$7.299 \cdot 10^{-2}$	$1.406 \cdot 10^{-1}$	$1.458 \cdot 10^{-1}$
200.0	$2.712 \cdot 10^{-2}$	$5.319 \cdot 10^{-2}$	$7.470 \cdot 10^{-2}$	$1.482 \cdot 10^{-1}$	$1.491 \cdot 10^{-1}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$4.232 \cdot 10^{-7}$	$1.367 \cdot 10^{-5}$	$1.937 \cdot 10^{-4}$	$1.368 \cdot 10^{-3}$	$4.507 \cdot 10^{-3}$	$6.288 \cdot 10^{-3}$
5.0	$1.825 \cdot 10^{-2}$	$5.179 \cdot 10^{-2}$	$8.417 \cdot 10^{-2}$	$1.136 \cdot 10^{-1}$	$1.414 \cdot 10^{-1}$	$1.648 \cdot 10^{-1}$
10.0	$7.221 \cdot 10^{-2}$	$1.311 \cdot 10^{-1}$	$1.790 \cdot 10^{-1}$	$2.162 \cdot 10^{-1}$	$2.479 \cdot 10^{-1}$	$2.749 \cdot 10^{-1}$
15.0	$1.201 \cdot 10^{-1}$	$1.918 \cdot 10^{-1}$	$2.437 \cdot 10^{-1}$	$2.837 \cdot 10^{-1}$	$3.175 \cdot 10^{-1}$	$3.449 \cdot 10^{-1}$
20.0	$1.601 \cdot 10^{-1}$	$2.376 \cdot 10^{-1}$	$2.920 \cdot 10^{-1}$	$3.341 \cdot 10^{-1}$	$3.678 \cdot 10^{-1}$	$3.957 \cdot 10^{-1}$
25.0	$1.945 \cdot 10^{-1}$	$2.756 \cdot 10^{-1}$	$3.317 \cdot 10^{-1}$	$3.734 \cdot 10^{-1}$	$4.071 \cdot 10^{-1}$	$4.348 \cdot 10^{-1}$
30.0	$2.243 \cdot 10^{-1}$	$3.073 \cdot 10^{-1}$	$3.638 \cdot 10^{-1}$	$4.060 \cdot 10^{-1}$	$4.394 \cdot 10^{-1}$	$4.661 \cdot 10^{-1}$
35.0	$2.503 \cdot 10^{-1}$	$3.354 \cdot 10^{-1}$	$3.918 \cdot 10^{-1}$	$4.333 \cdot 10^{-1}$	$4.662 \cdot 10^{-1}$	$4.926 \cdot 10^{-1}$
40.0	$2.730 \cdot 10^{-1}$	$3.590 \cdot 10^{-1}$	$4.155 \cdot 10^{-1}$	$4.564 \cdot 10^{-1}$	$4.885 \cdot 10^{-1}$	$5.147 \cdot 10^{-1}$
45.0	$2.941 \cdot 10^{-1}$	$3.803 \cdot 10^{-1}$	$4.365 \cdot 10^{-1}$	$4.767 \cdot 10^{-1}$	$5.088 \cdot 10^{-1}$	$5.336 \cdot 10^{-1}$
50.0	$3.128 \cdot 10^{-1}$	$3.993 \cdot 10^{-1}$	$4.552 \cdot 10^{-1}$	$4.945 \cdot 10^{-1}$	$5.261 \cdot 10^{-1}$	$5.502 \cdot 10^{-1}$
75.0	$3.878 \cdot 10^{-1}$	$4.728 \cdot 10^{-1}$	$5.250 \cdot 10^{-1}$	$5.613 \cdot 10^{-1}$	$5.889 \cdot 10^{-1}$	$6.096 \cdot 10^{-1}$
100.0	$4.414 \cdot 10^{-1}$	$5.224 \cdot 10^{-1}$	$5.714 \cdot 10^{-1}$	$6.044 \cdot 10^{-1}$	$6.283 \cdot 10^{-1}$	$6.457 \cdot 10^{-1}$
150.0	$5.169 \cdot 10^{-1}$	$5.895 \cdot 10^{-1}$	$6.305 \cdot 10^{-1}$	$6.563 \cdot 10^{-1}$	$6.742 \cdot 10^{-1}$	$6.857 \cdot 10^{-1}$
200.0	$5.676 \cdot 10^{-1}$	$6.324 \cdot 10^{-1}$	$6.659 \cdot 10^{-1}$	$6.857 \cdot 10^{-1}$	$6.981 \cdot 10^{-1}$	$7.047 \cdot 10^{-1}$

T_e [eV]	O^{+}	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$2.390 \cdot 10^{-7}$	$8.479 \cdot 10^{-6}$	$1.339 \cdot 10^{-4}$	$1.089 \cdot 10^{-3}$	$4.385 \cdot 10^{-3}$	$7.533 \cdot 10^{-3}$
5.0	$2.010 \cdot 10^{-2}$	$6.012 \cdot 10^{-2}$	$1.010 \cdot 10^{-1}$	$1.374 \cdot 10^{-1}$	$1.706 \cdot 10^{-1}$	$2.009 \cdot 10^{-1}$
10.0	$8.492 \cdot 10^{-2}$	$1.593 \cdot 10^{-1}$	$2.196 \cdot 10^{-1}$	$2.698 \cdot 10^{-1}$	$3.131 \cdot 10^{-1}$	$3.495 \cdot 10^{-1}$
15.0	$1.447 \cdot 10^{-1}$	$2.359 \cdot 10^{-1}$	$3.055 \cdot 10^{-1}$	$3.609 \cdot 10^{-1}$	$4.070 \cdot 10^{-1}$	$4.460 \cdot 10^{-1}$
20.0	$1.952 \cdot 10^{-1}$	$2.972 \cdot 10^{-1}$	$3.716 \cdot 10^{-1}$	$4.298 \cdot 10^{-1}$	$4.772 \cdot 10^{-1}$	$5.174 \cdot 10^{-1}$
25.0	$2.391 \cdot 10^{-1}$	$3.482 \cdot 10^{-1}$	$4.252 \cdot 10^{-1}$	$4.845 \cdot 10^{-1}$	$5.333 \cdot 10^{-1}$	$5.737 \cdot 10^{-1}$
30.0	$2.772 \cdot 10^{-1}$	$3.910 \cdot 10^{-1}$	$4.703 \cdot 10^{-1}$	$5.309 \cdot 10^{-1}$	$5.798 \cdot 10^{-1}$	$6.196 \cdot 10^{-1}$
35.0	$3.112 \cdot 10^{-1}$	$4.288 \cdot 10^{-1}$	$5.092 \cdot 10^{-1}$	$5.697 \cdot 10^{-1}$	$6.188 \cdot 10^{-1}$	$6.588 \cdot 10^{-1}$
40.0	$3.419 \cdot 10^{-1}$	$4.622 \cdot 10^{-1}$	$5.433 \cdot 10^{-1}$	$6.038 \cdot 10^{-1}$	$6.526 \cdot 10^{-1}$	$6.919 \cdot 10^{-1}$
45.0	$3.695 \cdot 10^{-1}$	$4.915 \cdot 10^{-1}$	$5.734 \cdot 10^{-1}$	$6.338 \cdot 10^{-1}$	$6.820 \cdot 10^{-1}$	$7.210 \cdot 10^{-1}$
50.0	$3.952 \cdot 10^{-1}$	$5.184 \cdot 10^{-1}$	$6.007 \cdot 10^{-1}$	$6.607 \cdot 10^{-1}$	$7.084 \cdot 10^{-1}$	$7.467 \cdot 10^{-1}$
75.0	$4.974 \cdot 10^{-1}$	$6.239 \cdot 10^{-1}$	$7.037 \cdot 10^{-1}$	$7.618 \cdot 10^{-1}$	$8.066 \cdot 10^{-1}$	$8.415 \cdot 10^{-1}$
100.0	$5.729 \cdot 10^{-1}$	$6.982 \cdot 10^{-1}$	$7.757 \cdot 10^{-1}$	$8.300 \cdot 10^{-1}$	$8.714 \cdot 10^{-1}$	$9.026 \cdot 10^{-1}$
150.0	$6.818 \cdot 10^{-1}$	$8.006 \cdot 10^{-1}$	$8.710 \cdot 10^{-1}$	$9.179 \cdot 10^{-1}$	$9.521 \cdot 10^{-1}$	$9.767 \cdot 10^{-1}$
200.0	$7.582 \cdot 10^{-1}$	$8.691 \cdot 10^{-1}$	$9.318 \cdot 10^{-1}$	$9.718 \cdot 10^{-1}$	$9.999 \cdot 10^{-1}$	$1.018 \cdot 10^{0}$

T_e [eV]	Cu^{+}	Cu^{2+}	Cu^{3+}	Cu^{4+}	Cu^{5+}	Cu^{6+}
1.0	$1.435 \cdot 10^{-12}$	$1.282 \cdot 10^{-10}$	$6.027 \cdot 10^{-9}$	$1.700 \cdot 10^{-7}$	$3.150 \cdot 10^{-6}$	$3.978 \cdot 10^{-5}$
5.0	$9.563 \cdot 10^{-3}$	$3.918 \cdot 10^{-2}$	$9.608 \cdot 10^{-2}$	$1.462 \cdot 10^{-1}$	$2.158 \cdot 10^{-1}$	$2.700 \cdot 10^{-1}$
10.0	$9.771 \cdot 10^{-2}$	$2.096 \cdot 10^{-1}$	$3.230 \cdot 10^{-1}$	$4.304 \cdot 10^{-1}$	$5.241 \cdot 10^{-1}$	$6.104 \cdot 10^{-1}$
15.0	$2.167 \cdot 10^{-1}$	$3.743 \cdot 10^{-1}$	$5.178 \cdot 10^{-1}$	$6.481 \cdot 10^{-1}$	$7.639 \cdot 10^{-1}$	$8.687 \cdot 10^{-1}$
20.0	$3.249 \cdot 10^{-1}$	$5.172 \cdot 10^{-1}$	$6.831 \cdot 10^{-1}$	$8.284 \cdot 10^{-1}$	$9.600 \cdot 10^{-1}$	$1.077 \cdot 10^{0}$
25.0	$4.246 \cdot 10^{-1}$	$6.394 \cdot 10^{-1}$	$8.238 \cdot 10^{-1}$	$9.844 \cdot 10^{-1}$	$1.127 \cdot 10^{0}$	$1.254 \cdot 10^{0}$
30.0	$5.109 \cdot 10^{-1}$	$7.471 \cdot 10^{-1}$	$9.475 \cdot 10^{-1}$	$1.121 \cdot 10^{0}$	$1.273 \cdot 10^{0}$	$1.406 \cdot 10^{0}$
35.0	$5.917 \cdot 10^{-1}$	$8.454 \cdot 10^{-1}$	$1.059 \cdot 10^{0}$	$1.242 \cdot 10^{0}$	$1.404 \cdot 10^{0}$	$1.544 \cdot 10^{0}$
40.0	$6.637 \cdot 10^{-1}$	$9.343 \cdot 10^{-1}$	$1.161 \cdot 10^{0}$	$1.353 \cdot 10^{0}$	$1.520 \cdot 10^{0}$	$1.667 \cdot 10^{0}$
45.0	$7.300 \cdot 10^{-1}$	$1.016 \cdot 10^{0}$	$1.254 \cdot 10^{0}$	$1.452 \cdot 10^{0}$	$1.627 \cdot 10^{0}$	$1.778 \cdot 10^{0}$
50.0	$7.934 \cdot 10^{-1}$	$1.094 \cdot 10^{0}$	$1.341 \cdot 10^{0}$	$1.546 \cdot 10^{0}$	$1.727 \cdot 10^{0}$	$1.882 \cdot 10^{0}$
75.0	$1.054 \cdot 10^{0}$	$1.415 \cdot 10^{0}$	$1.698 \cdot 10^{0}$	$1.932 \cdot 10^{0}$	$2.130 \cdot 10^{0}$	$2.302 \cdot 10^{0}$
100.0	$1.263 \cdot 10^{0}$	$1.666 \cdot 10^{0}$	$1.977 \cdot 10^{0}$	$2.227 \cdot 10^{0}$	$2.439 \cdot 10^{0}$	$2.618 \cdot 10^{0}$
150.0	$1.591 \cdot 10^{0}$	$2.056 \cdot 10^{0}$	$2.400 \cdot 10^{0}$	$2.672 \cdot 10^{0}$	$2.897 \cdot 10^{0}$	$3.085 \cdot 10^{0}$
200.0	$1.848 \cdot 10^{0}$	$2.355 \cdot 10^{0}$	$2.720 \cdot 10^{0}$	$3.003 \cdot 10^{0}$	$3.234 \cdot 10^{0}$	$3.424 \cdot 10^{0}$

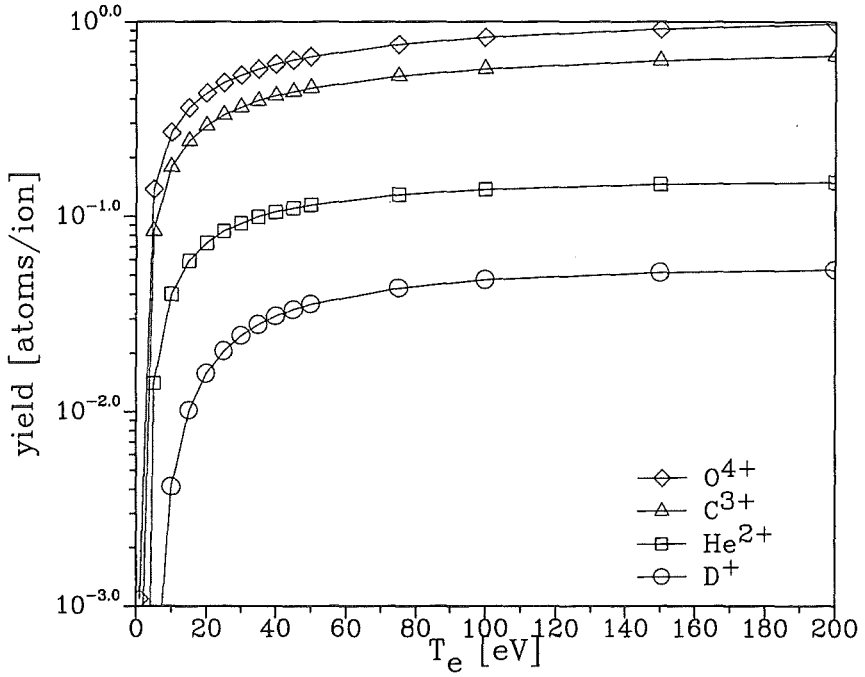


Figure C.15: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

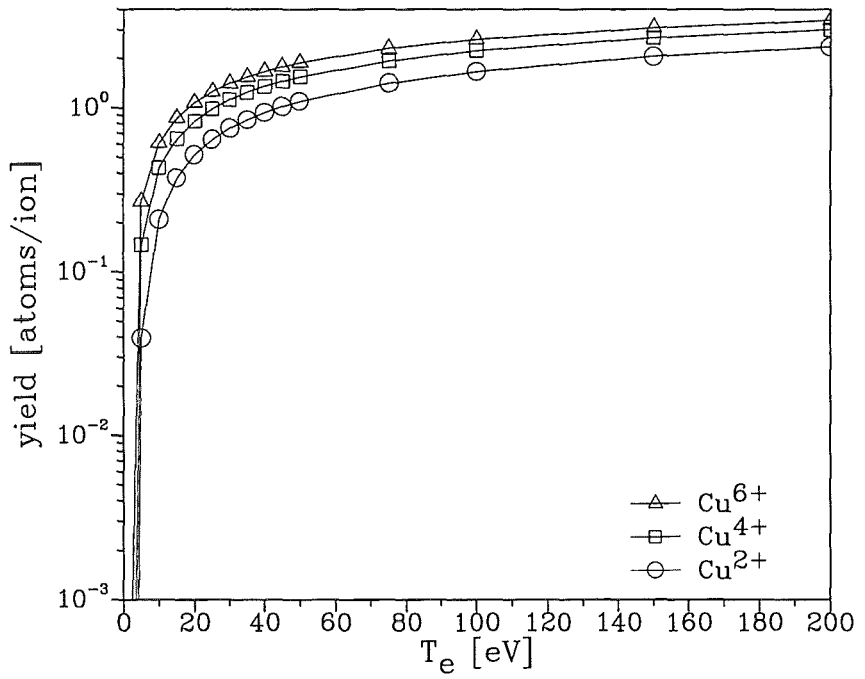


Figure C.16: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Molybdenum

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$1.765 \cdot 10^{-102}$	$8.541 \cdot 10^{-48}$	$9.870 \cdot 10^{-32}$	$1.751 \cdot 10^{-24}$	$7.955 \cdot 10^{-22}$
5.0	$3.140 \cdot 10^{-19}$	$2.378 \cdot 10^{-9}$	$1.199 \cdot 10^{-6}$	$1.977 \cdot 10^{-5}$	$2.674 \cdot 10^{-4}$
10.0	$8.132 \cdot 10^{-10}$	$2.202 \cdot 10^{-5}$	$3.209 \cdot 10^{-4}$	$1.312 \cdot 10^{-3}$	$5.609 \cdot 10^{-3}$
15.0	$5.657 \cdot 10^{-7}$	$2.994 \cdot 10^{-4}$	$1.790 \cdot 10^{-3}$	$5.305 \cdot 10^{-3}$	$1.474 \cdot 10^{-2}$
20.0	$1.117 \cdot 10^{-5}$	$1.052 \cdot 10^{-3}$	$4.293 \cdot 10^{-3}$	$1.076 \cdot 10^{-2}$	$2.299 \cdot 10^{-2}$
25.0	$5.867 \cdot 10^{-5}$	$2.255 \cdot 10^{-3}$	$7.240 \cdot 10^{-3}$	$1.654 \cdot 10^{-2}$	$2.984 \cdot 10^{-2}$
30.0	$1.680 \cdot 10^{-4}$	$3.759 \cdot 10^{-3}$	$1.026 \cdot 10^{-2}$	$2.190 \cdot 10^{-2}$	$3.612 \cdot 10^{-2}$
35.0	$3.508 \cdot 10^{-4}$	$5.395 \cdot 10^{-3}$	$1.309 \cdot 10^{-2}$	$2.679 \cdot 10^{-2}$	$4.142 \cdot 10^{-2}$
40.0	$6.081 \cdot 10^{-4}$	$7.059 \cdot 10^{-3}$	$1.565 \cdot 10^{-2}$	$3.139 \cdot 10^{-2}$	$4.574 \cdot 10^{-2}$
45.0	$9.348 \cdot 10^{-4}$	$8.702 \cdot 10^{-3}$	$1.792 \cdot 10^{-2}$	$3.556 \cdot 10^{-2}$	$4.992 \cdot 10^{-2}$
50.0	$1.320 \cdot 10^{-3}$	$1.027 \cdot 10^{-2}$	$2.014 \cdot 10^{-2}$	$3.921 \cdot 10^{-2}$	$5.346 \cdot 10^{-2}$
75.0	$3.685 \cdot 10^{-3}$	$1.657 \cdot 10^{-2}$	$2.839 \cdot 10^{-2}$	$5.400 \cdot 10^{-2}$	$6.637 \cdot 10^{-2}$
100.0	$6.060 \cdot 10^{-3}$	$2.101 \cdot 10^{-2}$	$3.384 \cdot 10^{-2}$	$6.419 \cdot 10^{-2}$	$7.453 \cdot 10^{-2}$
150.0	$9.582 \cdot 10^{-3}$	$2.659 \cdot 10^{-2}$	$4.035 \cdot 10^{-2}$	$7.739 \cdot 10^{-2}$	$8.428 \cdot 10^{-2}$
200.0	$1.188 \cdot 10^{-2}$	$2.986 \cdot 10^{-2}$	$4.393 \cdot 10^{-2}$	$8.553 \cdot 10^{-2}$	$8.969 \cdot 10^{-2}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$9.811 \cdot 10^{-14}$	$1.248 \cdot 10^{-11}$	$6.727 \cdot 10^{-10}$	$2.087 \cdot 10^{-8}$	$4.191 \cdot 10^{-7}$	$5.720 \cdot 10^{-6}$
5.0	$1.236 \cdot 10^{-3}$	$7.210 \cdot 10^{-3}$	$1.956 \cdot 10^{-2}$	$3.123 \cdot 10^{-2}$	$4.643 \cdot 10^{-2}$	$5.651 \cdot 10^{-2}$
10.0	$1.524 \cdot 10^{-2}$	$4.116 \cdot 10^{-2}$	$6.567 \cdot 10^{-2}$	$8.886 \cdot 10^{-2}$	$1.082 \cdot 10^{-1}$	$1.247 \cdot 10^{-1}$
15.0	$3.702 \cdot 10^{-2}$	$7.402 \cdot 10^{-2}$	$1.055 \cdot 10^{-1}$	$1.308 \cdot 10^{-1}$	$1.523 \cdot 10^{-1}$	$1.699 \cdot 10^{-1}$
20.0	$5.821 \cdot 10^{-2}$	$1.028 \cdot 10^{-1}$	$1.366 \cdot 10^{-1}$	$1.631 \cdot 10^{-1}$	$1.855 \cdot 10^{-1}$	$2.042 \cdot 10^{-1}$
25.0	$7.830 \cdot 10^{-2}$	$1.269 \cdot 10^{-1}$	$1.624 \cdot 10^{-1}$	$1.896 \cdot 10^{-1}$	$2.123 \cdot 10^{-1}$	$2.310 \cdot 10^{-1}$
30.0	$9.587 \cdot 10^{-2}$	$1.474 \cdot 10^{-1}$	$1.837 \cdot 10^{-1}$	$2.120 \cdot 10^{-1}$	$2.344 \cdot 10^{-1}$	$2.531 \cdot 10^{-1}$
35.0	$1.126 \cdot 10^{-1}$	$1.659 \cdot 10^{-1}$	$2.029 \cdot 10^{-1}$	$2.308 \cdot 10^{-1}$	$2.532 \cdot 10^{-1}$	$2.718 \cdot 10^{-1}$
40.0	$1.272 \cdot 10^{-1}$	$1.817 \cdot 10^{-1}$	$2.192 \cdot 10^{-1}$	$2.471 \cdot 10^{-1}$	$2.691 \cdot 10^{-1}$	$2.878 \cdot 10^{-1}$
45.0	$1.409 \cdot 10^{-1}$	$1.963 \cdot 10^{-1}$	$2.337 \cdot 10^{-1}$	$2.616 \cdot 10^{-1}$	$2.838 \cdot 10^{-1}$	$3.018 \cdot 10^{-1}$
50.0	$1.537 \cdot 10^{-1}$	$2.093 \cdot 10^{-1}$	$2.469 \cdot 10^{-1}$	$2.740 \cdot 10^{-1}$	$2.966 \cdot 10^{-1}$	$3.141 \cdot 10^{-1}$
75.0	$2.049 \cdot 10^{-1}$	$2.612 \cdot 10^{-1}$	$2.974 \cdot 10^{-1}$	$3.234 \cdot 10^{-1}$	$3.438 \cdot 10^{-1}$	$3.596 \cdot 10^{-1}$
100.0	$2.430 \cdot 10^{-1}$	$2.974 \cdot 10^{-1}$	$3.322 \cdot 10^{-1}$	$3.562 \cdot 10^{-1}$	$3.750 \cdot 10^{-1}$	$3.891 \cdot 10^{-1}$
150.0	$2.983 \cdot 10^{-1}$	$3.483 \cdot 10^{-1}$	$3.787 \cdot 10^{-1}$	$3.991 \cdot 10^{-1}$	$4.144 \cdot 10^{-1}$	$4.253 \cdot 10^{-1}$
200.0	$3.369 \cdot 10^{-1}$	$3.819 \cdot 10^{-1}$	$4.084 \cdot 10^{-1}$	$4.256 \cdot 10^{-1}$	$4.377 \cdot 10^{-1}$	$4.459 \cdot 10^{-1}$

T_e [eV]	O^+	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$1.700 \cdot 10^{-13}$	$2.122 \cdot 10^{-11}$	$1.122 \cdot 10^{-9}$	$3.411 \cdot 10^{-8}$	$6.703 \cdot 10^{-7}$	$8.929 \cdot 10^{-6}$
5.0	$1.581 \cdot 10^{-3}$	$9.199 \cdot 10^{-3}$	$2.498 \cdot 10^{-2}$	$3.968 \cdot 10^{-2}$	$5.933 \cdot 10^{-2}$	$7.340 \cdot 10^{-2}$
10.0	$1.912 \cdot 10^{-2}$	$5.216 \cdot 10^{-2}$	$8.429 \cdot 10^{-2}$	$1.146 \cdot 10^{-1}$	$1.400 \cdot 10^{-1}$	$1.622 \cdot 10^{-1}$
15.0	$4.637 \cdot 10^{-2}$	$9.443 \cdot 10^{-2}$	$1.358 \cdot 10^{-1}$	$1.699 \cdot 10^{-1}$	$1.990 \cdot 10^{-1}$	$2.236 \cdot 10^{-1}$
20.0	$7.303 \cdot 10^{-2}$	$1.317 \cdot 10^{-1}$	$1.772 \cdot 10^{-1}$	$2.128 \cdot 10^{-1}$	$2.446 \cdot 10^{-1}$	$2.704 \cdot 10^{-1}$
25.0	$9.841 \cdot 10^{-2}$	$1.633 \cdot 10^{-1}$	$2.118 \cdot 10^{-1}$	$2.499 \cdot 10^{-1}$	$2.817 \cdot 10^{-1}$	$3.081 \cdot 10^{-1}$
30.0	$1.210 \cdot 10^{-1}$	$1.908 \cdot 10^{-1}$	$2.408 \cdot 10^{-1}$	$2.803 \cdot 10^{-1}$	$3.126 \cdot 10^{-1}$	$3.395 \cdot 10^{-1}$
35.0	$1.424 \cdot 10^{-1}$	$2.152 \cdot 10^{-1}$	$2.667 \cdot 10^{-1}$	$3.066 \cdot 10^{-1}$	$3.392 \cdot 10^{-1}$	$3.662 \cdot 10^{-1}$
40.0	$1.617 \cdot 10^{-1}$	$2.361 \cdot 10^{-1}$	$2.896 \cdot 10^{-1}$	$3.297 \cdot 10^{-1}$	$3.624 \cdot 10^{-1}$	$3.893 \cdot 10^{-1}$
45.0	$1.797 \cdot 10^{-1}$	$2.564 \cdot 10^{-1}$	$3.101 \cdot 10^{-1}$	$3.506 \cdot 10^{-1}$	$3.832 \cdot 10^{-1}$	$4.100 \cdot 10^{-1}$
50.0	$1.962 \cdot 10^{-1}$	$2.749 \cdot 10^{-1}$	$3.284 \cdot 10^{-1}$	$3.689 \cdot 10^{-1}$	$4.018 \cdot 10^{-1}$	$4.283 \cdot 10^{-1}$
75.0	$2.647 \cdot 10^{-1}$	$3.466 \cdot 10^{-1}$	$4.009 \cdot 10^{-1}$	$4.407 \cdot 10^{-1}$	$4.721 \cdot 10^{-1}$	$4.971 \cdot 10^{-1}$
100.0	$3.169 \cdot 10^{-1}$	$3.993 \cdot 10^{-1}$	$4.524 \cdot 10^{-1}$	$4.902 \cdot 10^{-1}$	$5.207 \cdot 10^{-1}$	$5.441 \cdot 10^{-1}$
150.0	$3.939 \cdot 10^{-1}$	$4.738 \cdot 10^{-1}$	$5.229 \cdot 10^{-1}$	$5.580 \cdot 10^{-1}$	$5.843 \cdot 10^{-1}$	$6.041 \cdot 10^{-1}$
200.0	$4.502 \cdot 10^{-1}$	$5.249 \cdot 10^{-1}$	$5.710 \cdot 10^{-1}$	$6.017 \cdot 10^{-1}$	$6.247 \cdot 10^{-1}$	$6.413 \cdot 10^{-1}$

T_e [eV]	Mo^+	Mo^{2+}	Mo^{3+}	Mo^{4+}	Mo^{5+}	Mo^{6+}
1.0	$5.927 \cdot 10^{-26}$	$2.433 \cdot 10^{-23}$	$4.645 \cdot 10^{-21}$	$4.903 \cdot 10^{-19}$	$3.367 \cdot 10^{-17}$	$1.642 \cdot 10^{-15}$
5.0	$1.495 \cdot 10^{-4}$	$1.495 \cdot 10^{-3}$	$8.594 \cdot 10^{-3}$	$2.184 \cdot 10^{-2}$	$3.997 \cdot 10^{-2}$	$7.777 \cdot 10^{-2}$
10.0	$1.284 \cdot 10^{-2}$	$4.545 \cdot 10^{-2}$	$1.061 \cdot 10^{-1}$	$1.572 \cdot 10^{-1}$	$2.299 \cdot 10^{-1}$	$2.910 \cdot 10^{-1}$
15.0	$5.296 \cdot 10^{-2}$	$1.365 \cdot 10^{-1}$	$2.292 \cdot 10^{-1}$	$3.171 \cdot 10^{-1}$	$3.992 \cdot 10^{-1}$	$4.752 \cdot 10^{-1}$
20.0	$1.098 \cdot 10^{-1}$	$2.230 \cdot 10^{-1}$	$3.433 \cdot 10^{-1}$	$4.504 \cdot 10^{-1}$	$5.492 \cdot 10^{-1}$	$6.378 \cdot 10^{-1}$
25.0	$1.710 \cdot 10^{-1}$	$3.134 \cdot 10^{-1}$	$4.465 \cdot 10^{-1}$	$5.699 \cdot 10^{-1}$	$6.821 \cdot 10^{-1}$	$7.824 \cdot 10^{-1}$
30.0	$2.306 \cdot 10^{-1}$	$3.936 \cdot 10^{-1}$	$5.409 \cdot 10^{-1}$	$6.757 \cdot 10^{-1}$	$7.994 \cdot 10^{-1}$	$9.099 \cdot 10^{-1}$
35.0	$2.864 \cdot 10^{-1}$	$4.671 \cdot 10^{-1}$	$6.274 \cdot 10^{-1}$	$7.756 \cdot 10^{-1}$	$9.076 \cdot 10^{-1}$	$1.025 \cdot 10^{0}$
40.0	$3.382 \cdot 10^{-1}$	$5.366 \cdot 10^{-1}$	$7.108 \cdot 10^{-1}$	$8.673 \cdot 10^{-1}$	$1.006 \cdot 10^{0}$	$1.130 \cdot 10^{0}$
45.0	$3.904 \cdot 10^{-1}$	$6.001 \cdot 10^{-1}$	$7.879 \cdot 10^{-1}$	$9.511 \cdot 10^{-1}$	$1.098 \cdot 10^{0}$	$1.228 \cdot 10^{0}$
50.0	$4.364 \cdot 10^{-1}$	$6.624 \cdot 10^{-1}$	$8.591 \cdot 10^{-1}$	$1.030 \cdot 10^{0}$	$1.181 \cdot 10^{0}$	$1.318 \cdot 10^{0}$
75.0	$6.404 \cdot 10^{-1}$	$9.242 \cdot 10^{-1}$	$1.162 \cdot 10^{0}$	$1.363 \cdot 10^{0}$	$1.541 \cdot 10^{0}$	$1.698 \cdot 10^{0}$
100.0	$8.086 \cdot 10^{-1}$	$1.137 \cdot 10^{0}$	$1.404 \cdot 10^{0}$	$1.630 \cdot 10^{0}$	$1.826 \cdot 10^{0}$	$1.997 \cdot 10^{0}$
150.0	$1.080 \cdot 10^{0}$	$1.477 \cdot 10^{0}$	$1.792 \cdot 10^{0}$	$2.048 \cdot 10^{0}$	$2.267 \cdot 10^{0}$	$2.457 \cdot 10^{0}$
200.0	$1.301 \cdot 10^{0}$	$1.751 \cdot 10^{0}$	$2.095 \cdot 10^{0}$	$2.373 \cdot 10^{0}$	$2.609 \cdot 10^{0}$	$2.809 \cdot 10^{0}$

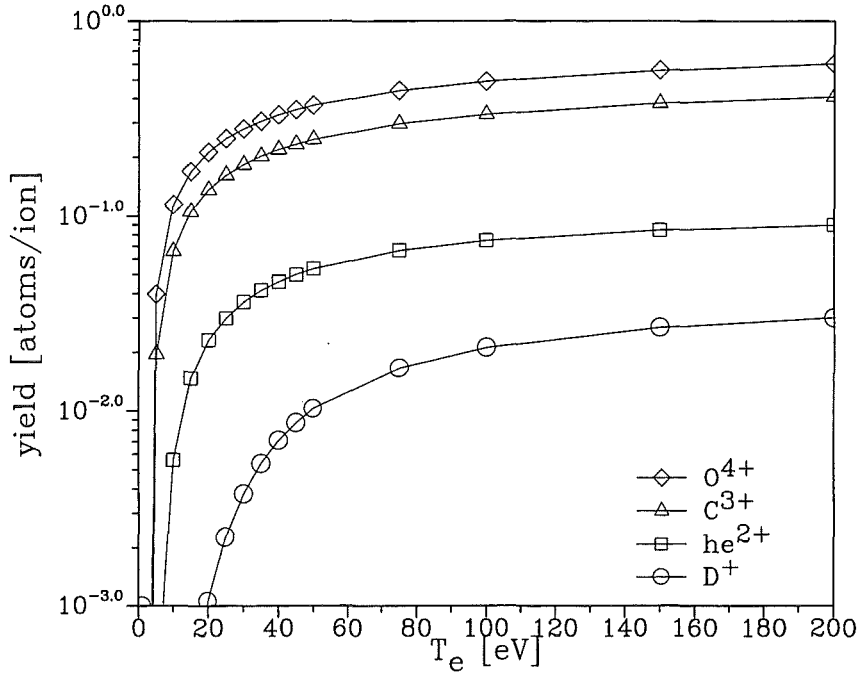


Figure C.17: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

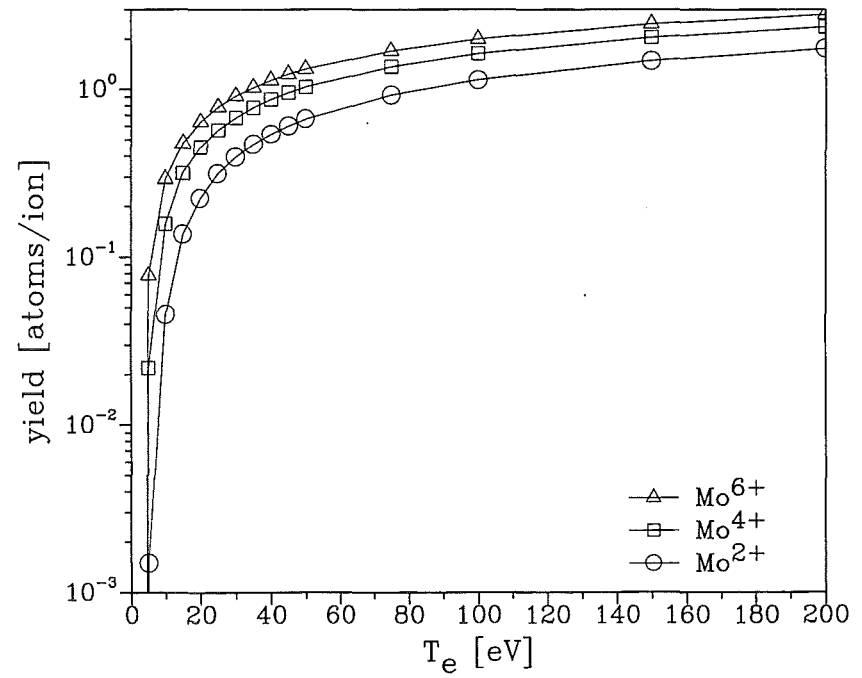


Figure C.18: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

- Tungsten

T_e [eV]	H^+	D^+	T^+	He^+	He^2+
1.0	$8.153 \cdot 10^{-275}$	$2.338 \cdot 10^{-125}$	$5.034 \cdot 10^{-80}$	$5.402 \cdot 10^{-59}$	$3.487 \cdot 10^{-55}$
5.0	$1.001 \cdot 10^{-51}$	$2.304 \cdot 10^{-23}$	$5.431 \cdot 10^{-15}$	$3.680 \cdot 10^{-11}$	$2.647 \cdot 10^{-9}$
10.0	$3.591 \cdot 10^{-25}$	$1.172 \cdot 10^{-11}$	$8.431 \cdot 10^{-8}$	$4.592 \cdot 10^{-6}$	$8.014 \cdot 10^{-5}$
15.0	$1.115 \cdot 10^{-16}$	$4.630 \cdot 10^{-8}$	$1.110 \cdot 10^{-5}$	$1.344 \cdot 10^{-4}$	$1.068 \cdot 10^{-3}$
20.0	$1.370 \cdot 10^{-12}$	$2.127 \cdot 10^{-6}$	$1.007 \cdot 10^{-4}$	$6.468 \cdot 10^{-4}$	$3.232 \cdot 10^{-3}$
25.0	$3.174 \cdot 10^{-10}$	$1.799 \cdot 10^{-5}$	$3.514 \cdot 10^{-4}$	$1.661 \cdot 10^{-3}$	$6.291 \cdot 10^{-3}$
30.0	$1.050 \cdot 10^{-8}$	$6.868 \cdot 10^{-5}$	$8.004 \cdot 10^{-4}$	$3.150 \cdot 10^{-3}$	$1.004 \cdot 10^{-2}$
35.0	$1.169 \cdot 10^{-7}$	$1.724 \cdot 10^{-4}$	$1.443 \cdot 10^{-3}$	$5.000 \cdot 10^{-3}$	$1.335 \cdot 10^{-2}$
40.0	$6.682 \cdot 10^{-7}$	$3.387 \cdot 10^{-4}$	$2.260 \cdot 10^{-3}$	$7.071 \cdot 10^{-3}$	$1.637 \cdot 10^{-2}$
45.0	$2.474 \cdot 10^{-6}$	$5.710 \cdot 10^{-4}$	$3.210 \cdot 10^{-3}$	$9.278 \cdot 10^{-3}$	$1.960 \cdot 10^{-2}$
50.0	$6.812 \cdot 10^{-6}$	$8.689 \cdot 10^{-4}$	$4.243 \cdot 10^{-3}$	$1.150 \cdot 10^{-2}$	$2.248 \cdot 10^{-2}$
75.0	$1.160 \cdot 10^{-4}$	$3.096 \cdot 10^{-3}$	$9.743 \cdot 10^{-3}$	$2.201 \cdot 10^{-2}$	$3.392 \cdot 10^{-2}$
100.0	$4.395 \cdot 10^{-4}$	$5.794 \cdot 10^{-3}$	$1.453 \cdot 10^{-2}$	$3.059 \cdot 10^{-2}$	$4.250 \cdot 10^{-2}$
150.0	$1.665 \cdot 10^{-3}$	$1.064 \cdot 10^{-2}$	$2.168 \cdot 10^{-2}$	$4.334 \cdot 10^{-2}$	$5.386 \cdot 10^{-2}$
200.0	$3.199 \cdot 10^{-3}$	$1.429 \cdot 10^{-2}$	$2.656 \cdot 10^{-2}$	$5.233 \cdot 10^{-2}$	$6.115 \cdot 10^{-2}$

T_e [eV]	C^+	C^{2+}	C^{3+}	C^{4+}	C^{5+}	C^{6+}
1.0	$1.543 \cdot 10^{-22}$	$5.988 \cdot 10^{-20}$	$8.473 \cdot 10^{-18}$	$6.598 \cdot 10^{-16}$	$3.342 \cdot 10^{-14}$	$1.198 \cdot 10^{-12}$
5.0	$5.078 \cdot 10^{-5}$	$6.454 \cdot 10^{-4}$	$3.008 \cdot 10^{-3}$	$6.301 \cdot 10^{-3}$	$1.396 \cdot 10^{-2}$	$2.022 \cdot 10^{-2}$
10.0	$2.587 \cdot 10^{-3}$	$1.133 \cdot 10^{-2}$	$2.466 \cdot 10^{-2}$	$3.780 \cdot 10^{-2}$	$4.978 \cdot 10^{-2}$	$5.952 \cdot 10^{-2}$
15.0	$1.012 \cdot 10^{-2}$	$2.928 \cdot 10^{-2}$	$4.782 \cdot 10^{-2}$	$6.402 \cdot 10^{-2}$	$7.882 \cdot 10^{-2}$	$9.259 \cdot 10^{-2}$
20.0	$2.043 \cdot 10^{-2}$	$4.650 \cdot 10^{-2}$	$6.822 \cdot 10^{-2}$	$8.791 \cdot 10^{-2}$	$1.037 \cdot 10^{-1}$	$1.183 \cdot 10^{-1}$
25.0	$3.166 \cdot 10^{-2}$	$6.218 \cdot 10^{-2}$	$8.620 \cdot 10^{-2}$	$1.068 \cdot 10^{-1}$	$1.244 \cdot 10^{-1}$	$1.390 \cdot 10^{-1}$
30.0	$4.246 \cdot 10^{-2}$	$7.603 \cdot 10^{-2}$	$1.025 \cdot 10^{-1}$	$1.240 \cdot 10^{-1}$	$1.417 \cdot 10^{-1}$	$1.564 \cdot 10^{-1}$
35.0	$5.267 \cdot 10^{-2}$	$8.947 \cdot 10^{-2}$	$1.168 \cdot 10^{-1}$	$1.384 \cdot 10^{-1}$	$1.567 \cdot 10^{-1}$	$1.718 \cdot 10^{-1}$
40.0	$6.275 \cdot 10^{-2}$	$1.011 \cdot 10^{-1}$	$1.295 \cdot 10^{-1}$	$1.514 \cdot 10^{-1}$	$1.697 \cdot 10^{-1}$	$1.850 \cdot 10^{-1}$
45.0	$7.195 \cdot 10^{-2}$	$1.118 \cdot 10^{-1}$	$1.408 \cdot 10^{-1}$	$1.631 \cdot 10^{-1}$	$1.815 \cdot 10^{-1}$	$1.969 \cdot 10^{-1}$
50.0	$8.084 \cdot 10^{-2}$	$1.217 \cdot 10^{-1}$	$1.514 \cdot 10^{-1}$	$1.737 \cdot 10^{-1}$	$1.923 \cdot 10^{-1}$	$2.075 \cdot 10^{-1}$
75.0	$1.180 \cdot 10^{-1}$	$1.627 \cdot 10^{-1}$	$1.925 \cdot 10^{-1}$	$2.153 \cdot 10^{-1}$	$2.335 \cdot 10^{-1}$	$2.485 \cdot 10^{-1}$
100.0	$1.476 \cdot 10^{-1}$	$1.927 \cdot 10^{-1}$	$2.227 \cdot 10^{-1}$	$2.450 \cdot 10^{-1}$	$2.626 \cdot 10^{-1}$	$2.768 \cdot 10^{-1}$
150.0	$1.926 \cdot 10^{-1}$	$2.365 \cdot 10^{-1}$	$2.652 \cdot 10^{-1}$	$2.859 \cdot 10^{-1}$	$3.020 \cdot 10^{-1}$	$3.146 \cdot 10^{-1}$
200.0	$2.256 \cdot 10^{-1}$	$2.677 \cdot 10^{-1}$	$2.945 \cdot 10^{-1}$	$3.134 \cdot 10^{-1}$	$3.279 \cdot 10^{-1}$	$3.391 \cdot 10^{-1}$

T_e [eV]	O^{+}	O^{2+}	O^{3+}	O^{4+}	O^{5+}	O^{6+}
1.0	$1.822 \cdot 10^{-19}$	$5.048 \cdot 10^{-17}$	$5.354 \cdot 10^{-15}$	$3.185 \cdot 10^{-13}$	$1.239 \cdot 10^{-11}$	$3.403 \cdot 10^{-10}$
5.0	$1.594 \cdot 10^{-4}$	$1.603 \cdot 10^{-3}$	$5.683 \cdot 10^{-3}$	$1.349 \cdot 10^{-2}$	$2.318 \cdot 10^{-2}$	$2.877 \cdot 10^{-2}$
10.0	$4.907 \cdot 10^{-3}$	$1.956 \cdot 10^{-2}$	$3.618 \cdot 10^{-2}$	$5.396 \cdot 10^{-2}$	$7.122 \cdot 10^{-2}$	$8.625 \cdot 10^{-2}$
15.0	$1.654 \cdot 10^{-2}$	$4.263 \cdot 10^{-2}$	$6.929 \cdot 10^{-2}$	$9.150 \cdot 10^{-2}$	$1.109 \cdot 10^{-1}$	$1.284 \cdot 10^{-1}$
20.0	$3.111 \cdot 10^{-2}$	$6.593 \cdot 10^{-2}$	$9.650 \cdot 10^{-2}$	$1.219 \cdot 10^{-1}$	$1.434 \cdot 10^{-1}$	$1.628 \cdot 10^{-1}$
25.0	$4.588 \cdot 10^{-2}$	$8.724 \cdot 10^{-2}$	$1.202 \cdot 10^{-1}$	$1.474 \cdot 10^{-1}$	$1.706 \cdot 10^{-1}$	$1.909 \cdot 10^{-1}$
30.0	$5.972 \cdot 10^{-2}$	$1.059 \cdot 10^{-1}$	$1.409 \cdot 10^{-1}$	$1.701 \cdot 10^{-1}$	$1.942 \cdot 10^{-1}$	$2.151 \cdot 10^{-1}$
35.0	$7.372 \cdot 10^{-2}$	$1.230 \cdot 10^{-1}$	$1.602 \cdot 10^{-1}$	$1.899 \cdot 10^{-1}$	$2.148 \cdot 10^{-1}$	$2.360 \cdot 10^{-1}$
40.0	$8.632 \cdot 10^{-2}$	$1.385 \cdot 10^{-1}$	$1.770 \cdot 10^{-1}$	$2.076 \cdot 10^{-1}$	$2.331 \cdot 10^{-1}$	$2.546 \cdot 10^{-1}$
45.0	$9.864 \cdot 10^{-2}$	$1.524 \cdot 10^{-1}$	$1.924 \cdot 10^{-1}$	$2.238 \cdot 10^{-1}$	$2.496 \cdot 10^{-1}$	$2.712 \cdot 10^{-1}$
50.0	$1.101 \cdot 10^{-1}$	$1.660 \cdot 10^{-1}$	$2.065 \cdot 10^{-1}$	$2.382 \cdot 10^{-1}$	$2.644 \cdot 10^{-1}$	$2.863 \cdot 10^{-1}$
75.0	$1.590 \cdot 10^{-1}$	$2.207 \cdot 10^{-1}$	$2.638 \cdot 10^{-1}$	$2.965 \cdot 10^{-1}$	$3.231 \cdot 10^{-1}$	$3.450 \cdot 10^{-1}$
100.0	$1.980 \cdot 10^{-1}$	$2.623 \cdot 10^{-1}$	$3.061 \cdot 10^{-1}$	$3.389 \cdot 10^{-1}$	$3.654 \cdot 10^{-1}$	$3.867 \cdot 10^{-1}$
150.0	$2.582 \cdot 10^{-1}$	$3.240 \cdot 10^{-1}$	$3.672 \cdot 10^{-1}$	$3.989 \cdot 10^{-1}$	$4.243 \cdot 10^{-1}$	$4.445 \cdot 10^{-1}$
200.0	$3.037 \cdot 10^{-1}$	$3.686 \cdot 10^{-1}$	$4.103 \cdot 10^{-1}$	$4.410 \cdot 10^{-1}$	$4.645 \cdot 10^{-1}$	$4.832 \cdot 10^{-1}$

T_e [eV]	W^{+}	W^{2+}	W^{3+}	W^{4+}	W^{5+}	W^{6+}
1.0	$6.908 \cdot 10^{-34}$	$7.380 \cdot 10^{-31}$	$2.564 \cdot 10^{-28}$	$4.578 \cdot 10^{-26}$	$5.135 \cdot 10^{-24}$	$4.020 \cdot 10^{-22}$
5.0	$6.725 \cdot 10^{-6}$	$1.306 \cdot 10^{-4}$	$1.388 \cdot 10^{-3}$	$7.490 \cdot 10^{-3}$	$1.823 \cdot 10^{-2}$	$2.532 \cdot 10^{-2}$
10.0	$4.017 \cdot 10^{-3}$	$2.029 \cdot 10^{-2}$	$5.490 \cdot 10^{-2}$	$1.130 \cdot 10^{-1}$	$1.531 \cdot 10^{-1}$	$2.239 \cdot 10^{-1}$
15.0	$3.147 \cdot 10^{-2}$	$9.041 \cdot 10^{-2}$	$1.738 \cdot 10^{-1}$	$2.586 \cdot 10^{-1}$	$3.359 \cdot 10^{-1}$	$4.035 \cdot 10^{-1}$
20.0	$9.055 \cdot 10^{-2}$	$1.896 \cdot 10^{-1}$	$2.979 \cdot 10^{-1}$	$3.999 \cdot 10^{-1}$	$4.952 \cdot 10^{-1}$	$5.838 \cdot 10^{-1}$
25.0	$1.697 \cdot 10^{-1}$	$2.859 \cdot 10^{-1}$	$4.124 \cdot 10^{-1}$	$5.298 \cdot 10^{-1}$	$6.358 \cdot 10^{-1}$	$7.339 \cdot 10^{-1}$
30.0	$2.563 \cdot 10^{-1}$	$3.858 \cdot 10^{-1}$	$5.185 \cdot 10^{-1}$	$6.487 \cdot 10^{-1}$	$7.639 \cdot 10^{-1}$	$8.752 \cdot 10^{-1}$
35.0	$3.425 \cdot 10^{-1}$	$4.731 \cdot 10^{-1}$	$6.204 \cdot 10^{-1}$	$7.576 \cdot 10^{-1}$	$8.806 \cdot 10^{-1}$	$1.002 \cdot 10^{0}$
40.0	$4.238 \cdot 10^{-1}$	$5.597 \cdot 10^{-1}$	$7.139 \cdot 10^{-1}$	$8.585 \cdot 10^{-1}$	$9.938 \cdot 10^{-1}$	$1.120 \cdot 10^{0}$
45.0	$4.998 \cdot 10^{-1}$	$6.358 \cdot 10^{-1}$	$7.949 \cdot 10^{-1}$	$9.518 \cdot 10^{-1}$	$1.097 \cdot 10^{0}$	$1.230 \cdot 10^{0}$
50.0	$5.704 \cdot 10^{-1}$	$7.104 \cdot 10^{-1}$	$8.790 \cdot 10^{-1}$	$1.040 \cdot 10^{0}$	$1.192 \cdot 10^{0}$	$1.332 \cdot 10^{0}$
75.0	$8.563 \cdot 10^{-1}$	$1.017 \cdot 10^{0}$	$1.223 \cdot 10^{0}$	$1.418 \cdot 10^{0}$	$1.606 \cdot 10^{0}$	$1.778 \cdot 10^{0}$
100.0	$1.069 \cdot 10^{0}$	$1.259 \cdot 10^{0}$	$1.499 \cdot 10^{0}$	$1.729 \cdot 10^{0}$	$1.945 \cdot 10^{0}$	$2.140 \cdot 10^{0}$
150.0	$1.381 \cdot 10^{0}$	$1.646 \cdot 10^{0}$	$1.954 \cdot 10^{0}$	$2.234 \cdot 10^{0}$	$2.493 \cdot 10^{0}$	$2.729 \cdot 10^{0}$
200.0	$1.618 \cdot 10^{0}$	$1.960 \cdot 10^{0}$	$2.318 \cdot 10^{0}$	$2.643 \cdot 10^{0}$	$2.939 \cdot 10^{0}$	$3.202 \cdot 10^{0}$

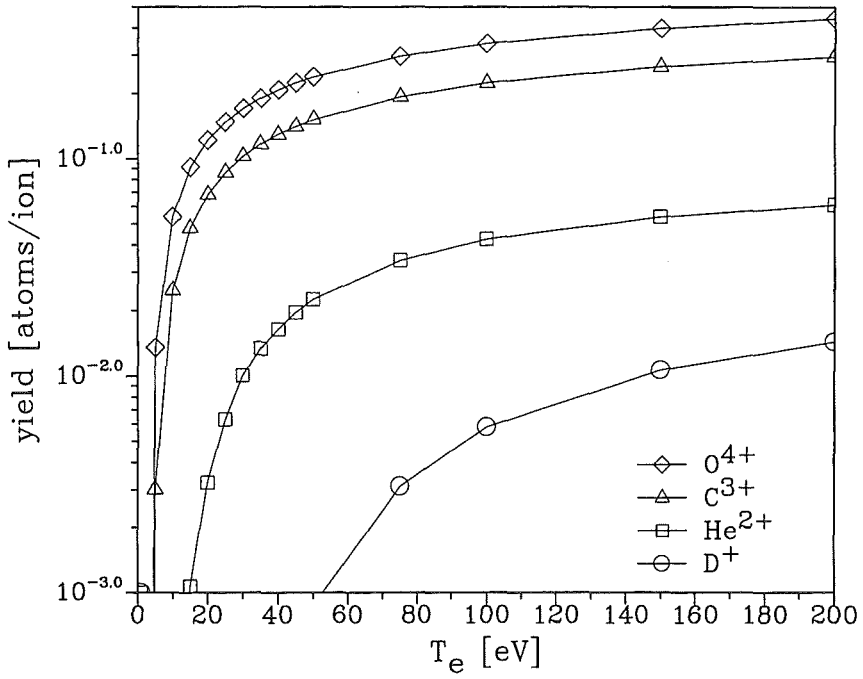


Figure C.19: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for D^+, He^2+, C^{3+}, O^{4+} impact. The lines are connection lines of the calculated values indicated by the symbols.

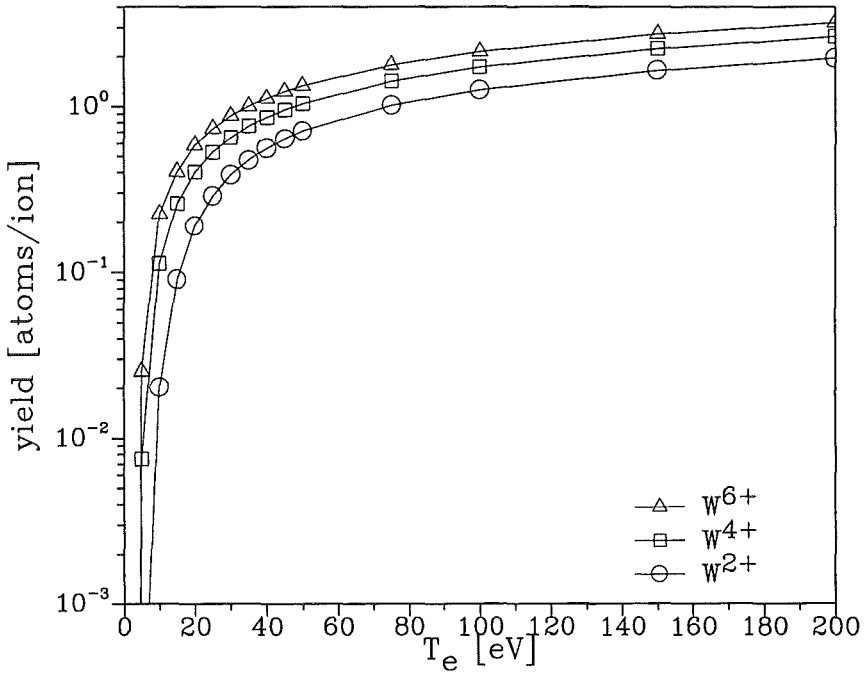


Figure C.20: Twiced averaged sputtering yield $\bar{Y}(T_e, q)$ according eqn.(2.114) for self sputtering. The lines are connection lines of the calculated values indicated by the symbols.

Appendix D

Organisation of C-Files and Databases

D.1 Index of C-Files, Header-and Make-Files

C -Source/Header-File	Description
Cpsi.c	standard ERO-TX: routine(s): Cpsi-handle(...)
bildreflection.c	standard ERO-TX: routine(s): bild-molecule(...)
bildtherm.c	standard ERO-TX: routine(s): bildtherm(...)
bildvert.c	standard ERO-TX: routine(s): bild-vert1(...)
c-chemie.c	standard ERO-TX: routine(s): change-status(...), chem-rate(...) chem-rate-prot(...), prozesses(...)
chainjob.c	standard ERO-TX: routine(s):chain-submitter(...)
chequer.c	standard ERO-TX: routine(s): cheque(...)
distrib.c	standard ERO-TX: routine(s): concentrate(...), powerload(...), marking(...), s-koord(...), s-angle(...), s-energy(...), NoElem(...), ElemNo(...), flux-to-cell(...)
emission.c	standard ERO-TX: routine(s): load-emission(...), free-emission(...) emission(..)
extrafieldsII.c	standard ERO-TX: routine(s): B-field(...), E-field(...)
extrafieldsIII.c	non-standard ERO-TX utility: routine(s): B-field(...),E-field(...)
flow.c	standard ERO-TX: routine(s): molecular-flow(...)
impurity.c	standard ERO-TX: routine(s): impurity-percent(...), fiqi(...), z-effective(...), q-plasma(...)
limgrid.c	standard ERO-TX: routine(s): s-grid-maker(...),s-grid-free(...), v-grid-maker(...), v-grid-free(...), volume-out(...), surface-in(...), surface-out(...), B-field1(...), emission-out(...)
local-cs.c	standard ERO-TX: routine(s): local-flow-velocity(...), relaxation-times(...)
local-dl.c	standard ERO-TX: routine(s): local-dl(...)
main.c	standard ERO-TX: routine(s): main(...)
mathhlp.c	standard ERO-TX: routine(s): runden(...), maximum(...), minimum(...), maxwell-energy(...), win-gleichverteilt-for(...), win-cosineverteilt-for(...), gauss-ran(...)[1], gauss-ran-ltd(...), plumin-ran(...), determinate(...), G(...), Phi-G(...), ran1-(...)[1]
move2sputter.c	non-standard ERO-TX utility: routine(s): move-ctrl(...)
move-ctrl.c	standard ERO-TX: routine(s): move-ctrl(...)

[1] this routines are taken from the Numerical Recipes [45]

C -Source/Header-File	Description
nT.c	standard ERO-TX: routine(s): ne(...), Te(...), Ti(...), grad-Te(...), grad-Ti(...)
nr.c	standard ERO-TX: routine(s): gaussj(...)[2]erf1(...)[2], gammp(...)[2], gser(...)[2], gcf(...)[2], gammln(...)[2]
nrutil.c	standard ERO-TX: routine(s): ref. Numerical Recipes[2]
param.c	standard ERO-TX: routine(s): datapool1(...)
part-go.c	standard ERO-TX: routine(s): part-go1(...), part-go-chem(...)
psi.c	standard ERO-TX: routine(s): psi-handle(...)
rates.c	standard ERO-TX: routine(s): rates(...), l-ratecoeff(...)
read-reflection.c	standard ERO-TX: routine(s): read-reflection(...)
reflector.c	standard ERO-TX: routine(s): reflect-mker(...), Rprobability(...)
si-chemie.c	standard ERO-TX: routine(s): si-chem-rate(...), si-chem-rate-prot(...), prozesses(...)
sort-in.c	standard ERO-TX: routine(s): sort-in(...)
sputt.c	non-standard ERO-TX utility: routine(s): integral(...), integrand1(...), integrand2(...), phys-sputt(...),chem-sputt(...), subl-sputt(...), res-sputt(...)
sputtern.c	standard ERO-TX: routine(s): local-yield(...), rt-phys-sputt(...)
thermal-force.c	standard ERO-TX: routine(s): thermal-force(...)
const.h	standard ERO-TX: standard constants used in ERO-TX TRUE, FALSE,INF-, RAD, GRAD, E-V, V-E, EPSILON0, MPROTON, EELEKTRON, NM,
distrib.h	standard ERO-TX: prototypes for: concentrate(...), powerload(...), marking(...), NoElem(...), ElemNo(...), bild-molecule(...), bild-reflection(...), bild-vert1(...), bildtherm(...), s-koord(...), s-angle(...), smolecular-flow(...), s-energy(...), rates(...), l-ratecoeff(...), read-reflection(...), flux-to-cell(...)
emission.h	standard ERO-TX: prototypes for: load-emission(...), free-emission(...) emission(..)
lim-go.h	standard ERO-TX: prototypes for: move-ctrl(...), part-go1(...), part-go-chem(...), psi-handle(...), Cpsi-handle(...), B-field(...), E-field(...), change-status(...), chem-rate(...), chem-rate-prot(...), prozesses(...), si-chem-rate(...), si-chem-rate-prot(...), si-prozesses(...), cheque(...), fiqi(...), impurity-percent(...), z-effective(...), q-plasma(...),
limgrid.h	standard ERO-TX: prototypes for: s-grid-maker(...), s-grid-free(...), surface-out(...), surface-in(...), emission(...) v-grid-maker(...), v-grid-free(...), volume-out(...), B-field1(...),
local-cs.h	standard ERO-TX: prototypes for: local-flow-velocity(...), local-dl(...), relaxation-times(...), thermal-force(...),
main.h	standard ERO-TX: prototypes for: datapool1(...), chain-submitter(...),
mathhlp.h	standard ERO-TX: prototypes for: runden(...), maximum(...), minimum(...), maxwell-energy(...), win-gleichverteilt-for(...), win-cosineverteilt-for(...), ran1-(...), plumin-ran(...), gauss-ran(...), gauss-ran-ltd(...), G(...), Phi-G(...), determinate(...)

[2]this routines are taken from the Numerical Recipes [45]

Header-File	Description
nT.h	standard ERO-TX: prototypes for: ne(...), Te(...), Ti(...), grad-Te(...), grad-Ti(...)
nr.h	standard ERO-TX: prototypes for: gaussj(...)gammp(...), gser(...), gcf(...), gammln(...), erf1(...)
nrutil.h	standard ERO-TX: prototypes for: refer to Numerical Recipes
reflector.h	standard ERO-TX: prototypes for: Rprobability(...), reflect-mker(...),
sort-in.h	standard ERO-TX: prototypes for: sort-in(...)
sputt.h	standard ERO-TX: prototypes for: integral(...), integrand1(...), integrand2(...), phys-sputt(...), chem-sputt(...), res-sputt(...), subl-sputt(...),
sputtern.h	standard ERO-TX: prototypes for: local-yield(...), rt-phys-sputt(...),
struct.h	standard ERO-TX: structure prototypes for: struct FileDialogData struct LoopDialogData struct NumericDialogData struct PlasmaDialogData struct ElementDialogData struct LimiterDialogData struct TeilchenDialogData struct MarkerDialogData struct ChemieDialogData struct SurfaceGridData struct VolumenGridData struct V-TraceData struct TimeGridData struct Particle Struct struct Ion struct Numeric struct Plasma struct nTdata struct Magnetic struct Electric struct Diffusion struct Ort struct Spezies struct BRotator struct V-PM struct BField struct EField struct Force struct PConst struct Contents struct SurfaceCell struct VolumeCell struct Base struct VolumeAll struct Emission struct Chemie struct ChemConst struct MaterialData struct SputterData struct ReflectionData struct LostParticle

Makefile	Description
make.Efeld	non-standard ERO-TX utility: makefile for inclusion of the parallel electric field due to the source itself
makefile	standard ERO-TX: standard makefile

D.2 Index of Supplied Sputtering Data

All data of the report "Sputtering Data" [21] have been transfered to a file. In addition the sputtering yields of H,C,O,Si on Si have been calculated with TRIM.VMC and fitted with the revised Bodhansky ref. eqn.(2.112) formula to specific needs.

FILE	DESCRIPTION
sput.kor	data as published (at best knowledge) including density
sputsi1ev.dat	same as sput.kor, but for sputtering of: D,C,O,Si → Si surface binding energy was set to $E_{sb}(Si) = 1eV$ (calculation done with TRIM.VMC and then fitted with revised Bodhansky formula
sputsi2ev.dat	same as sput.kor, but for sputtering of: D,C,O,Si → Si surface binding energy was set to $E_{sb}(Si) = 2eV$ (calculation done with TRIM.VMC and then fitted with revised Bodhansky formula

D.3 Index of Supplied Integral Sputtering Yield Data

The integral sputtering yield has been calculated for all projectile/target-combinations given in [21] according to eqn.(2.114) up to charge state $q = 4$, 6 and for electron temperatures $T_e = 5 - 200eV$ (ref. to Appendix C for the values of the most important elements in fusion physics).

FILE	DESCRIPTION
i-sputt.dat	integral yield for all elements given in Eckstein report up to charge state 4 and electron temperature $Te = 200eV$ (but integration only up to $E = 900eV$)
integral.all	integral yield for all elements given in Eckstein report (projectiles H,D,T,C,O, and self sputtering) up to charge state 6 and electron temperature $T_e = 200eV$ (integration up to $E = 3600eV$)
sputt-si1ev.all	same as previous, but sputtering D,C,O,Si → Si with $E_{sb}(Si) = 1eV$
sputt-si2ev.all	same as previous, but sputtering D,C,O,Si → Si with $E_{sb}(Si) = 2eV$

D.4 Index of Supplied Reflection Data

The reflection yield for the following projectile/target combinations was calculated with TRIM.VMC.

FILE	Description
-b-on-b.dat	Reflection of boron on boron
-c-on-c.dat	Reflection of carbon on carbon
-c-on-w.dat	Reflection of carbon on tungsten
-c-on-cu.dat	Reflection of carbon on copper
-c-on-fe.dat	Reflection of carbon on iron
-c-on-mo.dat	Reflection of carbon on molybdenum
-c-on-si.dat	Reflection of carbon on silicon
-w-on-c.dat	Reflection of tungsten on carbon
-w-on-w.dat	Reflection of tungsten on tungsten
cu-on-c.dat	Reflection of copper on carbon
cu-on-si.dat	Reflection of copper on silicon
fe-on-c.dat	Reflection of iron on carbon
fe-on-si.dat	Reflection of iron on silicon
mo-on-c.dat	Reflection of molybdenum on carbon
mo-on-mo.dat	Reflection of molybdenum on molybdenum
si-on-c.dat	Reflection of silicon on carbon
si-on-cu.dat	Reflection of silicon on copper
si-on-fe.dat	Reflection of silicon on iron
si-on-si.dat	Reflection of silicon on silicon

D.5 Index of Supplied Ionization Rate Coefficients

The data to calculate the ionization rate coefficients published by Bell et al., [6], [30],[27] have been transfered to files. In addition the ionization rates of molybdenum and tungsten have been calculated from a Lotz-formulae and fitted to the rate coefficient scheme of Bell et al. given by eqns.(3.1,3.2) (ref. Appendix B).

available elements			
-b.dat	-c.dat	-f.dat	-h.dat
-k.dat	-n.dat	-o.dat	-p.dat
-s.dat	-v.dat	-w.dat	al.dat
ar.dat	ba.dat	be.dat	ca.dat
cd.dat	cl.dat	co.dat	cr.dat
cs.dat	fe.dat	he.dat	hg.dat
kr.dat	li.dat	mg.dat	mn.dat
mo.dat	na.dat	ne.dat	ni.dat
rb.dat	sc.dat	si.dat	sr.dat
ti.dat	xe.dat	zn.dat	

Appendix E

Final Remarks

The following is a list of further information, which does not fit in any previous chapters and contains also some comments on bugs and other flies:

E.0.1 Physical Information

- Chemical erosion of carbon due to oxygen is dealt in that way, that each impinging oxygen ion is assumed to form a volatile carbonmonoxyde molecule (CO) with a probability of 0.5. It is assumed that no effective redeposition takes place (each redeposited CO-molecule releases a carbon atom and forms again a new CO-molecule), therefore only the erosion effect due to background oxygen impact is considered.
- Depth profiling is possible using the `xxxfilm.tmp`-temporary file. Since this file could become very large on deposition dominated areas after several dozens of time steps it is compressed if more than 48 subsequent layers are reached. Compression is handled in a very simple way, namely that the lowest five layers are compressed to a single one.
- The bulk material is always assumed to have a resource capacity of 10^{100} atoms. If bookkeeping on the bulk material erosion is desired it should be specified as a marker of finite thickness (e.g. some microns).
- All particles released by a simulation particles impact on the material surface are collected in a special bin and transported away with the particles sputtered by the background plasma impact in the next time step.

E.0.2 Error Handling & Runtime Errors

- If a particle starts with direction into the colder and less dense plasma areas and therefore is not redeposited a special notice will be printed:
  `Warning : neutral "_c" to wall in cell (x= "15" y= "12")` where the values in quotation marks above are examples for illustration
- If a particle starts with but the ionization length is too long, so that the atom leaves the computational before it is ionized a special notice will be printed:
  `Warning : neutral "_c" out of box in cell (x= "15" y= "12")` where the values in quotation marks above are examples for illustration
- If an ion is supposed to be ionized to higher charge states than supplied in the database, the code will not recover. Usually the search routines searches then until infinity (or until cpu-time limit is exceeded).
- If `particle_sputtering` is swichted on, the user should be careful, that the relevant data exist, otherwise an runtime error will occur.
- If `particle_reflection` is switched on although the relevant data are not supplied to the database, the code will set the reflection coefficient R to $R \equiv 0$.

E.0.3 Numerical Information

- Although already mentioned it should be repeated, that the code is able to deal with sequences in electron density, ion and electron temperature, source rates from the external source and the length of the time steps. These values are updated after each time step, when the code is started newly.

- The density of different elements, compounds and alloys can differ quite significantly, therefore the definition of the parameter `interaction_depth` is somewhat unspecified. Since ERO-TEXTOR deals with numbers of atoms, rather than with thickness or density per cm^2, the problem is solved in that way that a mean density of $\rho = 5 \cdot 10^{15} \frac{\text{atoms}}{cm^2}$ is assumed. The specific interaction volume is thus merely another word for a specific number of atoms per cm^2 considered in the calculations.
- There are some hard wired numbers (in the `move_ctrl()`-routine), which should be mentioned here:

  - `win_max = cos(RAD * 80.)` - which is the cutoff angle if angle($B_{\text{Field}}, n_{\text{surface}}$) > $win_{\max}$
  - `LAMBDA = 15.` - the Coulomb-logarithm
  - `max_charge = 10` -maximum considerable charge state
  - `angle_c = 0.5` -mean angle of incidence of background plasma ions onto surface ($0.5 \equiv \cos(60^o)$)

  some other hard wired numbers not mentioned here result from the limitation on the amount of different species the code can deal with.
- Although not explicitly stated, the code (version 2.0) can only handle magnetic field vectors parallel to the x-axis ($\vec{B} \parallel \vec{e}_x$), yet. If other magnetic topologies are required the following routines have to be rewritten, respecting that the decay of these plasma parameter does not depend on the local z-value alone:

  - `double Te(...)`
  - `double Ti(...)`
  - `double ne(...)`
  - `void grad_Te(...)`
  - `void grad_Ti(...)`

  In addition the calculation of the z-value of the lcfs is no longer independent of the local x- and y-coordinates, so that this calculation has also be done locally.
- Since the ability to handle limiter surfaces of different shapes is restricted to some simple linear, spherical or toroidal shapes, it may be desirable from case to case to implement different geometries. ERO-TEXTOR calculates these surface properties in the following routines, which then must be changed:

  - `void move_ctrl(...)`
  - `int local_flow_velocity(...)`
  - `int local_dl(...)`
  - `E_field(...)`
  - `void bild_vert1(...)`
  - `void bild_therm(...)`
  - `void s_koord(...)`
  - `struct SurfaceCell ***s_grid_maker(...)`
  - `struct VolumeAll *v_grid_maker(...)`

Bibliography

[1] V.A. Abramov, Y.L. Igitkhanov, V.I. Pistunovich, V.A. Pozharov, *First Wall and Divertor Plate Sputtering in a Tokamak Reactor*, J.Nucl.Mat. 162-164 (1989) 462

[2] M.Baelmans, *Code Improvements and Applications of a Two-dimensional Edge Plasma Model for Toroidal Devices*, Ph.D. thesis, Universiteit Leuven 1994 and Jül-Rep. 2891, Forschungszentrum Jülich GmbH, Jülich (1994)

[3] M.I. Baskes, *Dynamical Calculation of Low Energy Hydrogen Reflection*, J.Nucl.Mat. 128-129(1984) 676-680

[4] R. Behrisch, *Evaporation for Heat Pulses on Ni, Mo, W and ATJ Graphite as First Wall Materials*, J.Nucl.Mat. 93&94 (1980) 498-504

[5] R. Behrisch, D. Naujoks, Contrib. Plasma Phys. 34 (1994) 293

[6] K.L. Bell, H.B. Gilbody, J.G. Hughes et al., *Atomic and Molecular Data For Fusion, Part I*, Rep. CLM-R216 (1982)

[7] C.K. Birdsall, *Plasma physics via computer simulation*, McGraw-Hill, Singapore (1985)

[8] B.J. Braams, *A Multi Fluid Code for Simulation of the Edge Plasma in Tokamaks*, Net-Rep EUR-FU IXII-80-87-68 (1987)

[9] S.I. Braginskii, *Transport Processes in Plasmas* in *Reviews of Plasma Physics*, Vol. I, ed. M.A. Leontovich, Consultants Bureau, New York (1965)

[10] J. Brooks, *Near Surface Sputtered Particle Transport for an Oblique Incidence Magnetic Field Plasma*, Phys.Fluids B2,8 (1990) 1858

[11] S. Chapman, Proc.Phys.Soc. 72 (1958) 353

[12] H. Chatam, D. Hills, R. Robertson, A. Gallagher, *Total and Partial Electron Collisional Ionization Cross Sections for CH_4, C_2H_6, SiH_4 and Si_2H_6*, J.Chem.Phys. 81 (1984) 1770

[13] R. Chodura, Phys. Fluids, 25 (1982) 1628

[14] R. Chodura, *Plasma Flow in the Sheath and the Presheath of a Scrape-Off Layer* in *Physics of Plasma-Wall Interactions in Controlled Fusion*, eds. D.E.Post and R.Behrisch, NATO ASI Series, Plenum Press, N.Y. (1986)

[15] P.Clausing, Ann. Physik 12 (1932) 961-989

[16] E.W.McDaniel, *Collision Phenomena in Ionized Gases*, J.Wiley & Sons, N.Y. (1964)

[17] B.B. Dayton, *Gas Flow Patterns at Entrance and Exit of Cylindrical Tubes*, Vac. Symp. Trans. (1956) 5-11

[18] W. Eckstein, D.B. Heifetz, *Data Sets for Hydrogen Reflection and their Use in Neutral Transport Calculations*, J.Nucl.Mat. 145-47 (1987) 332-328

[19] W. Eckstein, *Reflection*, Suppl. to Nucl.Fus. I (1991) 17-31

[20] W. Eckstein, *Computer Simulation of Ion-Solid Interaction*, Springer, Berlin (1991)

[21] W. Eckstein, C. Garcia-Rosales, J. Roth, W.Ottenberger, *Sputtering Data*, Max-Planck- Institut für Plasmaphysik, Report IPP 9/82 (1993)

[22] A.B. Ehrhardt, W.D. Langer, *Collisional Processes of Hydrocarbons in Hydrogen Plasmas*, PPPL Report, Princeton (1986)

[23] U. Fantz, *Spektroskopische Untersuchung und Modellierung von mikrowellen-angeregten Silanplasmen*, Ph.D. thesis, Universität Stuttgart (1995)

[24] M.S. Gordon, *Structure and Stability of SiH_4^+*, Chem.Phys.Lett. 59 (1978) 410

[25] A.A. Haasz, E.Vietzke, in *Physical Processes of the Interaction of Fusion Plasma with Solids*, eds. J. Roth, W. Hofer, Academic Press (1995) 135

[26] M.F.A Harrison, E.S. Hotston, NET Rep. EUR-FU/80/90-97, Max-Planck-Institut für Plasmaphysik, Garching (1989)

[27] M.J. Higgins, M.A. Lennon, K.L. Bell, et al., *Atomic and Molecular Data For Fusion, Part III*, Rep. CLM-R294 (1989)

[28] U.Kögler, in preparation: Ph.D. thesis, Universität Düsseldorf, Düsseldorf

[29] B. LaBombard, J.A. Goetz, I. Hutchinson, et al. *Experimental Investigation of Transport Phenomena in the Scrape-off Layer and Divertor* 12th. PSI conference, St. Raphael, France (1996), to be published in J.Nucl.Mat.

[30] M.A. Lennon, K.L. Bell, H.B. Gilbody, J.G. Hughes, et al., *Atomic and Molecular Data For Fusion, Part II*, Rep. CLM-R270 (1986)

[31] W. Lotz, *An Empirical Formula for the Electron Impact Ionization Cross Section*, Z.Physik, 220 (1969) 466

[32] D.C. Montgomery, D.A. Tidman, *Plasma Kinetic Theory*, McGraw-Hill, New York (1964)

[33] J.P. Morrison, J.C. Traeger, *Ionization and Dissoziation by Electron Impact III. CH_4 and SiH_4*, J.Mass Spec. Ion Phys. 11 (1973) 289

[34] D. Naujoks, R. Behrisch, J.P. Coad, L. deKock, *Material Transport by Erosion and Redeposition on Surface Probes in the Scrape-Off Layer of JET*, Nucl. Fus. 33 (1993) 581

[35] D. Naujoks, R. Behrisch, V. Philipps, B.Schweer, *Erosion and Redeposition on Carbon Probes in the Boundary Plasma of TEXTOR*, Plasma Phys. Control. Fus. 36 (1994) 2021

[36] D. Naujoks, J. Roth, K. Krieger, G. Lieder, M. Laux, *Erosion and Redeposition in the ASDEX Upgrade Divertor*, J.Nucl.Mat. 210 (1994) 43

[37] D. Naujoks, R. Behrisch, *Erosion and Redeposition at the Vessel Walls in Fusion Devices*, J.Nucl.Mat.220-222 (1995) 227

[38] D. Naujoks, W. Eckstein, *Sputtering of Plasma Facing Material by Simultaneous Bombardment with Carbon and Deuterium Ions*, J.Nucl.Mat.220-222 (1995) 993

[39] J. Neuhauser, W. Schneider, R. Wunderlich, K. Lackner, *Modeling of Impurity Flow in the Tokamak Scrape-Off-Layer*, Nucl.Fus. 24 (1984) 39-47

[40] J. Perrin, J.P.M. Schmitt, G. de Rosny, B. Drevillon, J.Huc, A. Lloret, *Dissociation Cross Sections of Silane and Disilane by Electron Impact*, Chem.Phys. 73 (1982) 383

[41] J. Perrin, J.F.M. Aarts, *Dissociative Excitation of SiH_4, SiD_4, Si_2H_6 and GeH_4 by 0-100eV Electron Impact*, Chem.Phys. 80 (1983) 351-365

[42] J. Perrin, O.Leroy, M.C. Bordage, *Cross Sections, Rate Constants and Transport Coefficients in Silane Plasma Chemistry*, Contrib. Plasma Phys. 36 (1996) 3-49

[43] V. Philipps, A. Pospieszczyk, *Observations on Impurity Release and Deposition Processes close to Limiter Surfaces in TEXTOR-94*, 12th. PSI conference, St. Raphael, France (1996), to be published in J.Nucl.Mat.

[44] A. Pospieszczyk, V. Philipps, E.Casarotto et al., *Chemical Erosion Measurements from various Carbon based Limiters and Coatings in TEXTOR*, Europhysics Conference Abstracts II (1995) 309

[45] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes*, Cambridge University Press, (1992)

[46] T.Pütz, *Entwicklung eines Finite Elemente Verfahrens zur Modellierung der Plasmaströmung in der Randschicht von Tokamaks*, Ph.D. thesis, Universität Düsseldorf, 1994 and Jül-Rep., Forschungszentrum Jülich GmbH, Jülich (1994)

[47] D. Reiser, *Eine Monte-Carlo-Methode zur Untersuchung von Verunreinigungsionen in Tokamak-Plasmen*, Jül-Rep. 3173, Forschungszentrum Jülich GmbH, Jülich (1996)

[48] J. Roth, E. Vietzke, A.A. Haasz, *Erosion of Graphite due to Particle Impact*, Suppl. to Nucl.Fus. I (1991) 63 -79

[49] J. Roth, D. Naujoks, K. Krieger, A. Field, G. Lieder, S. Hirsch, *Experimental Investigations of high-Z Materials in the ASDEX Upgrade Divertor*, J. Nucl. Mat.220-222 (1995) 231

[50] G. Schneider, *Modellierung eines ECR-Silanplasmas*, Diplomarbeit, Universität Stuttgart (1994)

[51] I.I. Sobelman, L.A. Vainshtein, E.A. Yukov, *Excitation of Atoms and Boradening of Spectral Lines*, Springer, Berlin (1981)

[52] L. Spitzer, jr., *Physics of Fully Ionized Gases*, J.Wiley&Sons, New York (1962)

[53] P.C. Stangeby, C.M. Farrell, *Description of the LIM and DIVIMP Codes* (1989)

[54] P.C. Stangeby, C.S. Pitcher, J.D. Elder, *The Nature of Plasma Fluxes to Surfaces Nearly Tangential to the Magnetic Field*, Nucl.Fus. 32 (1992) 2079

[55] P.C. Stangeby, *Large Probes in Tokamak Scrape-Off Plasmas. Analytic Model for the Collisionless Scrape-Off Plasma*, Phys. Fluids 28 (1985) 644-651

[56] P.C. Stangeby, *The plasma sheath* in *Physics of Plasma-Wall Interactions in Controlled Fusion*, eds. D.E.Post and R.Behrisch, NATO ASI Series, Plenum Press, N.Y. (1986)

[57] P.C. Stangeby, G.M. McCracken, *Plasma Boundary Phenomena*, Nucl.Fus. 30 (1990) 1225-1377

[58] P.C. Stangeby, J.D. Elder, *Impurity Retention by Divertors, Part I: One Dimensional Models*, Nucl.Fus. 35 (1995) 1391-1412

[59] T. Tabata, R. Ito, Y.Itikawa, W.Itoh, K. Morita, *Data on the Backscattering Coefficients of Light Ions from Solids*, Rep. IPPJ-AM 34, Institute of Plasma Physics, Naghoya (1984)

[60] M.Z. Tokar', *Plasma Behavior near strong Sources of Impurities*, Contrib.Plas.Phys. 36 (1996) 250

[61] B.A. Trubnikov, *Particle Interactions in a Fully Ionized Plasma* in *Reviews of Plasma Physics*, Vol. I, ed. M.A. Leontovich, Consultants Bureau, New York (1965)

[62] E. Vietzke, A. Refke, V. Philipps, M. Hennes, *Energy Distributions and Yields of Sputtered C_2 and C_3 Clusters* , 12th. PSI conference, St. Raphael, France (1996), to be published in J.Nucl.Mat.

[63] R. Wedell, Appl.Phys. A 35 (1984) 91

[64] M. Wutz, H. Adam, W. Walcher, *Theorie und Praxis der Vakuumtechnik*, Vieweg&Sohn, Braunschweig (1986)

[65] Y. Yamamura, Y. Itikawa, N. Itoh, Report IPPJ-AM-26, Nagoya (1983)

Forschungszentrum Jülich

Jül-3361
March 1997
ISSN 0944-2952