IPP-Report

MAX-PLANCK-INSTITUT
FÜR PLASMAPHYSIK

A. Mutzke

SDTrimSP-2D: Simulation of Particles Bombarding on a Two Dimensional Target Version 3.0

IPP 2024-31
November 2024

SDTrimSP-2D: Simulation of Particles Bombarding on a Two Dimensional Target Version 3.0

A. Mutzke

November 19, 2024

Contents

1. Introduction

The new program version is based on the one-dimensional TRIMM, TRIDYN and SDTrimSP (Static and Dynamic Trim for Sequential and Parallel computer) codes described in [2], [3], [4] and [5]. All these programs simulate the interaction of energetic particles with solids using the binary collision approximation.

The particle (projectile), impinging from outside on the solid (target), can be a neutral atom or an ion. If a projectile penetrates a solid target it will be scattered due to collisions with target atoms, which lead to an elastic energy loss and to a change of direction. In addition, the projectiles suffer an inelastic energy loss due to collisions with electrons. If the projectiles have lost all their energy, they are stuck in the target. Other possibilities are that the projectiles are backscattered after some collisions or that they are transmitted.

The energy lost by a projectile in a collision is transferred to a target atom (recoil), which itself can collide with other target atoms. If its energy is large enough the recoil can also leave the target (backward or transmission sputtering).

The movement of particles take place in three dimensions. The target is resolved in only one dimension in the program SDTrimSP (layers). It consists of layers, which have different thickness and varying composition.

In this new 2-D version the target is resolved in two dimensions (SDTrimSP-2D), allowing to account for effects of roughness on the scattering and sputtering and to model dynamical morphological changes [6], [10].

This report is an extension of the previous version (IPP 12/11).

2. Geometry

The main difference between SDTrimSP [1] and SDTrimSP-2D is the mapping of the target geometry. In SDTrimSP the target is one dimensional (Fig. 1a) and treated as a stack of layers in X direction, while Y and Z direction are taken as infinite. The code can be used in two different modes: In 'static mode' the target is fixed, while in 'dynamical mode' the thickness of layers is changeable.
The expansion into a second dimension makes it possible to describe target surface profiles and their dynamic change. The target in SDTrimSP-2D is therefor a regular grid in X- and Z-direction, while Y direction stays infinite, shown in Fig. 1b.

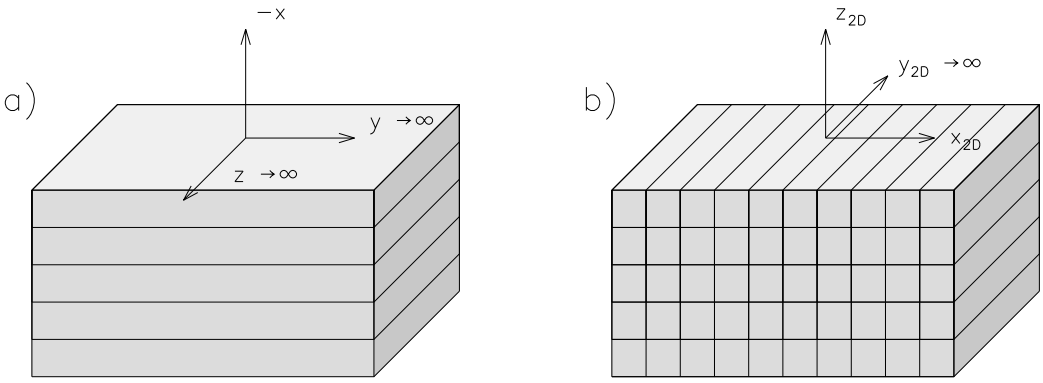


Figure 1: Geometry of one dimensional target a) and two dimensional target b)

A right-handed coordinate system (X_{2D}, Y_{2D}, Z_{2D}) is used in the new code to describe particle positions and cell geometry. Fig. 2 shows the definition of polar and azimuthal angles (blue) in comparison to the coordinate system used in SDTrimSP (red) and SDTrimSP-2D (black). The coordinate origin is fixed and not movable, like in SDTrimSP.

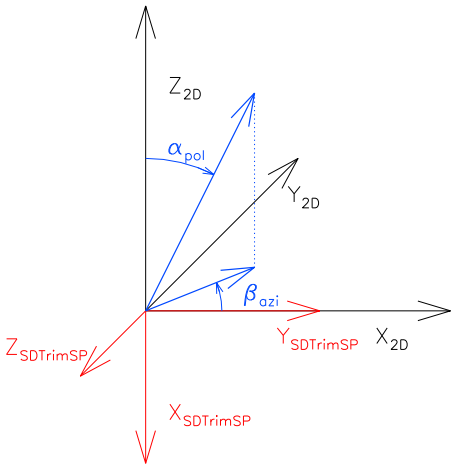


Figure 2: Definition of the coordinates, polar (α) and azimuthal (β) angle

Each cell in the two dimensional grid has 4 active surfaces and directions (left, right, bottom and top), that are described in Fig. 3. The surface-indices are counted from one to four in the program, while for better handling direction-indices are ranging from -2 to +2.

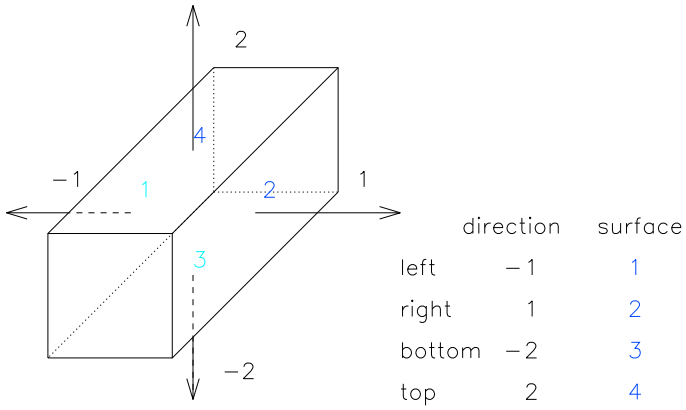


Figure 3: Indices of directions and surfaces of one cell used in the program.

To be able to treat the 2D problem additional features were introduced. There are two kinds of cells, depending on their location in the target labeled as active or passive. Passive cells, P, can not change their geometry and their volume is constant. If these cells swell or shrink, a relaxation is only possible due to volume- and particle-fluxes to neighboring cells, Fig. 4. The handling of this process is described in Chapter 4.

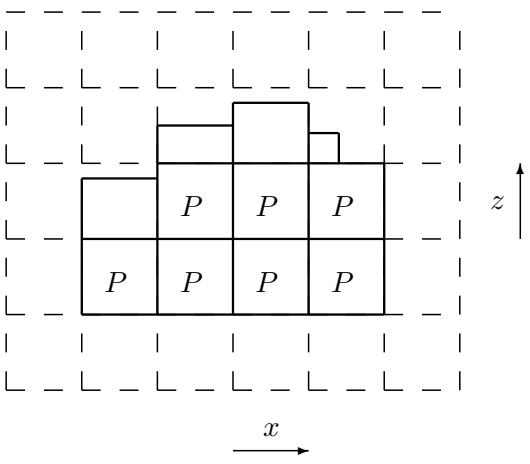


Figure 4: Target geometry with passive (P) or active grid cells.

Active cells, A, have at minimum one open boundary and their size can even be smaller than the mesh, thus their volume isn't constant in contradiction to passive cells. Active cells can only have one or two open boundaries. Some possible settings of active cells are shown in Fig. 5. Case (a) demonstrates an active surface cell with open boundaries to the upper and right side. Due to the neighboring active cell on the left side this boundary is not open. In case (b) the active cell is pinched between two passive cells. The resulting open boundaries are at top and bottom. Case (c) and (d) are examples for active cells with only one open boundary. This is possible, if the active cell has three or only one neighboring cell.
Growth and shrinkage processes of the target can only occur at open boundaries.

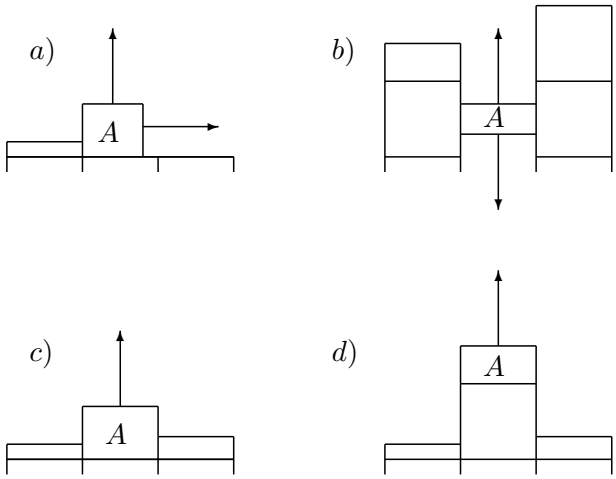


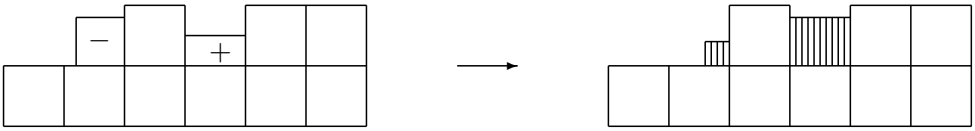
Figure 5: Examples of active cells (A) with two (a, b) and one (c, d) open boundaries.

During a simulation run cells might change their status from active to passive and backwards, depending on the movement of particles in the surrounding cells. Three different examples of active cell behaviour are shown in Fig. 6.

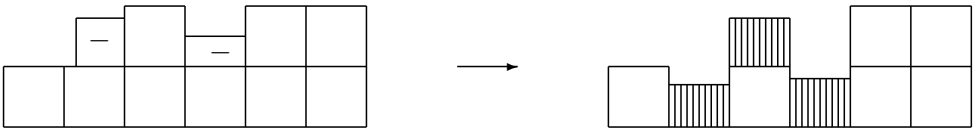
Case 1 is the simple volume change of active cells, that are able to swell (+) or shrink (-) inside the grid. If the reduction of the volume is larger than the cell-volume the cell is deleted and the next cell gets the status active (boundary) cell (case 2). If the new volume is greater than the standard cell-volume new active cells are created (case 3). This calculation is performed in subroutine '*change_vol*'.

A closer look into calculation of the cell volume change is given in the next chapter.

case 1: cells swell or shrink



case 2: cells shrink until deletion and neighboring cells shrink further



case 3: cells swell to maximum and new cells are created

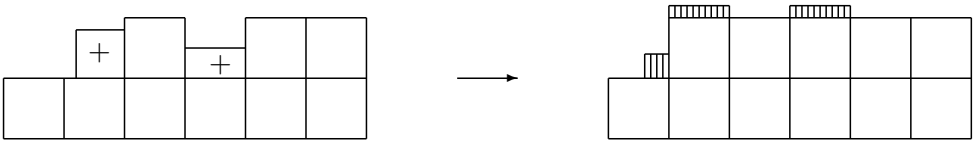


Figure 6: Change of geometry of active cells, right side: cells after relaxation.

3. Change of cell volume

Due to the constant flux of particles onto the target during a simulation, its composition and thus the volume of layers (cells) changes dynamically.

The number of implanted projectiles u_{im}, the number of recoils transferred into u_{in} and the number of atoms removed from a layer (cell) into another layer (cell) or sputtered u_{out} are determined after the bombardment of a target with n_r pseudo-particles (in program: nr_pproj). The probability that one incident particle changes the number of atoms in one layer or cell is N_0:

$$N_0(j) = \frac{u_{im}(j) + u_{in}(j) - u_{out}(j)}{n_r} \qquad \text{j ... species of atoms } (j = 1...ncp) \tag{3.1}$$

The composition of the target is updated after one fluence step, which is the quotient of fluence flc and number of fluence steps $maxhist$:

$$\Delta flc = \frac{flc}{maxhist} \qquad (\text{ in program: } fluc\_step)$$

The atomic composition of target determines the atomic fraction qu and the number of species is ncp. The sum of all atomic fractions qu is one.

$$\sum_{j=1}^{ncp} qu(j) = 1$$

In some cases it is necessary to define a maximum allowed atomic fraction. This maximum allowed atomic fraction qu_a^{max} can be defined in order to simulate local saturation phenomena, see [3].
If a maximum for one species (a) is given, the number of particles can be calculated accordingly:

$$N_m(a)^{max} = \frac{qu_a^{max}}{1 - qu_a^{max}} \cdot \Big(\sum_{j=1}^{ncp} N_0(j) - N_0(a) \Big)$$

$$N_0'(a) = max(N_m(a)^{max}, N_0(a))$$

If two maxima qu_a^{max} and qu_b^{max} are given for two species a and b, then the number of particles are :

$$\begin{aligned} N_0'(a) &= \frac{Q_a \cdot Q_b + Q_a}{1 - Q_a \cdot Q_b} \cdot \Big(\sum_{j=1}^{ncp} N_0(j) - N_0(a) - N_0(b) \Big) \\ N_0'(b) &= Q_b \cdot \Big(\sum_{j=1}^{ncp} N_0(j) - N_0(a) - N_0(b) + N_0'(a) \Big) \end{aligned}$$

with:

$$Q_a = \frac{qu_a^{max}}{1 - qu_a^{max}}$$

$$Q_b = \frac{qu_b^{max}}{1 - qu_b^{max}}$$

3. Change of cell volume

In the following 1D and 2D solution to the change of 'volume' are described.

SDTrimSP

In the program SDTrimSP [1] the new absolute number of atoms after one fluence-step Δflc is $N_1^{1D}(j)$ (normalized to $\Delta x \cdot \Delta y$), i.e. it is equivalent to the areal density. The thickness of one layer is Δz, the atomic fraction is qu and the number density of the pure solid is ϱ_0. The new thickness of the layer z_{new} and the new volume V_{new} of one layer can be calculated according to the following formula:

$$V_{new} = z_{new} \cdot \Delta x \cdot \Delta y = \sum^{j} \frac{N_1^{1D}(j)}{\varrho_0(j)} \cdot \Delta x \cdot \Delta y$$

$$z_{new} = \sum^{j} \frac{N_1^{1D}(j)}{\varrho_0(j)}$$

with:

$$N_1^{1D}(j) = \Delta N + N = N_0(j) \cdot \Delta flc + qu_{old}(j) \cdot \varrho_{old} \cdot \Delta z$$

SDTrimSP-2D

In the program with 2D-targets the number of atoms in one cell is the sum of implanted projectiles, the number of recoils transferred into the cell, the number of atoms removed from the cell (eq. 3.1) and the number of existing atoms (old) after one fluence step is:

$$N_1(j) = N_0(j) \cdot \Delta flc \cdot A_{flc} + qu_{old}(j) \cdot \varrho_{old} \cdot V_c$$

The new volume is:

$$V_{new} = \Delta V_c + V_c = \sum^{j} \frac{N_1(j)}{\varrho_0(j)}$$

If V_c and V_{new} are known, the change of the volume ΔV_c can be calculated.

To calculate the volume change a value for the real beam surface is needed in simulations with SDTrimSP-2D. Besides one has to keep in mind that the target in SDTrimSP-2D can be one (like SDtrimSP) or two dimensional.
For 1-D cases, see Fig. 47, the cells are so large that they can be treated as a layer. In this case the Δx of the cell must be much larger than the maximum of x-values during a particle cascade. This is the same geometry as in the code SDTrimSP. The infinite extension in the y-direction is achieved by a large value of Δy_{cell} of all cells. In this case, the real beam surface A_{flc} can be calculated using the cell geometry:

$$A_{flc} = \Delta x_{cell} \cdot \Delta y_{cell} \qquad \text{(1-D)}$$

In the 2-D case, the real beam surface A_{flc} can be calculated using the cell and beam geometry :

$$A_{flc} = \Delta x_{beam} \cdot \Delta y_{cell} \qquad \text{(2-D)} \qquad \text{(in program: } A_{flc} \text{ ... beam\_geo\_fac)}$$

4. Target relaxation

Implantation of projectiles and relocation or sputtering of recoils caused by bombardment produce vacancies and additional atoms in the target. Therefore the cell volume is changed and the target is allowed to relax.

SDTrimSP

In the 1D model, the layers can swell and shrink. A new target thickness z_{new} can be calculated. The starting point of the x-axis is removed on the surface. Fig. 7 shows the simple relaxation of a 1D-target with three layers (d_1, d_2, d_3), which swell (+) or shrink (-). The thickness of the whole target is the sum of all layer-thickness. The origin of the coordinate system is moved to the new surface.

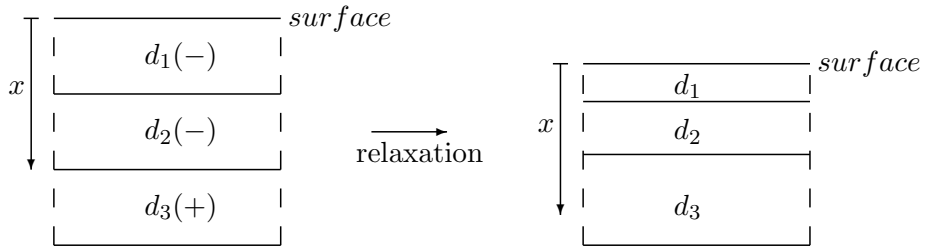


Figure 7: Relaxation of 1D-target with three layers (d_1, d_2, d_3), moving of coordinate origin and change of thick d_1, d_2, d_3, compare with 2D-target Fig. 10

SDTrimSP-2D

The relaxation method used in 1D is not suitable for the 2D case. Therefore, a new concept of relaxation had to be found:
The dynamic relaxation of the target is acting on the volumes. Particles are transported into cells in the neighborhood. The assumption is that the cells are squares and the increase or decrease of their volumes are the sum of the transfers from all directions. the change of volume ΔV_c is given and and can be split in V_l, V_r, V_t and V_b. An example of a volume change is given in Fig. 8 with the expansion of a cell.
The additional volume can be calculated accordingly:

$$\begin{aligned}
\Delta V_c &= V_l + V_r + V_t + V_b \qquad (r...right, l...left, t...top, b...bottom) \\
\Delta V_c &= (dz \cdot dl + dz \cdot dl + dx \cdot dl + dx \cdot dl) \cdot dy \\
\Delta V_c &= dl \cdot (2 \cdot dz + 2 \cdot dx) \cdot dy
\end{aligned}$$

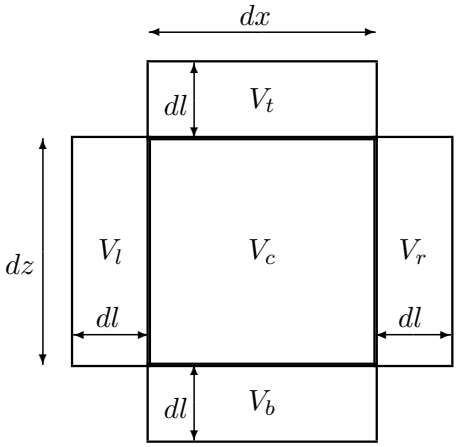


Figure 8: The expansion of the volume

Under the assumption of an equally expanding cell towards the four surfaces:

$$\begin{aligned} V_{l/r} &= dl \cdot dz \cdot dy \\ V_{t/b} &= dl \cdot dx \cdot dy, \end{aligned} \tag{4.2}$$

it is possible to eliminate dl and calculate the volume changes:

$$\begin{aligned} V_{l/r} &= \Delta V_c \cdot \frac{dz}{2(dz+dx)} \\ V_{t/b} &= \Delta V_c \cdot \frac{dx}{2(dz+dx)}. \end{aligned} \tag{4.3}$$

To get a consistent treatment of the target, volume changes have to be converted into 'fluxes'. At the beginning of a fluence step the volume of every cell is smaller/greater (active) or equal (passive cell) to the grid cell size and therefore the divergence of the fluxes between cells is zero.

$$\nabla \cdot \vec{F} = 0$$

After bombardment of the target, there might exist defects (sinks) or particles were moved, respectively implanted (sources). Thus, the target is no longer divergence-free. The additional flux is proportional to the volume change.

$$\nabla \cdot \vec{F} = Q = k \cdot \Delta V \qquad \text{k...dimension-factor}$$

The relaxation process now has the task to eliminate sources and sinks to restore a divergence-free target.

$$\Delta V(\tau \to \infty) = 0$$

The elimination of sources and sinks and therefore the calculation of fluxes is carried out by a diffusion process with a pseudo relaxation-time $\tau\prime$. The volume fluxes are calculated in the subroutine 'flux_vol'.

$$\frac{\partial V}{\partial \tau\prime} = \frac{\partial D \cdot \frac{\partial V}{\partial x}}{\partial x} + \frac{\partial D \cdot \frac{\partial V}{\partial y}}{\partial y} + \frac{\partial D \cdot \frac{\partial V}{\partial z}}{\partial z}$$

If the target is two dimensional (like in SDTrimSP-2D) the equation has only two terms:

$$\frac{\partial V}{\partial \tau\prime} = \frac{\partial D \cdot \frac{\partial V}{\partial x}}{\partial x} + \frac{\partial D \cdot \frac{\partial V}{\partial z}}{\partial z} \tag{4.4}$$

The finite-difference form is upstream in pseudo-time and centered in space:

$$\begin{aligned} \frac{\Delta V}{\Delta \tau\prime} &= \frac{D_{i+1/2} \cdot (V_{i+1} - V_i) - D_{i-1/2} \cdot (V_i - V_{i-1})}{\Delta x^2} \\ &+ \frac{D_{k+1/2} \cdot (V_{k+1} - V_k) - D_{k-1/2} \cdot (V_k - V_{k-1})}{\Delta z^2} \end{aligned}$$

with:

$$\begin{aligned} \Delta V(\tau' = 0)) &= \Delta V_c \qquad \text{(initial condition, after bombardment)} \\ \Delta V(\tau' \to \infty) &= 0 \qquad \text{(end condition, after relaxation)} \end{aligned}$$

If $\Delta x = \Delta z$ and $\tau = \tau\prime/\Delta x^2$ then:

$$\begin{aligned} \frac{\Delta V}{\Delta \tau} &= D_{i+1/2} \cdot V_{i+1} - D_{i+1/2} \cdot V_i - D_{i-1/2} \cdot V_i + D_{i-1/2} \cdot V_{i-1} \\ &+ D_{k+1/2} \cdot V_{k+1} - D_{k+1/2} \cdot V_k - D_{k-1/2} \cdot V_k + D_{k-1/2} \cdot V_{k-1} \end{aligned}$$

The reduction of tension (or ΔV_c) of the target takes place mainly in the direction of the surface. It is possible to control this anisotropy with different diffusion-coefficients or the weight-factors wig (see Chapter 6.1). Replacing the Volume V by V_l, V_r, V_t, V_b and the diffusion-coefficient D by $wig_l, wig_r, wig_t, wig_b$ one gets:

$$\begin{aligned} \frac{\Delta V}{\Delta \tau} &= wig_{l,i+1} \cdot V_{l,i+1} - wig_{r,i} \cdot V_{r,i} - wig_{l,i} \cdot V_{l,i} + wig_{r,i-1} \cdot V_{r,i-1} \\ &+ wig_{b,k+1} \cdot V_{b,k+1} - wig_{t,k} \cdot V_{t,k} - wig_{b,k} \cdot V_{b,k} + wig_{t,k-1} \cdot V_{t,k-1} \end{aligned}$$

All terms of the finite-difference equation can be written in a flux-form:

$$\frac{V_{\tau+1,i} - V_{\tau,i}}{\Delta\tau} = F_{l,i+1} - F_{r,i} - F_{l,i} + F_{r,i-1} + F_{b,k+1} - F_{t,k} - F_{b,k} + F_{t,k-1}$$

Here, F...volume-flux, r...right, l...left, t...top, b...bottom.

Finite-difference equation for a cell with number C_{Nr}=0 according to Fig. 9 is:

$$V_{0,\tau+1} = V_{0,\tau} + (F_{2,l} - F_{0,r} + F_{1,r} - F_{0,l} + F_{4,b} - F_{0,t} + F_{3,t} - F_{0,b}) \cdot \Delta\tau \tag{4.5}$$

With help of equations 4.3 the fluxes are:

$$\begin{aligned} F_{0,l} &= \Delta V_{0,\tau} \cdot \frac{dz_0}{2(dz_0 + dx_0)} \cdot wig_{l,0} \\ F_{0,r} &= \Delta V_{0,\tau} \cdot \frac{dz_0}{2(dz_0 + dx_0)} \cdot wig_{r,0} \\ F_{0,t} &= \Delta V_{0,\tau} \cdot \frac{dx_0}{2(dz_0 + dx_0)} \cdot wig_{t,0} \\ F_{0,b} &= \Delta V_{0,\tau} \cdot \frac{dx_0}{2(dz_0 + dx_0)} \cdot wig_{b,0} \end{aligned}$$

If the target is periodical in x-direction cells on the left side of the target are directly connected to their counterparts on the right side.

The solution of equations 4.4 and 4.5 are the transport fluxes (F) represented as volumes. If the solution is stationary, a full relaxation of the target is achieved. This is calculated in the subroutine '$flux\_vol$'.

$$\Delta V(\tau \to \infty) = 0$$

$$F_{right} = \sum_{\tau=0}^{\infty} (F_{2,l} - F_{0,r}) \tag{4.6}$$

$$F_{left} = \sum_{\tau=0}^{\infty} (F_{1,r} - F_{0,l}) \tag{4.7}$$

$$F_{top} = \sum_{\tau=0}^{\infty} (F_{4,b} - F_{0,t}) \tag{4.8}$$

$$F_{bottom} = \sum_{\tau=0}^{\infty} (F_{3,t} - F_{0,b}) \tag{4.9}$$

According to volume-flux in x and z directions the particle transport is calculated to the neighbour cells.

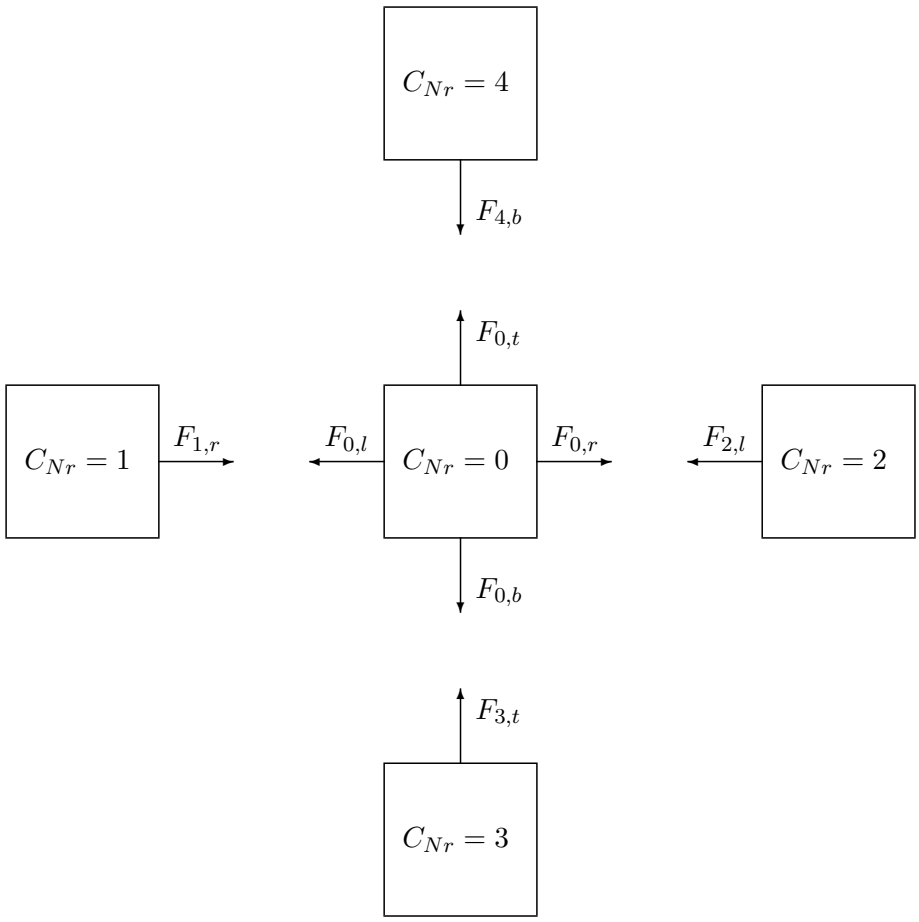


Figure 9: Volume-flux F from and to a cell with cell-number C_{Nr}=0

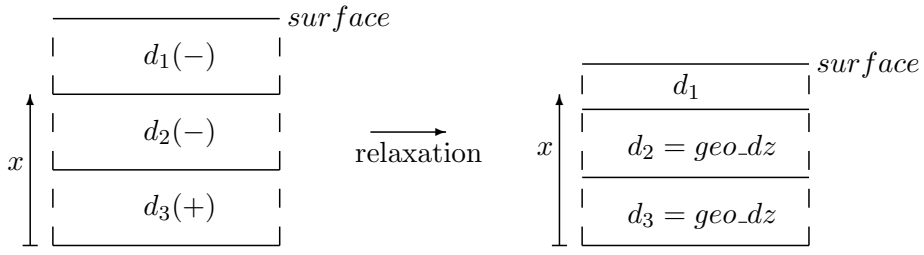


Figure 10: Relaxation of 2D-target with three layers (d_1, d_2, d_3), **no** moving of coordinate origin and , **no** change of thick d_2, d_3, compare with 1D-target Fig. 7

5. Change of density and target fraction in a cell

There is more than one species in almost all simulations, Thus, density and atomic fraction in the target and the single cells have to be taken into account. Especially the relaxation process has to consider the physical limits.

The number of particles, which originate from the neighboring cells (C_{Nr}) is

$$N_{in}(j) = \sum_{C_{Nr}=1}^{4} \frac{F_{in}(C_{Nr})}{V(C_{Nr}) + dV_c(C_{Nr})} \cdot N_1(C_{Nr}, j) \quad j...species\ of\ atoms,\ \ F...volume - flux$$

while the incoming volume is

$$dV_{in} = \sum_{C_{Nr}=1}^{4} F_{in}(C_{Nr}).$$

The Volume of a passive cell in the target is constant and independent of flux. The number of particles in the cell is:

$$N_2(j) = N_{in}(j) + \frac{(V_c - dV_{in})}{V_c + dV_c} \cdot N_1(j).$$

The number of particles in the active cell (surface-cell) and its additional volume is:

$$N_2(j) = N_{in}(j) + \frac{(V_c - dV_{in})}{V_c + dV_c} \cdot N_1(j) + \frac{dV_{add}}{V_c + dV_c} \cdot N_1(j)$$

$$dV_{add} = \sum_{ob} F_{out}(ob) \quad ob...flux\ over\ open\ boundary$$

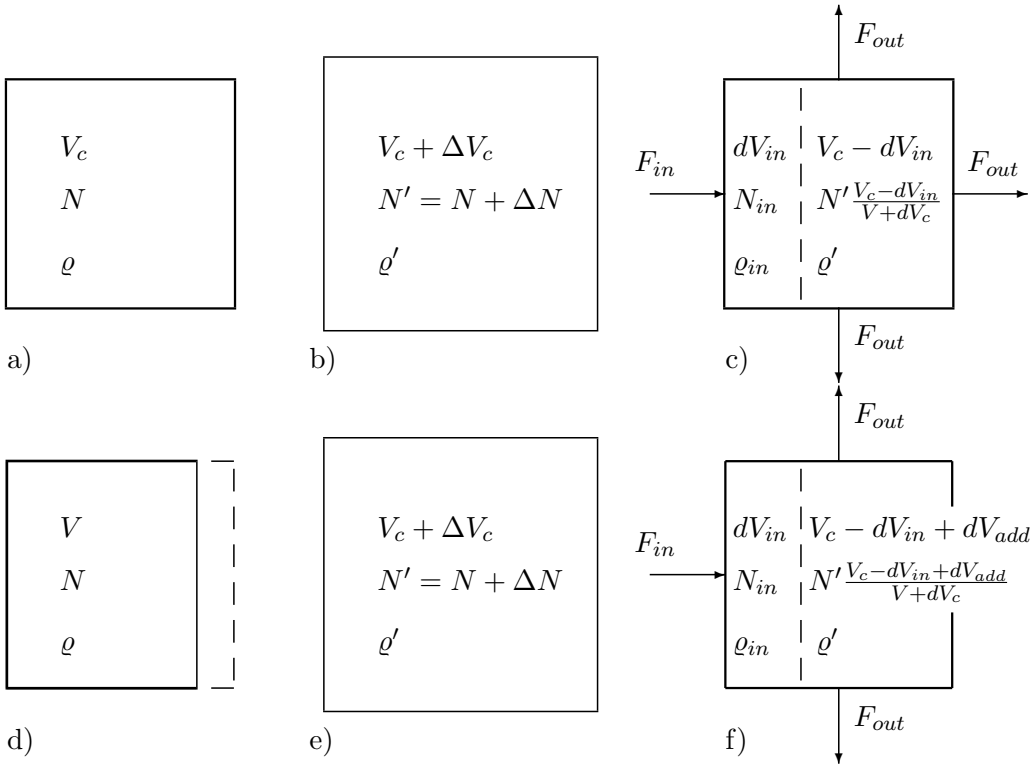


Figure 11: Parameters of a passive cell a) before bombardment, b) after collision and c) after relaxation, V(a) = V(c) and of an active cell with right open boundary d) before bombardment e) after collision and f) after relaxation, V(d) < V(f).

The new mean atomic density in each cell is therefore:

$$\frac{1}{\varrho_{new}} = \sum_{k=1}^{n} \frac{qu(k)}{\varrho_0(k)}$$

with composition

$$qu(j) = N_2(j) / \sum_{k=1}^{ncp} (N_2(k)) \quad j...species\ of\ atoms \quad k...number\ of\ all\ species.$$

6. Determination of the anisotropy

6.1. Anisotropy weight factor

The relaxation is strongly depending on the used weighting scheme, which will be described in this chapter.

First step is a categorization of all cells in terms of their distance to the surface. Different levels are introduced, where the surface cells get the value one. The next 'layer' of cells connected to those surface cells, get the value two and so on. A smaller level corresponds to a cell closer to the surface. Fig. 12 shows the procedure for an example target.

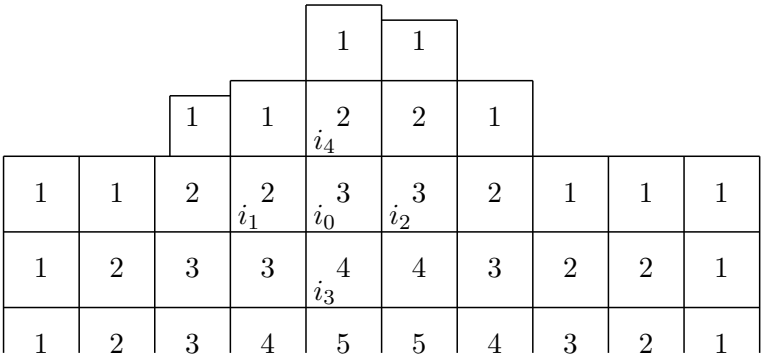


Figure 12: Determine levels for cells in the target (Cell with 1 are surface-cells).

Once the level of all cells are known, different rules for weighting factors can be applied. The weighting of the flux-connections, wig, between neighbour cells is depending on the level of the cells (see Fig. 12 and Fig. 13).
If the level is smaller then the neighbour cell than the reciprocal level of the cell is used, e.g. $i_0 > i_1$ the weighting is $1/i_0 = 1/3$.
If the level is greater or equal then the neighbour cell than the reciprocal number of distance to the surface is used, e.g. $i_0 = i_2$ the weighting is $1/6$, i_0 is 6th cell from right surface (see Fig. 12). Minimum of weighting is $1/40$.
Only the surface cells get a special weighting. The weighting is 0.15. This choice produces more stable numerical solutions.

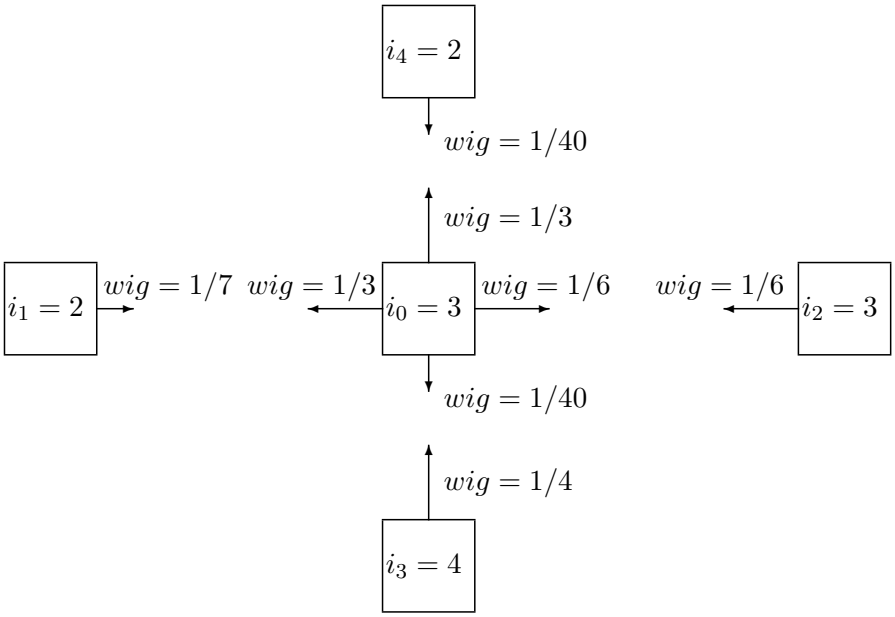


Figure 13: Determine the anisotropy weight factors wig for cell i_0

The difference between the solutions with an isotropic and anisotropic weighting scheme is small for one expanded cell and for one fluence step, but important for the whole target. This is caused due to non-local effects of the relaxation process across neighbours. In contradiction to the Eckstein ansatz in 1D the swelling happens not only perpendicular to the surface, but also to the sides ('smearing').
To show the influence of the weighting scheme on the simulation a silicon target was prepared (see Fig. 14) and bombarded with 5 keV Argon atoms under normal incidence.

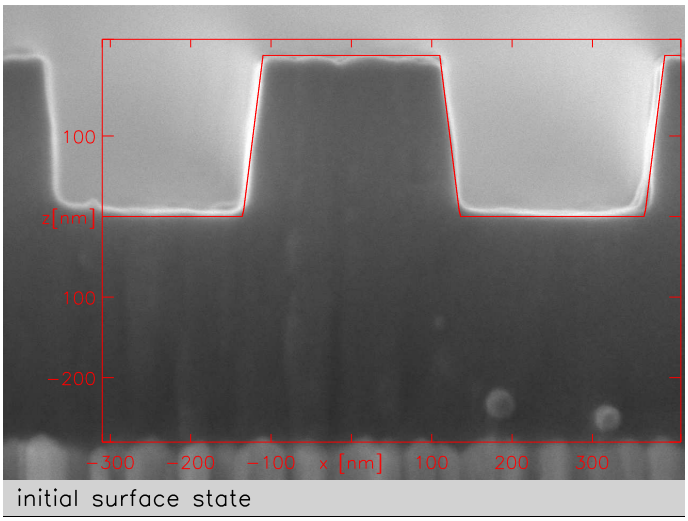


Figure 14: SEM images of a cross-section of initial state and model surface (red)

Fig. 15 shows simulation of this surface without (black) and with weighting (red line). A good agreement is achieved only with the use of anisotropic relaxation.

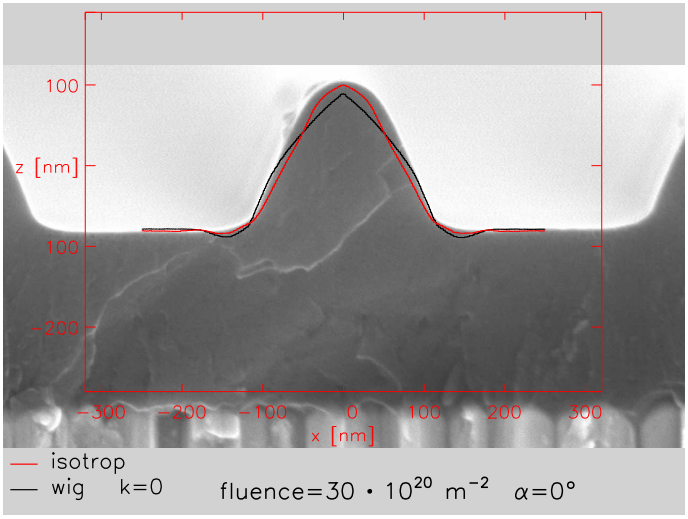


Figure 15: Comparison of calculated surface without (black) and with weighting (red) with experiment [6] 6 keV Ar on Si normal incident

6.2. Relaxation schema

The dynamic relaxation of increase or decrease target volumes takes place in two steps, see Fig. 16. The first step is a direct transport towards the surface. Between these calculation-steps a diffusive transport is carried out in all directions. The result is a more or less diffusive transport in the direction of the surface. The number of diffusion-step is k.

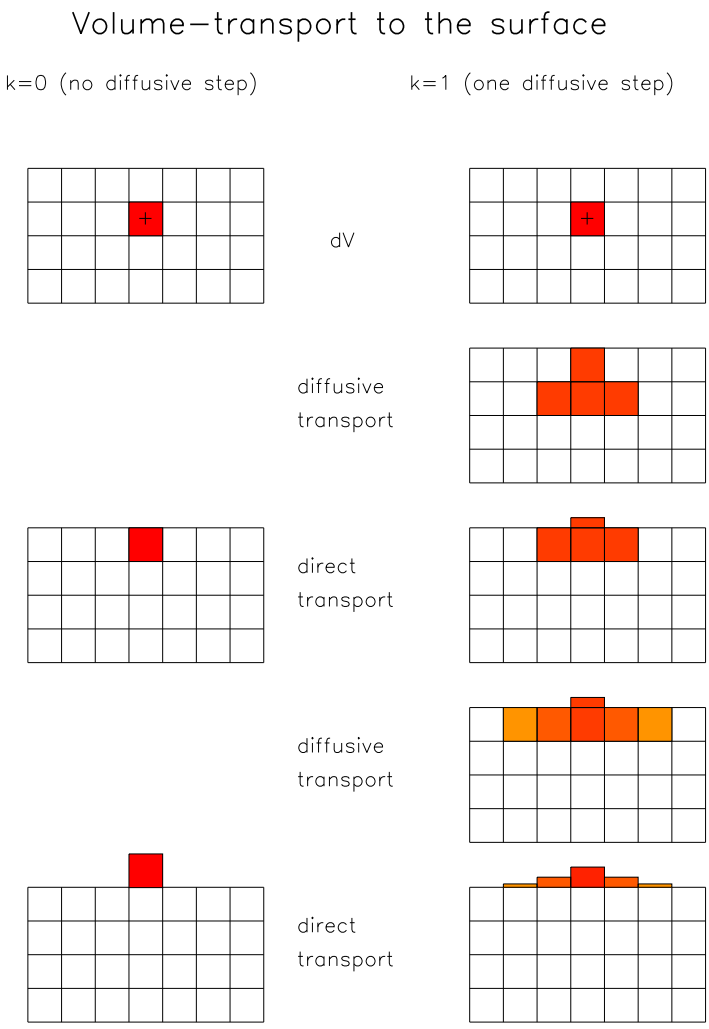


Figure 16: Volume transport in the 2D-model, left side direct transport to surface, right side transport with diffusive (smooth) transport-steps.

Fig. 17 shows calculations with smooth (number of diffusion steps $k = 5$) and hard (no diffusion-step, $k = 0$) relaxation.
The comparison with measurement shows that $k = 5$ is a good value.
Without a diffusion component ($k = 0$) the calculated target tends to oscillate surfaces.

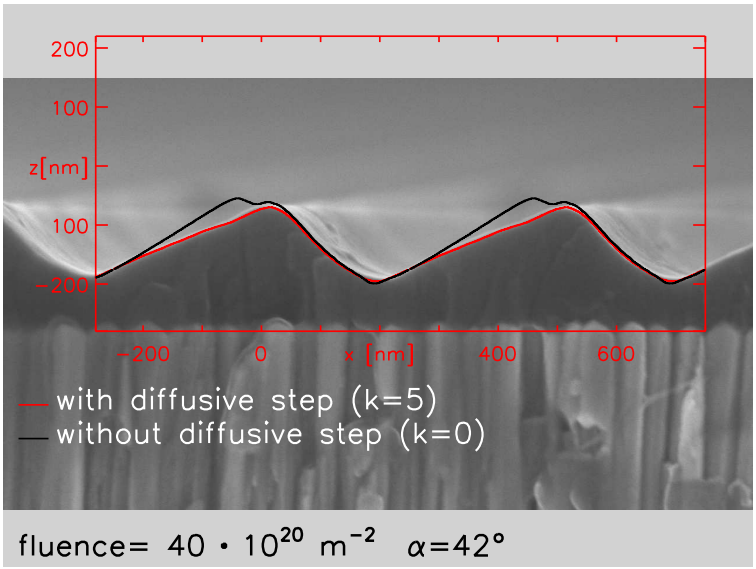


Figure 17: Comparison of calculated surface without (black) and with (red) diffusive (red) transport-steps with experiment [6] 6 keV Ar on Si normal incident

7. Transport of non-bounded particles (outgasing of noble gas ions)

The implantation of atoms in the target changes the density and the composition inside the solid and therefore has an influence on the collision cascade, on the depth profile and on sputtering. Due to the low binding energy of gas atoms (nearly zero for noble gases) they can get easily sputtered. Therefore, the gas concentration near the surface is lower than in deeper layers. The sputtering happens almost exclusively in the near-surface layers, so the sputtering-yield differs only slightly from static calculations. Another effect of the low binding energy is the possible out-gasing of noble gas atoms.
Only the presence of defects are cause a local diffusion. The damage is created during the collision-cascade and exists only temporarily. The diffusion depends on the concentration of damage and the number density of the unbounded gas atoms. The total transport of gas atoms can happen only within the range of the depth of penetration and is composed of two parts, the pressure-driven transport and the damage-driven diffusion.
The input (option) variable is **loutgas**.
default: **loutgas = .false.**

7.1. Damage-driven diffusion, DDF

According to Fick's diffusion law the flux J is:

$$J = -\frac{\partial(\eta(z) \cdot n(z))}{\partial z} \quad \text{, with} \tag{7.10}$$

η the diffusion coefficient, n the concentration (number density) and z denoting the depth. The fluence-dependent equation is:

$$\frac{\partial n}{\partial \phi} = -\frac{\partial(\eta(z) \cdot \frac{\partial n(z)}{\partial z})}{\partial z} \tag{7.11}$$

Although the atoms (Xe, Ar) are not bounded, however they cannot diffuse freely. Only due to the collision damage (like damage diffusion) the atoms can move in the target. The relative probability for the diffusion is P_{dam}. Therefore the diffusion-coefficient can be express as:

$$\eta(z) = \eta_0 \cdot P_{dam} \tag{7.12}$$

A series of calculations has shown that the probability depends on both the number of the damage as well as the concentration of the particles. N_{dam} is the number density of damage in a layer and $qu(z)$ the atomic fraction. The area in which diffusion occurs is the range of the defect profile. Therefore the relative probability for diffusion is given by:

$$P_{dam} = \frac{N_{dam}(z)}{max[N_{dam}]} \cdot qu(z) \tag{7.13}$$

The input variable of η_0 is **diff_koeff1**.
default: **diff_koeff1(:)=0.0**

7.2. Pressure-driven transport, PDT

The pressure is generated by the particles coming from the outside. A high pressure is created at the implantation position. The target responds by expansion and the pressure reduces after a certain period. But before the target relaxes fully, this pressure generate a transport non bounded particles (Xe, Ar) towards the surface. The flux J, is:

$$J = -K(z) \cdot n(z) \tag{7.14}$$

K is the transport coefficient, n is the concentration (number density) and z is the depth. ϕ is the fluence. The fluence-dependent equation is:

$$\frac{\partial n}{\partial \phi} = \frac{\partial (K(z) \cdot n(z))}{\partial z} \tag{7.15}$$

The range of the transport is limited by the range of the implanted profile z_{max}. Therefore the transport-coefficient is:

$$K(z \leq z_{max}) = K_0 \tag{7.16}$$
$$K(z > z_{max}) = 0 \tag{7.17}$$

The flux increases monotonically with the number of particles and the flux dependent on the form of the profile.
The input variable of K_0 is **diff_koeff2**.
default: **diff_koeff2(:)=0.0**

7.3. Results

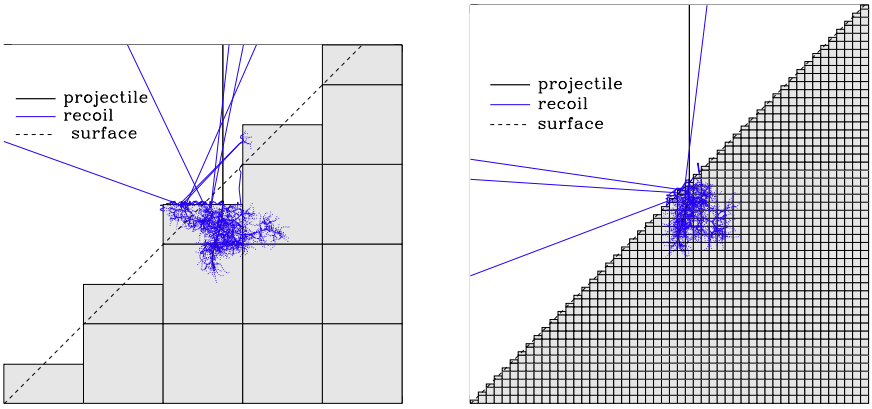
The agreements of calculated surfaces with measured values are very good, see Fig. 52 and 53.

The coefficients η_0 (diff_koeff1) and K_0 (diff_koeff2) for Xe and Ar [1] are:

$$\begin{aligned} \eta_0(Xe) &= 1.65 \cdot 10^6 \ \mathring{A}^4/ion \\ K_0(Xe) &= 95 \ \mathring{A}^3/ion \\ \eta_0(Ar) &= 1.65 \cdot 10^5 \ \mathring{A}^4/ion \\ K_0(Ar) &= 15 \ \mathring{A}^3/ion \end{aligned}$$

8. Influence of resolution

If the grid resolution of a target with inclined surface is insufficient the value of the sputter yield is more and more close to the value of a non-inclined surface, Fig. 19. The inclination of the surface is γ . The size of the grid should be much smaller than the extension of the trajectories, see Fig. 18. The roughness size has a similar influence [12].



(a) (b)

Figure 18: Trajectories at inclined plane of $\gamma = 45°$ (dash line) with a grid resolution (a) dx=dz=100 Å and (b) dx=dz=10 Å (outputfiles: E0_31_3D.dat, T_10_2D.dat)

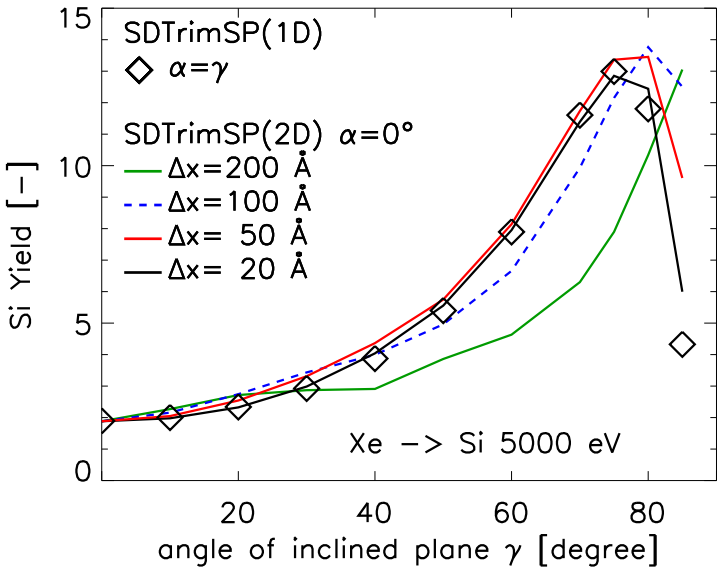


Figure 19: Sputter Yield of Si dependent on angle of incidence α at different grid resolutions (outputfile: E0_31_3D.dat)

9. Comparison of results from SDtrimSP and SDTrimSP-2D

In case of a static calculation (no target change) of smooth surfaces, the use of a two-dimensional target and its resolution is not important, unless a profile in x-direction are already available. Due to possibility of running SDTrimSP-2D with one dimensional geometry and in a dynamic-mode direct comparisons with SDTrimSP may be performed. A first dynamical test is to simulate a 1-D target, which is shown in Fig. 20.

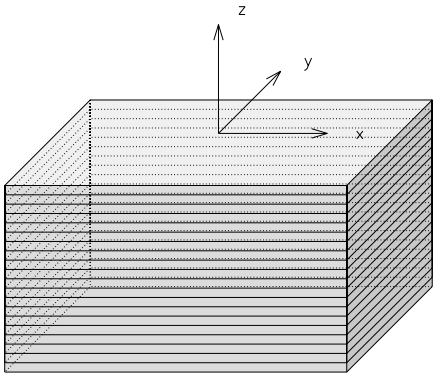


Figure 20: Geometry of one dimensional target in the 2D-program, according to SDTrimSP

The target consist of Ta_2O_5 and was bombarded with 1000 eV He. Fig. 21 shows the surface elevations, the depth profiles, the sputtering yields and the atomic fraction at surface for SDTrimSP and SDTrimSP-2D. The differences between the results are small.

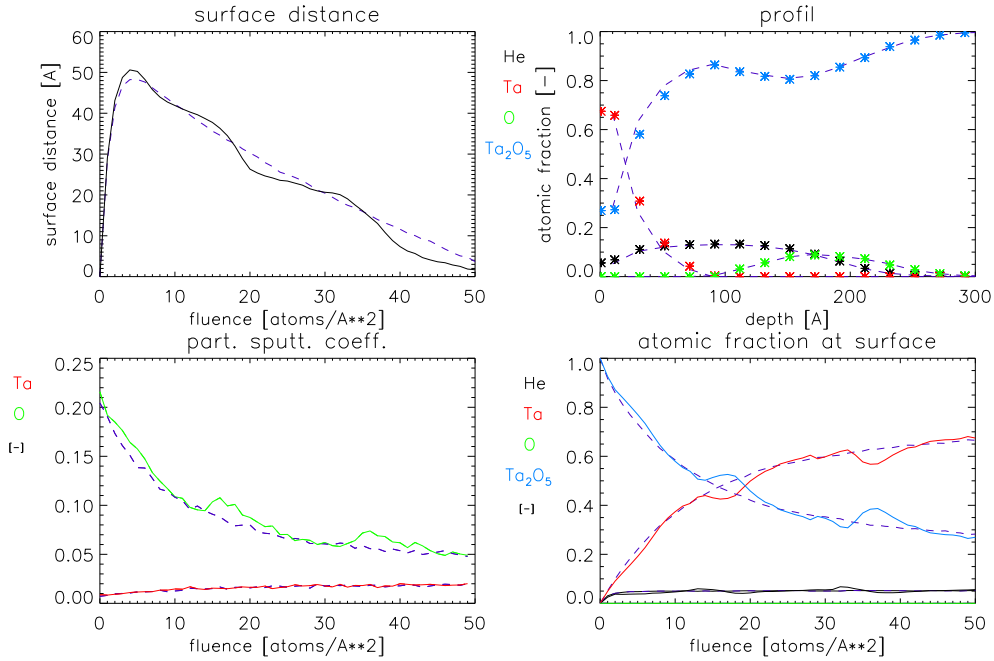


Figure 21: 1000 eV He on Ta_2O_5, Comparison of results from SDTrimSP (dashed line) and from SDTrimSP-2D (outputfiles: T_10_2D.dat, E0_31_3D.dat)

Another dynamical test-case is the bombardment of silicon target with 2000 eV Xe. In this simulation Xe is implanted and the damage-driven-diffusion is used to simulate the outgasing process of Xe. The comparison of the results from simulations with SDTrimSP and SDTrimSP-2D shows Fig. 22.

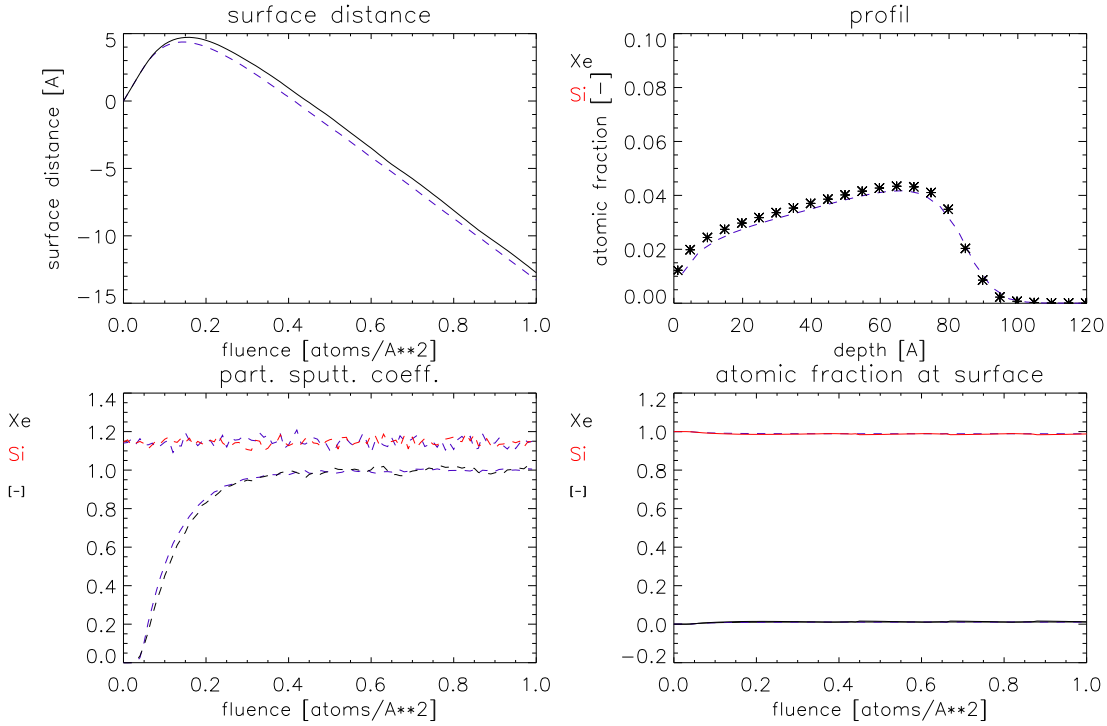


Figure 22: 2000 eV Xe on Si, Comparison of results from SDTrimSP-2D and from SDTrimSP (dashed line) (outputfiles: T_10_2D.dat, E0_31_3D.dat)

10. SDTrimSP-2D: input-parameter in tri.inp

10.1. Input-parameter of target-geometry

Geometry of target
geo_x ... half X-width of target [A]
geo_z ... depth (-Z) of target (negative) [A]
geo_dx ... grid-length (cell-width Δx) in X-direction
geo_dz ... grid-length (cell-width Δz) in Z-direction

unit of length: angstrom [A]

default: geo_dx=geo_dz

Switch:
imax = 1 : without memory-reduction
imax = 2 : with memory-reduction

example:
geo_x = 1000.
geo_z = -2800.
geo_dx = 25.0
geo_dz = 25.0
imax = 1

10.2. Input-parameter of surface (roughness), parameter: 'case_geo'

10.2.1. case_geo=10 (layers)

Target like in SDTrimSP-1D:

case_start =1
geo_x = geo_dx
example: Chapter 11.1 and Fig. 47

10.2.2. case_geo=20 (smooth surface)

Target with smooth surface , see Chapter 11.1 and Fig. 48 - 50

10.2.3. case_geo=24 (roughness: $z_{surf} = f(x)$, $x >= 0$)

This option use the inputfile 'rauh.inp'.
The first line is a comment. The number in the second line gives the number of points that are given, followed the x and z-values (all values ≥0).
Example of inputfile 'rauh.inp' (# ... Comment line):

#pitch grating (x,z Values)
6
0.0 000.0
100.0 000.0
140.0 200.0
360.0 200.0
385.0 100.0
500.0 50.0

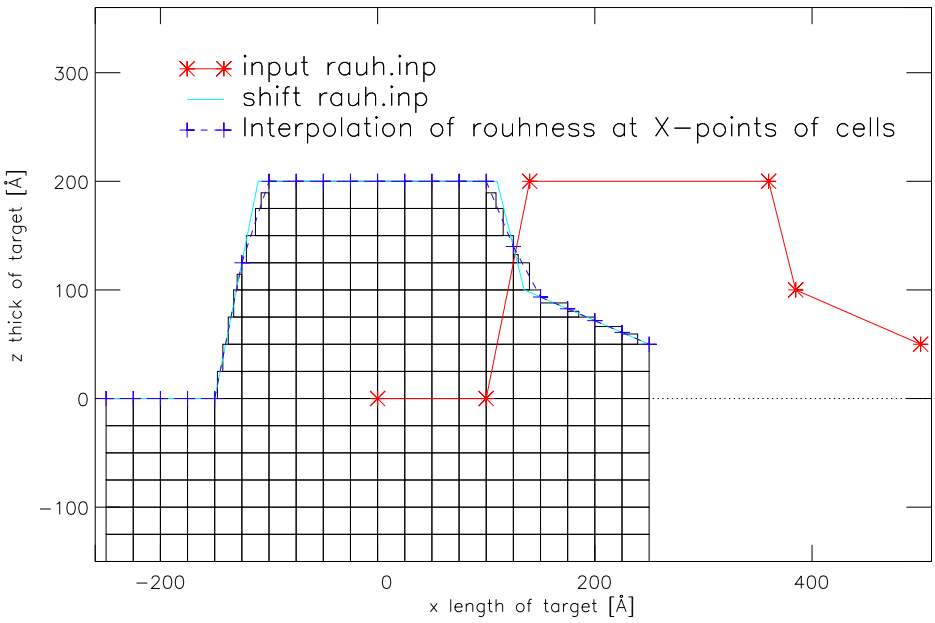


Figure 23: Input of rauh.inp (red), shift (light blue), interpolation at corner (blue) and simulated target-cells (black grids), option case_geo=24

10.2.4. case_geo=25 (roughness: $z_{surf} = f(x)$)

This option use the inputfile 'rauh.inp'.

This example use a similar structure of input-file 'rauh.inp' as 'case_geo'=24, but the X-values may be also negative.

Example of inputfile 'rauh.inp' (# ... Comment line):

```
#pitch grating (x,y Values)
6
 -250.0   000.0
 -150.0   000.0
  -10.0   200.0
  110.0   200.0
  135.0   100.0
  250.0    50.0
```

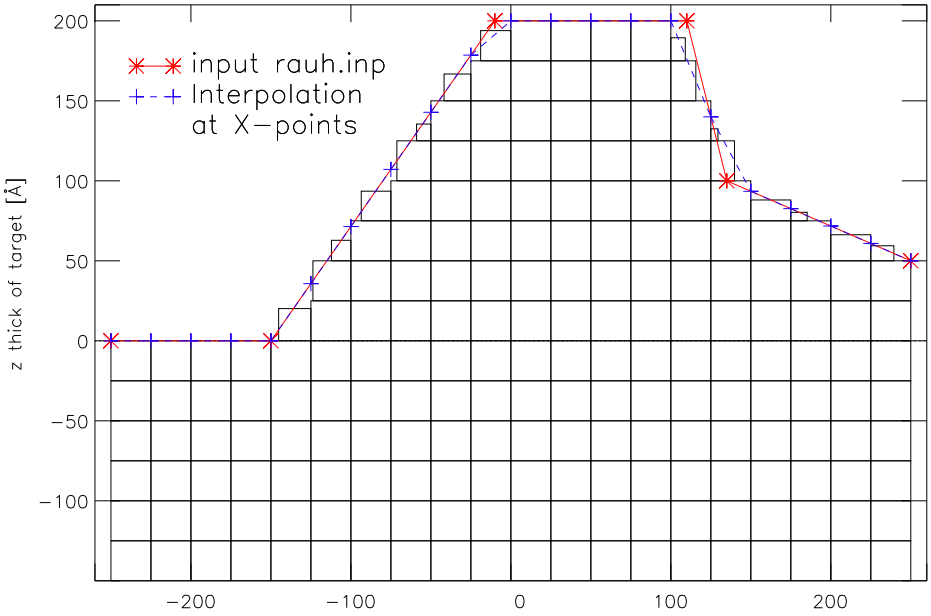


Figure 24: Input of rauh.inp (red), interpolation at corner (blue) and simulated target-cells (black grids), option case_geo=25

10.2.5. case_geo=25 with switch l_alpha_rough_kor=.true.

This option is for surfaces with different boundary conditions (left, right) of roughness (mostly measurements) to avoid errors in periodic calculations. To obtain correct results the coordinate system has to be rotated. Also the angle of incidence α must be changed. The picture Fig. 25 shows the rotation of the coordinate system by the angle γ.

Procedure of new roughness for calculation z_c in rotated system:

1. Interpolation of roughness $z_0 = f(x_0)$ (rauh.inp) at X-points of cells like case_geo=25
2. Calculation of line between left and right boundary roughness, see Fig. 25 (black dash-line)
3. Calculation of γ, see Fig. 25
4. shift roughness to the coordinate origin

$$\begin{aligned} x_1 &= x_0 \\ y_1 &= y_0 \\ z_1 &= z_0 - dz \end{aligned}$$

5. rotation of roughness around y-axis and shift back

$$\begin{aligned} x_2 &= x_1 * cos(-\gamma) - z_1 * sin(-\gamma) \\ y_2 &= y_1 \\ z_2 &= \Big(z_1 * cos(-\gamma) + x_1 * sin(-\gamma)\Big) + dz \end{aligned}$$

6. shift roughness to positive values (if $min(z_2) < 0$)

$$\begin{aligned} x_c &= x_2 \\ y_c &= y_2 \\ z_c &= z_2 - dz2 \qquad dz2 = min(0, min(z_2)) \end{aligned}$$

Calculation of new start-position (xs_c, ys_c, zs_c) of incident projectiles in rotated system:

$$\begin{aligned} xs_1 &= xs_0 \\ ys_1 &= ys_0 \\ zs_1 &= zs_0 - dz \end{aligned}$$

$$\begin{aligned} xs_c &= xs_1 * cos(-\gamma) - zs_1 * sin(-\gamma) \\ ys_c &= ys_1 \\ zs_c &= \Big(zs_1 * cos(-\gamma) + xs_1 * sin(-\gamma)\Big) + dz - dz2 \end{aligned}$$

Calculation cos-values of new incident angles ($cosx_c, cosy_c, cosz_c$) in rotated system:

$$\begin{aligned} cosz &= cos(180-\alpha) && \alpha \ ... \ polar-angle \\ sinz &= \sqrt{(1.-cosz*cosz)} \\ cosx &= sinz*cos(180+\beta) && \beta \ ... \ azimuthal-angle \\ cosy &= sinz*sin(180+\beta) \end{aligned}$$

$$\begin{aligned} cosx_c &= cosx*cos(-\gamma)-cosz*sin(-\gamma) \\ cosy_c &= cosy \\ cosz_c &= cosz*cos(-\gamma)+cosx*sin(-\gamma) \end{aligned}$$

Calculation of the polar and azimuthal angle of sputtered particles back from the calculation system (x_c, y_c, z_c, Fig. 27) to the normal system (x_0, y_0, z_0):

$$\begin{aligned} cosx &= cosx_c*cos(\gamma)-cosz_c*sin(\gamma) \\ cosy &= cosy_c \\ cosz &= cosz_c*cos(\gamma)+cosx_c*sin(\gamma) \\ sinz &= \sqrt{(1.-cosz*cosz)} \end{aligned}$$

$$\begin{aligned} cos_p &= cosz \\ cos_a &= 0 && if\ (sinz=0) \\ sin_a &= 0 && if\ (sinz=0) \\ cos_a &= cosx/sinz && if\ (sinz>0) \\ sin_a &= cosy/sinz && if\ (sinz>0) \end{aligned}$$

$$\begin{aligned} \alpha &= arccos(cos_p) \\ \beta &= arctan(sin_a/cos_a) \end{aligned}$$

note: If $\alpha = 0$ then β=undefined (may be set =0)

Fig. 27 - 30 show the different results of bombardment with the same boundary conditions of roughness (l_alpha_rough_kor=.true.) and with large differences at the edges of the surface.

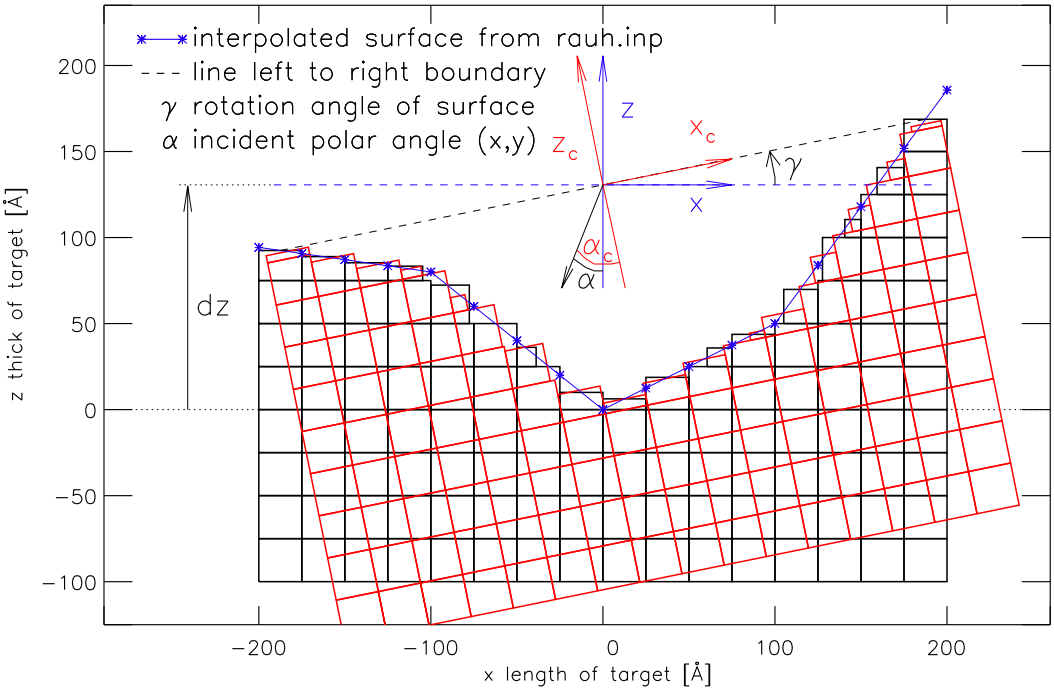


Figure 25: Rotation of coordinate system to get the same left and right roughness (z-values)

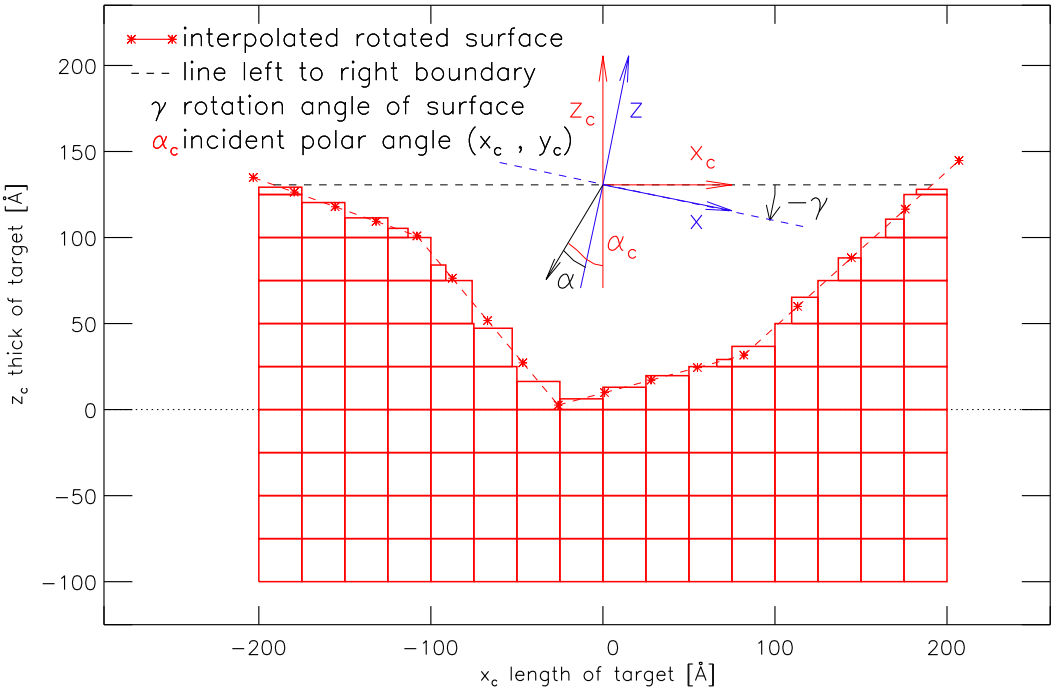


Figure 26: Surface in rotated coordinate system x_c, z_c with corrected incident angle α_c

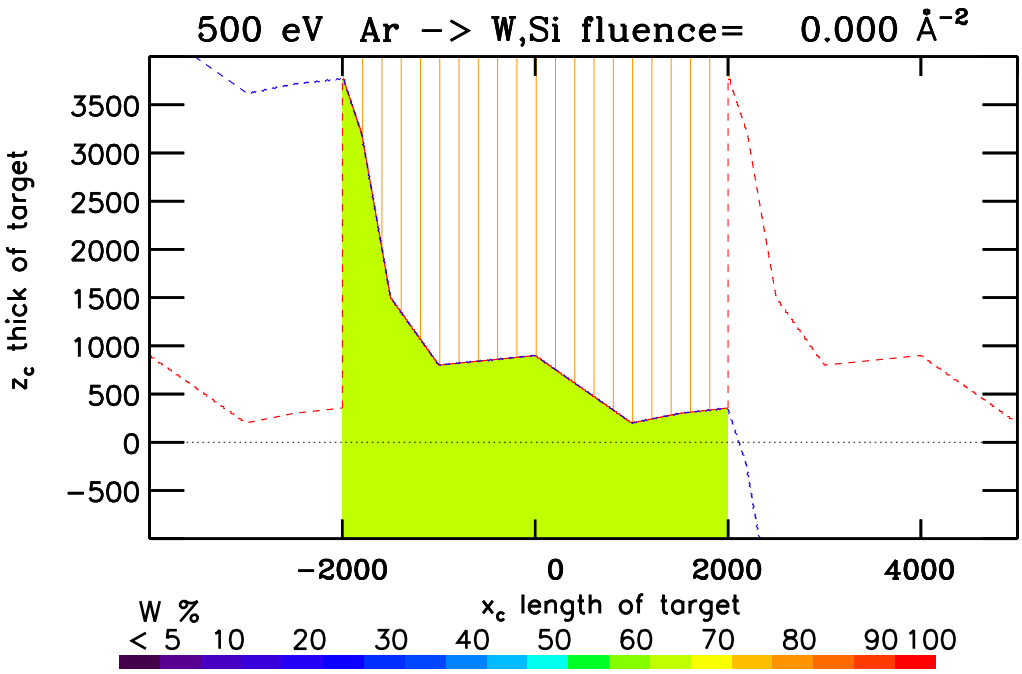


Figure 27: Simulated periodical roughness (red dash-line) in coordinate system x, z with start-target (green area) , simulated periodical start-surface from rotated system (blue dash-line), like Fig. 28

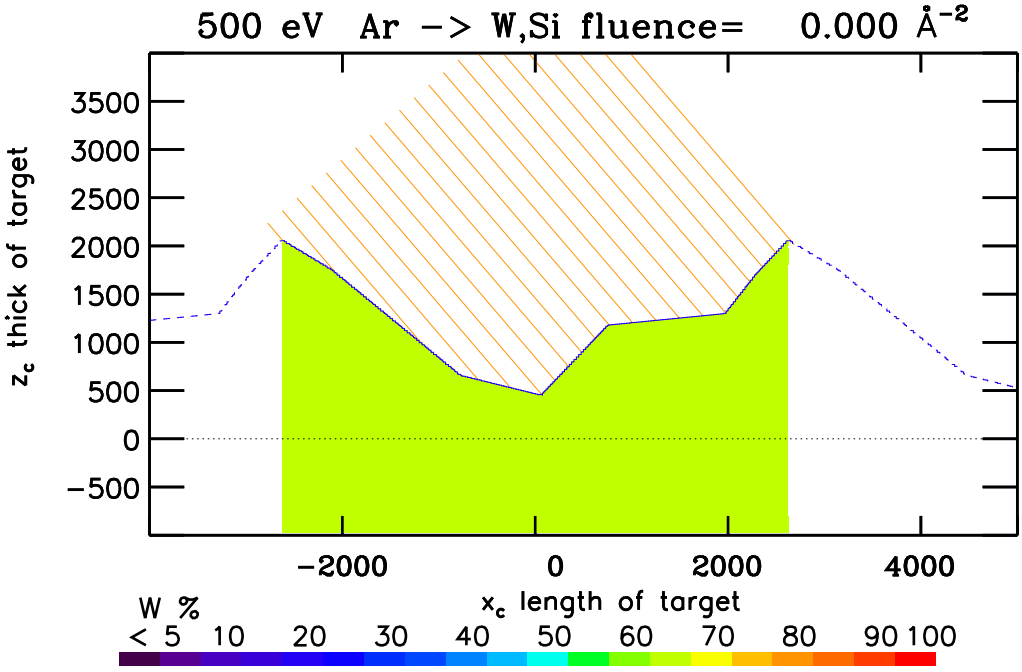


Figure 28: Surface in rotated coordinate system x_c, z_c with corrected incident angle α_c, left and right boundary roughness are the same

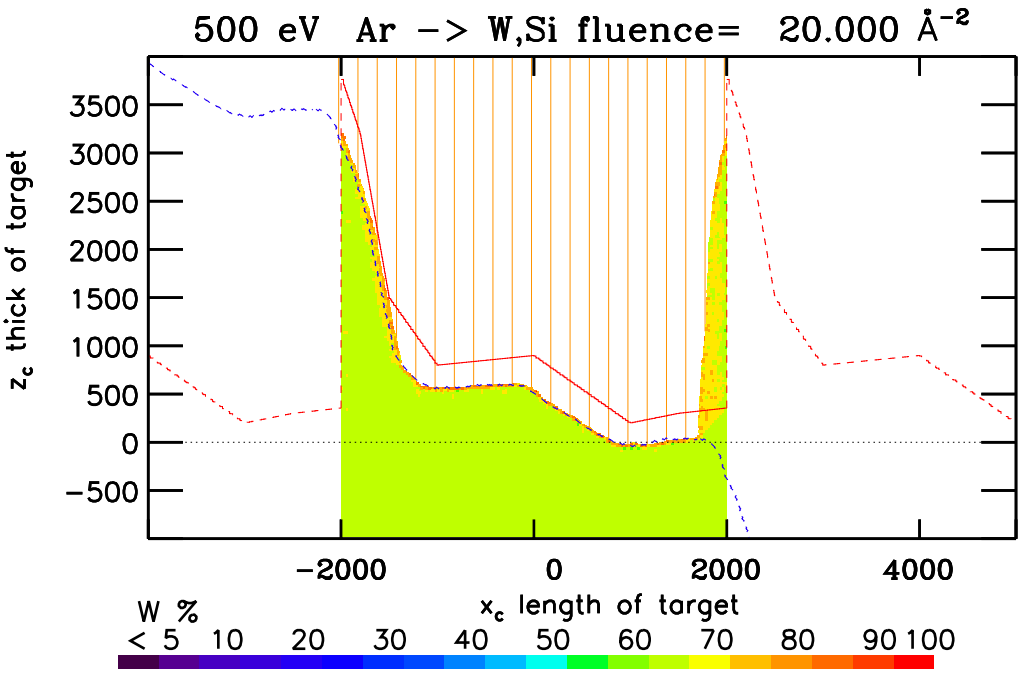


Figure 29: Surface after bombardment in coordinate system x, z, with start-surface (red line) and surface after bombardment in rotated system (blue dash-line) like Fig. 30

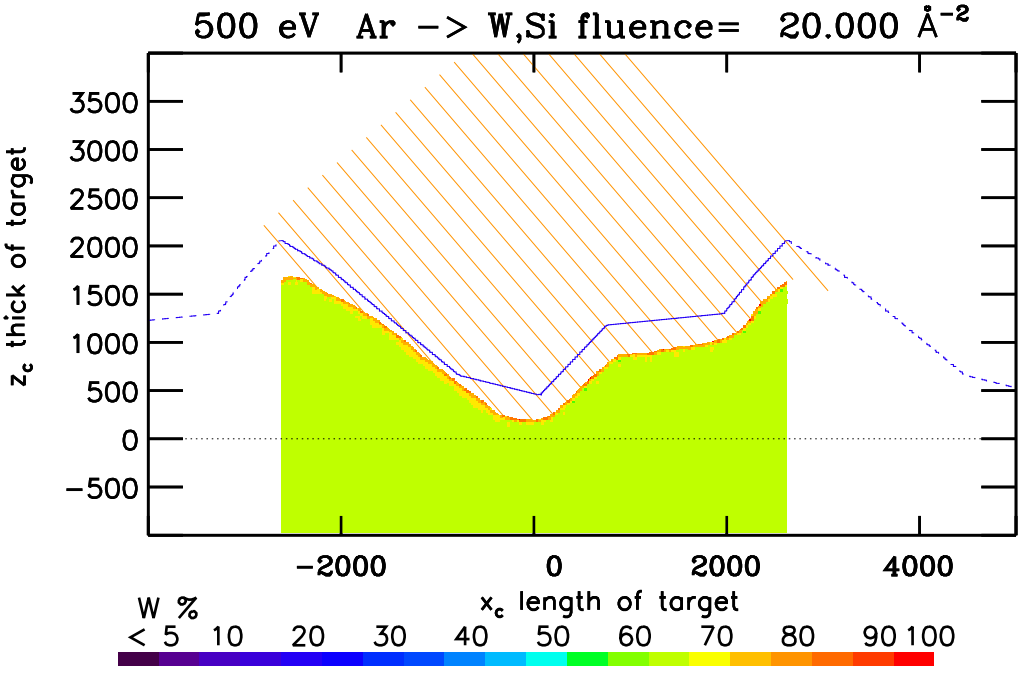


Figure 30: Surface in rotated coordinate system x_c, z_c after bombardment with corrected incident angle α_c and start-surface (blue line)

10.2.6. case_geo=26 (voxel) $z_{surf} = f(x, z)$

Given are voxels (cuboids) in inputfile 'fig.inp'. The option offers the possibility of input any cell structure. Start-target is given cell-structure (x_1, x_2, z_1, z_2).

Condition: All voxel inside the calculated range $(-geo\_x \leq x \leq geo\_x,\ z > 0)$ and no cells beyond the periodical start-target (base plate $(z \leq 0)$).
$\Delta x \leq$ geo_dx and $\Delta z \leq$ geo_dz.

Example 1
'tri.inp': ncp=3
symbol = "Ar","W","Si"
qu_tar = 0.00, 0.00, 1.00
'fig.inp:'

```
# i(mirror), i(qu), Nr(cells)                (# ... Comment line)
# cell-coordinates(x1 x2 z1 z2) [A] , qu(ncp) [-]
0  !—1... with mirror (coordinates x,z > 0)  0...without mirror
0  !—1... with qu(1:ncp)                     0...without qu (qu from tri.inp)
8  !—number cells
  00   25 00 25  0.0 0.9 0.1
  25   50 00 25  0.0 0.8 0.2
  50   75 00 25  0.0 0.7 0.3
  75  100 00 25  0.0 0.6 0.4
  25   50 25 50  0.0 0.5 0.5
  50   75 25 50  0.0 0.4 0.6
  75  100 25 50  0.0 0.3 0.7
 100  125 25 50  0.0 0.2 0.8
```

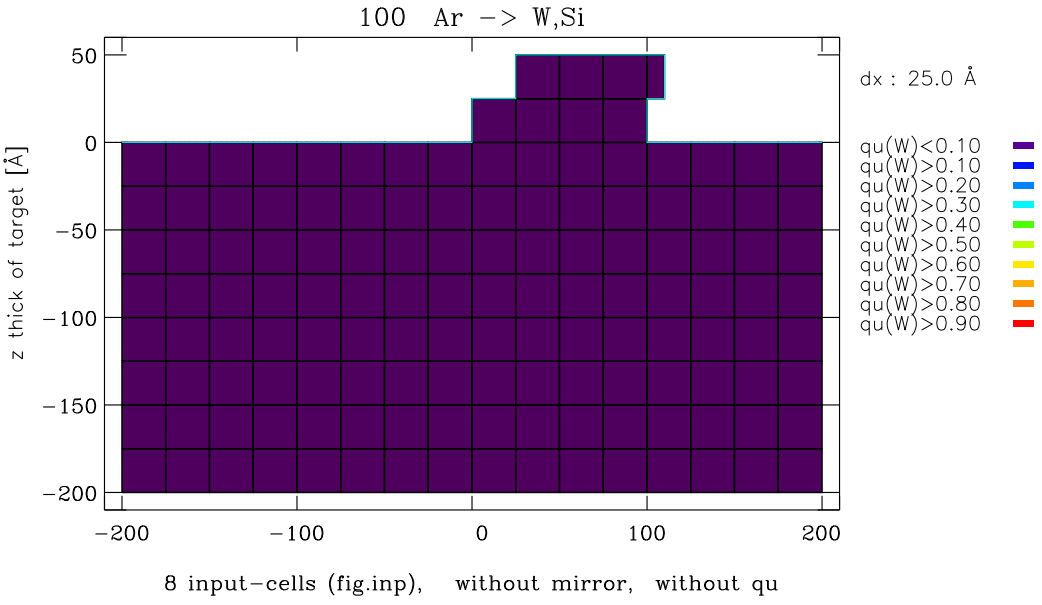


Figure 31: Start-target, input of fig.inp without mirror and qu (outputfile: T_10_2D.dat)

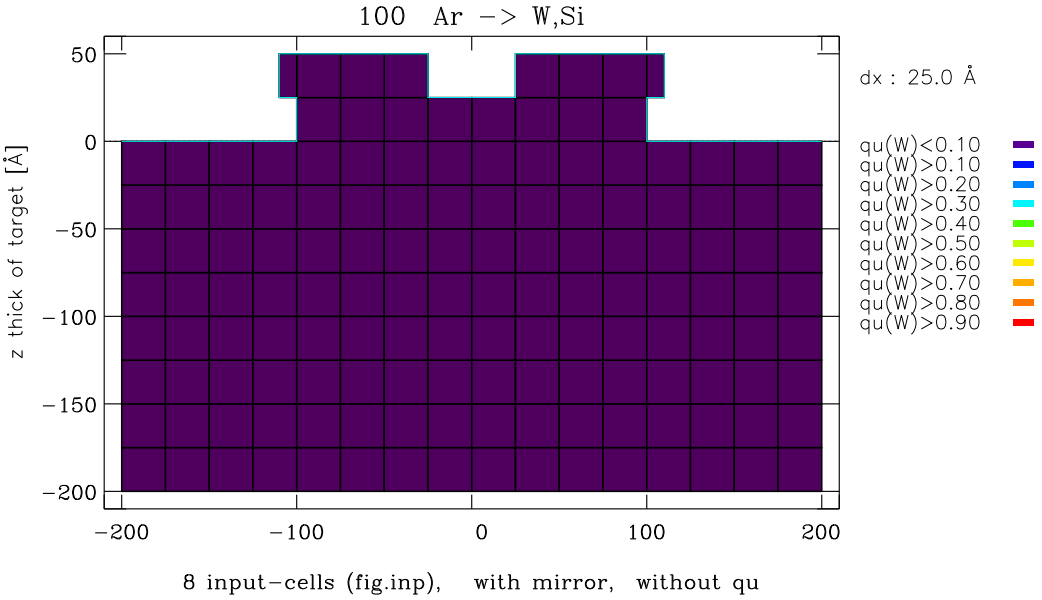


Figure 32: Start-target, input of fig.inp (8 cells) with mirror and without qu (outputfile: T_10_2D.dat)

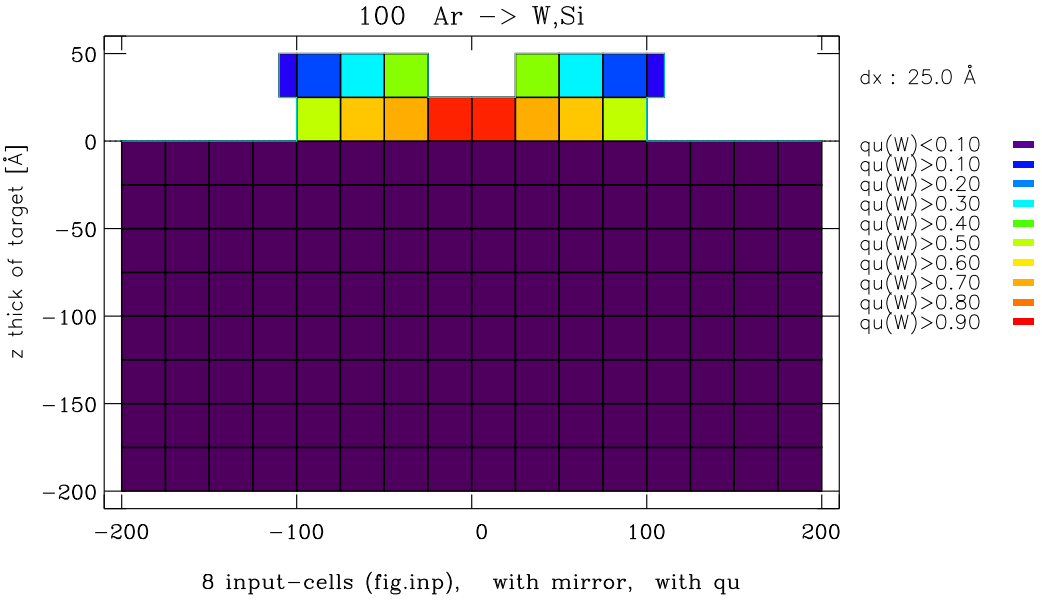


Figure 33: Start-target, input of fig.inp (8 cells) with mirror and qu (outputfile: T_10_2D.dat)

Example 2: Xe → Ta,Si $\alpha = 30°$ (target with a hole)

'tri.inp': ncp=3
symbol = "Xe","Ta","Si"
qu_tar = 0.00, 0.00, 1.00

'fig.inp:' # Xe → Ta,Si
cell-coordinates(x1 x2 z1 z2) [A] , qu(ncp) [-]
0 !—1... with mirror (coordinates x,z > 0) 0...without mirror
1 !—1... with qu(1:ncp) 0...without qu (qu from tri.inp)
16 !—number cells

```
-75  -50   0 25   0.0 0.1 0.9
 12   25   0 25   0.0 0.2 0.8
 25   50   0 25   0.0 0.3 0.7
 50   75   0 25   0.0 0.4 0.6
 75  100   0 25   0.0 0.1 0.9
-65  -50  25 50   0.0 0.2 0.8
-50  -25  25 50   0.0 0.4 0.6
-25    0  25 50   0.0 0.6 0.4
  0   25  25 50   0.0 0.9 0.1
 25   50  25 50   0.0 0.7 0.3
 50   75  25 50   0.0 0.5 0.5
 75  100  25 50   0.0 0.3 0.7
100  110  25 50   0.0 0.2 0.8
-25    0  50 75   0.0 0.3 0.7
  0   25  50 75   0.0 0.4 0.6
 25   50  50 75   0.0 0.5 0.5
```

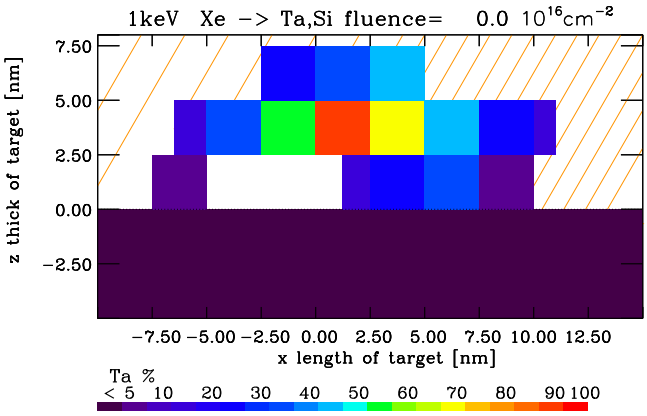


Figure 34: Start-Target
(outputfile: T_10_2D.dat)

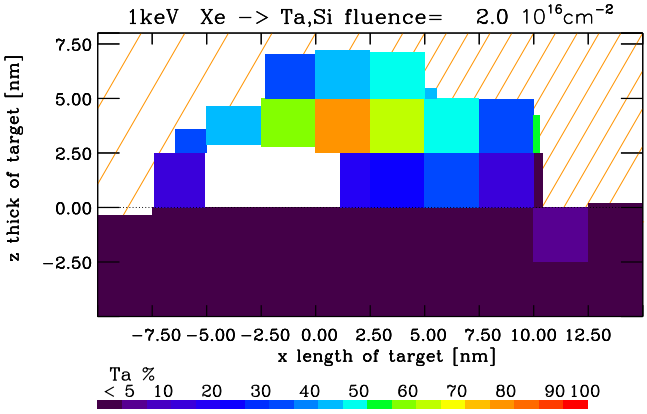


Figure 35: Target after bombardment
(outputfile: T_10_2D.dat)

10.2.7. case_geo=20 with Cosine distribution

Besides defining a rough surface with the input file rauh.inp, a periodic cosine-structures is possible with the input parameters **pr_amp** and **pr_wl** in **tri.inp**. They represent the amplitude and wavelength of a cosine that is added on top of the predefined surface structure, whether this is given as flat or as rough from rauh.inp. The new surface $z_{new}(x)$ can then be described as:

$$z_{new}(x) = z(x) + pr\_amp \cdot \left[1 + \cos\left(\frac{2\pi}{pr\_wl} \cdot x + \pi\right)\right]$$

with $z(x)$ describing the surface before adding the periodic roughness. It should be noted that in order to preserve the periodicity of the whole geometry, the periodic wavelength **pr_wl** has to be a divider of the total x-length **2** · **geo_x**.
The switch for pure cos-distribution is **case_geo=27** in **tri.inp**.
Fig. 36 shows a sketch of how the two parameters control the periodic roughness features.

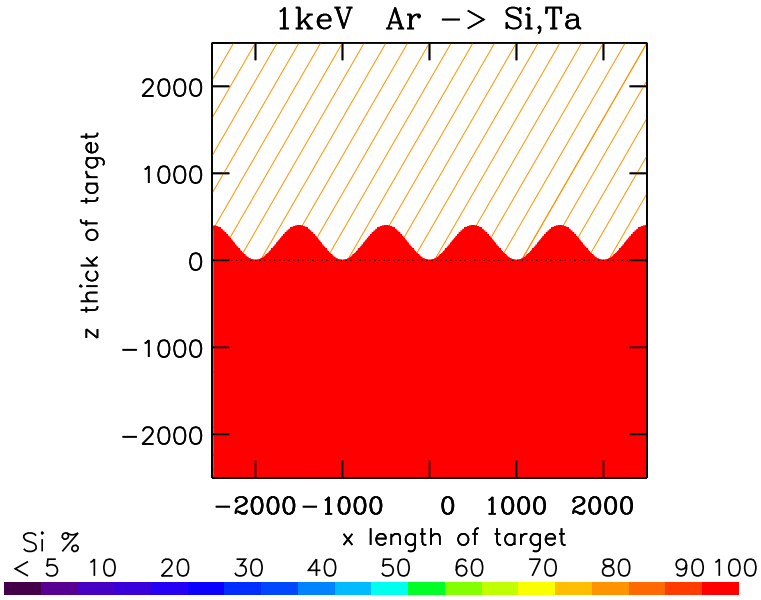


Figure 36: A cosine roughness structure, parameters are pr_amp = 200 and pr_wl = 1000 (outputfile: T_10_2D.dat).

10.2.8. case_geo=25 with cosine distribution

This case combine periodic roughness and cosine distribution
Fig. 37 shows an example that combines a surface roughness taken from **rauh.inp** and cosine distribution with pr_amp = 200 and pr_wl = 1000. The switch for this case **case_geo=25** and the input parameters **pr_amp** and **pr_wl** in **tri.inp**.

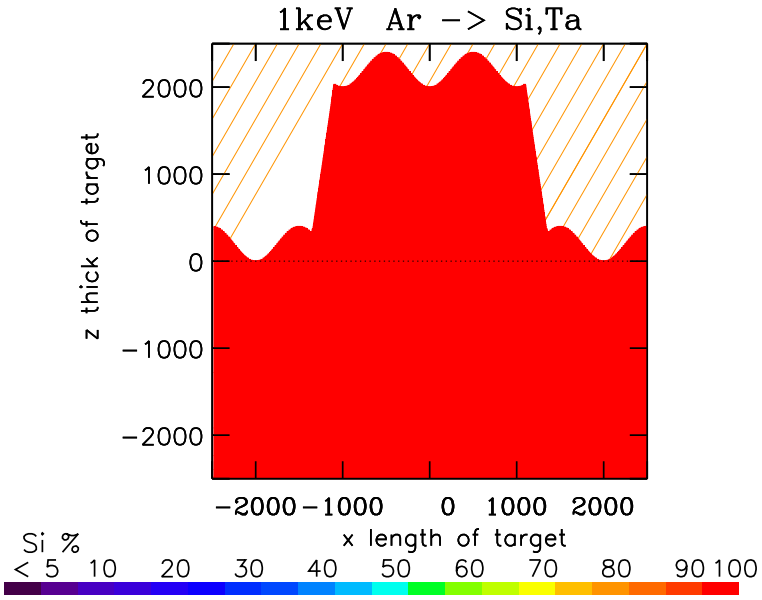


Figure 37: A combination of a rauh.inp input and added cosine distribution, the periodic roughness parameters were defined pr_amp = 200 and pr_wl = 1000 (outputfile: T_10_2D.dat).

10.3. Input-parameter of beam, parameter: 'case_start'

case_start ... case of start position of projectiles (x_start, z_start)
- 1 : x_start and z_start are constant and given in tri.inp (z_start> target)
- 2 : not used
- 3 : x_start and z_start are constant and given in tri.inp (start also **inside** target)
- 4 : z_start is constant and given in tri.inp
  x(start)=x_start ± dx_start/2*random

x_start ... x-center of beam (if case_start=4)

dx_start ... x-width of beam (use only case_start=4)
dx_start < geo_dx : dx_start = x-target
(e.g.: dx_star=0.0 calculate: dx_start=2 geo_x)

z_start ... z-start-position of projectiles

example 1 (whole target-surface), Fig. 38,Fig. 39) :
X_{target}=1600: geo_x = 800.
global beam (projectiles bombard the whole target-surface):
case_start= 4
x_start = 0.
dx_start= 0. or dx_start=1600
z_start = 900.

example 2(part of target-surface):
X_{target}=1600: geo_x = 800.
global beam (projectiles bombard the whole target-surface):
case_start= 4
dx_start= 800.
z_start = 900.

example 2a: Fig. 40 **(Is not ever CORRECT or what the user wants)**
x_start = 0.

example 2b: Fig. 41 **(CORRECT)**
x_start = 300.

In Fig. 39 and 40 the projectile appears to leave the computing area on the left side and come in on the right. Due to the periodicity, these trajectories are identical.

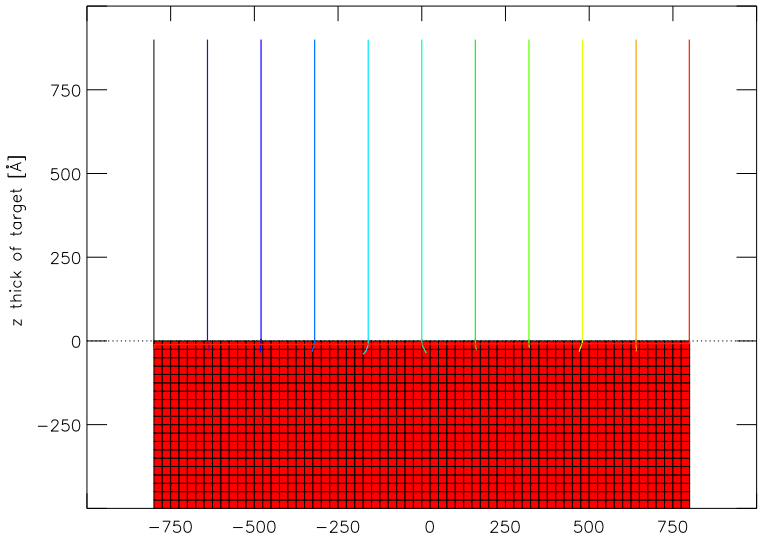


Figure 38: The target simulation $\Delta x_{beam} = 1600\mathring{A}$ $\alpha = 0°$ or whole target-surface, this corresponds to a periodic target (or infinite target) and infinite width beam.

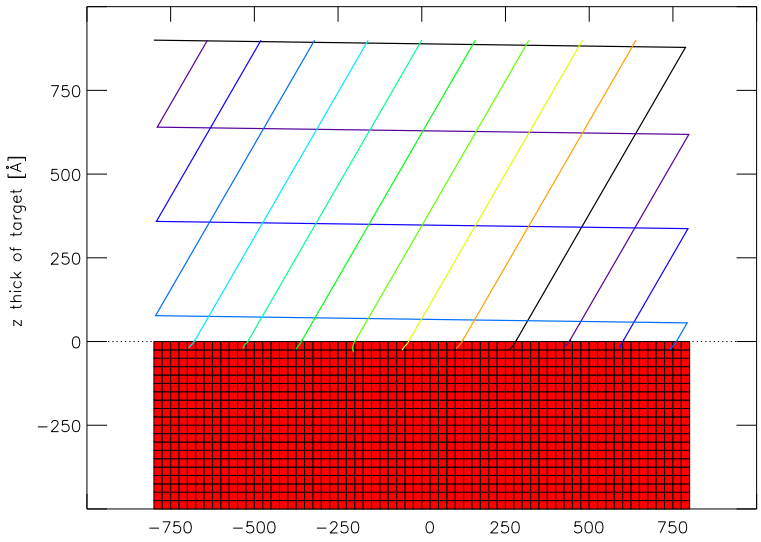


Figure 39: The target simulation $\Delta x_{beam} = 1600\mathring{A}$ $\alpha = 30°$ or whole target-surface this corresponds to a periodic target (or infinite target) and infinite width beam.

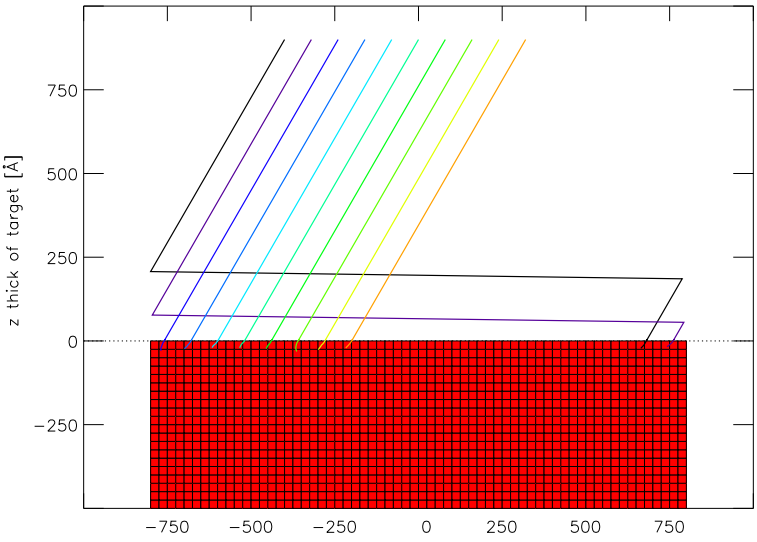


Figure 40: The target simulation $\Delta x_{beam} = 800\mathring{A}$ $\alpha = 30°$ this corresponds to a periodic target (or infinite target) and periodic beam with constant width

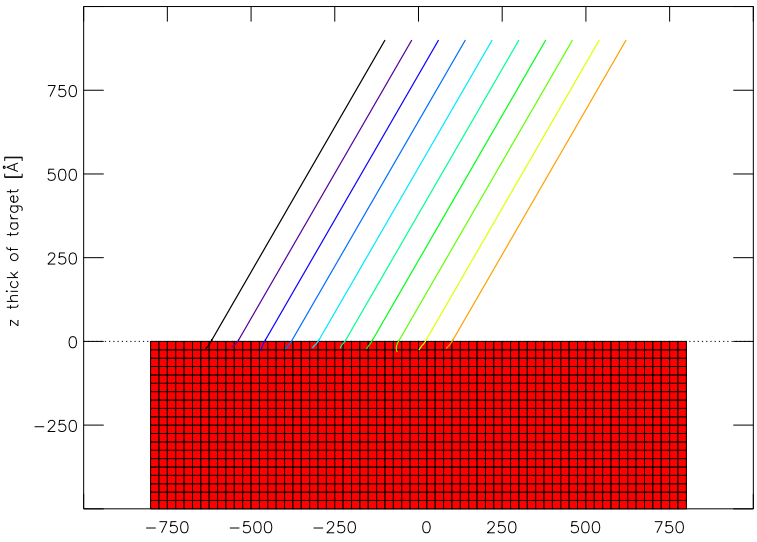


Figure 41: The target simulation $\Delta x_{beam} = 800\mathring{A}$ $\alpha = 30°$ this corresponds to a periodic target or infinite target and periodic beam with constant width

10.4. Initialization data of target

10.4.1. Composition of the target bulk

The bulk composition is controlled by the input parameter **qu_tar** in **tri.inp**. qu_tar has to be provided with one value for each components of the calculation. The sum must be one. A maximum ratio of **two** components in the target can be controlled by the input parameter **qu_max**, its ratio during a dynamic calculation will not exceed this value. 0.0 or 1.0 can use for all components.

Example with **ncp** = 5:
qu_tar = 0.000, 0.017, 0.850, 0.051, 0.082

qu_max = 0.500, 0.017, 1.000, 1.000, 1.000

10.4.2. Initialization of a layered target with 'initial_composition.inp'

A target with several horizontal layers of different composition can be defined by using the input parameter **iq0** in **tri.inp**. With **iq0 = 0** the target composition is uniform, taken from **qu_tar**. By setting **iq0 > 0**, the layer information is read from the additional input file '**initial_composition.inp**'. An example of inputfile 'initial_composition.inp' is given as follows:

```
#number layer, z, qu(1..ncp) Ar -> Si Ta      ! --text
2                                             ! --number layers
  -500                                        ! --start depth 1.layer (may be < surface target)
    0.000     0.500     0.400 ! --qu(1..ncp)
  -1000                                       ! --start depth 2.layer
    0.000     0.200     0.800 ! --qu(1..ncp)
  -1500                                       ! --end last layer (may be > target)
```

The first line is a comment. The number in the second line gives the number of layers that are given, followed by the borders of the layers and the concentration of all elements. The sum of all concentrations is 1.

Fig. 42 shows a plot of the target corresponding to the example given above. The color in the picture indicates the Si concentration and thus shows the two layers defined in '**initial_composition.inp**'. Areas outside of the defined layers are filled with the bulk composition from **qu_tar**, which was chosen to be 100% Si here.

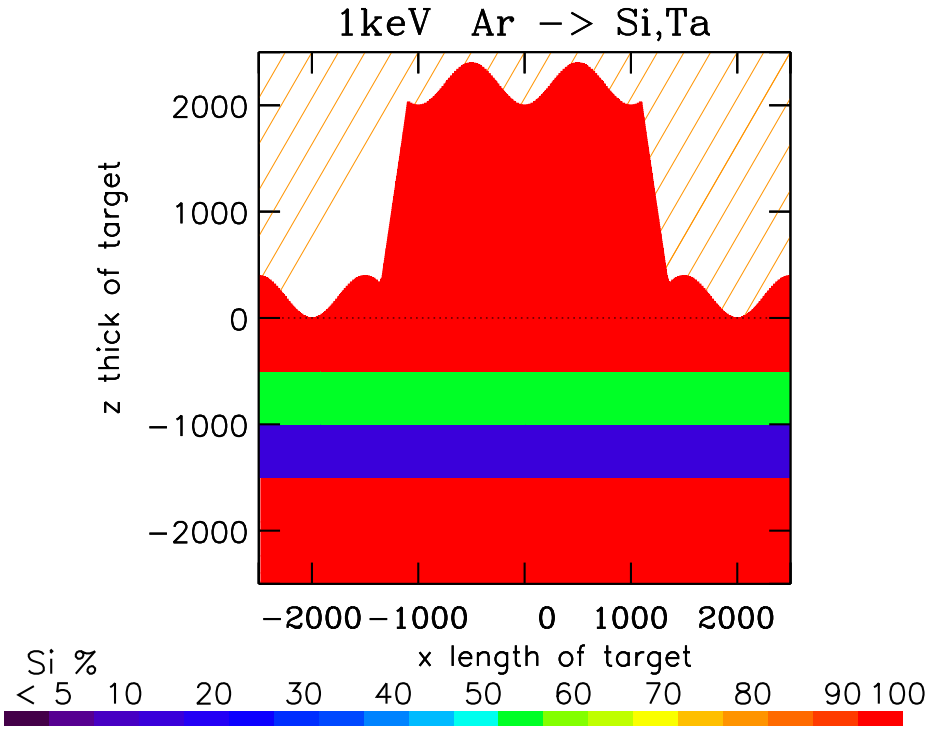


Figure 42: An example with layer-composition. A plot of a target corresponding to the given example 'initial_composition.inp'. The elements of this calculation were chosen to be Ar, Si and Ta, with the Si concentration is plotted (outputfile: T_10_2D.dat).

10.4.3. Initialization of surface layers with 'initial_composition_surf.inp'

By setting the input parameter **iq0_surf = 1**, it is possible to create surface layers that follow the structure of the surface, with an example being shown in Fig. 43. This method can be used to take into account an oxide layer or a deposited film on top of a rough surface.
The input file '**initial_composition_surf.inp**' that was used for the example in Fig. 43 looks as following (symbol ! is only the begin of comment) :

```
number, distant from surface, qu(1..ncp) Ar -> Si,Ta      !  —text
2                                        ! —number layers
 0.0                                     ! —start at surface
    0.000      0.350      0.650  ! —qu(1..ncp)
-100.0                                   ! —start 2. layer from surface
    0.000      0.450      0.550  ! —qu(1..ncp)
-200.0                                   ! —end last layer from surfac
```

The structure of this file is the same as for '**initial_composition.inp**', which was presented in the previous chapter. But the z-values are the distance from the local surface.

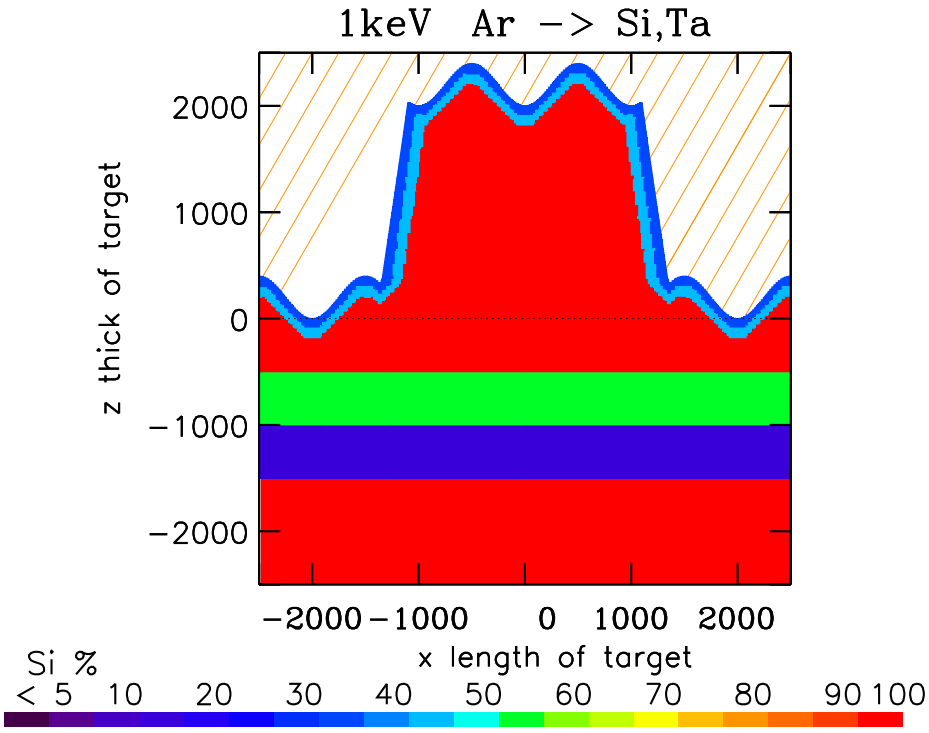


Figure 43: An example with both input 'initial_composition.inp' and 'initial_composition_surf.inp'. The plot of a target corresponding to the given examples. The elements of this calculation were chosen to be Ar, Si and Ta. The Si concentration is plotted (outputfile: T_10_2D.dat).

10.4.4. Initialization of doped target (example W cluster in Fe-target)

calculate cluster
Switch of calculate cluster-cells are: l_clust
n_clust
qu_clust

Ratio of cluster cells to normal cells is 1:9. The size of the cluster are one cell (1x1), 4 cells (2x2), 9 cells (3x3), 16 cells (4x4) and 36 cells (6x6). The places of cluster is random, see Fig. 44 and 45. After start it is generated a outputfile 'doped_zellen_2D.inp.dat'. This may be use as input for next run (copy 'doped_zellen_2D.inp.dat' to' doped_zellen_2D.inp' and use option l_clust_read=.true.).

Example: D − > Fe with W-cluster
ncp = 3
symbol ="D","Fe","W"
qu_tar = 0.000, 1.000 ,0.000 ... Fe-target
l_clust=.true.
n_clust=36
qu_clust=0.000, 0.000 ,1.000 ... W-cluster

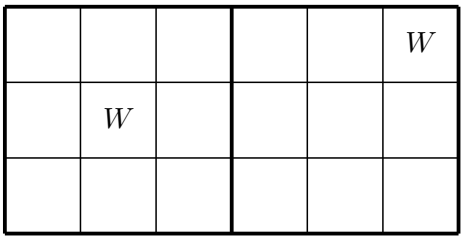


Figure 44: Possible places of one W-cell in two cell-cluster if n_clust=1
(one W - cell in 9 cells − > 1:9)

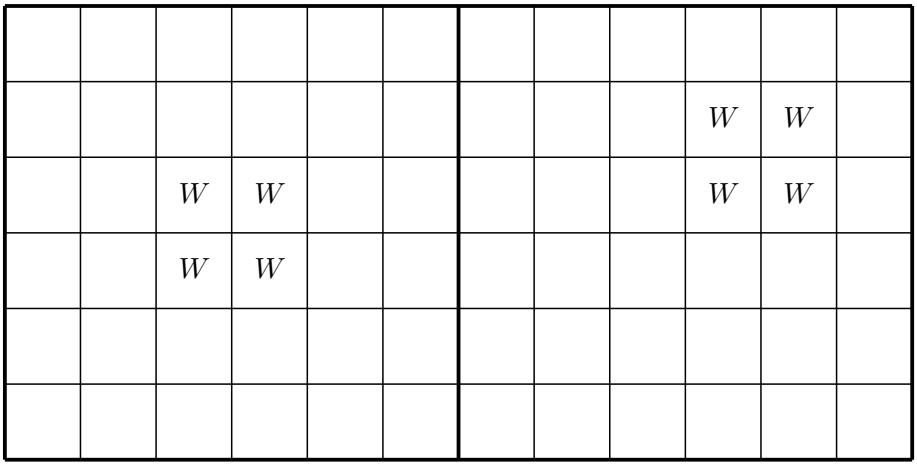


Figure 45: Possible places of four W-cells in two cell-cluster if n_clust=4
(four W-cells in 36 cells − > 1:9)

It may be run with cluster-inputfile (point 1-3 of procedure) or without cluster-inputfile (only point 1 of procedure).

read cluster-inputfile
Switch of read cluster-inputfile 'doped_zellen_2D.inp' are:
l_clust
l_clust_read

procedure:
1. first run without inputfile, option:
   l_clust=.true.
   n_clust=9 (example)
   qu_clust=0.000, 0.000 ,1.000 (example))
   l_clust_read=.false.

2. copy 'doped_zellen_2D.inp.dat' to ' doped_zellen_2D.inp'

3. next run with inputfile ' doped_zellen_2D.inp', option::
   l_clust=.true.
   l_clust_read=.true.

Example: D − > Fe with W-cluster
tri.inp:
ncp = 3
symbol =”D”,”Fe”,”W”
l_clust=.true.
l_clust_read=.true.

doped_zellen_2D.inp (only four cluster) :

```
#doped cell: ix, iz, qu(1:ncp)          !text
5 216 216                               !number cells, max(ix), max(iz)
  23 150      0.00000 0.00000 1.00000
 110  16      0.00000 0.00000 1.00000
 119  78      0.00000 0.00000 1.00000
 196  99      0.00000 0.00000 1.00000
 212 176      0.00000 0.00000 1.00000
```

with:

$$max(ix) = 2 \cdot geo\_x/geo\_dx$$
$$max(iz) = geo\_z/geo\_dz$$

Fig. 45 shows the change of the target if bombarded with D-atoms. The sputtering of Fe is greater than sputtering of W.

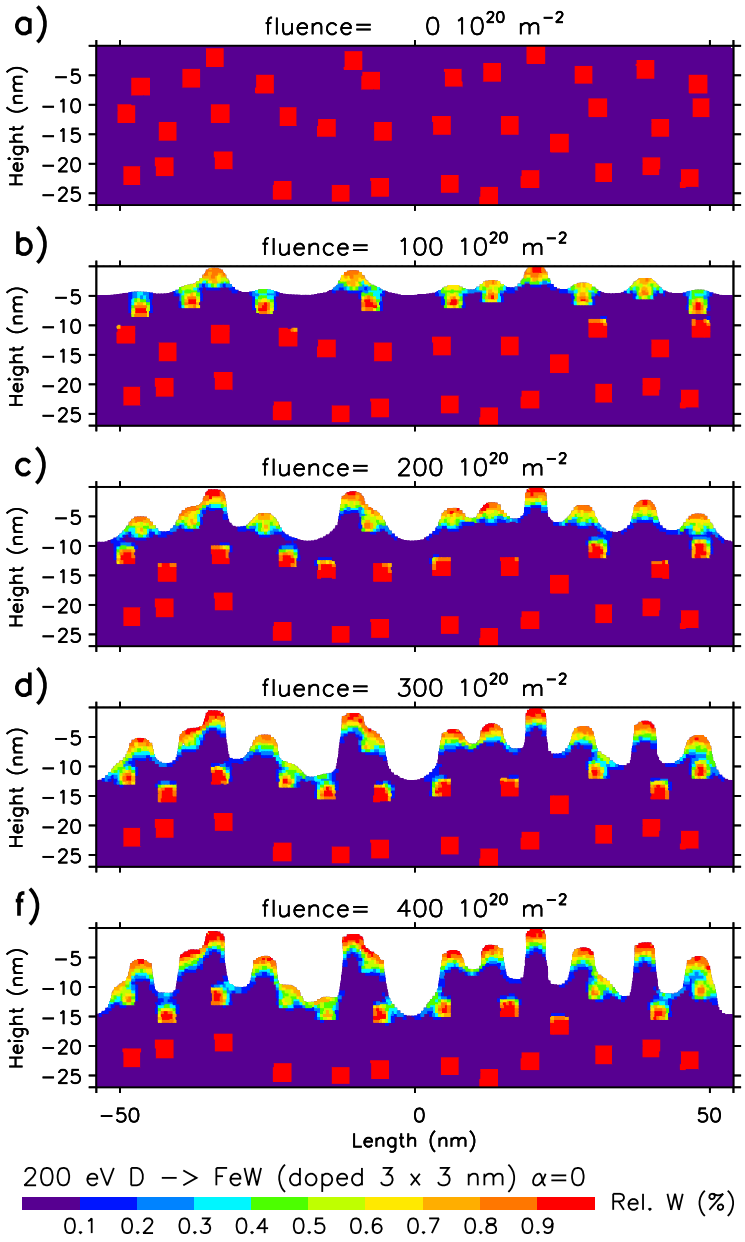


Figure 46: Simulation of Fe-Target with doped W cluster, [11], (outputfile: T_10_2D.dat).

11. Two-dimensional examples with SDTrimSP-2D

11.1. Target with smooth start-surface

Target with smooth start-surface can be calculated with the switch:
'case_geo = 10'
or
'case_geo = 20' .

11.1.1. 5000 eV Xe on Si

One example is the bombarding of a silicon-target with xenon. The thickness of cells is $\Delta z = 10$ Å$= 1\ nm$. The energy of the beam is 5000 eV, the fluence is 2.0 atoms/Å^2. The Fig. 47 (case_geo = 10) and Fig. 48 (case_geo = 20) show the simulation of target with constant layers like 1-D-simulation (Fig. 20).

The decrease of surface (sputter depth) in the model SDTrimSP is:

$$SD_{1D} = 57.5 \text{ Å} \tag{11.18}$$

The decrease of surface of the 2-D-Model in case of 1-D-simulation (case_geo = 10) with $\Delta x, \Delta y >> \Delta z$ is shown in Fig. 20 and Fig. 47. The surface is ideal smooth and the decrease of surface is:

$$SD_{2D}(smooth) = 58.7 \text{ Å} \tag{11.19}$$

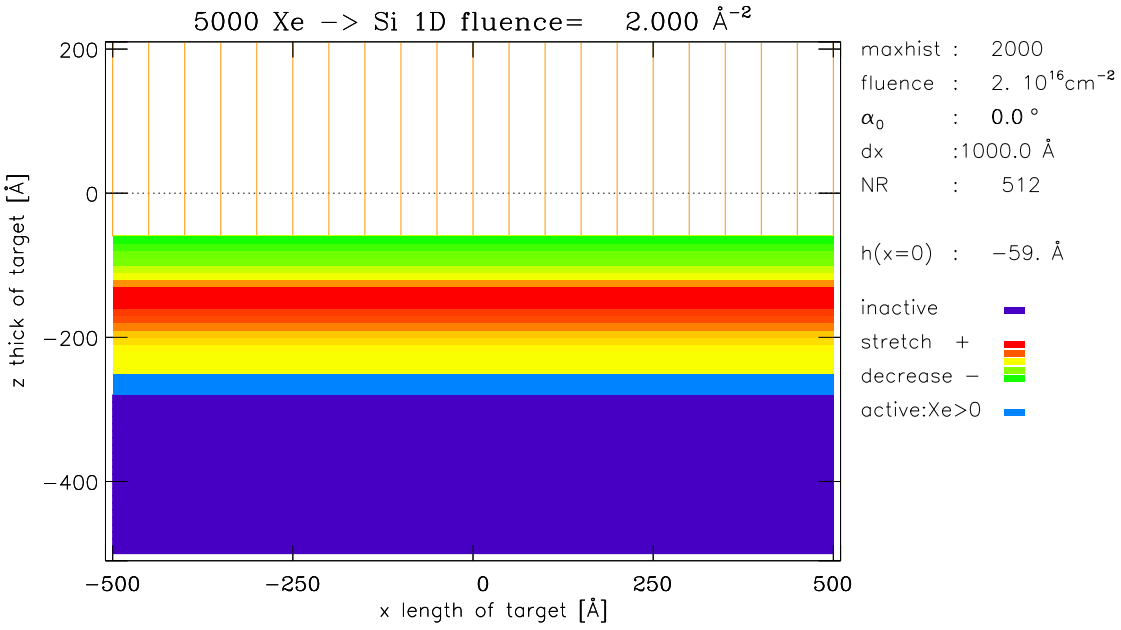


Figure 47: The target simulation with SDTrimSP-2D in case of 1-D-structure (case_geo = 10) (outputfiles: T_10_2D.dat, activ_cell.dat).

The change of target of a 2-D simulation ($\Delta y >> \Delta x, \Delta z$; case_geo = 20) with different beam widths are shown in Fig. 48 and Fig. 49.
The decrease of surface or maximum sputtered depth (SD) in the center of the beam $x = 0,\ y = 0$ are listed for various beam widths in Tab. 1. In the case of very small width of the beam the calculated depth of material loss is greater, than in the cases with broader beams. The simulation with a periodical beam (infinity in X (and Y) direction) produce similar results as the 1-D or the 2-D(smooth) calculations with appropriate resolution in X-direction.

nr.	Δx of beam	maximum sputtered depth (SD)
1	50 Å	115 Å
2	100 Å	111 Å
3	200 Å	78 Å
4	400 Å	72 Å
5	periodic: 1000 Å	57.7 Å
6	1-D-layers	58.7 Å

Table 1: Maximum of sputtered depth in the center of beam

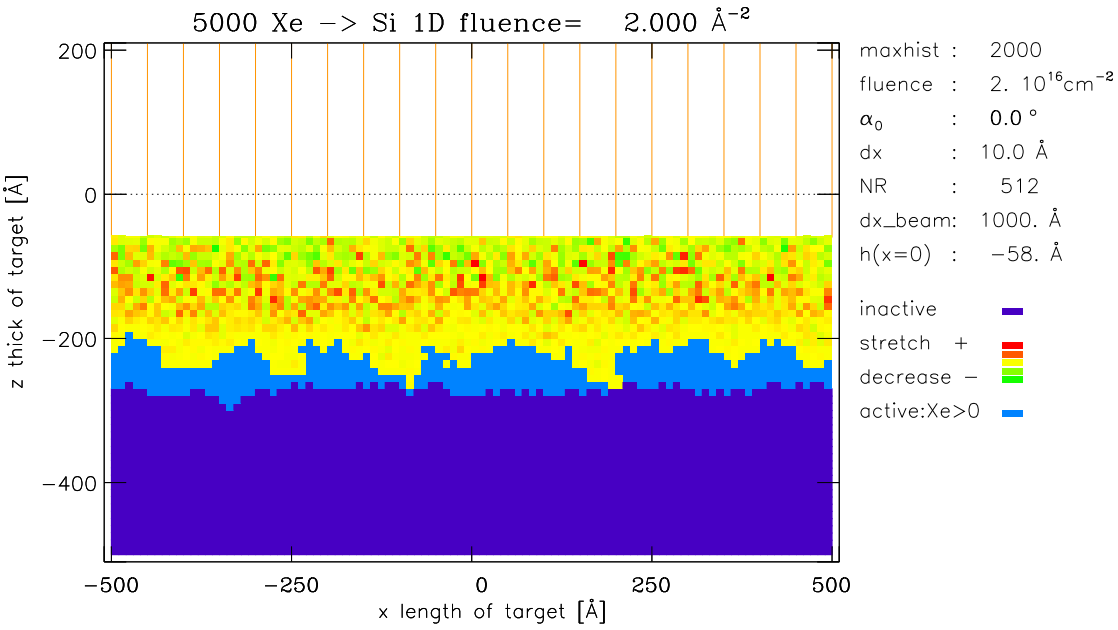


Figure 48: The target simulation $\Delta x_{beam} = 1000\mathring{A}$ this corresponds to a periodic target or infinite target (outputfiles: T_10_2D.dat, activ_cell.dat).

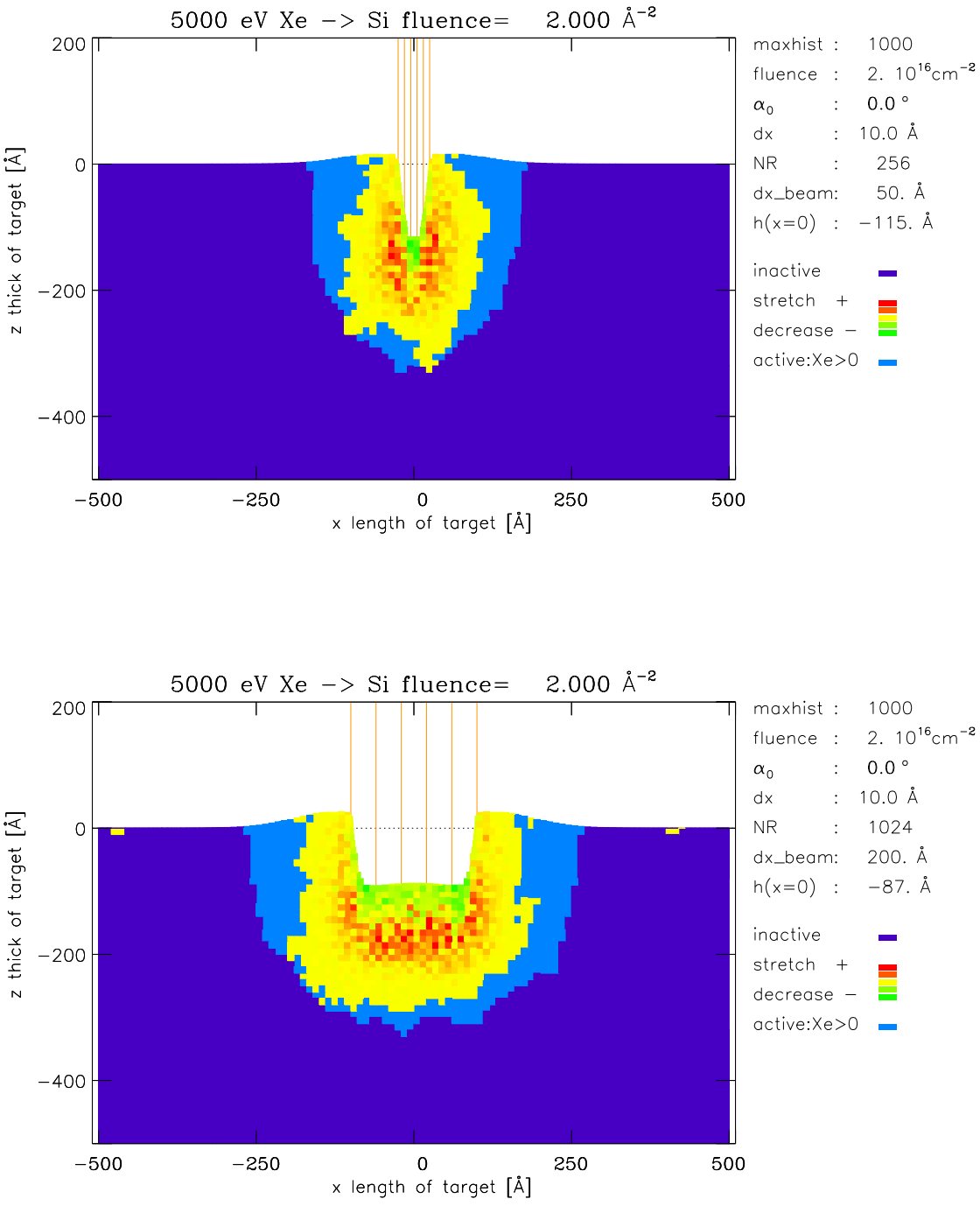


Figure 49: The target simulation $\Delta x_{beam} = 50$ Å and $\Delta x_{beam} = 200$ Å (outputfiles: T_10_2D.dat, activ_cell.dat).

11.1.2. 5000 eV W on C-target with incidence angle of α=30°

Another interesting case is the bombardment with an incident angle greater than zero, like the bombardment of C by W with an incident angle of $\alpha = 30°$. The fluence dependent deposition of tungsten and the appearance of a hole is shown in Fig. 50. The start-target has a smooth surface (case_geo = 20).

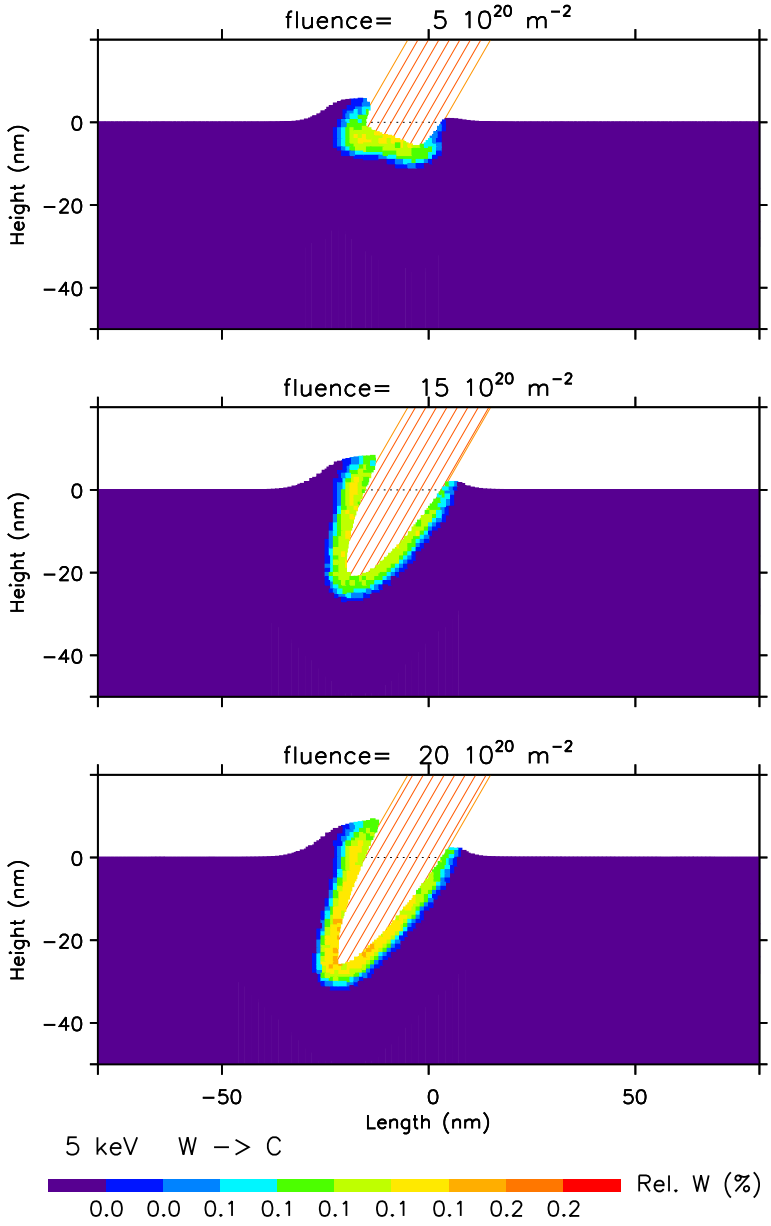


Figure 50: Simulated fluence dependent deposition of tungsten at 5 keV W on C target for an incident angle of $\alpha = 30°$ (case_geo = 20) (outputfile: T_10_2D.dat)

11.2. Target with start structure (roughness)

Target with a start-surface (roughness) can be calculated with the switch 'case_geo' greater then 20.

11.2.1. 6000 eV Ar on Si target with periodic structure

The structure of the used real target (pitch grating) is shown in Fig. 51. The surface is periodic in the X-direction and independent of the Y-direction (infinite), i.e a perfect 2D surface.

The switch for the calculation is 'case_geo'=25. This option use the inputfile 'rauh.inp'. An example of input-file 'rauh.inp' of real pitch grating, Fig. 14 and Fig. 51, is:

```
pitch grating (x,y Values)
6
-2500.0   0000.0
-1350.0   0000.0
-1100.0   2000.0
 1100.0   2000.0
 1350.0   0000.0
 2500.0   0000.0
```

The first line is a comment. The number in the second line gives the number of points that are given, followed the x and z-values.

The nano-structured specimen is fabricated on a Si wafer with an intermediate Ta layer. The Ta layer is used as a reference marker to allow quantitative measurements of the Si layer thick. Fig. 14 shows the cross-section at the beginning of bombardments.

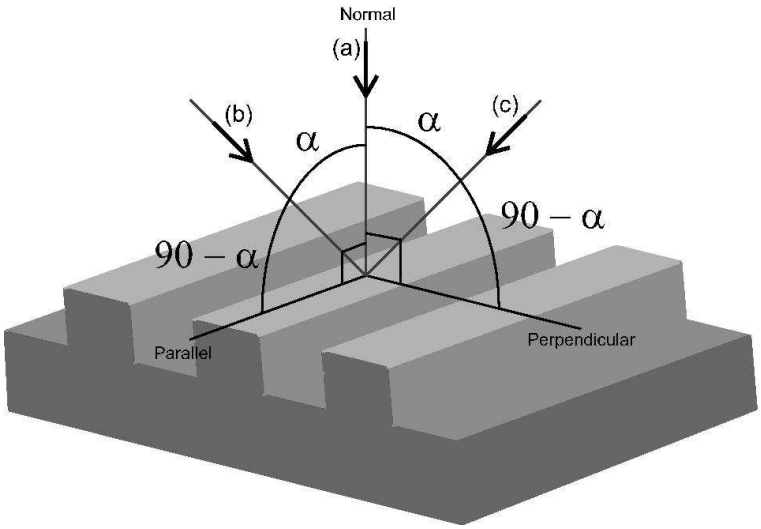


Figure 51: Structure of pitch grating and direction of normal incident $\alpha = 0°$ (a) and $\alpha = 42°$ (c)

With increasing fluence the target shrinks. The agreement of calculated surface (red line) after bombardment with the experiment is very good, Fig. 52 and 53 a-b.

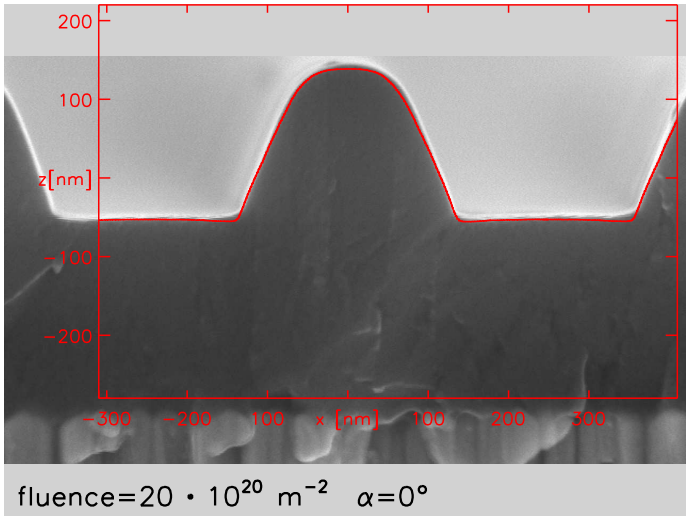


Figure 52: Comparison of calculated surface (red) with experiment from [6] of 6 keV Ar on Si normal incident at fluence $20 \cdot 10^{20}$ $atoms/m^2$ (outputfile: surf.dat).

The next experiment is the bombardment of the same target at an incident angle $\alpha = 42°$, Fig. 53 c and 53 d. The agreement of calculated surface (red line) with the experiment is also very good.

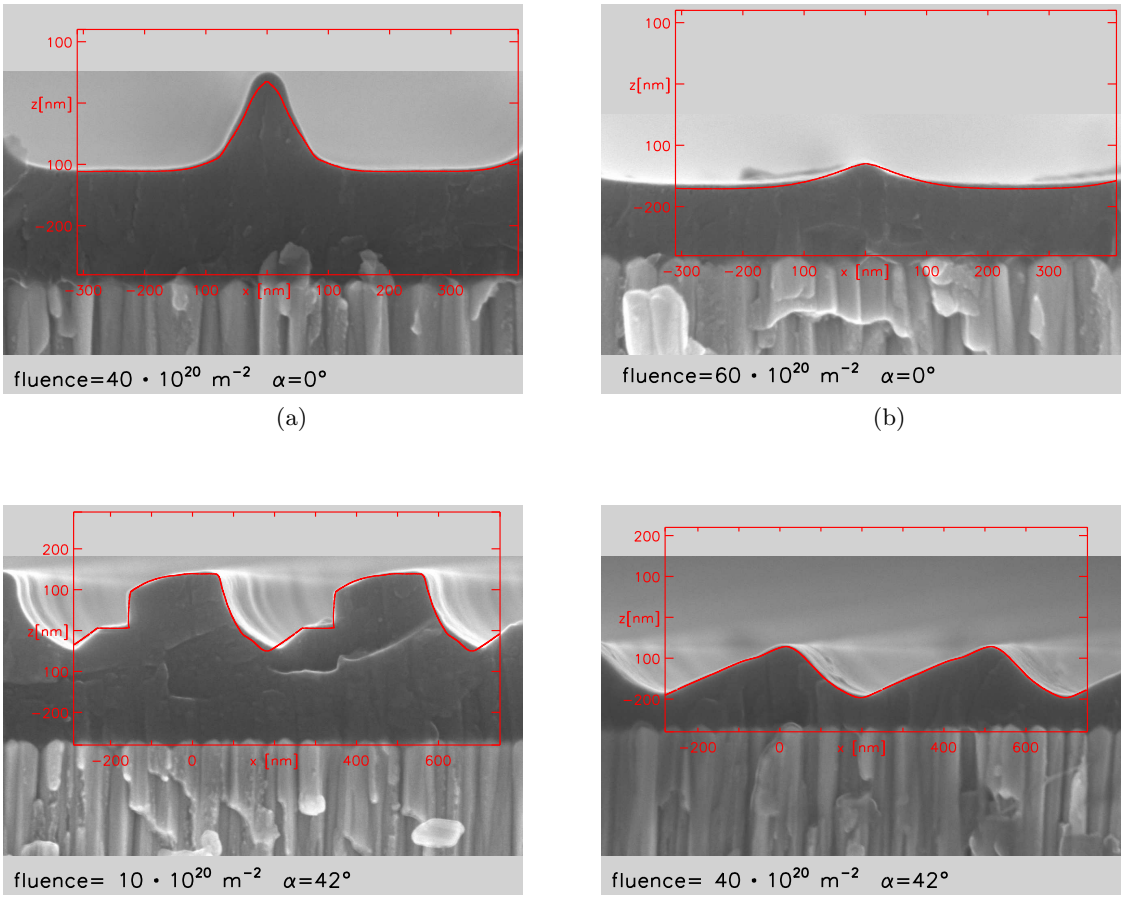
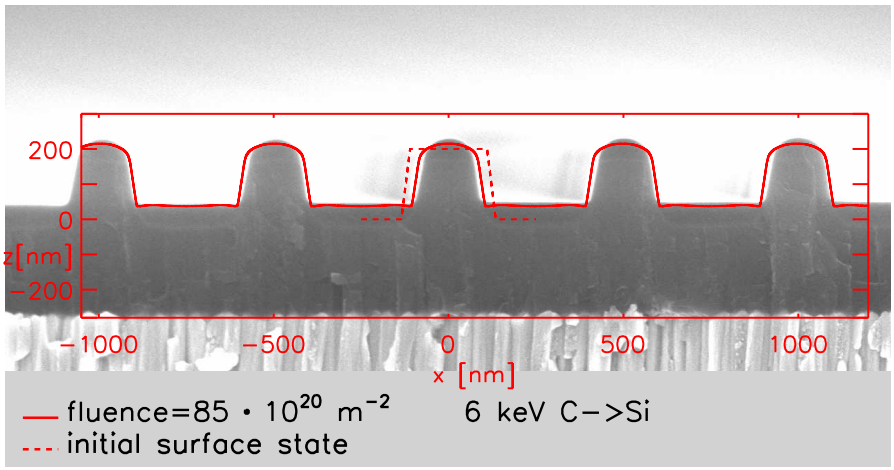


Figure 53: Comparison of calculated surface (red) with experiment from [6] of 6 keV Ar on Si, (a) and (b) incident angle $\alpha = 0°$, (c) and(d) incident angle $\alpha = 42°$ (outputfile: surf.dat).

11.2.2. 6000 eV C on Si target with structure

The structure of the used target (pitch grating) is the same as in Chapter 11.2.1 and in Fig. 51. If a silicon-target is bombarded with carbon more material is implanted as sputtered. Fig. 54 shows the comparison of simulated surface with the experiment for a fluence of $85 \cdot 10^{20}$ $atoms/m^2$. The agreement of the calculated surface (red line) with the experiment is very good.
The comparisons of experimental data and simulation of areal density are shown in Fig. 55. The areal density of C atoms is growing due to the implantation. The surface density of Si is reduced by sputtering.



left

Figure 54: Comparison of calculated surface (red) with experiment [8] of 6 keV C on Si normal incident at fluence $85 \cdot 10^{20}$ $atoms/m^2$ (outputfile: surf.dat).

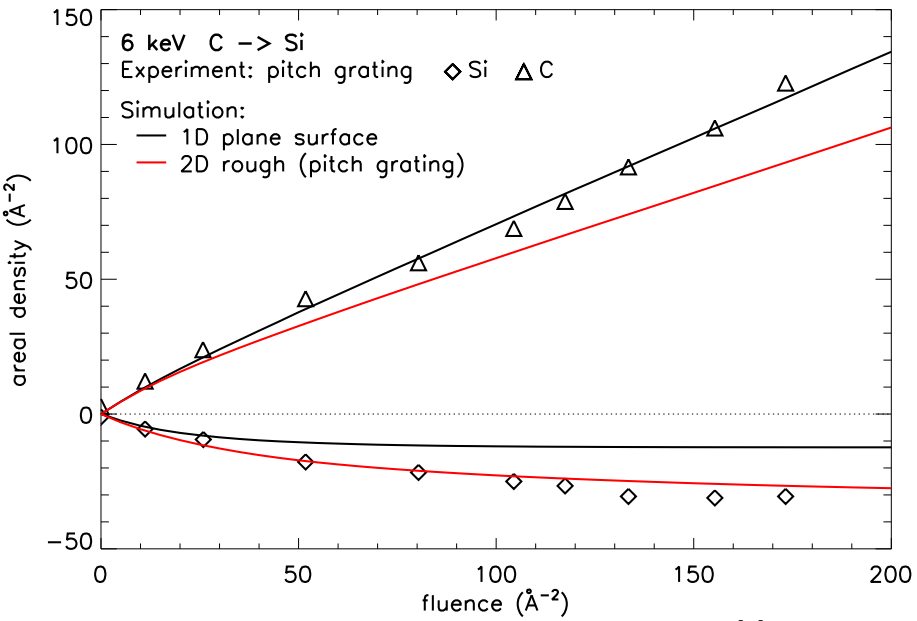


Figure 55: Comparison of the experimentally measured [9] and simulated fluence dependent variation of Si and C areal density normal angle of incidence (outputfile: E0_31_3D.dat).

The fluence dependent deposition of carbon is shown in Fig. 56. The simulation shows the implantation of carbon. At the end of the calculation the surface is completely covered with carbon.

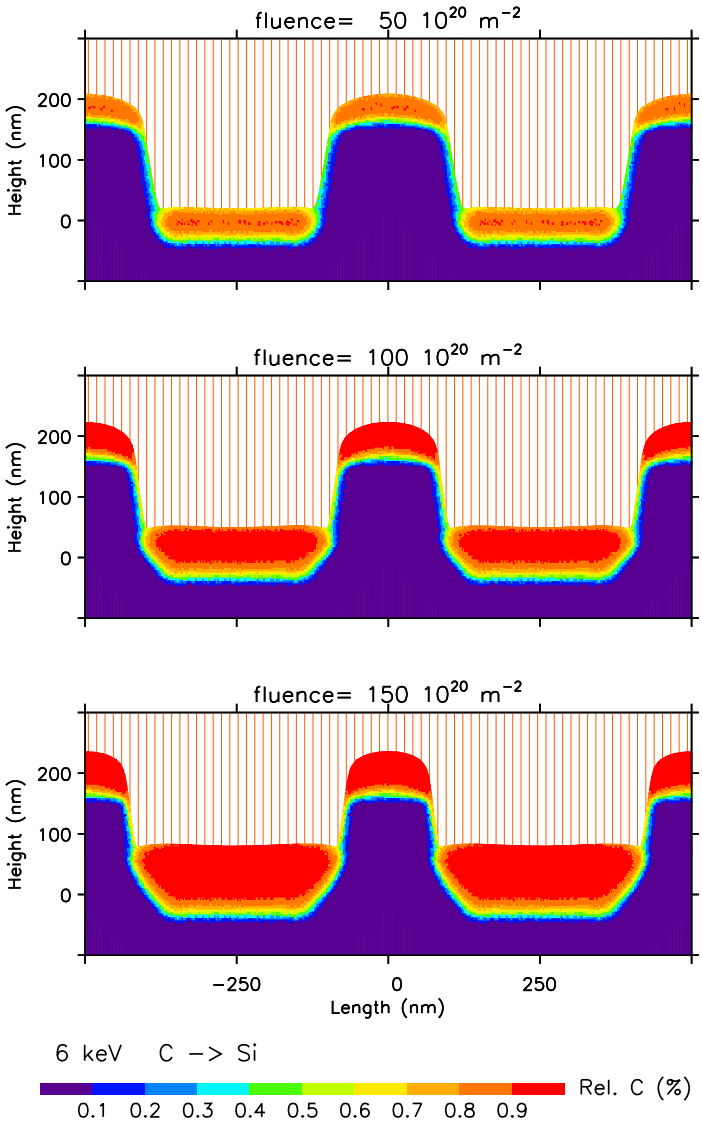


Figure 56: Simulated fluence dependent deposition of Carbon at 6 keV C on Si target normal angle of incidence (outputfile: T_10_2D.dat).

11.3. Restart-File

11.3.1. Output of Restart-File

The use of a restart-file is important for long runs. The program can be break, parameter can be changed and the program-run can be continued at a later date.

Default is: **l_write_restart=.false.**

In combination with the switch **ihist_out** the output is continuous during the run. The name of the output-file is formed with the history-number.

If ihist_out = 100 then:
Restart_0000000102
Restart_0000000202
Restart_0000000302 and so on.

If the programme is interrupted at ihist = 363 the program can be continued from ihist=302. All output-files til ihist = 300 are ok and can be updated for next output-step ihist=400.
Another possibility is a new start with the target from restart-file with ihist = 1. In this case all output-files are deleted (or overwritten).

The program uses the restart-input-file **Restart_data.inp**. Therefor it needs to be renamed the restart-file and the first word in the tri.inp must be 'RESTART'

copy or rename 'Restart_0000000302' to 'Restart_data.inp'
change 'tri.inp'

The second necessary input-file is **restart.inp**. If the first word is **.false.** then the program continued.
If the first word is **.true.** then the program start new and the user can change some input-value like in tri.inp.

11.3.2. Continue of program with Restart-File

Necessary files : 'tri.inp', 'restart.inp' and 'Restart_data.inp'

1. copy or rename 'Restart_0000000302' to 'Restart_data.inp'
2. change 'tri.inp'
3. change 'restart.inp'

New (changed) **Inputfile 'tri.inp'** with namelist of TRI_INP:

```
RESTART 10 keV Ar -> Si
&TRI_INP
   text='---elements---'
      ncp = 2,
      symbol = "Ar", "Si"
            .
            .
            .
      idout = 100
      l_write_restart=.true.
            .
            .
/
```

Inputfile 'restart.inp' with empty namelist of INPUTR:

```
.false.
&INPUTR
/
```

11.3.3. New-start of program with Restart-File

Necessary files : 'tri.inp', 'restart.inp' and 'Restart_data.inp'

1. copy or rename 'Restart_0000000302' to 'Restart_data.inp'
2. change 'tri.inp'
3. change 'restart.inp'

New (changed) **Inputfile 'tri.inp'** with namelist of TRI_INP:

```
RESTART 10 keV Ar -> Si
&TRI_INP
   text='---elements---'
      ncp = 2,
      symbol = "Ar", "Si"
            .
            .
            .
      idout = 100
      l_write_restart=.true.
            .
            .
/
```

Inputfile 'restart.inp' with namelist of INPUTR:

```
.true.
&INPUTR
         tar_dynamic = .false.
         lmatrices = .true.
/
```

The value of tar_dynamic and lmatrices from the restart-file will overwrite. All other values from Input-file 'tri.inp' are ignored.

Some of input-values from 'tri.inp' (find in the restart-file) may be overwrite.
Optional input variables for namelist INPUTR in 'restart.inp' are:

maxhist, ihist_out,nr_pproj, flc, ipot, isbv, irand, case_e0, de0_beam, e0_beam, matrix_e_min_p, matrix_e_max_p, matrix_e_min_r, matrix_e_max_r, case_alpha, alpha0, dalpha0, l_write_restart, e_surfb, e_displ, e_cutoff, tttt, tar_dynamic, lmatrices

11.4. Local yield

Locale Y is only useful if a homogeneous beam hits the surface. N is number atoms.

$$Y_{local}(dA_i) = \frac{N_{out}(dA_i) - Nin(dA_i)}{N_{Projectile}(dA_i)} \tag{11.20}$$

$$N_{Projectile}(dA_i) = N_{Projectile}(all) \cdot \frac{dA_i}{A_{beam}} \qquad A_{beam} = k \cdot dA_i \tag{11.21}$$

The total Yield Y is the average of all local yields:

$$Y = \sum_{i=1}^{k} Y_{local}(dA_i)/k \tag{11.22}$$

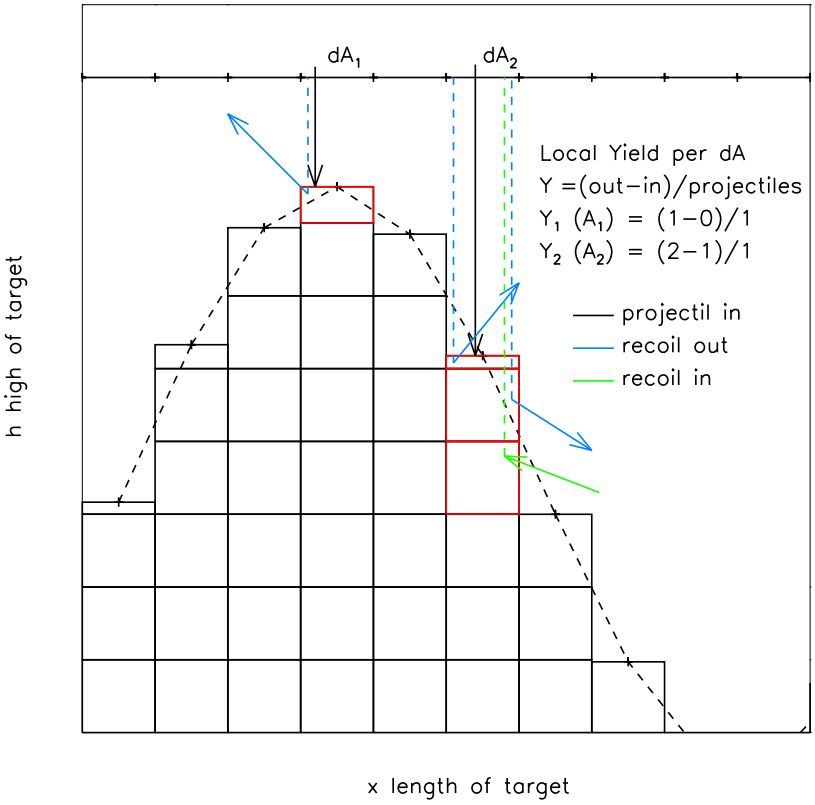


Figure 57: Definition of local yield Y_i

Fig. 58 shows the structure of a target with 45° walls with cascade of one incident projectile. The projectile is implanted on the right side and create some recoils (blue cascade). One recoil leave the right side and is implanted on the opposite side (left side), where it again excites atoms. One of this recoil leave the target (back-sputtered).

The graphic below shows the local-yield. There is a large different between local 2D-Yield (black line) and the calculated yield of 1D-Model (red line).

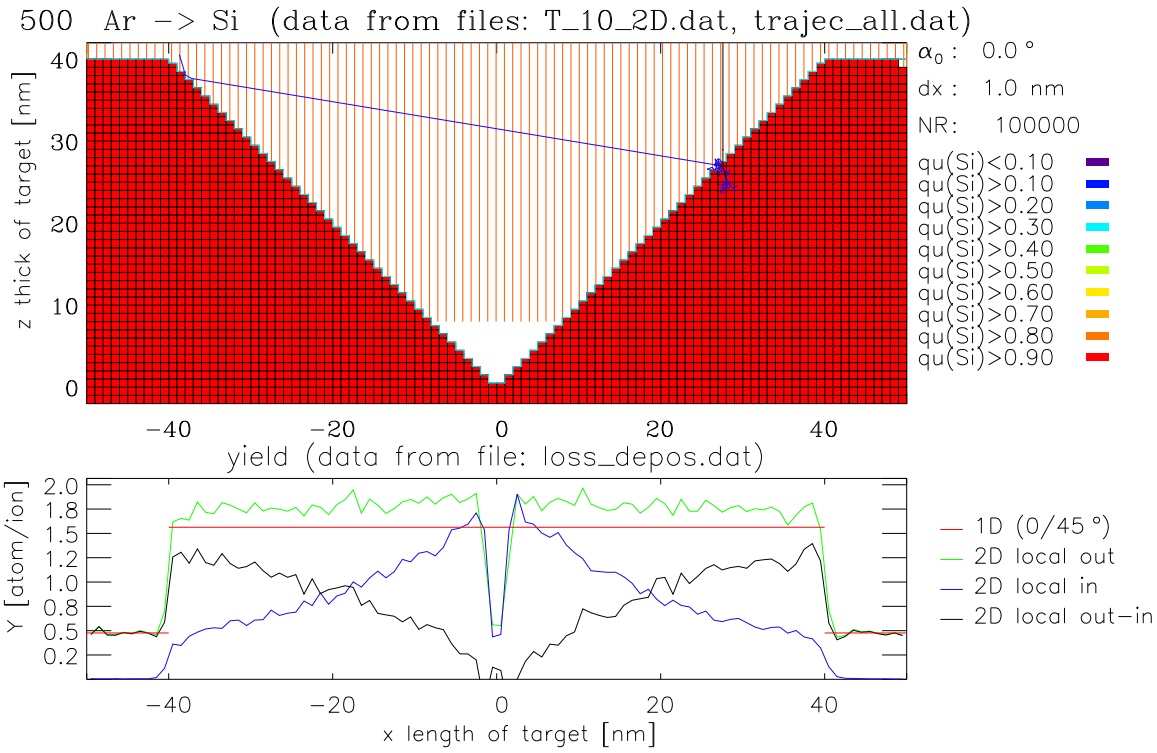


Figure 58: Example of target with one local cascade and local Yield=out/in of a 2D-target with 45° walls
(outputfiles: T_10_2D.dat, trajec_all.dat, loss_depos.dat).

11.5. Comparison of rough and smooth yield and reflection distribution

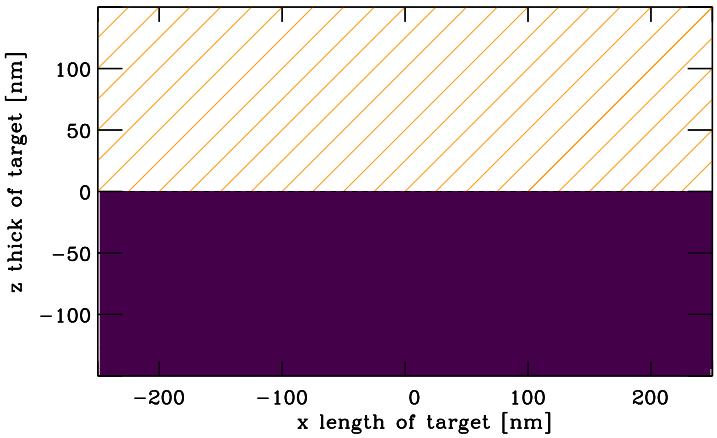


Figure 59: Example 1:
target smooth,
$\alpha = 45°$
reflection: Fig. 62, 68,
yield : Fig. 65, 71

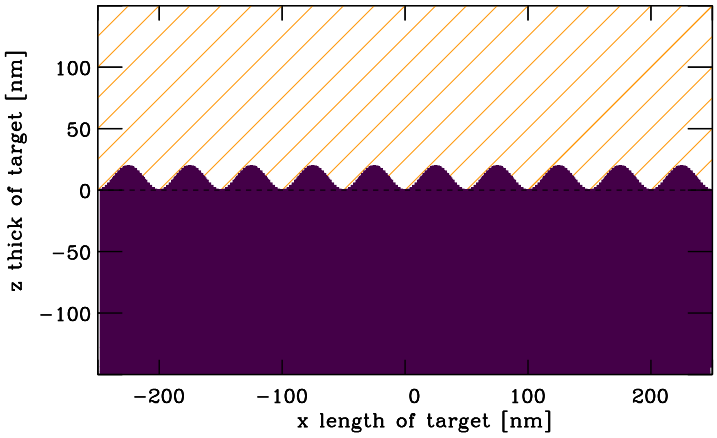


Figure 60: Example 2:
target rough,
amplitude=10 nm,
reflection: Fig. 63, 69
yield : Fig. 66, 72

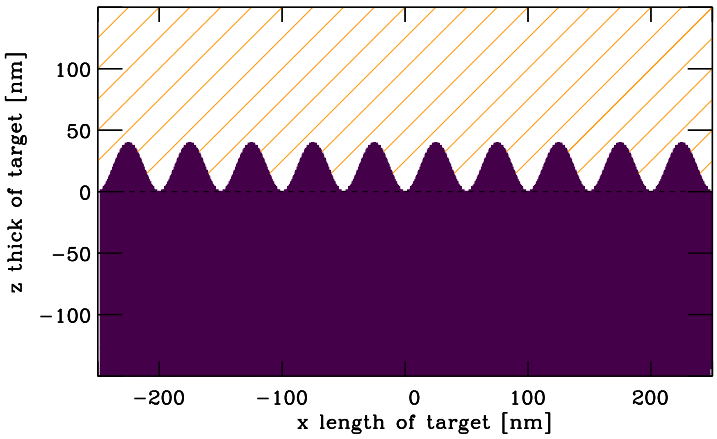


Figure 61: Example 3:
target rough,
amplitude=20 nm,
reflection: Fig. 64, 70
yield : Fig. 67, 73

11.5.1. 1 keV Ar on Si $\alpha = 45°$

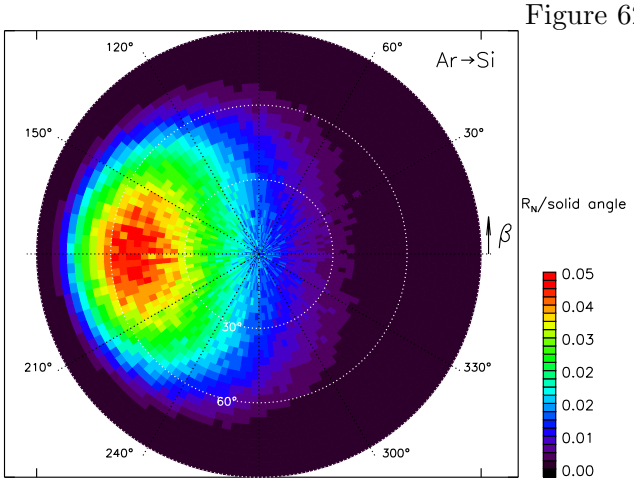


Figure 62: Ar contour-plot of the angular distribution of reflection per solid angle.
1 keV Ar on Si $\alpha = 45°$
smooth surface,
surface and beam see Fig. 59
(outputfile: meagb_p.dat)

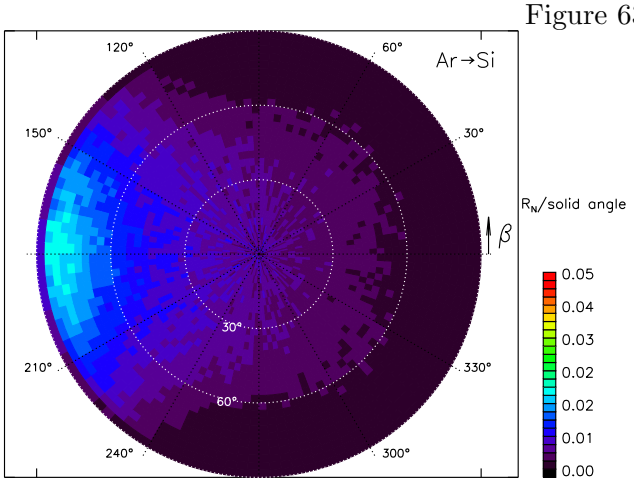


Figure 63: Ar contour-plot of the angular distribution of reflection per solid angle.
1 keV Ar on Si $\alpha = 45°$
rough surface, amplitude=10 nm,
surface and beam see Fig. 60
(outputfile: meagb_p.dat)

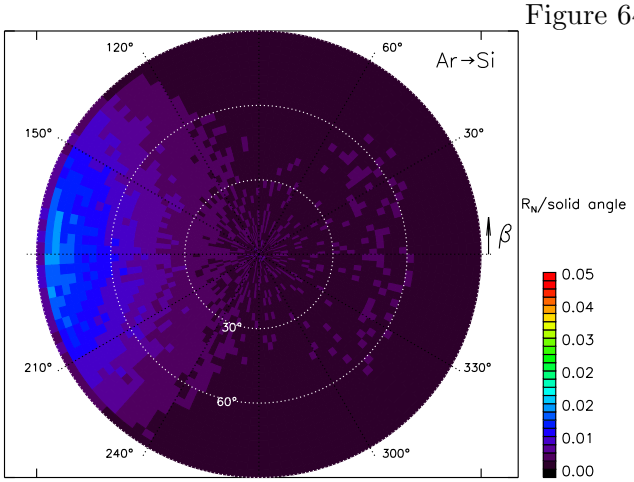


Figure 64: Ar contour-plot of the angular distribution of reflection per solid angle.
1 keV Ar on Si $\alpha = 45°$
rough surface, amplitude=20 nm
surface and beam see Fig. 61
(outputfile: meagb_p.dat)

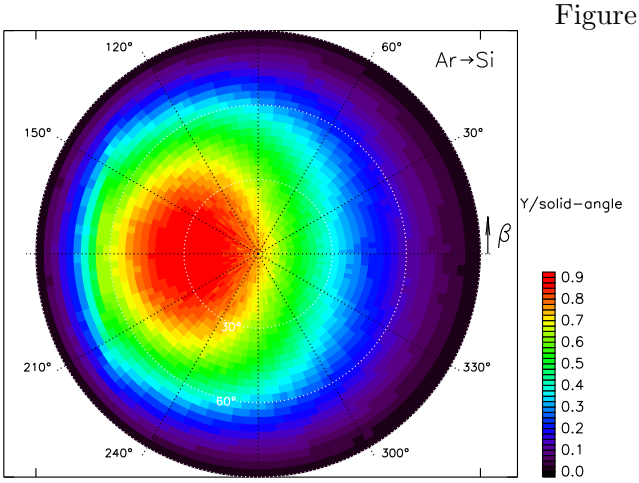


Figure 65: Si contour-plot of the angular distribution of yield per solid angle.
1 keV Ar on Si $\alpha = 45°$
smooth surface,
surface and beam see Fig. 59
(outputfile: meagb_s.dat)

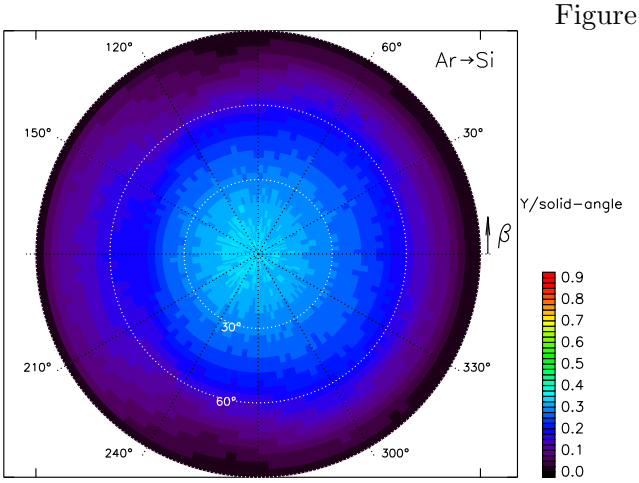


Figure 66: Si contour-plot of the angular distribution of yield per solid angle.
1 keV Ar on Si $\alpha = 45°$
rough surface, amplitude=10 nm,
surface and beam see Fig. 60
(outputfile: meagb_s.dat)

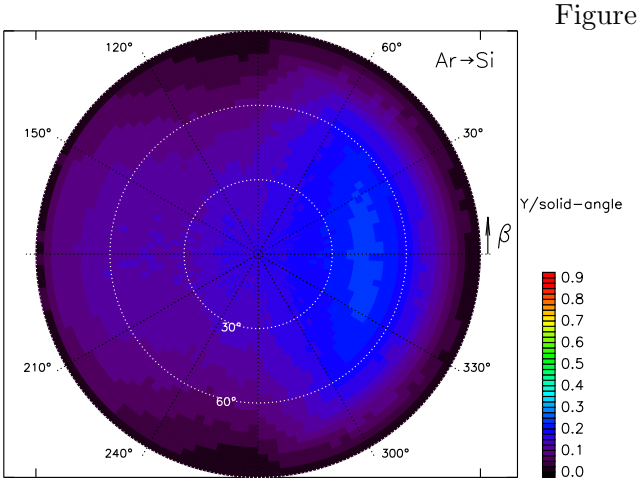


Figure 67: Si contour-plot of the angular distribution of yield per solid angle.
1 keV Ar on Si $\alpha = 45°$
rough surface, amplitude=20 nm
surface and beam see Fig. 61
(outputfile: meagb_s.dat)

11.5.2. 1 keV He on Si $\alpha = 45°$

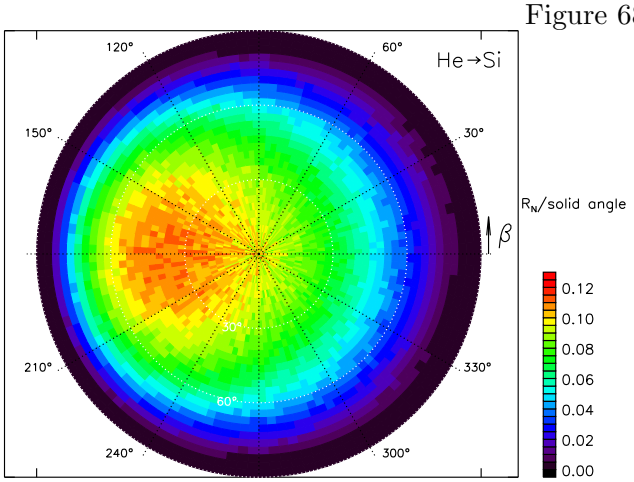


Figure 68: He contour-plot of the angular distribution of reflection per solid angle.
1 keV He on Si $\alpha = 45°$
smooth surface,
surface and beam see Fig. 59
(outputfile: meagb_p.dat)

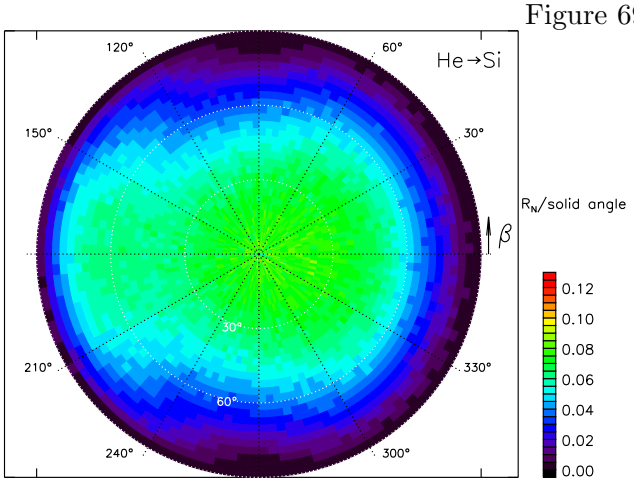


Figure 69: He contour-plot of the angular distribution of reflection per solid angle.
1 keV He on Si $\alpha = 45°$
rough surface, amplitude=10 nm,
surface and beam see Fig. 60
(outputfile: meagb_p.dat)

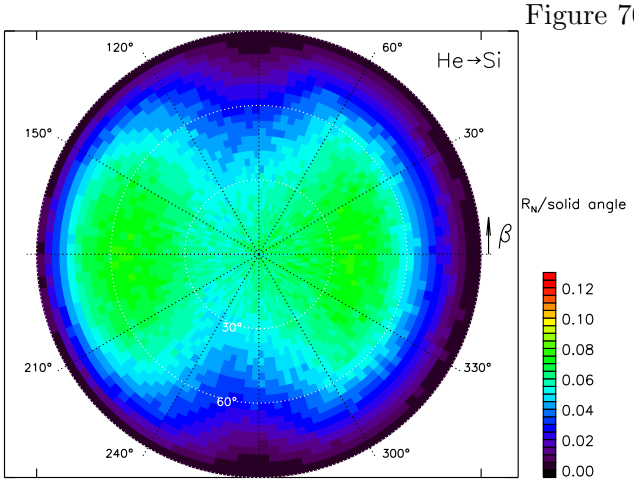


Figure 70: He contour-plot of the angular distribution of reflection per solid angle.
1 keV He on Si $\alpha = 45°$
rough surface, amplitude=20 nm
surface and beam see Fig. 61
(outputfile: meagb_p.dat)

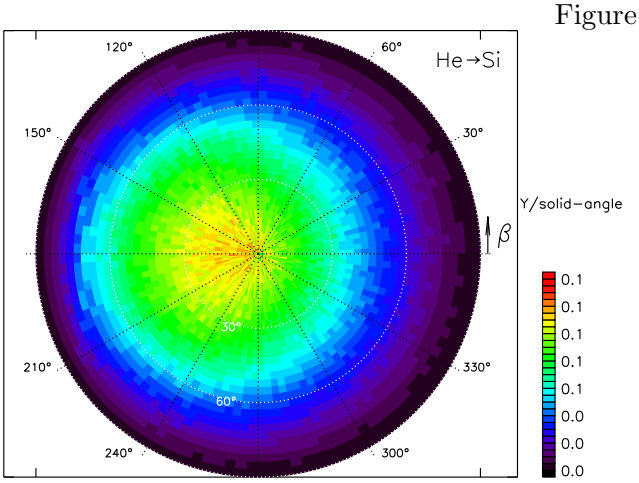


Figure 71: Si contour-plot of the angular distribution of yield per solid angle.
1 keV He on Si $\alpha = 45°$
smooth surface,
surface and beam see Fig. 59
(outputfile: meagb_s.dat)

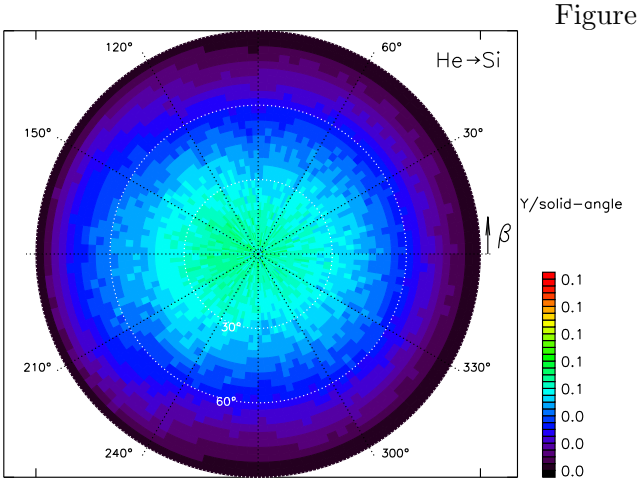


Figure 72: Si contour-plot of the angular distribution of yield per solid angle.
1 keV He on Si $\alpha = 45°$
rough surface, amplitude=10 nm,
surface and beam see Fig. 60
(outputfile: meagb_s.dat)

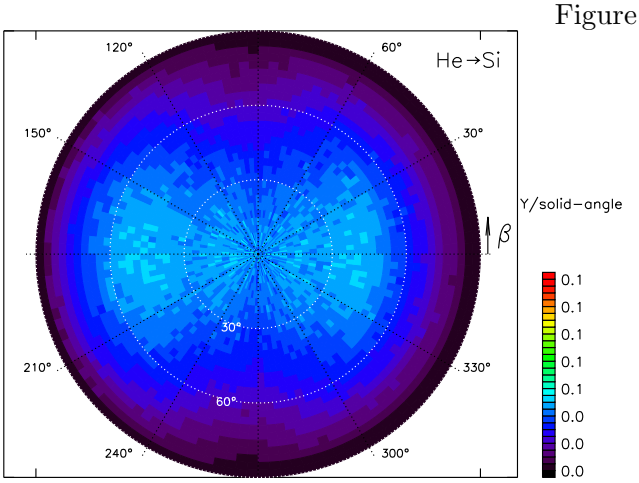


Figure 73: Si contour-plot of the angular distribution of yield per solid angle.
1 keV He on Si $\alpha = 45°$
rough surface, amplitude=20 nm
surface and beam see Fig. 61
(outputfile: meagb_s.dat)

References

[1] A. Mutzke, U. von Toussaint, W. Eckstein, R. Dohmen, K. Schmid, SDTrimSP Version 7.00, IPP Report 2024-06

[2] W. Eckstein, *Computer Simulation of Ion-Solid Interactions*, Springer Series in Material Science, Vol. 10, Springer Berlin, Heidelberg 1991

[3] W. Moeller,W. Eckstein, *TRIDYN-Binary Collision*, IPP 9/64, 1988

[4] W. Moeller,W. Eckstein, *Tridyn-fzr-2001*, 2001

[5] H. Schlager,W. Eckstein, *The scattering Integrals*, IPP 9/69, 1991

[6] I. Bizyukov, A. Mutzke, R. Schneider, J. Davis, *Evolution of the 2D surface structure of a silicon pitch grating under argon ion bombardment: experiment and modeling*, Nucl. Instrum. Meth. vol.B 268, pp. 2631-2638, 2010

[7] I. Bizyukov, A. Mutzke, R. Schneider, A.M. Gigler , K. Krieger, *Morphology and changes of elemental surface composition of tungsten bombarded with carbon ions*, Nucl. Instrum. Meth. vol.B 266, pp. 1979-1986, 2008

[8] A. Mutzke, I. Bizyukov, R. Schneider, J. Davis, *Nano-scale modification of 2D surface structures exposed to 6 keV carbon ions: Experiment and modeling* , Nucl. Instrum. Meth. vol.B 269 pp. 582-589,(2011)

[9] A. Mutzke, I. Bizyukov, H. Langhuth , M. Mayer, K. Krieger, R. Schneider , *Study of interaction of C+ ion beam with a Si pitch grating on a macro-scale level* , Nucl. Instrum. Meth. vol.B 293 pp.11-15 (2012)

[10] M. Wagner, M. Mayer, U. von Toussaint and A. Mutzke (2022), *Simulation of the evolution of rough surfaces by sputtering using the binary collision approximation*, Radiation Effects and Defects in Solids, DOI: 10.1080/10420150.2022.2098751

[11] U. von Toussaint, A. Mutzke, *Fluence dependent changes of erosion yields and surface morphology of the iron-tungsten model system: SDTrimSP-2D simulation studies*, Nuclear Materials and Energy 12 (2017) 318-322

[12] I. Bizyukov, A. Mutzke, R. Schneider, *effect of increasing surface roughness on sputtering and reflection*, ISSN 1562-6016. BAHT. 2012. No6(82)

A. Units of physics terms in the code

name	symbol	unit
mean free path length	l_m	Å$= 10^{-10}m$
random number	R_{random}	$-$
impact parameter	P_{impact}	Å$= 10^{-10}m$
number of atoms	N	$-$
number of projectiles	n_r	$-$
fluence	flc	$atom/$Å^2
geometry of cell	$\Delta x, \Delta y, \Delta z$	Å
pseudo time	τ	s
beam area	A_{flc}	Å^2
volume	V	Å$^3 = 10^{-30}m^3$
volume-change	ΔV	Å^3
atomic fraction	qu	$-$
length	L	Å$= 10^{-10}m$
density	ϱ	$g/$Å$^3 = g/10^{-30}m^3$
atomic density	ϱ_0	$atoms/$Å^3
energy	E	eV

Table 2: Units of physics terms in the code

B. Search algorithm of local cell

The mapping of particles to their corresponding cells is essential, hence the algorithms used in SDTrimSP and SDTrimSP-2D are described in the following chapter.

SDTrimSP

The search algorithm of the actual layer in SDTrimSP is very simple and sketched in Fig. 74. The x-coordinate (x_p) of every particle is known, so that a simple comparison is possible.

The particle is outside the target if x_p is smaller than $-d_{refraction}$. It is part of the refraction layer if $x_p < 0$ *and* $x_p > -d_{refraction}$ is true. If $x_p > 0$ *and* $x_p < -d_1$ than the particle is in the first layer and so on.

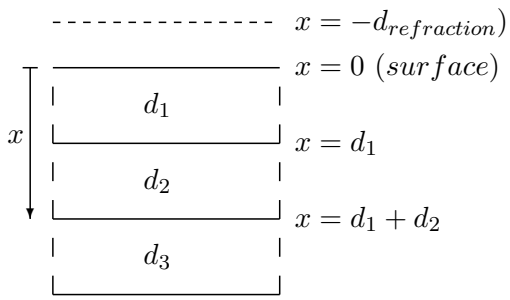


Figure 74: Depth geometry of 1D-target with three layers (d_1, d_2, d_3)

SDTrimSP-2D

The search algorithm of the actual cell in SDTrimSP-2D, in which the particle exists or moves, is computed in the subroutine '*xyz_in_cel*'. A particle can be inside a cell (black line) but also inside of the refraction-thick (red dash line), due to a planar surface potential, Fig. 75. In all other cases it is outside of the target. The x- and z-coordinates of particle (x_p, z_p) and the coordinates of each mesh (x_i, z_k) are known.

The search algorithm can be divided into different steps or cases, that are shown (Fig. 75) and in the following:
In most cases, it is known in which cell the particle was located before it moves (Nr-cell $\neq 0$). Hence, case 1 is the trivial test if the particle is still in the initial cell.

If the cell number is not known or the test of case 1 is negative, the mesh-number is determined due to particles coordinates in x and z direction ($x_{i-1} < x_p < x_i$, $z_{k-1} < z_p < z_k$).

B. Search algorithm of local cell

Exist a cell in this mesh (index-cell $\neq 0$) it is checked whether the particle is in this cell (case 2) or inside the refraction-zone (case 3).

If this test is also negative (no cell inside the mesh, index-cell = 0) then all cells are checked in the neighboring area. The cases (4-11) are top, bottom, left, right, top-left, top-right, bottom-left and bottom-right. It is possible that the particle is inside of the refraction-zone of other cells.

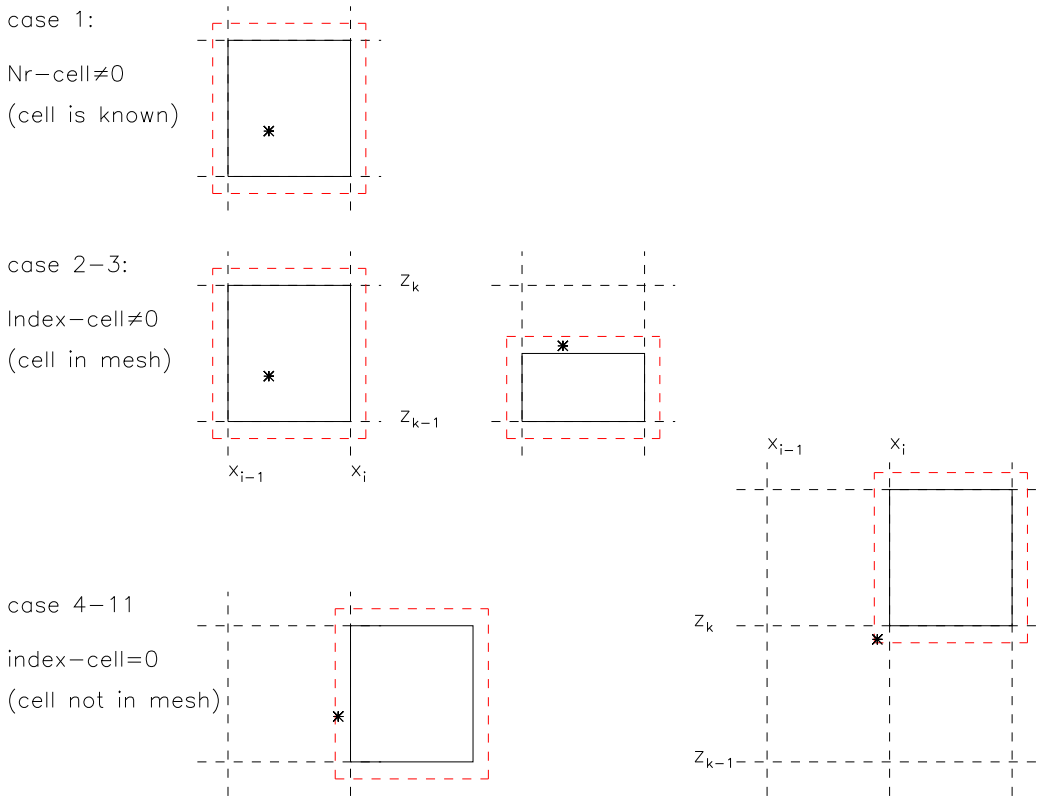


Figure 75: Cases of location of a particle in program. Red dash line is the refraction-zone at surface, due to a planar surface potential

C. Geometry of a binary collision

Fig. 76 shows the points P_0, P_1 and P_2 and T_0 and T_1 where computing takes place in the programs 'projectile' and 'recoil'.

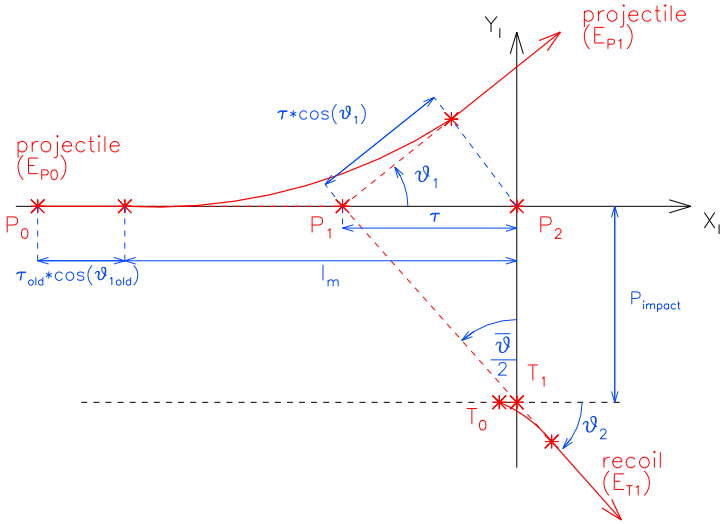


Figure 76: Trajectories of the projectiles and recoil particle in the laboratory system

D. Gaussian distribution of incidence angle energy

D.1. Gaussian distribution of incidence angle

Input values are case_alpha=5, alpha0 and dalpha0. The polar angle alpha0 (α_0) has a range of [0...90°]. The azimuthal angle φ_{azi} may be 0°($\alpha >= 0$) or 180°($\alpha < 0$).

$$\text{mean:} \quad \alpha_0 = \quad alpha0 = \frac{1}{n} \cdot \sum \alpha$$

$$\text{standard deviation:} \quad \Delta\alpha = \quad dalpha0 = \frac{1}{n-1} \cdot \sum (\alpha - \alpha_0)$$

$$\text{frequency function:} \quad p = \quad \frac{1}{\sqrt{2 \cdot \Pi \Delta\alpha^2}} \cdot e^{\frac{-(\alpha-\alpha_0)^2}{2\Delta\alpha^2}}$$

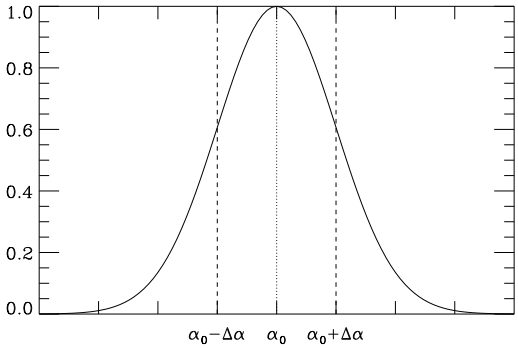


Figure 77: Gauss frequency distribution ($p \cdot \sqrt{2 \cdot \Pi \Delta\alpha^2}$) of polar angles

D.2. Gaussian distribution of incidence energy

Input values are case_e0=5, e0_beam and de0_beam. The beam energy e0_beam (e_0)has a range of [$1eV$...$100MeV$].

$$\text{mean:} \quad e_0 = \quad e0\_beam = \frac{1}{n} \cdot \sum e$$

$$\text{standard deviation:} \quad \Delta e = \quad de0\_beam = \frac{1}{n-1} \cdot \sum (e - e_0)$$

$$\text{frequency function:} \quad p = \quad \frac{1}{\sqrt{2 \cdot \Pi \Delta e^2}} \cdot e^{\frac{-(e-e_0)^2}{2\Delta e^2}}$$

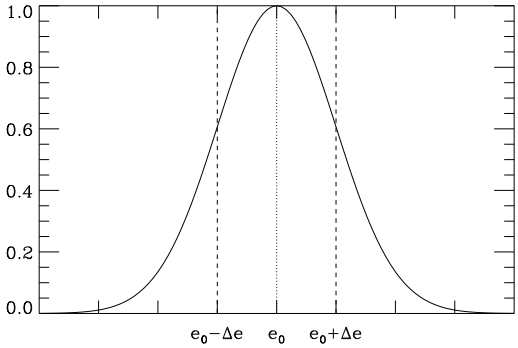


Figure 78: Gauss frequency distribution ($p \cdot \sqrt{2 \cdot \Pi \Delta e^2}$) of energy (case 6)

E. input and output files

All input files have the extension '.inp'.
All output files have the extension '.dat'.

E.1. List of input-files

name	description
tri.inp	main input-file
rauh.inp	x,z value of roughness
ininitial_composition.inp	input of initial composition of target dependent on depth layers
ininitial_composition_surf.inp	input of initial composition near surface
stop_run.inp	possible to break (0/1) during the run
restart.inp	restart-file generated with flag l_write_restart=.true.
Restart_data.inp	change some input-values from tri.inp after restart
doped_zellen_2D.inp	input of cells with other composition as target
energy.inp	input of incident energy distribution
angle.inp	input of incident angular distribution
fig.inp	input of roughness cells
mat_surfb.inp	input matrix of surface-binding-energy (isbv=6)

Table 3: List of input files

E.2. List of Output files

name	description
output.dat	main in and out key figures
T_10_2D.dat ... f(flc)	cell-coordinates(x_1, x_2, y_1, y_2), atomic-fraction (qu(1:ncp)), density (dns), see Fig. 46 - 50, Fig. 56, Fig. 31 - 37 or Fig. 21 - 22 (profil/atomic fraction)
E0_31_3D.dat ... f(flc)	surface, scattering and sputtering values (for calculate coefficients use: /post/readtridyn31.F90), see Fig. 55 or Fig. 21 - 22 (surface/sputt.-coeff)
surf.dat ... f(flc)	surface coordinates x z, see Fig. 52 - 54
surf_proj.dat ... f(flc)	surface-distribution of number incident projectiles
loss_depos.dat ... f(flc)	distribution of local in/out of particles at projected-surface f(x) see Fig. 58
loss_depos_2.dat ... f(flc)	static: distribution of local in/out of particles at surface f(x,y)
re_deposit.dat ... f(flc)	local re-deposition of particles
sputter_yield.dat ... f(flc)	local scattered and sputtering yield
activ_cell.dat ... f(flc)	corresponding cell to T_10_2D.dat pos...swell, neg...shrink, 0...inactive, see Fig. 46 - 49
profil.dat	one profile qu(z)

Table 4: List of main output files (f(flc) ... dependent on fluence)

Particle-information:
partic_stop_p.dat, partic_back_p.dat ,partic_tran_p.dat, partic_stop_r.dat, partic_back_r.dat, partic_tran_r.dat

Trajectories-information:
trajec_all.dat, trajec_stop_p.dat, trajec_back_p.dat, trajec_tran_p.dat, trajec_stop_r.dat, trajec_back_r.dat, trajec_tran_r.dat,
see Fig. 18 or Fig. 58

Help-files:
ausdat.dat, time_2.dat, ausdat.dat, ausgabenr.dat

Run-time information during the run of program:
time00.dat, time_2.dat

Matrix-file (see Chapter G) :
meagb_p.dat, meagb_s.dat, meagt_p.dat, meagt_s.dat,
see Fig. 62 - Fig. 73

Other:
doped_zellen_2D.inp.dat:
output of doped cells, may be used for input of doped_zellen_2D.inp,
see Chapter 10.4.4 and Fig. 46

mat_surfb.inp:
matrix of surface-bindig-energy [eV] —text
3 —number of elements
2 He —Z, symbol for check of input
74 W —Z, symbol for check of input
8 O —Z, symbol for check of input

He	W	O	
0.0000	0.0000	0.0000	for He
0.0000	8.6400	3.5	for W
0.0000	3.5	3.5	for O

F. Inputfiles 'energy.inp' and 'angle.inp'

F.1. Distribution-values per interval of energy or angle

Values of distribution are per interval of energy or interval of angle.

energy(eV)	distribution [-/dE]
100	1
200	2
300	2
500	5
600	8
700	12
800	20
900	25

Table 5: Inputfile 'energy.inp' values per energy-interval

angle(degree)	distribution [-/dangle]
30	10
60	20
70	20
90	70

Table 6: Inputfile 'angle.inp' per angle-interval

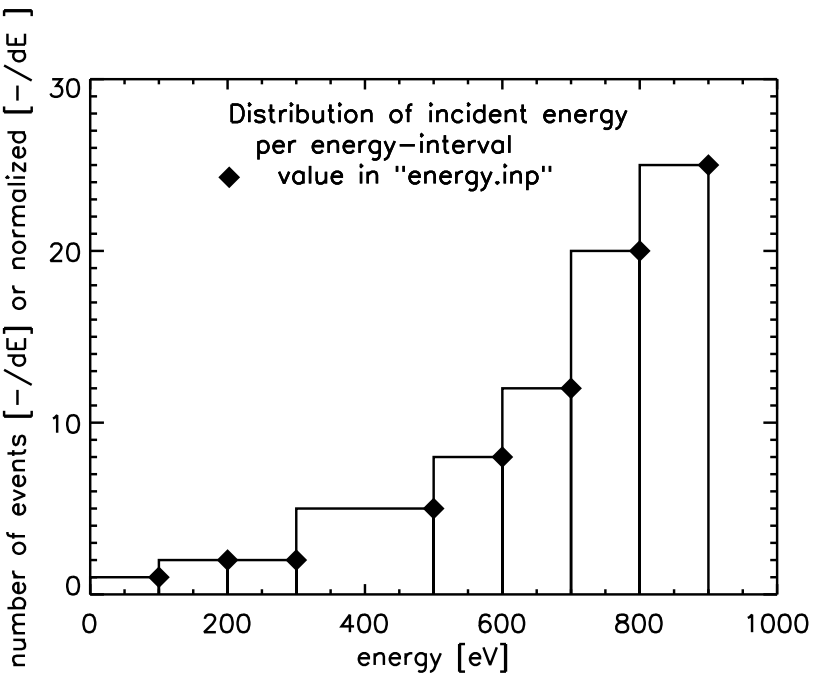


Figure 79: Distribution of incident energy according to Tab. 5

F.2. Distribution-values in constant intervals are event-values or in %

energy(eV)	distribution [-]
100	1
200	2
300	2
400	5
500	5
600	8
700	12
800	20
900	25

Table 7: The same distribution as Tab. 5 but constant interval of energy: the distribution may be number of events or measured values

energy(eV)	distribution [%]
100	1.25
200	2.50
300	2.50
400	6.25
500	6.25
600	10.00
700	15.00
800	25.00
900	31.25

Table 8: The same distribution as Tab. 5 but constant interval of energy: the distribution may be in % (sum is 100%)

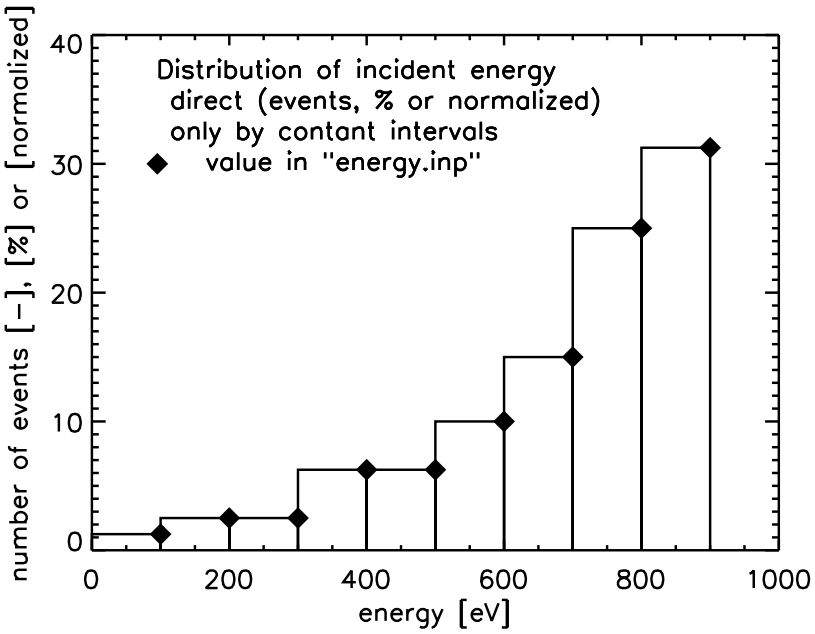


Figure 80: The same distribution of energy as Tab. 5 but values according to Tab. 8

G. Output of energy and angular distributions

G.1. Output-matrix-file

The option *lmatrices* initiates output of the energy and angular distributions of scattered and sputtered atoms into four files.

option:	lmatrices =.true.	
optional option:	lmatout_cos_angle=.true.	
output:	meagb_p.dat	...output of back-scattered particles (projectiles)
	meagb_s.dat	...output of all back-sputtered particles (recoils)
	meagt_p.dat	...output of all transmitted scattered particles (projectiles)
	meagt_s.dat	...output of all transmitted sputtered particles (recoils)

G.2. Post-processing of output-matrix-file with readmatrix4.F90

The FORTRAN program *readmatrix*4.*F*90 in the directory *post* splits the four matrices into individual matrices.

post program :	/post/readmatrix4.F90
input files:	meagb_p.dat
	meagb_s.dat
	meagt_p.dat
	meagt_s.dat

outputfiles:

name of outputfile	x-axis	y-axis	values
matrix__ag..	polar(lin)	azimuth(lin)	number of particles
matrix_Sag..	polar(lin)	azimuth(lin)	number of particles per solid angle
matrix__ea..	energy(lin)	polar(lin)	number of particles
matrix__eg..	energy(lin)	azimuth(lin)	number of particles
matrix__ee..	polar(lin)	azimuth(lin)	energy
matrix_lea..	energy(log)	polar(lin)	number of particles
matrix_leg..	energy(log)	azimuth(lin)	number of particles
matrixc_ag..	polar(cos)	azimuth(lin)	number of particles
matrixc_ea..	energy(lin)	polar(cos)	number of particles
matrixcSag..	polar(cos)	azimuth(lin)	number of particles per solid angle
matrixc_eg..	energy(lin)	azimuth(lin)	number of particles
matrixc_ee..	polar(cos)	azimuth(lin)	energy
matrixclea..	energy(log)	polar(cos)	number of particles
matrixcleg..	energy(log)	azimuth	number of particles

file extension:

..b_p..	back-scattered projectiles
..t_p..	transmitted projectiles
..b_s..	back-sputtered recoil
..t_s..	transmitted-sputtered recoil
..1.dat	number species

a ...polar angle
b ...back (-scattered/-sputtered)
p ...projectile
l ...log(e)
S ...values per solid angle
g ...azimuthal angle
t ...transmitted
s ...sputtered recoil
c ...cosine interval of polar angle

An example for the naming convention used is matrix__agb_p1.dat (number of 1. back-scattered-projectile dependent on polar- and azimuthal-angles)

H. Global parameters

parameter	value	description	program
ncpm	8	max. number of species	param.F90
pemax	256	max. number of cores for parallelization	param.F90
ntqmax	1000000	memory request limit for coll. cascades	default.F90

Table 9: Global parameters (Values are set in the program)

I. Input variables in 'tri.inp'

I.1. Necessary input variables in 'tri.inp'

The sequence of input values in the input file is arbitrary (namelist), Tab. 10 -18

variable	description
alpha0(ncp)	angle of incidence (degree) of ncp species in case_alpha=0,5
e0(ncp)	energies (eV) of projectiles (qubeam $>$ 0.) for case_e0=0,5 e0=ttemp $\cdot$ boltzm (e0 $<$ 0) of projectiles for case_e0=2,3 temperature (eV) (kT) (e0 $>$ 0) of projectiles for case_e0=2,3
flc	incident fluence (10^{16} atoms/cm^2 or atoms/Å^2)
case_geo	flag of geometry and input surface = 10 : one dimensional = 20 : two dimensional, periodic, with smooth surface = 24 : two dimensional, periodic, with surface profile in rauh.inp left: x=0, left values =right values is set in program = 25 : two dimensional, periodic, with surface profile in rauh.inp, centre: x=0 = 26 : input of two dimensional voxel, cells (x1,x2,z1,z2,qu(ncp)) in fig.inp, voxels on base plate = 27 : two dimensional, periodic, with surface profile cos(x), centre: x=0 parameter: pr_amp pr_wl
ipot	interaction potential: = 1 : KrC = 2 : Moliere = 3 : ZBL = 4 : Na-Ya = 5 : Si-Si = 6 : power
ncp	number of species (projectiles + target species) more than one projectile species is allowed
maxhist	number of histories (projectiles)

Table 10: Necessary input variables (no default values)

variable	description
isbv	atoms binding model, determines the composition dependent surface-binding-energy sbv(ncp,ncp) from the elemental surface binding energies e_surfb(ncp) and bulk-binding-energy e_bulkb(ncp) of target atoms = 1 : sbv(ip,jp)=e_surfb(jp) for ip=jp, =0 else e_bulkb(1:ncp)=0.0 e_surfb taken from table1 = 2 : sbv(ip,jp)=e_surfb(jp) for all ip, jp e_bulkb(1:ncp)=0.0 e_surfb taken from table1 = 3 : sbv(ip,jp)=0., if e_surfb(ip)=0 or e_surfb(jp)=0 sbv(ip,jp)=0.5*(e_surfb(ip)+e_surfb(jp)) else e_bulkb(1:ncp)=0.0 e_surfb taken from table1 = 4 : sbv(ip,jp)=e_surfb(jp) for ip=jp, =0 else (like isbv=1) e_bulkb(1:ncp)=0.0 e_surfb taken from table1 only for compound-atoms (ip): e_bulkb(ip)=e_bulkb(ip)+$\Delta H_f(ip)$/nm ΔH_f taken from table.compound = 6 : input of given matrix of the surface-binding-energy input-file: mat_surfb.inp = 8 : sbv(1:ncp,1:ncp)=0.0 (e_surfb=0.0) e_bulkb(1:ncp)=ΔH_{sub} e_cutoff(1:ncp)=$1/3 \cdot \Delta H_{sub}$ e_bulkb and e_cutoff taken from table1a only for compound-atoms (ip): e_bulkb(ip)=ΔH_{sub}+ΔH_f/nm ΔH_f taken from table.compound
qu_beam(ncp)	projectile atomic fractions (in incident beam) of ncp species, qu_beam $> 0.$, Note: sum(qubeam(1:ncp))=1 qu_beam $\leq 1.$ for projectiles, qu_beam $= 0.$ for target atoms
qu_tar(ncp)	initial target atomic fractions of ncp species in case of homogeneous initial composition (iq0 = 0)
symbol(ncp)	ncp chemical symbols of elements according to table1 (special symbol: 'H','D','T','He3','He','C_g', 'C_f','C_d')

Table 11: Necessary input variables (no default values)

I.2. Optional input variables in 'tri.inp'

These values have default values (see default_init.txt). If values different from the default values are needed, then these values have to be given explicitly in the input file.

variable	default value	description
beta0(ncp)	0	azimuthal angle of incidence (degree) of ncp species beta0 =0°...360° (starting above surface) (in case_alpha=1, 2 or 3 beta0=random-distribution)
case_alpha	0	flag for the choice of the angle of incidence = 0 : angles of incidence in degree alpha0 and beta0 are fixed during calculation α_{pol}=alpha0 =0°...90° (starting above surface) φ_{azi}=beta0 =0°...360° (starting above surface) = 1 : random distribution of angles $\alpha_{pol} = 0...alpha0$, $\varphi_{azi} = 0...360°$ = 2 : cosine distribution of angles of incidence $\alpha_{pol} = 0...90°$,max: by 45° , $\varphi_{azi} = 0...360°$ = 3 : cosine distribution of angles of incidence $\alpha_{pol} = 0...90°$,max: by 0° , $\varphi_{azi} = 0...360°$ = 4 : input of a given incident angular distribution from file angle.inp ($\varphi_{azi} = 0°$) = 5 : polar angle (α_{pol}) has a gaussian distribution $\varphi_{azi} = 0° or 180°$) input: α_0=alpha0 and $\Delta\alpha$=dalpha0
case_e0	0	flag for the choice of the incident energy = 0 : fixed incident energies(eV) of projectiles (qubeam>0) = 1 : input of a given energy distribution from file energy.inp = 2 : temperature (eV) of a Maxwellian velocity distribution kT=$ttemp \cdot boltzm$ [eV] (e0_beam < 0) of projectiles kT=e0_beam [eV] (e0_beam(> 0) of projectiles = 3 : temperature (eV) of a Maxwellian energy distribution kT=$ttemp \cdot boltzm$ [eV] (e0_beam < 0) of projectiles kT=e0_beam [eV] (e0_beam(> 0) of projectiles = 5 : energy has a gaussian distribution input: $e_0 = e0\_beam$ and $\Delta e = de0\_beam$

Table 12: Optional input variables with default values

variable	default value	description
case_start	3	input of start position of projectiles (x_start, z_start) = 1 : x_start and z_start are given in tri.inp z(start) = cel(x_start, z_start) +thick_deflec = 2 : not used = 3 : x_start and z_start are constant and given in tri.inp = 4 : z_start is constant and given in tri.inp x_start and dx_start are in tri.inp x= x_start-0.5*dx_start+random*dx_start = 5 : program intern x-distribution
charge(ncp)	0	charge of species if case_e0=2,3 and sheath>0 (plasma) $\geq$ 1. for qubeam>0 (projectiles) = 0. for qubeam=0 (target atoms)
dalpha0(ncp)	1	$\Delta\alpha$ [degree] for gaussian distribution, see case_alpha =5
de0_beam(ncp)	0	Δe [eV] for gauss distribution, see case_e0=5
deltahf	table	heat of formation (eV) of a molecular target default from table1
diff_koeff1(ncp)	1.0	transport-coefficient if loutgas true [$Å^3/ion$] (see also: loutgas)
diff_koeff2(ncp)	1.0	diffusion-coefficient if loutgas true [$Å^4/ion$] (see also: loutgas)
dns_0(ncp)	table	atomic density ($atoms/Å^3$) of ncp elements; default from table1
dx_start	0.	X-width of the beam [Å]
e_bulkb(ncp)	0.	bulk binding energy; if e_bulkb>0., e_bulk has to be subtracted from the surface binding energy e_surfb
e_cutoff(ncp)	table	cutoff energy (eV) of ncp species; defaults from table1 (0.05 eV for noble gases; 1 eV for H, D, T; e_surf - 0.05 eV for self-bombardment)
e_displ(ncp)	table	displacement energy (eV); default from table1 (if in table1 e_displ=0 then e_displ=15)
e_surfb(ncp)	table	surface binding energy (eV) (heat of sublimation);

Table 13: Optional input variables with default values

variable	default value	description
flc_flux	1.0	flux [atoms/(Å^2 s)], only to calculate time or fluence
flc_time	-1	time [s], only to calculate flux or fluence
geo_x	500	maximal width of target to right cell-expansion, unit =[Å] range of target=[-geo_x ... geo_x]
geo_z	500	maximal depth of target without cell-expansion, unit=[Å] range of depth=[0 ... geo_z]
geo_dx	10	cell-width in X, [Å]
geo_dz	10	cell-width in Z, [Å]
geo_imax	1	reduce target cell during calculation 1...no 2...yes
ihist_out	-1	control output, determines the outputfiles: E0_31_target.dat, E0_34_moments.dat, partic*.dat, trajec*.dat and restart_file = -1 : output after each fluence step of maxhist/100, 100 fluence steps = 0 : output only after the last fluence step > 0 : output after each ihist_out'th fluence step and last step
iintegral	0	integration method = 0 : MAGIC, only valid for KrC, ZBL, Moliere not recommended = 1 : Gauss-Mehler quadrature, ipivot $\geq$ 8 = 2 : Gauss-Legendre quadrature, ipivot $\leq$ 16 recommended
ioutput_hist(6)	10	number of traced trajectories for: stopped, backscattered and transmitted projectiles, stopped, backsputtered, transmission sputtered recoils (see also: ltraj_p, ltraj_r)
ioutput_part(6)	10	number of traced particles for: stopped, backscattered and transmitted projectiles, stopped, backsputtered, transmission sputtered recoils (see also: lparticle_p, lparticle_r)
ipivot	16	number of pivots in the Gauss-Mehler and Gauss-Legendre integration, the minimum number is 4 (larger numbers increase the computing time)

Table 14: Optional input variables with default values (continue)

variable	default value	description
inel0(ncp)	7,4,5	inelastic loss model = 1 : Lindhard-Scharff; necessary condition: $E < 25 \cdot Z^{4/3} \cdot M$ (in keV) where E, Z, M are the energy, the atomic number and the atomic mass of the moving particle = 2 : Oen-Robinson; necessary condition:$E < 25 \cdot Z^{4/3} \cdot M$ (in keV) = 3 : equally distributed of 1 and 2 = 4 : hydrogen (H,D,T): values from 'table3' = 5 : helium (He3,He): values from 'table4' = 6 : values is calculated for each element use values from 'table6a' and 'table6b' = 7 : values is calculated for each element with help of mix of 1 (low energy), 6 (high energy) and correctur-factors from 'table7_ck' and 'table7_a3a4a5' = 8 : use manual tabulated stopping power from database 'tbsp.db', to be found in tables-directory
iq0	0	initial composition flag = 0 : initial composition homogeneous, from qu_tar in 'tri.inp' > 0 : initial depth dependent composition taken from file 'initial_composition.inp'
iq0_surf	0	initial composition flag near surface = 0 : no composition homogeneous, one layer with > 0 : initial depth dependent composition taken from file 'initial_composition_surf.inp'
it_diff	10	number of iterations-steps for outgasing
irand	1	random seed
irc0	1	flag for subthreshold recoil atoms < 0 : subthreshold recoil atoms free ≥ 0 : subthreshold atoms bound
isot(ncp)	0	flag for isotope mass = 0 : natural isotope mixture (mass from table1) = 1 : isotope masses and natural abundances from table2 (valid for projectiles as well as for target species)

Table 15: Optional input variables with default values (continue)

variable	default value	description
iwc=2	2	number of ring cylinders for weak simultaneous collisions for projectiles; for high energies (MeV H or He) iwc can be reduced to 1 or 0 to reduce computing time
iwcr=2	2	number of ring cylinders for weak simultaneous collisions for recoils
l_alpha_rough_kor	.false.	coordinate system has to be rotated for correction of boundary conditions (left, right)
lchem_ch	.false.	calculation with chemical erosion H on C, D on C
l_clust	.false.	.true. : calculate of cluster-cells (see: n_clust, l_read, qu_clust) .false. : no cluster input of cells
l_clust_read	.false.	.true. : read inputfile 'doped_zellen_2D.inp' (seeL n_clust, qu_clust, l_clust) .false. : no read input-file
lmatrices	.false.	.true. : output of matrices, if idrel /= 0 .false. : no matrix output
lmatout_cos_angle	.false.	angular spacing = .false. : angle in degree intervals = .true. : cosine intervals
loutgas	.false.	calculation with outgasing transport and diffusion (see also diff_koeff1,diff_koeff2)
lparticle_p	.false.	.true. : output of projectile information .false. : no output of projectile information (see also: ioutput_part)
lparticle_r	.false.	.true. : output of recoil information .false. : no output of recoil information (see also: ioutput_part)
ltraj_p	.false.	.true. : output of projectile trajectories .false. : no output of projectile trajectories (see also: numb_hist, ioutput_hist)
ltraj_r	.false.	.true. : output of recoil trajectories .false. : no output of recoil trajectories (see also: numb_hist, ioutput_hist)

Table 16: Optional input variables with default values (continue)

variable	default value	description
l_two_comp	.false.	.true. : calculation density of one two-compound (eg. SiO2)
l_two_comp2	.false.	.true. : calculation density of two two-compound (eg. SiO2+CaO)
l_two_comp3	.false.	.true. : calculation density of three two-compound (eg. FeO+SiO2+CaO)
l_write_restart	.false.	.true. : output of restart-file
matrix_e_min_p	0	minimum of lin. energy distribution of projectiles in matrices
matrix_e_min_r	0	minimum of lin. energy distribution of recoils in matrices
matrix_e_max_p	max(e0)	maximum of lin. energy distribution of projectiles in matrices
matrix_e_max_r	max(e0)	maximum of lin. energy distribution of recoils in matrices
n_clust	1	number of cells inside a cluster (1,4,9,16,36) (see: l_clust, qu_clust, l_clust_read)
nr_pproj	10	number of projectiles between two target updates (idrel = 0)
numb_hist	20	number of traced trajectories of projectiles and recoils
pr_amp	0.	amplitude of a cosine structure (see also: case_geo =27)
pr_wl	1.	wavelength of a cosine structure (see also: case_geo =27)
qu_max(ncp)	1.	maximum atomic fractions in the target for ncp species, if idrel=0
qu_clust	1	composition of cluster-cell (see: l_clust, n_clust, l_clust_read)
rhom	-1	atomic density of a two-component target; default from table.compound $[g/cm^3]$
rhom2	-1	atomic density of a second two-component target; default from table.compound $[g/cm^3]$
rhom3	-1	atomic density of a third two-component target; default from table.compound $[g/cm^3]$

Table 17: Optional input variables with default values (continue)

variable	default value	description
sfin	0.	$= 0$: no inelastic energy loss outside the target surface ($x = 0.$) $= 1$: inelastic energy loss outside the target surface ($-su > x > 0.$)
shth	0.	$= 0$: no sheath potential > 0 : sheath potential (eV), usually $= 3 \cdot \|e0\| \cdot$ charge, only if case_e0=2,3 (Maxwellian distribution, plasma)
tableinp	'../tables'	directory of inputfile for tables
tar_dynamic	true	mode of simulation = true : full dynamic calculation (like TRIDYN) = false : suppression of dynamic relaxation (like TRIM), full static calculation
ttemp	0.	target temperature, only of interest at high temperatures, it reduces the surface binding energy according to a Maxwellian energy distribution
two_comp	'???'	name of first compound "SiO2" Note: only selected compounds in table.compound
two_comp2	'???'	name of second compound "CaO" Note: only selected compounds in table.compound
two_comp3	'???'	name of third compound "TaO2" Note: only selected compounds in table.compound
tttt		text in tri.inp
x_start	0.	x starting position of projectile
z_start	0.	z starting position of projectile

Table 18: Optional input variables with default values (continue)

J. Compiler information

The program needs FORTRAN-90 compiler and use for compile the program 'make'.
It turned out that the results are consistent for all three available fortran-compilers (GCC, NAG, INTEL) in the sequential setting, as well as the code results using 'use mpi' or 'use mpi_f08' for the parallel versions of intel and gcc/gfortran.
The Makefile in /src the mk-script in /bin/* and the run-script in case/ is more an example. You must modify these file according to your compiler and your computer-machine.

J.1. Makefile

Example of compiler options (set in /src/Makefile):

GCC:
gfortran -c -O2 -Wextra -I ../../src -DSEQ -DRAND2

INTEL:
ifort -c -w -FR -O2 -xP -I ../../src -DSEQ -DRAND2

NAG:
nagfor -c -O0 -g -C=undefined -mtrace -gline -quiet -ieee=stop -I ../../src -DSEQ -DRAND2

J.2. mk

The script 'mk' call 'make' and use as input the option from Makefile. Example for GCC sequential (set in /bin/gnu_GCC.SEQ) is:

#! /bin/sh -x
SRC_DIR=../../src
if [! -L Makefile]; then
ln -s $SRC_DIR/Makefile Makefile fi
make SRC_DIR=$SRC_DIR MODE=SEQ RAND=RAND2 COMPILER=GCC DEBUG=NO DEBUG2=NO -e -f $SRC_DIR/Makefile

option are: MODE= SEQ (sequential) or PPROJ (parallel)
RAND= RAND2 (32 bit) or CRAY (64 bit only intel)
COMPILER= GCC or INTEL or NAG
MPIMODE=MPI_90_8 (standard use mpi_f08) or MPI (use mpi)

J.3. run

The start of program in sequential mode from /case-directory is:
/2D-SDTrimSP/bin/gnu_GCC.SEQ/target.exe

The start of the program in parallel mode is very dependent on the computer-system and the organisation of queue system.
Example: salloc -n 64 -t 72:00:00
srun /2D-SDTrimSP/bin/gnu_GCC.PRO/2D-target.exe
or: mpirun -n 64 /SDTrimSP/bin/gnu_GCC.PRO/target.exe

K. Examples of Inputfile 'tri.inp'

Inputfile 'tri.inp' of smooth surface example Xe − > Si, see Fig. 48.

```
5000 eV Xe − > Si
&TRI_INP
tttt='—elements—'
      ncp = 2
      symbol ="Xe","Si"
tttt='—beam—'
      qu_beam = 1.000, 0.000
      case_e0 = 0
      e0_beam = 5000, 0.00
      case_alpha= 0
      alpha0 = 0.000, 0.000
tttt='—target—'
      qu_tar = 0.000, 1.000
      qu_max = 1.000, 1.000
tttt='—control—'
      flc = 2
      maxhist = 2000
      ihist_out = 200
      nr_pproj = 512
      tar_dynamic = .true.
      ipot = 1
      isbv = 1
tttt=' case_start: 4... given x:x+dx*random, z_start=constant'
      case_start= 4
      x_start = 0.
      dx_start= 1000.
      z_start = 500.
tttt=' case_geo: 10...one dim 20...smooth 25...rauh.inp '
      case_geo = 20
      geo_x = 500.
      geo_z = -1000.
      geo_dx = 10.0
      geo_dz = 10.0
tttt=' outgasing Xe'
      loutgas = .true.
      diff_koeff1 =1.65e06, 0.0
      diff_koeff2 = 95, 0.0
/
```

Inputfile 'tri.inp' of 1D example He − > Ta_2O_5, see Fig. 21.

```
1000 eV He − > Ta2O5
&TRI_INP
tttt='—elements—'
      ncp = 3
      symbol ="He", "Ta", "O"
tttt='—beam—'
      qu_beam = 1.000, 0.000, 0.000
      case_e0 = 0
      e0_beam = 1000, 0.00, 0.00
      case_alpha= 0
      alpha0 = 0.000, 0.000, 0.000
tttt='—target—'
      qu_tar = 0.000, 0.285714, 0.714286
      qu_max = 0.000, 1.000, 0.714286
      l_two_comp=.true.
      two_comp ="Ta2O5"
tttt='—control—'
      flc = 100
      maxhist = 4000
      ihist_out = 40
      nr_pproj = 512
      tar_dynamic = .true.
      ipot = 1
      isbv = 1
tttt=' case_start: 1...x=x_start, z_start=constant'
      case_start= 1
      x_start = 0.
      dx_start= 1000.
      z_start = 500.
tttt='geometry: case_geo = 10 ... 1-D'
      case_geo = 10
      geo_x = 500.
      geo_z = -1000.
      geo_dx = 1000.0
      geo_dz = 20.0
tttt=' outgasing He'
      loutgas = .true.
      diff_koeff1 = 0.50e5, 0.0
      diff_koeff2 = 60, 0.0
/
```

Inputfile 'tri.inp' of limited beam thickness Xe − > Si, see Fig. 49.

```
5000 eV Xe − > Si
&TRI_INP
tttt='—elements—'
      ncp = 2
      symbol ="Xe","Si"
tttt='—beam—'
      qu_beam = 1.000, 0.000
      case_e0 = 0
      e0_beam = 5000, 0.00
      case_alpha= 0
      alpha0 = 0.000, 0.000
tttt='—target—'
      qu_tar = 0.000, 1.000
      qu_max = 1.000, 1.000
tttt='—control—'
      flc = 2
      maxhist = 2000
      ihist_out = 200
      nr_pproj = 512
      tar_dynamic = .true.
      ipot = 1
      isbv = 1
tttt=' case_start: 4... given x:x+dx*random, z_start=constant'
      case_start= 4
      x_start = 0.
tttt=' limited beam: dx_start < 2*geo_x '
      dx_start= 200.
      z_start = 500.
tttt=' case_geo: 10 one dim, 20 smooth 25 rauh.inp '
      case_geo = 20
      geo_x = 500.
      geo_z = -1000.
      geo_dx = 10.0
      geo_dz = 10.0
tttt=' outgasing Xe'
      loutgas = .true.
      diff_koeff1 =1.65e06, 0.0
      diff_koeff2 = 95, 0.0
/
```

Inputfile 'tri.inp' of roughness example Ar − > Si,Ta, see Fig. 51 - 53.

```
6 keV Ar − > Si,Ta
&TRI_INP
tttt='—elements—'
      ncp = 3
      symbol = "Ar","Si","Ta"
tttt=' beam'
      qu_beam = 1.00, 0.00 , 0.00
      case_e0 = 0
      e0_beam = 6000.00, 0.00, 0.00
      case_alpha= 0
      alpha0 = 42.000, 0.000, 0.000
tttt='target'
      qu_tar = 0.00, 1.00, 0.00
      qu_max = 1.00, 1.00 , 1.00
tttt='control'
      flc = 60
      maxhist = 24000
      ihist_out = 1000
      nr_pproj = 960
      tar_dynamic = .true.
      ipot = 1
      isbv = 1
      geo_imax=2
tttt=' case_start: 4... given x:x+dx*random, z_start=constant'
      case_start= 4
      x_start = 0.
      dx_start= 5000.
      z_start = 2500.
tttt=' case_geo: 10 one dim, 20 smooth 25 with profile rauh.inp '
      case_geo=25
      geo_x = 2500.
      geo_z = -7000.
      geo_dx = 20.0
      geo_dz = 20.0
tttt=' outgasing Ar'
      loutgas = .true.
      diff_koeff1 = 1.65e5, 0, 0
      diff_koeff2 = 15 , 0, 0
/
```

Inputfile 'tri.inp' of roughness example 6000 eV C − > SiTi, see Fig. 54.

```
6 keV C − > Si,Ta
&TRI_INP
tttt='—elements—'
      ncp = 3
      symbol = "C_g","Si","Ta"
tttt=' beam'
      qu_beam = 1.00, 0.00 , 0.00
      case_e0 = 0
      e0_beam = 6000.00, 0.00, 0.00
      case_alpha= 0
      alpha0 = 0.000, 0.000, 0.000
tttt='target'
      qu_tar = 0.00, 1.00, 0.00
      qu_max = 1.00, 1.00 , 1.00
tttt='control'
      flc = 200
      maxhist = 20000
      ihist_out = 500
      nr_pproj = 960
      tar_dynamic = .true.
      ipot = 1
      isbv = 1
      geo_imax=2
tttt=' case_start: 4... given x:x+dx*random, z_start=constant'
      case_start= 4
      x_start = 0.
      dx_start= 5000.
      z_start = 2500.
tttt=' case_geo: 10...one dim, 20...smooth 25...rauh.inp '
      case_geo=25
      geo_x = 2500.
      geo_z = -7000.
      geo_dx = 20.0
      geo_dz = 20.0
/
```

Inputfile 'rauh.inp' of roughness example Ar − > Si,Ta, see Fig. 51 - 54.

```
pitch grating (x,y Values)
      6
     -2600.0   0000.0
     -1350.0   0000.0
     -1100.0   2000.0
      1100.0   2000.0
      1350.0   0000.0
      2600.0   0000.0
```

Inputfile 'tri.inp' of cluster example D − > Fe,W, see Fig. 46.

```
200 eV D − > FeW (doped with W 0.30 x 0.30 nm)
&TRI_INP
tttt='—elements—'
      ncp = 3
      symbol = "D","Fe","W"
tttt=' beam'
      qu_beam = 1.00, 0.00 , 0.00
      case_e0 = 0
      e0_beam = 200, 0.000, 0.000
      case_alpha= 0
      alpha0 = 0.000, 0.000, 0.000
tttt='target'
      qu_tar = 0.00, 1.00, 0.00
      qu_max = 0.00, 1.00 , 1.00
tttt='control'
      flc = 500
      maxhist = 25000
      ihist_out = 5000
      nr_pproj = 1024
      tar_dynamic = .true.
      ipot = 1
      isbv = 1
       l_clust=.true.
       n_clust=36
       qu_clust=0.000, 0.000 ,1.000
       l_clust_read=.false.
tttt=' case_start: 4... given x:x+dx*random, z_start=constant'
      case_start= 4
      x_start = 0.
      dx_start= 1080.
      z_start = 1000.
tttt=' case_geo: 10...one dim, 20...smooth 25...rauh.inp '
      case_geo=20
      geo_x = 540.
      geo_z = -500.
      geo_dx = 5.0
      geo_dz = 5.0
      geo_imax = 2
/
```

Inputfile 'tri.inp' of initial_composition C_g -> Si,Ta, similar Fig. 42 and Fig. 43.

```
6 keV C − > Si,Ta
&TRI_INP
tttt='—elements—'
      ncp = 3
      symbol = "C_g","Si","Ta"
tttt=' beam'
      qu_beam = 1.00, 0.00 , 0.00
      case_e0 = 0
      e0_beam = 6000, 0.000, 0.000
      case_alpha= 0
      alpha0 = 0.000, 0.000, 0.000
tttt='target'
      qu_tar = 0.00, 1.00, 0.00
      qu_max = 1.00, 1.00 , 1.00
       iq0=1          ! 0...homogeneous 1...initial_composition.inp (bulk)
       iq0_surf=1     ! 0...no 1... with initial_composition_surf.inp (surface)
tttt='control'
      flc = 2
      maxhist = 200
      ihist_out = 20
      nr_pproj = 960
      tar_dynamic = .true.
      ipot = 1
      isbv = 1
      geo_imax=2
tttt=' case_start: 4... given x:x+dx*random, z_start=constant'
      case_start= 4
      x_start = 0.
      dx_start= 5000.
      z_start = 3000.
tttt=' case_geo: 20...smooth 24...rauh.inp 25...rauh.inp 27...cos '
      case_geo=25
      geo_x = 2500.
      geo_z = -4000.
      geo_dx = 20.0
      geo_dz = 20.0
tttt='case_geo = 24 or 25 or 27 add surface cos-roughness'
       pr_amp=100
       pr_wl =500
/
```

Inputfile 'initial_composition.inp':

```
number layer, z, qu(1..ncp) ! —text
4                           ! —number layers
  10000.                    ! —z-start depth 1.layer (may be > surface target)
    0.00     0.30     0.70  ! —qu(1..ncp) C,Si,Ta
  -500.                     ! —z-start depth 2.layer
    0.00     1.00     0.00  ! —qu(1..ncp) C,Si,Ta
  -1000.                    ! —z-start depth 3.layer
    0.00     0.50     0.50  ! —qu(1..ncp) C,Si,Ta
  -1500.                    ! —z-start depth 4.layer
    0.00     0.00     1.00  ! —qu(1..ncp) C,Si,Ta
  -10000.                   ! —z-end last layer (may be < target)
```

Inputfile 'initial_composition_surf.inp'

```
number, distant from surface, qu(1..ncp) ! —text
2                                        ! —number layers
  0.0                                    ! —z-start at surface
    0.55     0.35     0.10               ! —qu(1..ncp) C,Si,Ta
  -100.                                  ! —z-start 2. layer from surface
    0.45     0.45     0.10               ! —qu(1..ncp) C,Si,Ta
  -200.                                  ! —z-end last layer from surface
```

Inputfile 'tri.inp' of yield distribution (lmatrices=.true.) Ar -> Si, see, Fig. 60.

```
1 keV Ar − > Si
&TRI_INP
tttt='—elements—'
      ncp = 2
      symbol = "Ar","Si"
tttt=' beam'
      qu_beam = 1.00, 0.00
      case_e0 = 0
      e0_beam = 1000, 0.000
      case_alpha= 0
      alpha0 = 45.00, 0.000
tttt='target'
      qu_tar = 0.00, 1.00
      qu_max = 1.00, 1.00
tttt='control'
      flc = 2
      maxhist = 1000
      ihist_out = 1000
      nr_pproj = 960
      tar_dynamic = .false.
      ipot = 1
      isbv = 1
tttt=' case_start: 4... given x:x+dx*random, z_start=constant'
      case_start= 4
      x_start = 0.
      dx_start= 5000.
      z_start = 3000.
tttt=' case_geo: 20...smooth 24...rauh.inp 25...rauh.inp 27...cos '
      case_geo=27
      pr_amp= 100
      pr_wl = 500
      geo_x = 2500.
      geo_z = -4000.
      geo_dx = 20.0
      geo_dz = 20.0
      lmatrices=.true.
/
```

Inputfile 'tri.inp' of voxel-input (fig.inp) Ar -> W,Si, see, Fig. 31-33.

```
100 eV Ar − > W,Si
&TRI_INP
tttt='—elements—'
      ncp = 3
      symbol = "Ar","W","Si"
tttt=' beam'
      qu_beam = 1.00, 0.00, 0.00
      case_e0 = 0
      e0_beam = 100.0, 0.0, 0.0
      case_alpha= 0
      alpha0 = 30.00, 0.000, 0.000
tttt='target'
      qu_tar = 0.00, 0.00, 1.00
      qu_max = 1.00, 1.00, 1.00
tttt='control'
      flc = 1
      maxhist = 100
      ihist_out = 10
      nr_pproj = 64
      tar_dynamic = .false.
      ipot = 1
      isbv = 1
tttt=' case_start: 4... given x:x+dx*random, z_start=constant'
      case_start= 4
      x_start = 0.
      dx_start= 400.
      z_start = 500.
tttt=' case_geo: 20...smooth 25...rauh.inp 26...fig.inp 27...cos '
      case_geo=26
      geo_x = 200.
      geo_z =-200.
      geo_dx = 25.0
      geo_dz = 25.0
/
```